Part III: Embedding-Driven Topic Discovery

AAAI 2022 Tutorial

Pre-Trained Language Representations for Text Mining

Yu Meng, Jiaxin Huang, Yu Zhang, Jiawei Han

Computer Science, University of Illinois at Urbana-Champaign

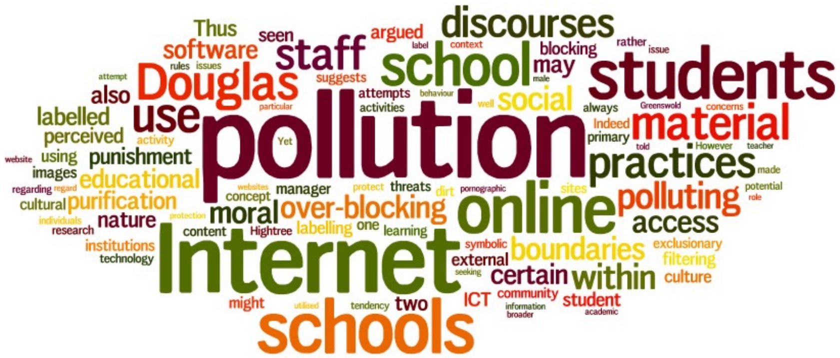
February 23, 2022

Outline

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

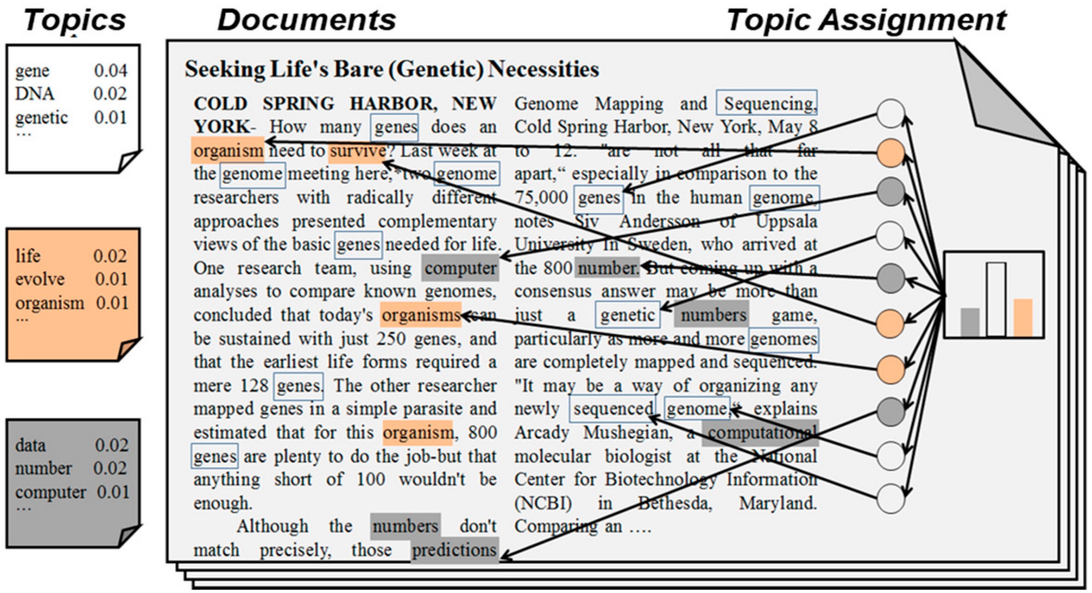
Topic Modeling: Introduction

- 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



Word-Topic Matrix

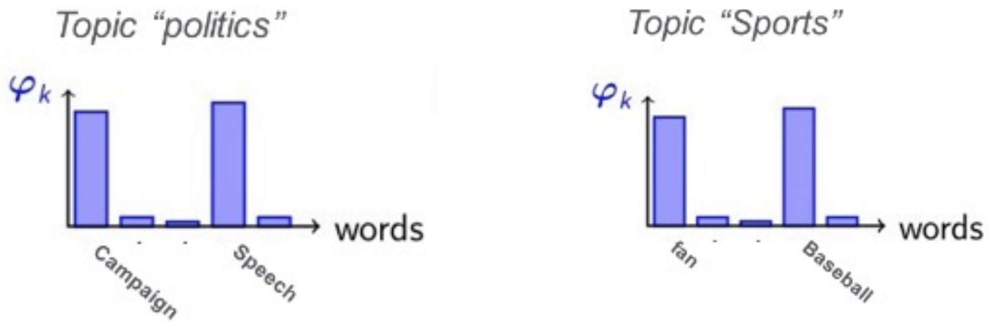
	Topic 1	…	Topic n
Word 1	0.09	…	0.06
…	….	…	…
Word n	0.08	…	0.01

Document-Topic Matrix

	Topic 1	…	Topic n
Doc 1	0.23	…	0.33
…	….	…	…
Doc n	0.15	…	0.28

Latent Dirichlet Allocation (LDA): Overview

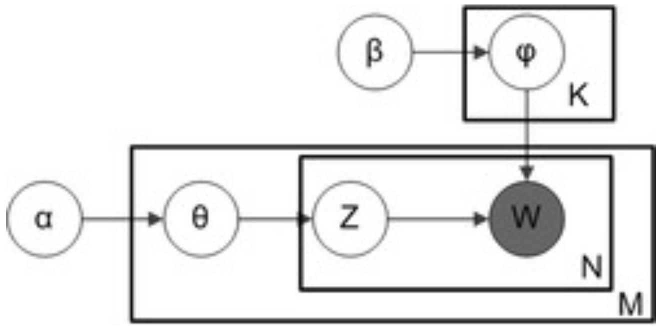
- Each document is represented as a mixture of various topics
  - Ex. A news document may be 40% on politics, 50% on economics, and 10% on sports
- Each topic is represented as a probability distribution over words
  - Ex. The distribution of “politics” vs. “sports” might be like:



- 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)

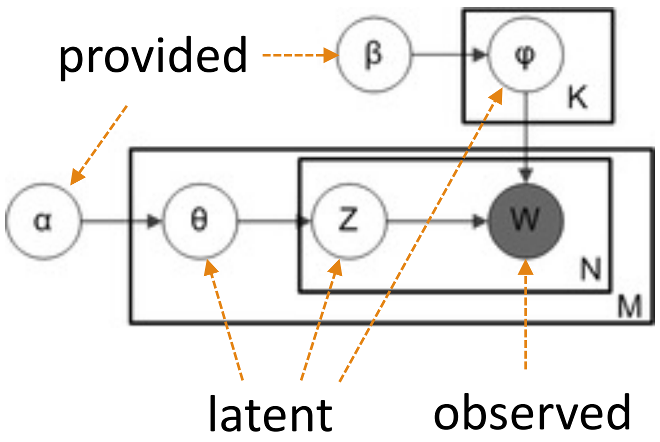
LDA: Generative Model

- 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
- Discriminative Topic Mining
- Clustering-Based Topic Discovery

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?

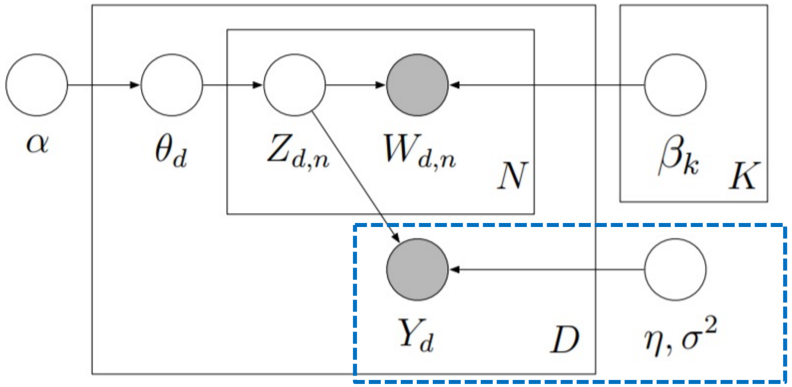
	Topic 1	Weight	Topic 2	Weight	Topic 3	Weight	Topic 4	Weight	Topic 5	Weight
0	life	0.018076	father	0.059603	official	0.017620	case	0.021908	art	0.010555
1	man	0.017714	graduate	0.048363	force	0.015388	law	0.020698	open	0.010413
2	woman	0.016657	son	0.042746	military	0.014587	court	0.019967	room	0.010363
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 $\boxed{y_i \sim N(\eta^\top \bar{z}_i, \sigma^2)}$, $\bar{z}_i = \frac{1}{L}\sum_{j=1}^{L} z_{i,j}$



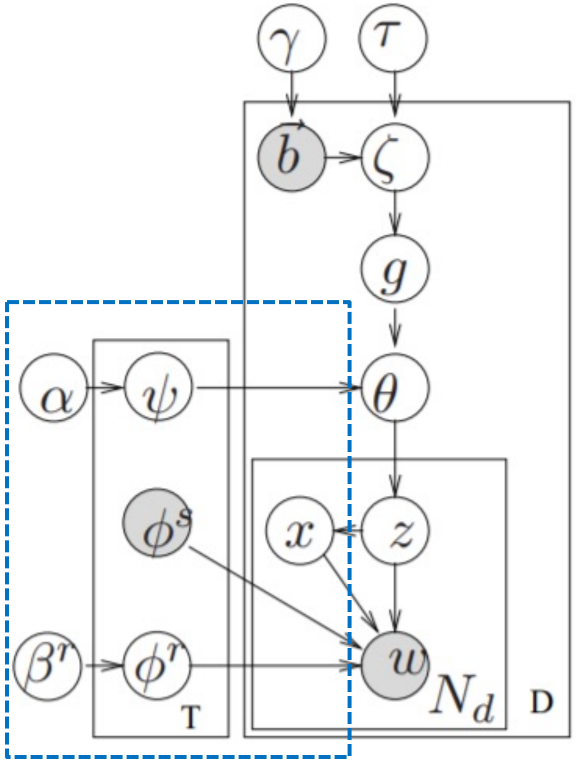
generate document's label

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 $\phi_k^r \sim \mathrm{Dir}(\beta_r)$.
   (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 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

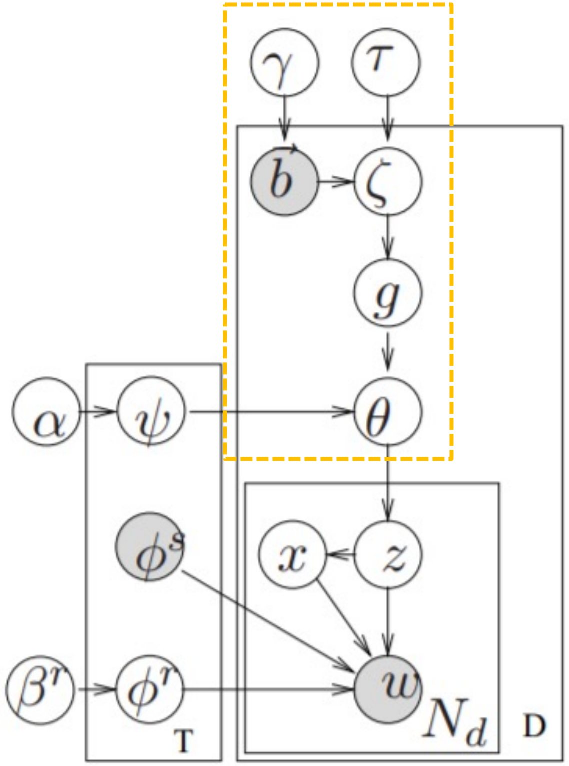


Seeded LDA: Guided Document-Topic Distribution

- 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
- Discriminative Topic Mining
  - Introduction of the Task
  - CatE: Discriminative Topic Mining via Category-Name Guided Text Embedding [WWW’20]
  - JoSH: Hierarchical Topic Mining via Joint Spherical Tree and Text Embedding [KDD’20]
- Clustering-Based Topic Discovery

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
- Discriminative Topic Mining
  - Introduction of the Task
  - CatE: Discriminative Topic Mining via Category-Name Guided Text Embedding [WWW’20]
  - JoSH: Hierarchical Topic Mining via Joint Spherical Tree and Text Embedding [KDD’20]
- Clustering-based Topic Discovery

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: 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 — 1. Topic assignment
  2. Each word w_i is generated conditioned on the semantics of the document d — 2. Global context
  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 — 3. Local context
- Likelihood of corpus generation conditioned on user-given categories

CatE Embedding: Objective

- Objective: negative log-likelihood

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

1. Topic assignment 2. Global context 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

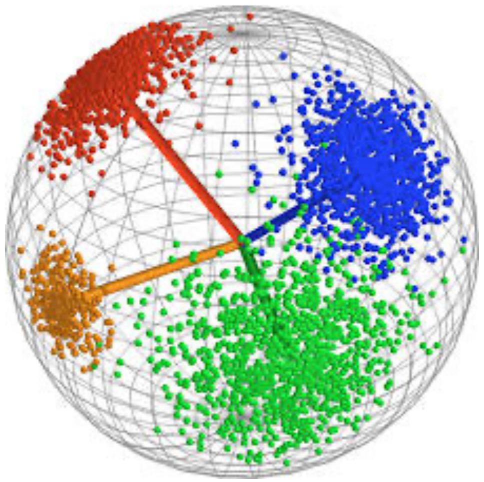
Interpreting The Model

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

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

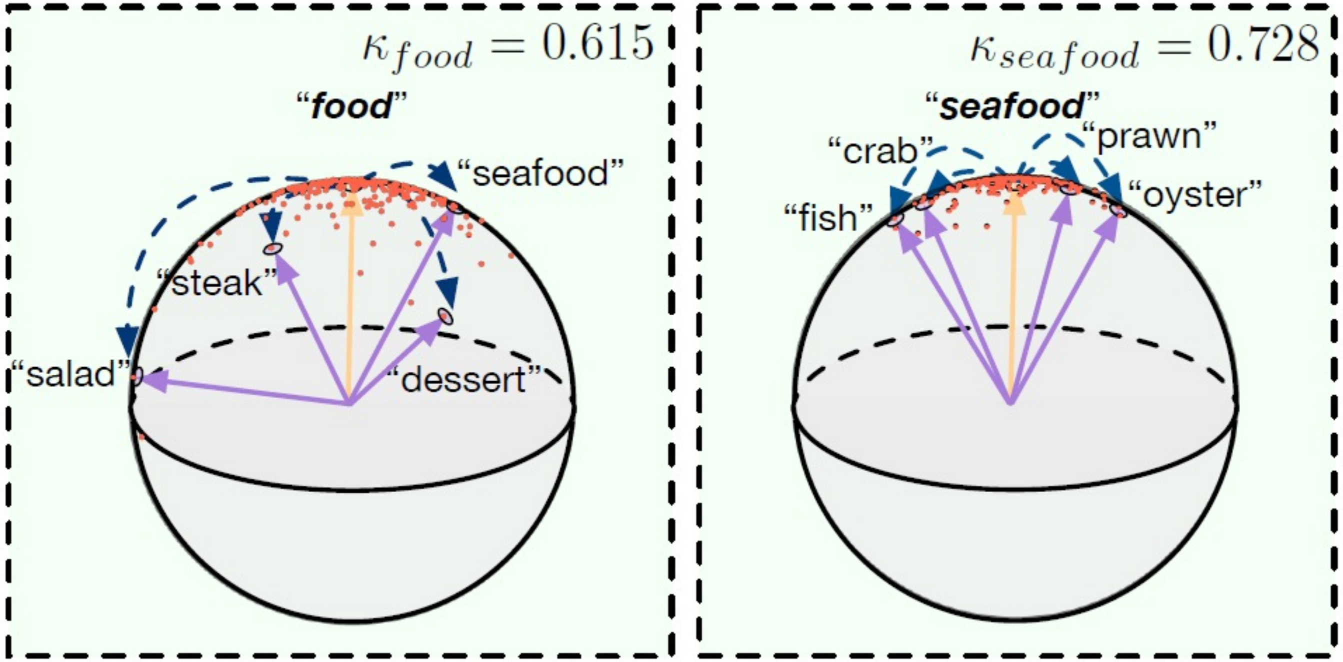
Concentration Parameter

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

Experiment Settings

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

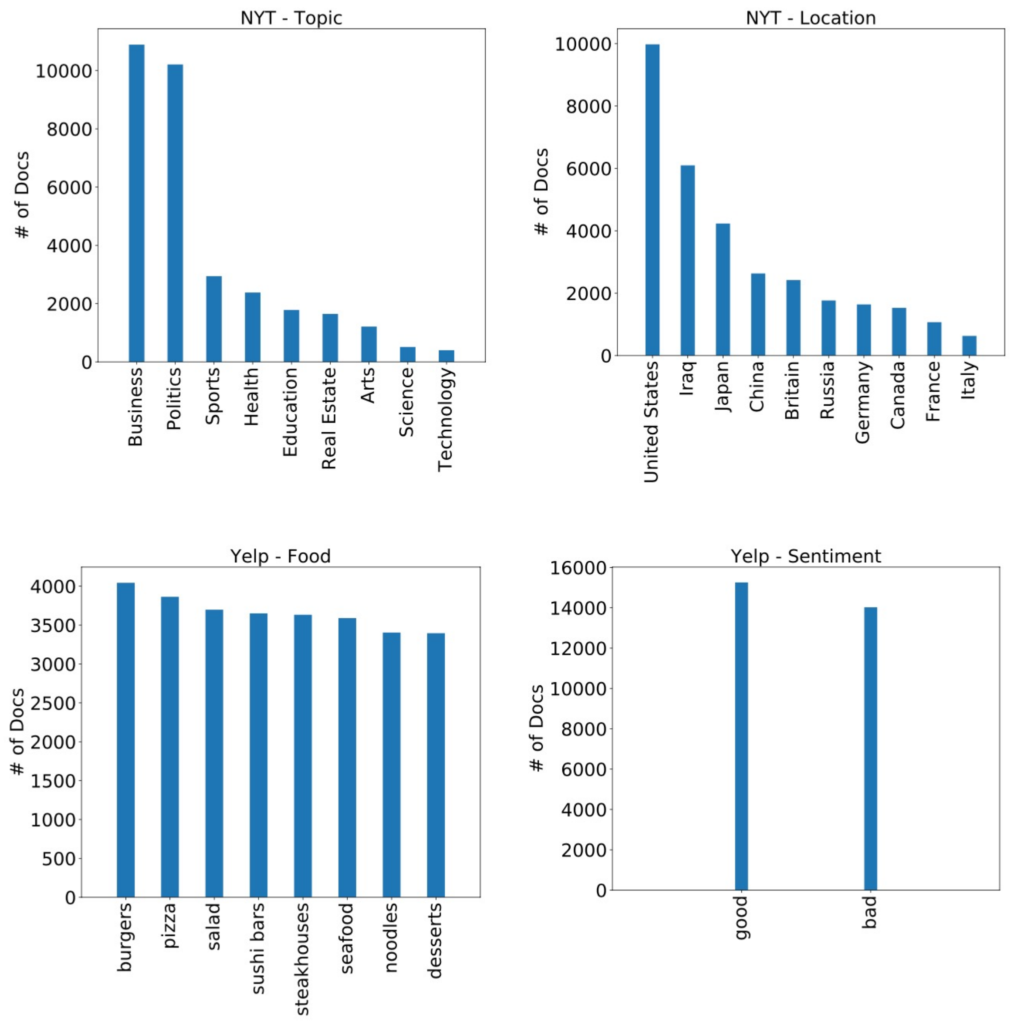


Figure 2: Dataset statistics.

Qualitative Results

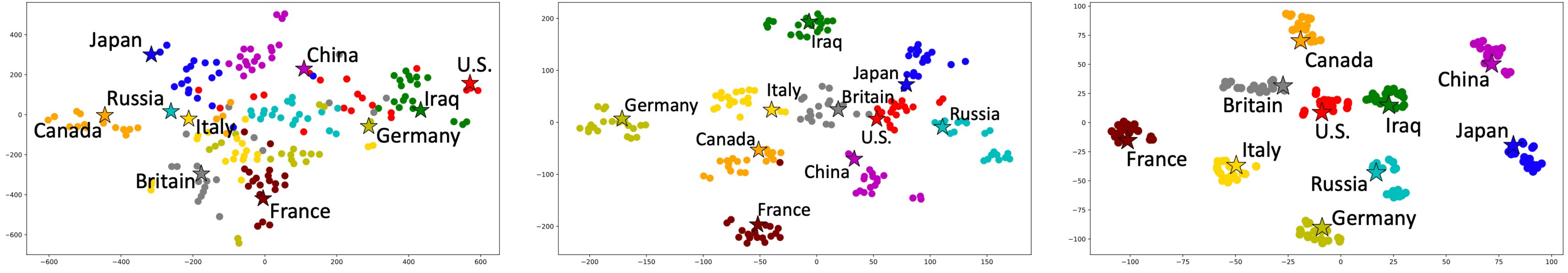
Methods	NYT-Location: britain	NYT-Location: canada	NYT-Topic: education	NYT-Topic: politics	Yelp-Food: burger	Yelp-Food: desserts	Yelp-Sentiment: good	Yelp-Sentiment: bad
LDA	company (×) companies (×) british shares (×) great britain	percent (×) economy (×) canadian united states (×) trade (×)	school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

Case Study

- Discriminative Embedding Space



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

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

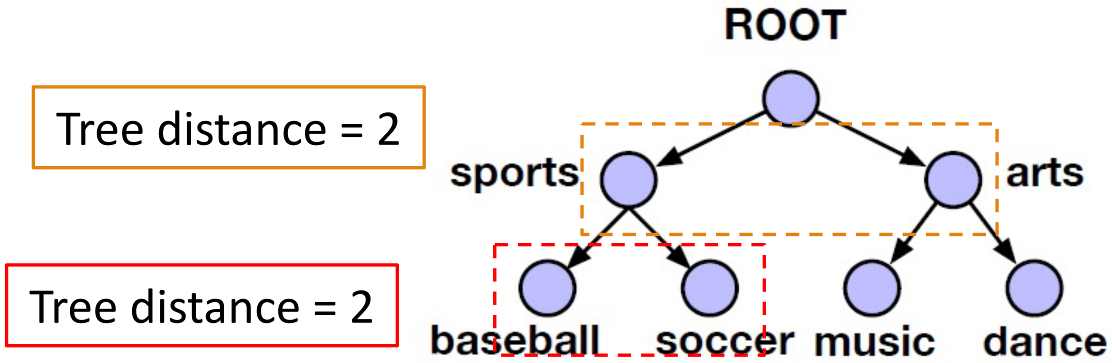
- Unsupervised Topic Modeling
- Supervised & Seed-Guided Topic Modeling
- Discriminative Topic Mining
  - Introduction of the Task
  - CatE: Discriminative Topic Mining via Category-Name Guided Text Embedding [WWW'20]
  - JoSH: Hierarchical Topic Mining via Joint Spherical Tree and Text Embedding [KDD'20]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

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

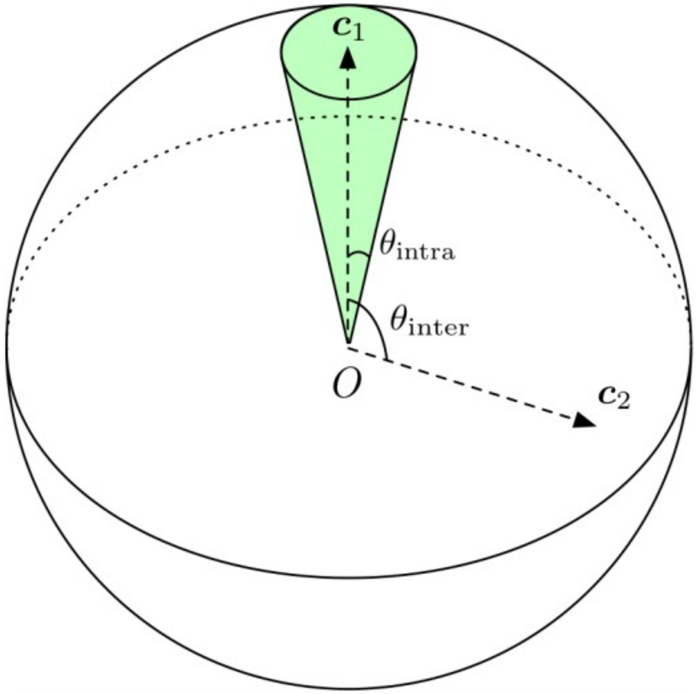
$$\mathcal{L}_{\text{intra}} = \sum_{c_i \in \mathcal{T}} \sum_{w_j \in C_i} \min(0, \boldsymbol{u}_{w_j}^\top \boldsymbol{c}_i - m_{\text{intra}}),$$

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

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

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

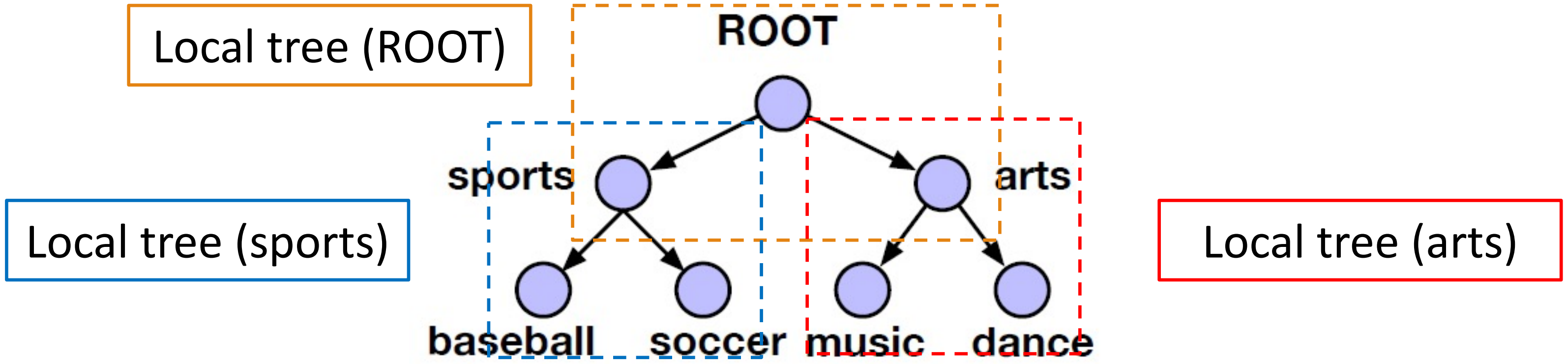
$$\theta_{\text{inter}} \ge \arccos(1 - m_{\text{inter}})$$



(a) Intra- & Inter-Category Configuration.

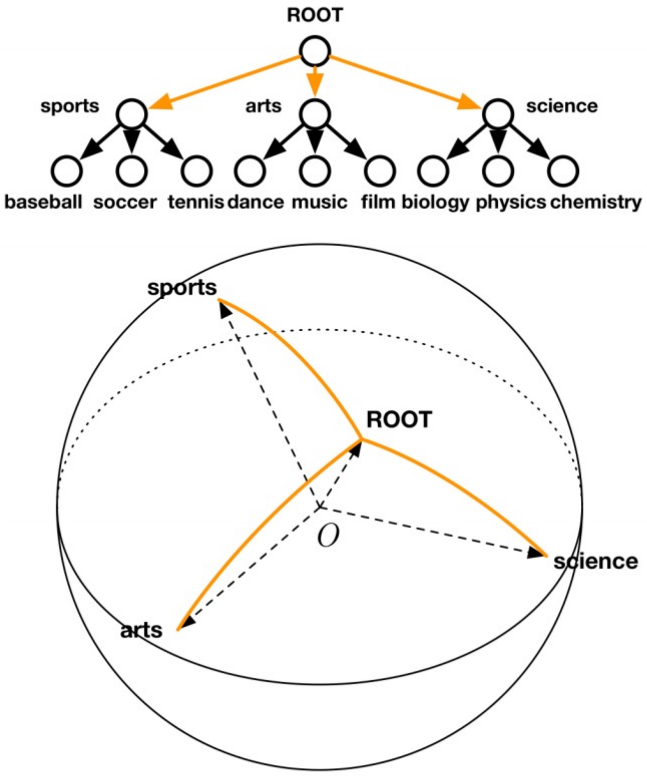
JoSH Tree Embedding

- **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

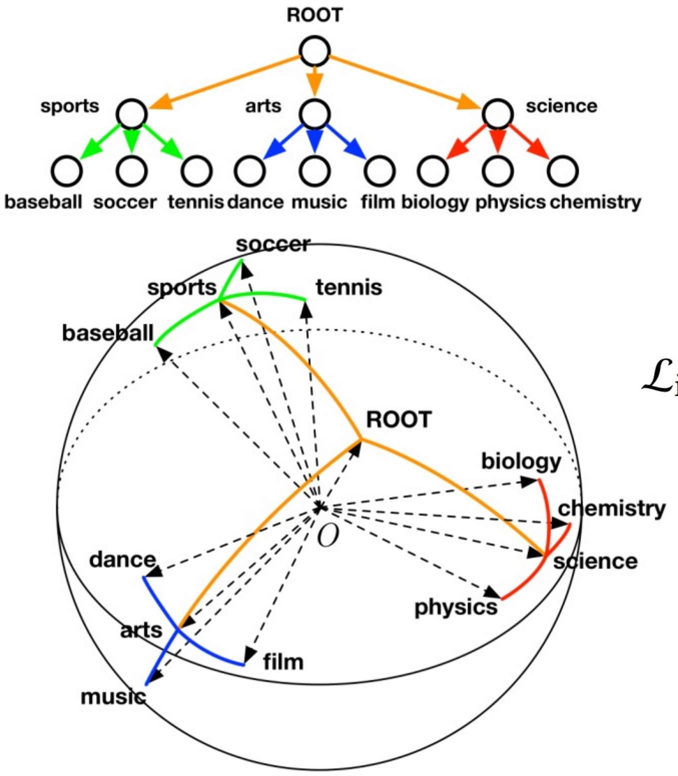


JoSH Tree Embedding

- **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

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

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

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

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

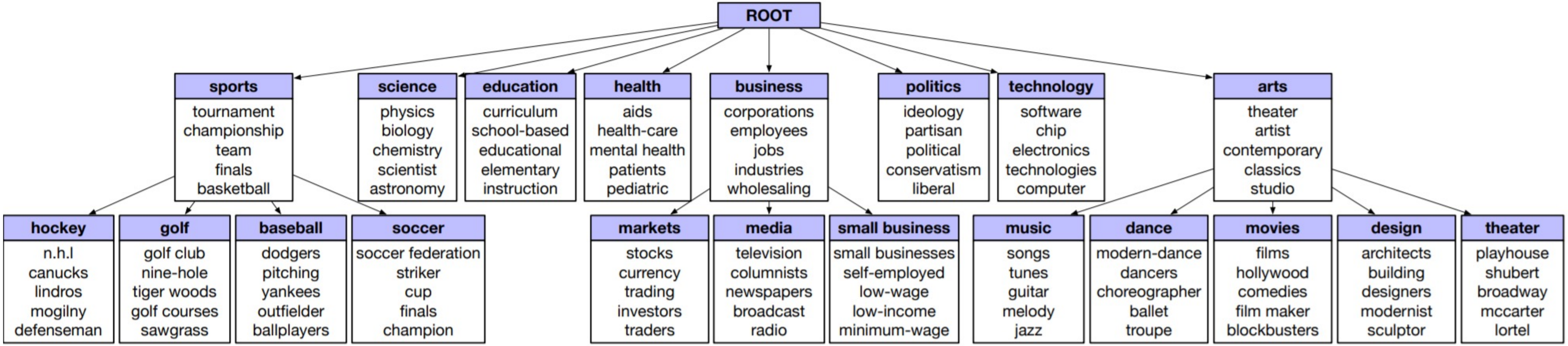
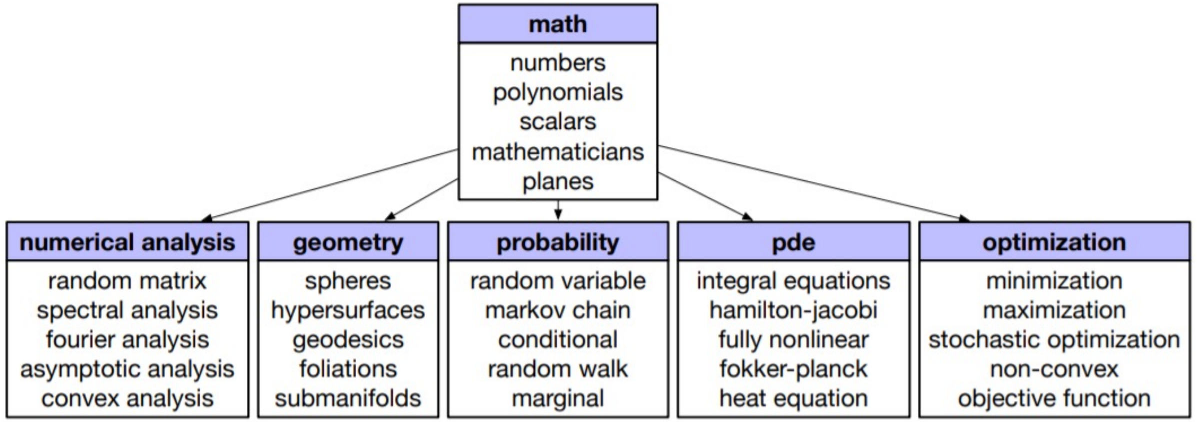
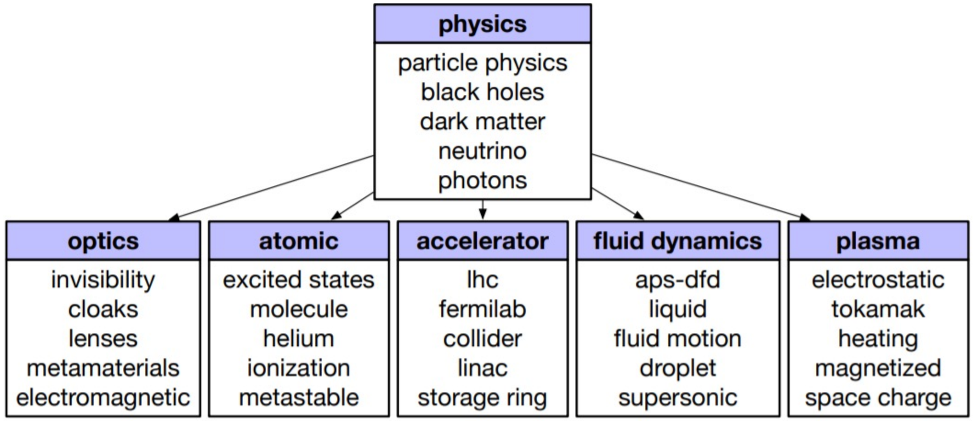


Figure 3: Hierarchical Topic Mining results on NYT.

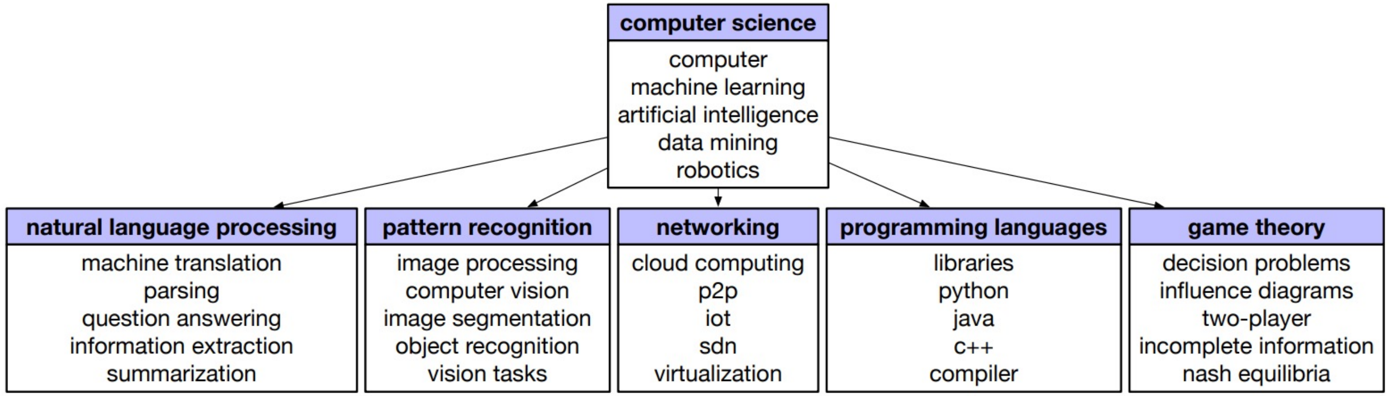
Experiments: Qualitative Results



(a) "Math" subtree.



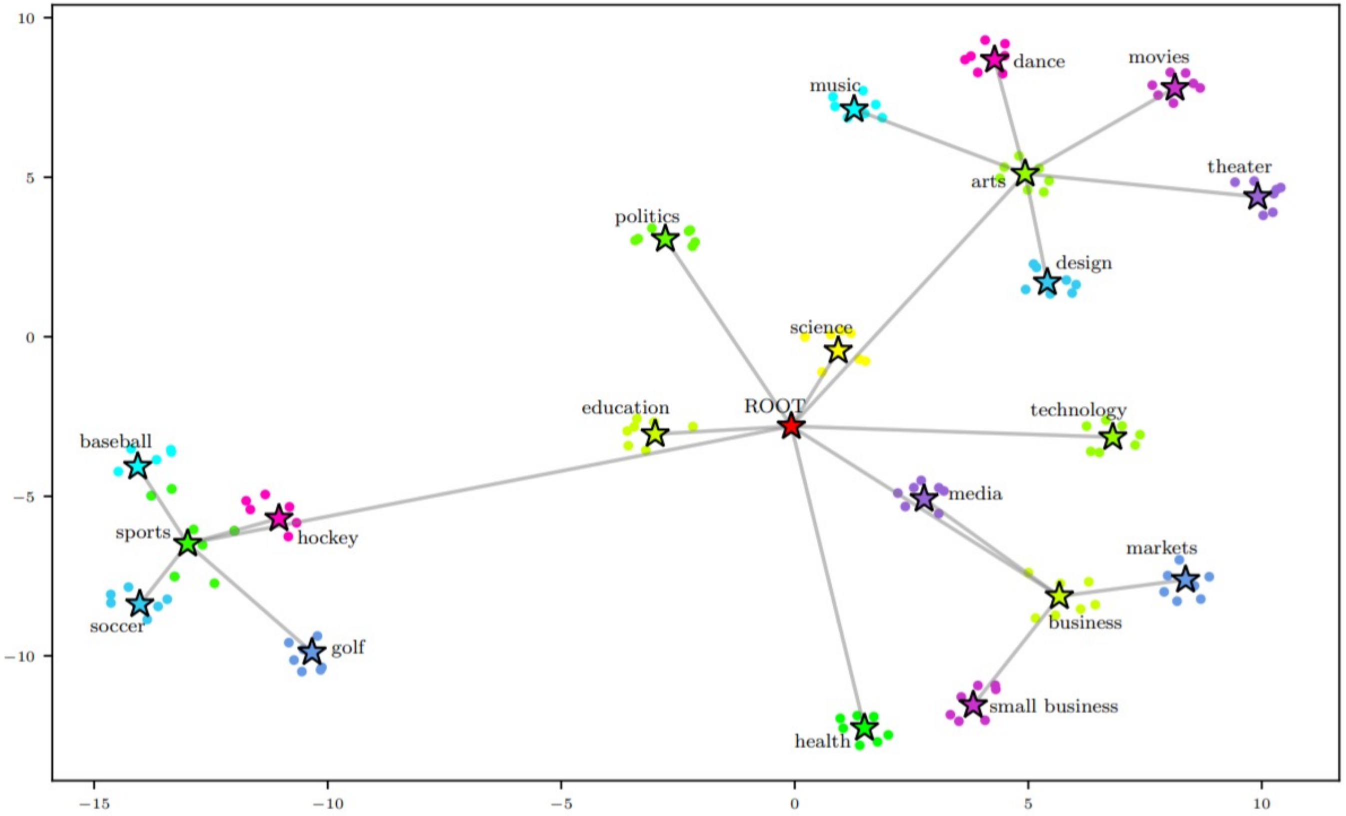
(b) "Physics" subtree.



(c) "Computer Science" subtree.

Experiments: Joint Embedding Space Visualization

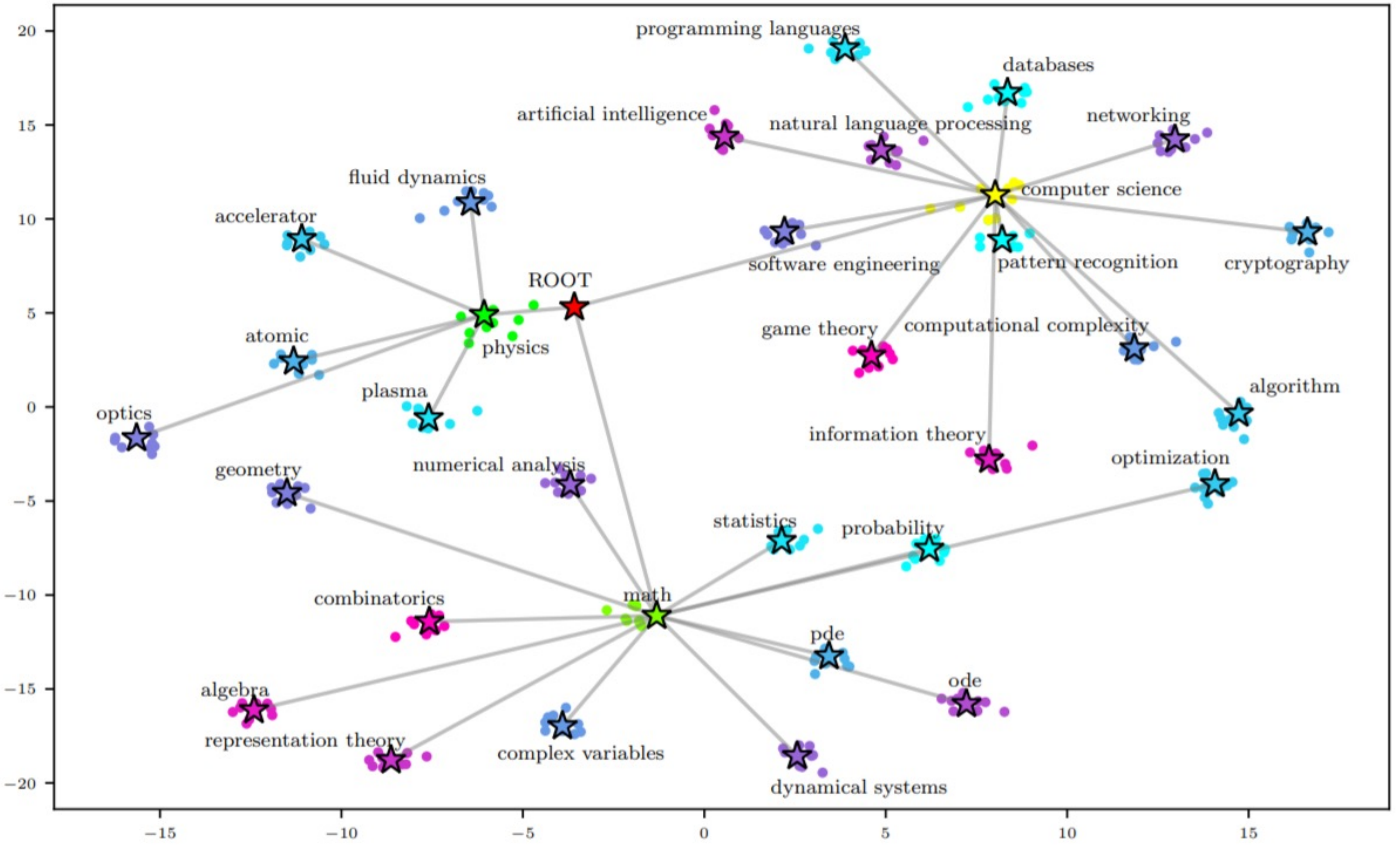
- T-SNE visualization (stars=category embeddings; dots=representative word embeddings)



(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.

Outline

- Unsupervised Topic Modeling
- Supervised & Seed-Guided Topic Modeling
- Discriminative Topic Mining
- Clustering-Based Topic Discovery
  - TopClus: Topic Discovery via Latent Space Clustering of Pretrained Language Model Representations [WWW’22]

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

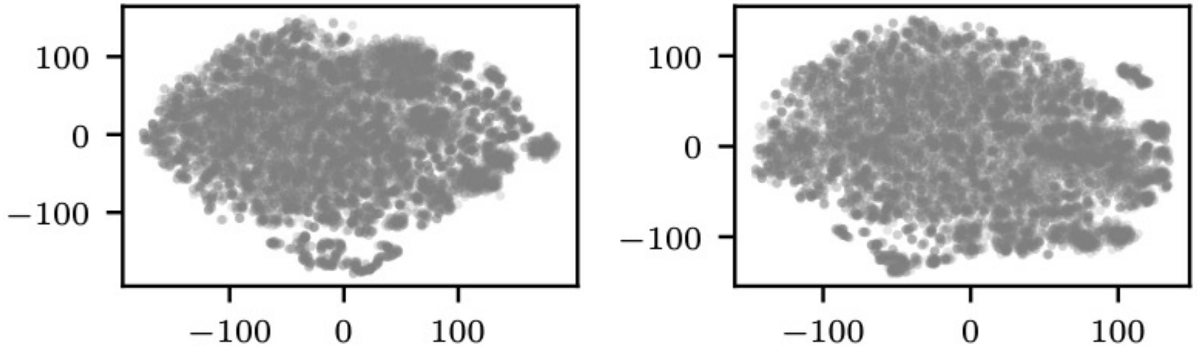
- Unsupervised Topic Modeling
- Supervised & Seed-Guided Topic Modeling
- Discriminative Topic Mining
- Clustering-Based Topic Discovery
  - TopClus: Topic Discovery via Latent Space Clustering of Pretrained Language Model Representations [WWW'22]

Motivation

- Recently, pre-trained language models (LMs) have achieved enormous success in lots of tasks
  - They employ Transformer as the backbone architecture for capturing the **long-range, high-order** semantic dependency in text sequences, yielding superior representations
  - They are pre-trained on large-scale text corpora like Wikipedia, they carry **generic linguistic features** that can be generalized to almost any text-related applications
- Given the strong representation power of the contextualized embeddings, it is natural to consider simply **clustering** them as an alternative to topic models
- Topics are essentially interpreted via clusters of semantically coherent and meaningful words
- Interestingly, such an attempt has not been reported successful yet

The Challenges

- Why not naively cluster pre-trained embeddings?
- Visualization: The embedding spaces do not exhibit clearly separated clusters
- Applying K-means with a typical K (e.g., K=100) to these spaces leads to low-quality and unstable clusters



(a) **New York Times**. (b) **Yelp Review**.

Figure 1: Visualization using t-SNE of 10,000 **randomly sampled contextualized word embeddings of BERT on (a) NYT and (b) Yelp datasets, respectively. The embedding spaces do not have clearly separated clusters.**

The Challenges

- Theoretically, such embedding space structure is due to **too many clusters**
- **Theorem**: The MLM pre-training objective of BERT assumes that the learned contextualized embeddings are generated from a Gaussian Mixture Model (GMM) with $|V|$ mixture components where $|V|$ is the vocabulary size of BERT.
- **Mismatch** between the number of clusters in the pre-trained LM embedding space and the number of topics to be discovered
  - If a smaller K ($K << |V|$) is used, the resulting partition will not fit the original data well, resulting in unstable and low-quality clusters
  - If a bigger K ($K \approx |V|$) is used, most clusters will contain only one unique term, which is meaningless for topic discovery

The Latent Space Model

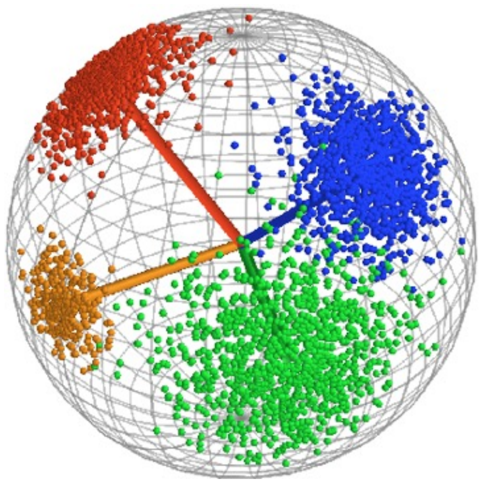
- We propose to project the original embedding space into a latent space with K clusters of words corresponding to K latent topics
- We assume that the latent space is **lower-dimensional** and **spherical**, with the following preferable properties:
  - **Spherical latent space** employs angular similarity between vectors to capture word semantic correlations, which works better than Euclidean metrics
  - **Lower-dimensional space** mitigates the "curse of dimensionality"
  - Projection from high-dimension to lower-dimension space forces the model to discard the information that is not helpful for forming topic clusters (e.g., syntactic features, "play", "plays" and "playing" should not represent different topics)

Latent Topic Space

- We propose a generative model for the joint learning

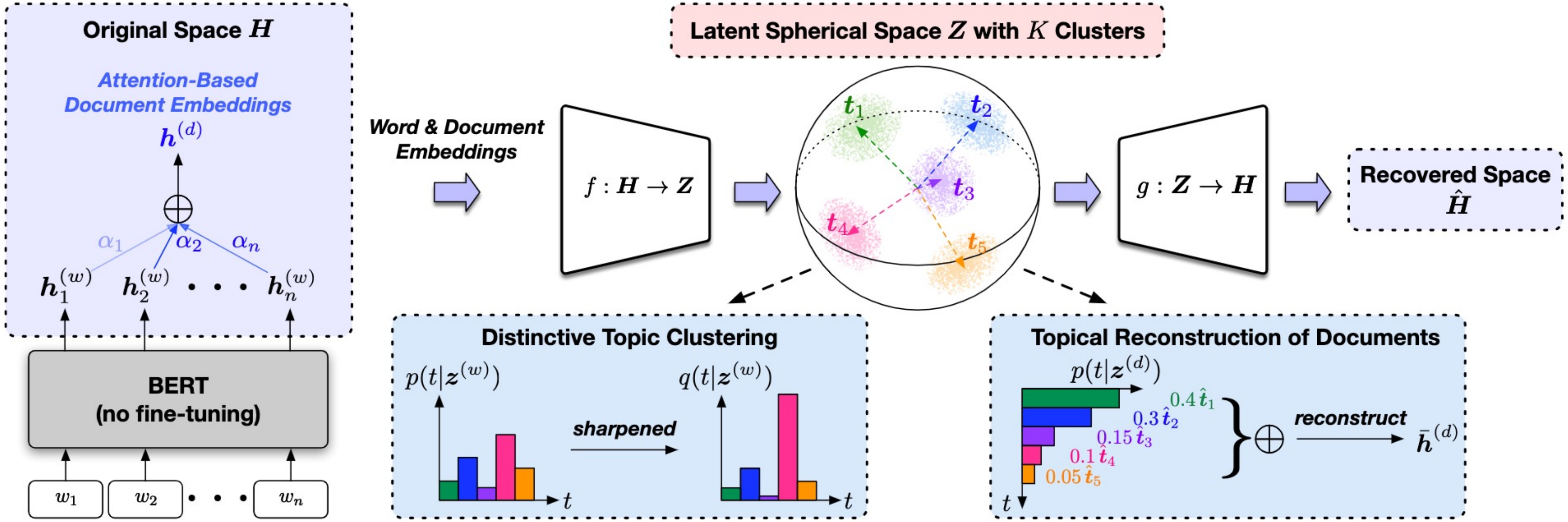
$$t_k \sim \mathrm{Uniform}(K),\ \boldsymbol{z}_i \sim \mathrm{vMF}_{d'}(\boldsymbol{t}_k, \kappa),\ \boldsymbol{h}_i = g(\boldsymbol{z}_i).$$

- A topic t is sampled from a uniform distribution over the K topics
- A latent embedding z is generated from the vMF distribution associated with topic t
- A function g maps the latent embedding z to the original embedding



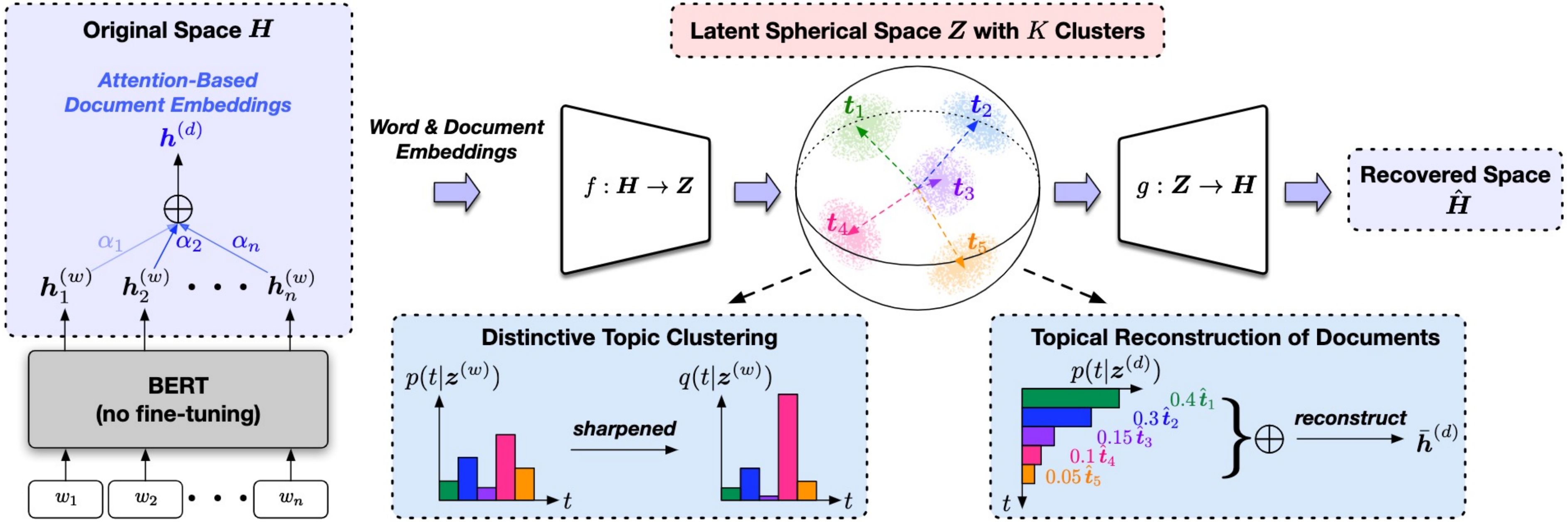
The Latent Space Model

- We propose to **jointly** learn the latent space projection and cluster in the latent space
  - The latent representation learning is guided by the clustering objective
  - The cluster quality benefits from the well-separated structure of the latent space
  - Achieve a mutually-enhanced effect



The Latent Space Model

- How to train the generative model?
  - A preservation loss that encourages the latent space to preserve the semantics of the original pre-trained LM induced embedding space **(preservation of original PLM embeddings)**
  - A reconstruction loss to ensure the learned latent topics are meaningful summaries of the documents **(Topic reconstruction of documents)**
  - A clustering loss that enforces separable cluster structures in the latent space for distinctive topic learning **(clustering)**



Preservation of Original PLM Embeddings

- Motivated by the general idea of generative model training that optimizes the model to **faithfully generate** the original data
- We encourage the output of the autoencoder to recover the structure of the original embedding space by minimizing the cosine distance between the generated and the original embedding

$$\mathcal{L}_{\text{pre}} = \sum_{i=1}^{N} \|\boldsymbol{h}_i^{(w)} - g\left(f\left(\boldsymbol{h}_i^{(w)}\right)\right)\|^2$$

Topic Reconstruction of Documents

- We aim to reconstruct document semantics with topic representations so that the learned latent topics are meaningful summaries of the documents.
- We require the reconstructed document embedding to be a good approximation of the original content by minimizing the following reconstruction loss:

$$\mathcal{L}_{\mathrm{rec}} = \sum_{d \in \mathcal{D}} \left\| \boxed{\hat{\boldsymbol{h}}^{(d)}} - \boxed{\bar{\boldsymbol{h}}^{(d)}} \right\|^2$$

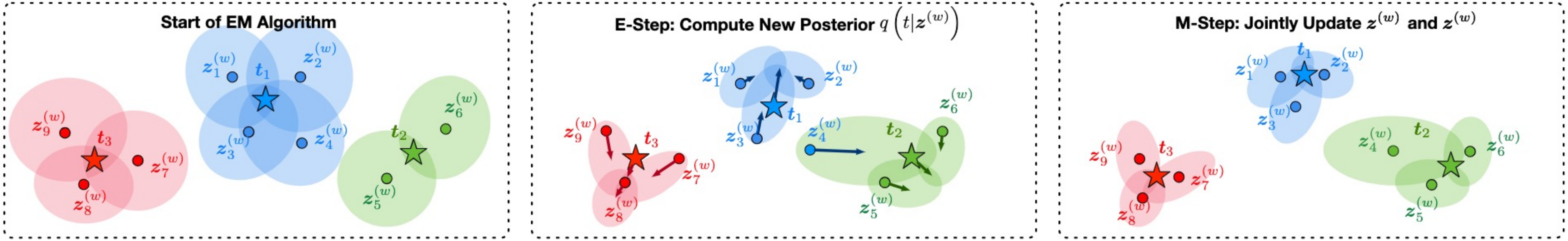
reconstructed document embedding

$$\boxed{\hat{\boldsymbol{h}}^{(d)} = \sum_{k=1}^{K} p\left(t_k \middle| \boldsymbol{z}^{(d)}\right) \hat{\boldsymbol{t}}_k, \quad \hat{\boldsymbol{t}}_k = g(\boldsymbol{t}_k),}$$

average of original word embeddings in the document

The Clustering Loss

- An EM algorithm, analogous to K-means
  - The E-step estimates a new cluster assignment of each word based on the current parameters
  - The M-step updates the model parameters given the cluster assignments



(a) Start of EM Algorithm. (b) E-Step. (c) M-Step.

Clustering EM

- E-step:
- Use the current posterior to derive a new posterior as the new cluster assignment

$$p(t_k|\boldsymbol{z}_i) = \frac{p(\boldsymbol{z}_i|t_k)p(t_k)}{\sum_{1 \le k' \le K} p(\boldsymbol{z}_i|t_{k'})p(t_{k'})}$$

$$t_k \sim \text{Uniform}(K),\ \boldsymbol{z}_i \sim \text{vMF}_{d'}(\boldsymbol{t}_k, \kappa),\ \boldsymbol{h}_i = g(\boldsymbol{z}_i).$$

$$p(t_k) = 1/K$$

$$p(\boldsymbol{z}_i|t_k) = \text{vMF}_{d'}(\boldsymbol{t}_k, \kappa) = n_{d'}(\kappa) \exp\left(\kappa \cdot \cos(\boldsymbol{z}_i, \boldsymbol{t}_k)\right)$$

$$p(t_k|\boldsymbol{z}_i) = \frac{\exp\left(\kappa \cdot \cos(\boldsymbol{z}_i, \boldsymbol{t}_k)\right)}{\sum_{1 \le k' \le K} \exp\left(\kappa \cdot \cos(\boldsymbol{z}_i, \boldsymbol{t}_{k'})\right)}$$

Clustering EM

- E-step:
- Use the current posterior to derive a new posterior as the new cluster assignment

$$p(t_k|z_i) = \frac{p(z_i|t_k)p(t_k)}{\sum_{1 \le k' \le K} p(z_i|t_{k'})p(t_{k'})}$$

$$\Downarrow$$

$$q(t_k|z_i) = \frac{p(t_k|z_i)^2/s_k}{\sum_{1 \le k' \le K} p(t_{k'}|z_i)^2/s_{k'}}, \quad s_k = \sum_{1 \le i \le N} p(t_k|z_i).$$

- Such a new posterior has the following advantages:
  - Distinctive topic learning: Squaring-then-normalizing the current posterior distribution has a **sharpening** effect that skews the distribution towards its most confident cluster assignment
  - Topic prior regularization: Dividing by the soft cluster frequency s_k encodes the uniform topic prior

Clustering EM

- M-step:
- Update the model parameters according to the new cluster assignment

$$\mathcal{L}_{\text{clus}} = -\sum_{1\le i\le N}\sum_{1\le k\le K} q(t_k|z_i)\log p(t_k|z_i),$$

- Both the topic center vectors and latent representations are updated to fit the new estimate
- This is the joint learning of latent space mapping functions and cluster structures

Experiments

- Topic Discovery

Quantitative

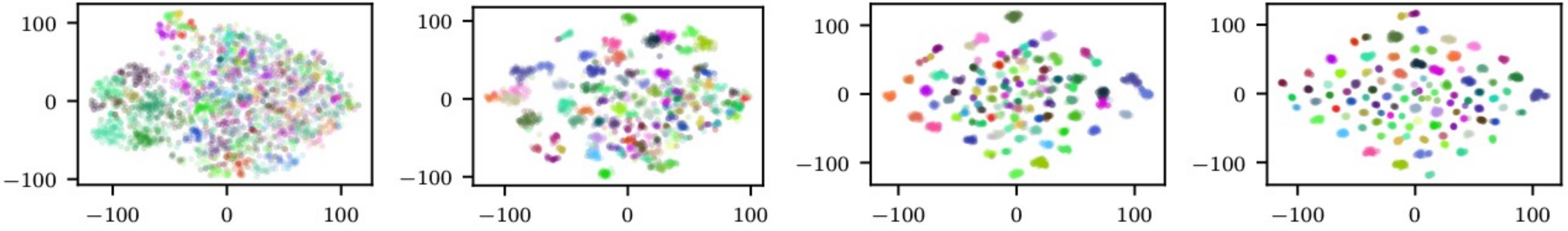
Methods	NYT UMass	NYT UCI	NYT Int.	NYT Div.	Yelp UMass	Yelp UCI	Yelp Int.	Yelp Div.
LDA	-3.75	-1.76	0.53	0.78	-4.71	-2.47	0.47	0.65
CorEx	-3.83	-0.96	0.77	-	-4.75	-1.91	0.43	-
ETM	-2.98	-0.98	0.67	0.30	-3.04	-0.33	0.47	0.16
BERTopic	-3.78	-0.51	0.70	0.61	-6.37	-2.05	0.73	0.36
TopClus	**-2.67**	**-0.45**	**0.93**	**0.99**	**-1.35**	**-0.27**	**0.87**	**0.96**

Qualitative

Methods	NYT Topic 1 (sports)	NYT Topic 2 (politics)	NYT Topic 3 (research)	NYT Topic 4 (france)	NYT Topic 5 (japan)	Yelp Topic 1 (positive)	Yelp Topic 2 (negative)	Yelp Topic 3 (vegetables)	Yelp Topic 4 (fruits)	Yelp Topic 5 (seafood)
LDA	olympic	*mr*	*said*	french	japanese	amazing	loud	spinach	mango	fish
	year	bush	report	*union*	tokyo	*really*	awful	carrots	strawberry	*roll*
	said	president	evidence	*germany*	*year*	*place*	*sunday*	greens	*vanilla*	salmon
	games	white	findings	*workers*	matsui	phenomenal	*like*	salad	banana	*fresh*
	team	house	defense	paris	*said*	pleasant	slow	*dressing*	*peanut*	*good*
CorEx	baseball	house	possibility	french	japanese	great	*even*	garlic	strawberry	shrimp
	championship	white	challenge	*italy*	tokyo	friendly	bad	tomato	*caramel*	*beef*
	playing	support	reasons	paris	*index*	*atmosphere*	mean	onions	*sugar*	crab
	fans	*groups*	*give*	francs	osaka	love	cold	*toppings*	fruit	*dishes*
	league	*member*	planned	jacques	*electronics*	favorite	*literally*	*slices*	mango	*salt*
ETM	olympic	government	approach	french	japanese	nice	disappointed	avocado	strawberry	fish
	league	national	problems	*students*	*agreement*	worth	cold	*greek*	mango	shrimp
	national	*plan*	experts	paris	tokyo	*lunch*	*review*	salads	*sweet*	lobster
	basketball	public	*move*	*german*	*market*	recommend	*experience*	spinach	*soft*	crab
	athletes	support	*give*	*american*	*european*	friendly	bad	tomatoes	*flavors*	*chips*
BERTopic	swimming	bush	researchers	french	japanese	awesome	horrible	tomatoes	strawberry	lobster
	freestyle	democrats	scientists	paris	tokyo	*atmosphere*	*quality*	avocado	mango	crab
	popov	white	cases	lyon	ufj	friendly	disgusting	*soups*	*cup*	shrimp
	gold	bushs	*genetic*	*minister*	*company*	*night*	disappointing	kale	lemon	oysters
	olympic	house	study	*billion*	yen	good	*place*	cauliflower	banana	*amazing*
TopClus	athletes	government	hypothesis	french	japanese	good	tough	potatoes	strawberry	fish
	medalist	ministry	methodology	seine	tokyo	best	bad	onions	lemon	octopus
	olympics	bureaucracy	possibility	toulouse	osaka	friendly	painful	tomatoes	apples	shrimp
	tournaments	politicians	criteria	marseille	hokkaido	cozy	frustrating	cabbage	grape	lobster
	quarterfinal	electoral	assumptions	paris	yokohama	casual	brutal	mushrooms	peach	crab

Experiments

- Visualization



(a) Epoch 0. (b) Epoch 2. (c) Epoch 4. (d) Epoch 8.

Figure 5: Visualization using t-SNE of 10, 000 randomly sampled latent embeddings during the course of TopClus training. Embeddings assigned to the same cluster are denoted with the same color. The latent space gradually exhibits distinctive and balanced cluster structure.

Advantages of TopClus over topic models

- TopClus works with contextualized embeddings which provide better word representations than the “bag-of-words” assumption of topic models
- TopClus employs pre-trained LMs to bring in general linguistic knowledge which helps generate more reliable and stable word representations on the target corpus than training topic models from scratch on it
- TopClus does not involve any probabilistic approximations, and is computationally and conceptually simpler than variational inference in topic models

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.
- Meng, Y., Zhang, Y., Huang, J., Zhang, Y., & Han, J. (2022). Topic Discovery via Latent Space Clustering of Pretrained Language Model Representations. WWW.
- 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.