Yale

Document Graph Representation Learning: A Topic Modeling Perspective

Delvin Ce Zhang

Yale University

Textual Documents are Ubiquitous

- Ubiquity of unstructured textual documents
  - Unstructured, noisy, dynamic, multi-lingual, …

Today was a great day for your restaurant. There was a huge buffet with important guests. It seems that your team coped perfectly, after all, the customers left satisfied. Now it only remains to wash the dishes and prepare for the next day. Previously, the sheer amount of dirty dishes would have caused problems.

But not now, now you have Sun Eco extra power dishwasher tablets. They save time and money, because you only need one tablet (one pack contains 175 pieces). Moreover, do not worry about having to clean the machine after - as the tablet and it's plastic wrap dissolve completely during the washing process.

After use, all that remains is to admire the result: all the dirt on the dishes, including soot on the pans, oil, grease and food residues will simply disappear. You will be left with just sparklingly clean dishes without marks and stains. Even pre-rinsing is not required, which saves water and reduces company expenses. In addition, the composition does not include phosphates, so there are no harmful substances or tastes left on the dishes

Use a secret weapon against massive loads of dirty dishes!

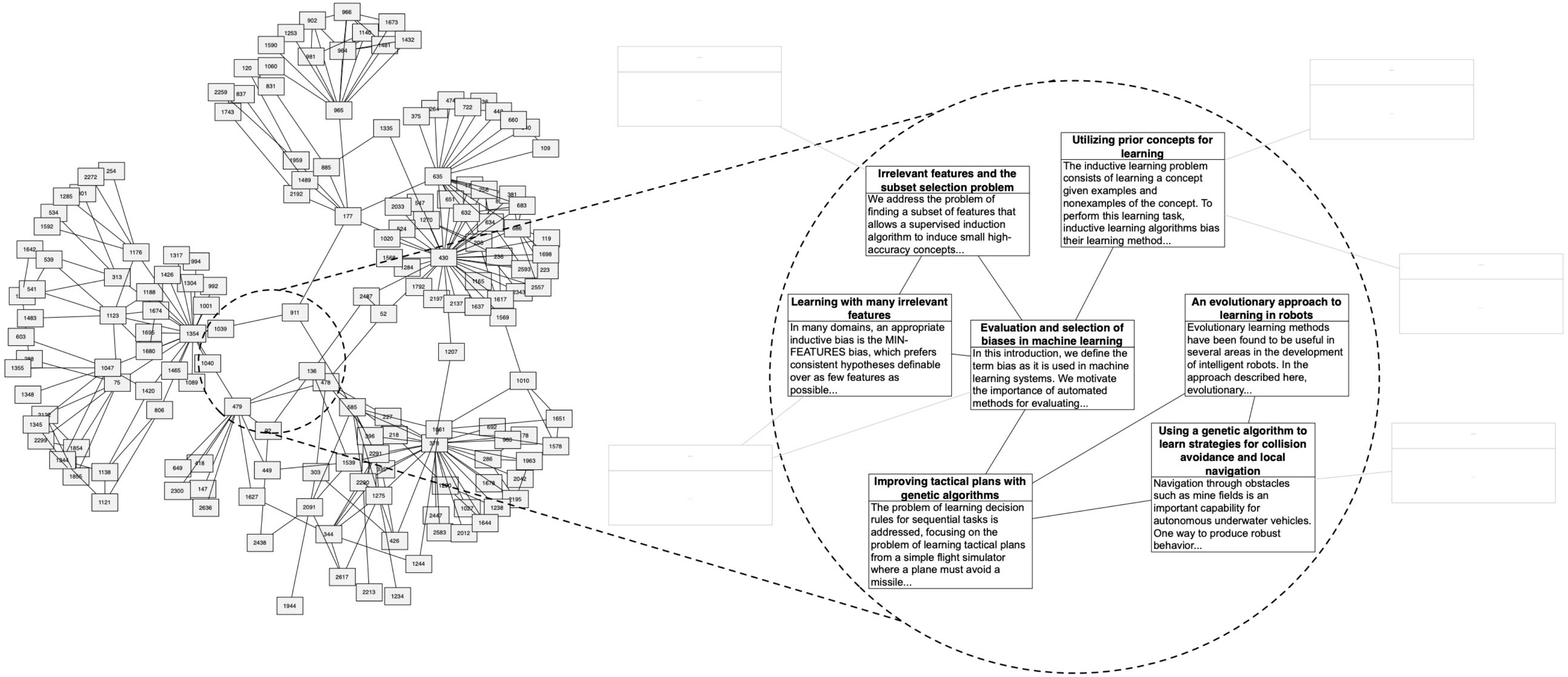
News articles

Academic papers

Product descriptions

Document Graph

- Documents are also interconnected in a graph structure – **Document Graph**
  - Academic citation graph, news article hyperlink graph, product contextual graph…
  - Consisting of a corpus of documents $\mathcal{D}$, and graph structure $\mathcal{E}$.



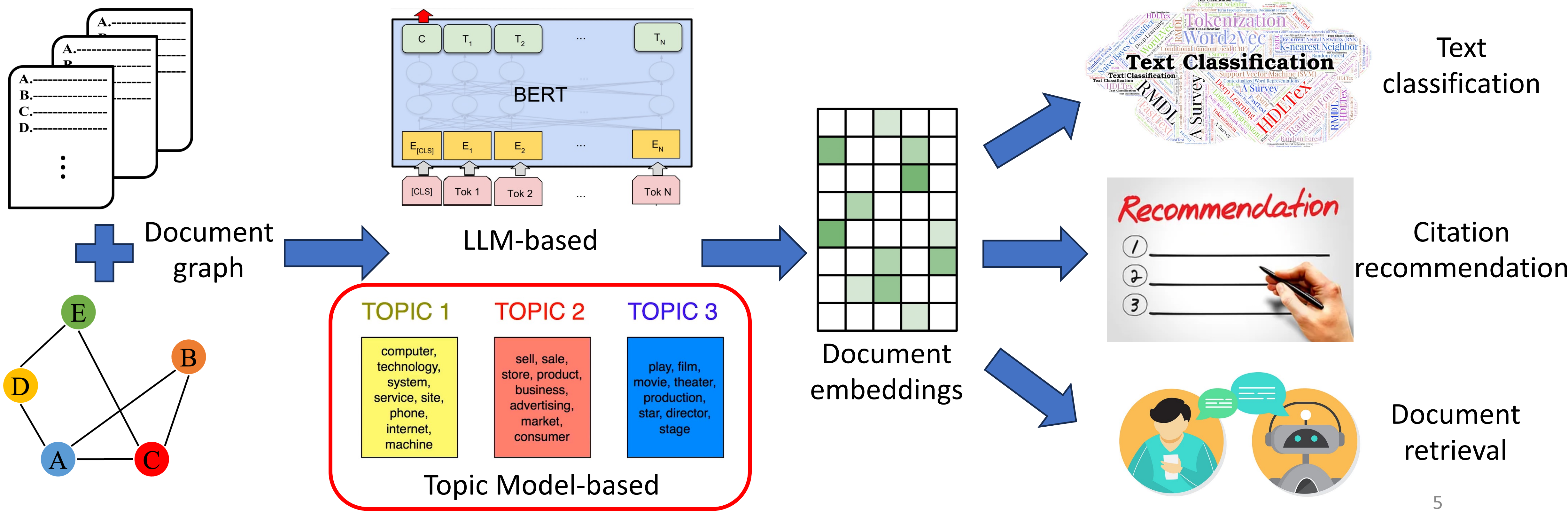
How to Process Document Graph

- Existing works

	Pros	Cons
Graph Neural Networks (GNNs)	Capture vertex attribute and graph structure	No language representations or linguistic semantics
Large Language Models (LLMs)	Learn contextualized document representation	No document graph structure
Topic Models (TMs)	Infer topic representation for documents	No document graph structure

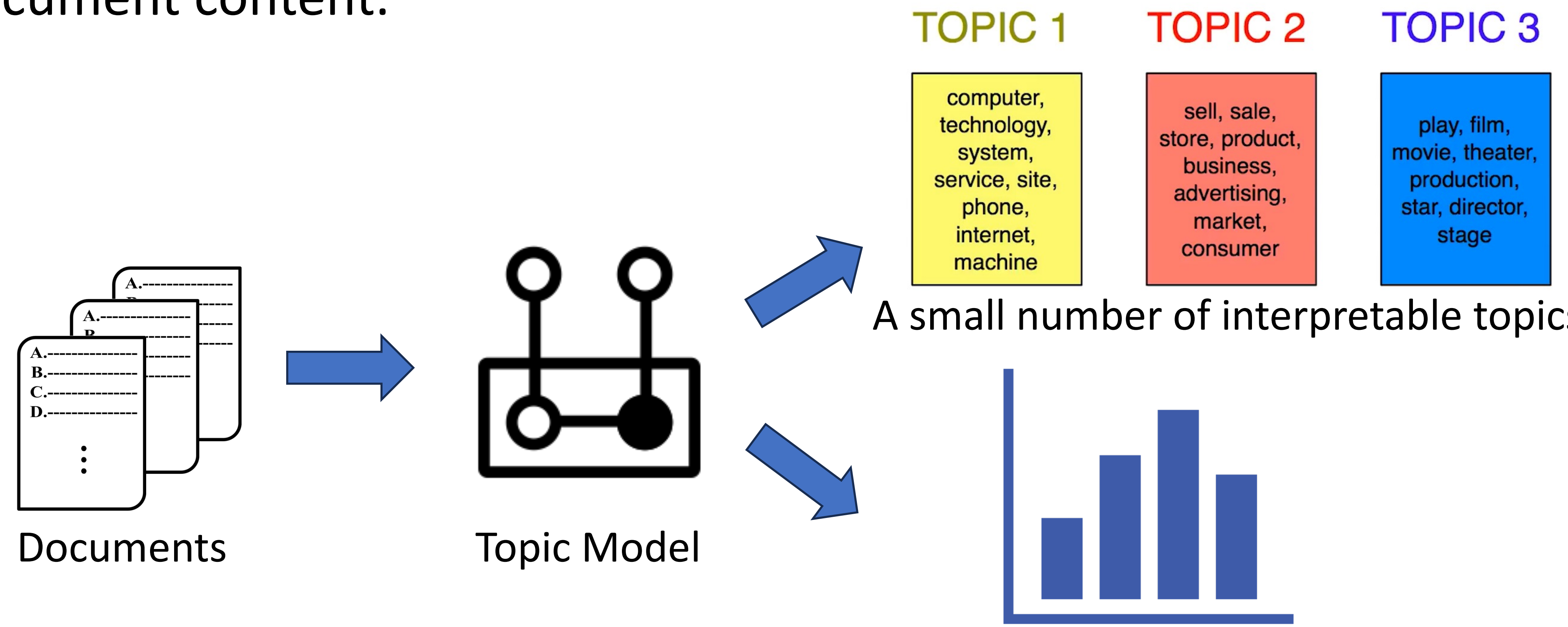
How to Process Document Graph

- This tutorial – **Document Graph Representation Learning**
  - Infer document embeddings that preserve **both i)** contextualized semantics contained in rich text documents, **and ii)** graph connectivity across documents.



Overview of Topic Modeling

- Topic Models (TMs) assume a small number of **latent topics** to generate document content.



Topic distribution
(Document representation)

Topic Modeling on Document Networks with Dirichlet Optimal Transport Barycenter

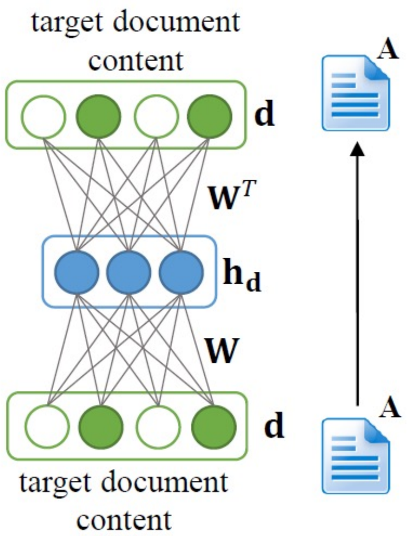
Delvin Ce Zhang[1] and Hady W. Lauw[2]

[1]Yale University, [2]Singapore Management University

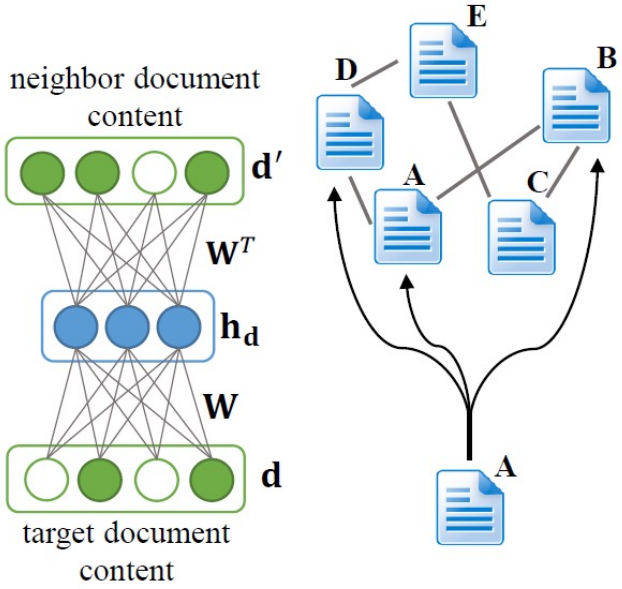
IEEE Transactions on Knowledge Discovery and Data Engineering (TKDE-23)

Model Architecture

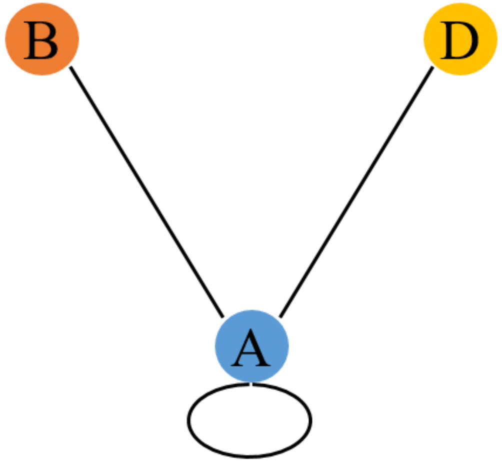
- Overall framework: generate both input document and its neighbors.



(a) Existing topic models



(b) Our proposed idea



(c) Simplified diagram

Model Architecture

- 1. GNN encoding

$$\mathbf{h}_i = GNN(i, N(i))$$

- 2. Dirichlet sampling

$$\boldsymbol{\alpha}_i = \max(10^{-12}, \mathrm{softplus}(\mathbf{h}_i))$$

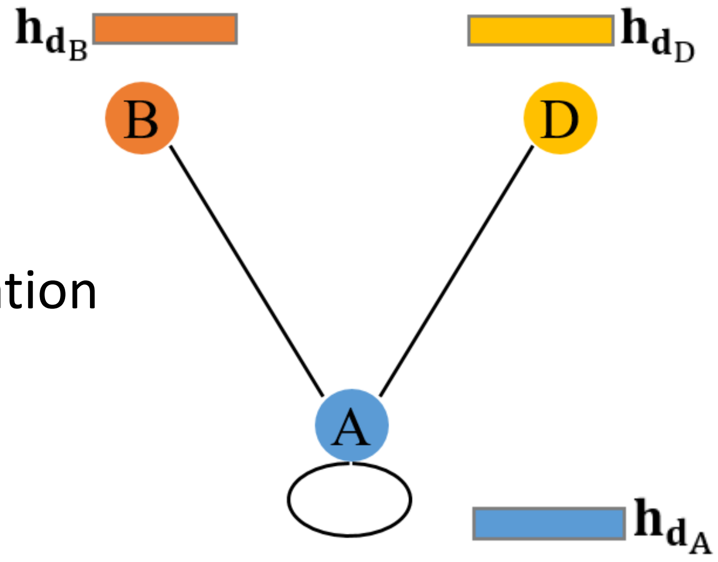
$$\mathbf{z}_i \sim \mathrm{Dir}(\boldsymbol{\alpha}_i)$$

$\mathbf{z}_i$ is K-dimensional document representation

- 3. Dirichlet reparameterization

$$z_{i,k} \sim \Gamma(\alpha_{i,k}) \longleftarrow \text{Gamma distribution}$$

$$\mathbf{z}_i = \left[\frac{z_{i,k}}{\sum_{k'=1}^{K} z_{i,k'}}, \ldots, \frac{z_{i,K}}{\sum_{k'=1}^{K} z_{i,k'}} \right] \sim \mathrm{Dir}(\boldsymbol{\alpha}_i)$$

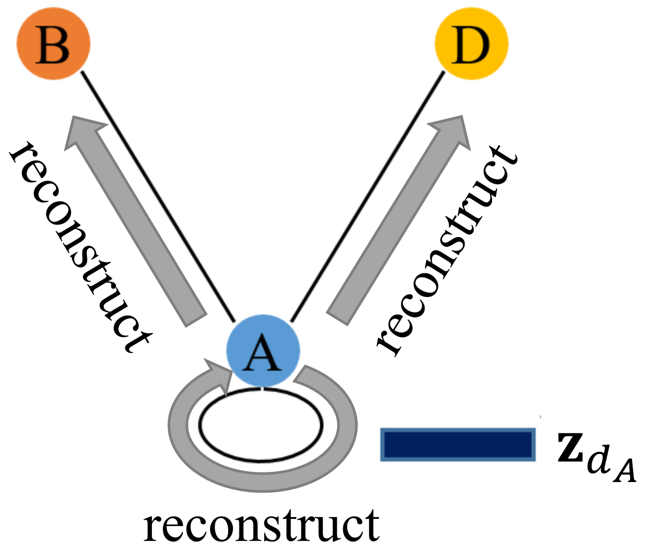


Model Architecture

- 4. Neighbor generation with Optimal Transport

$$\min \sum_{j \in \mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j)$$

Optimal Transport distance

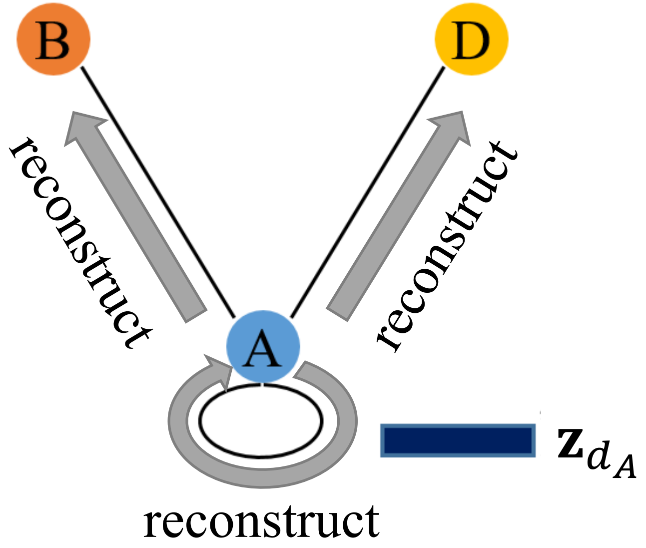


Model Architecture

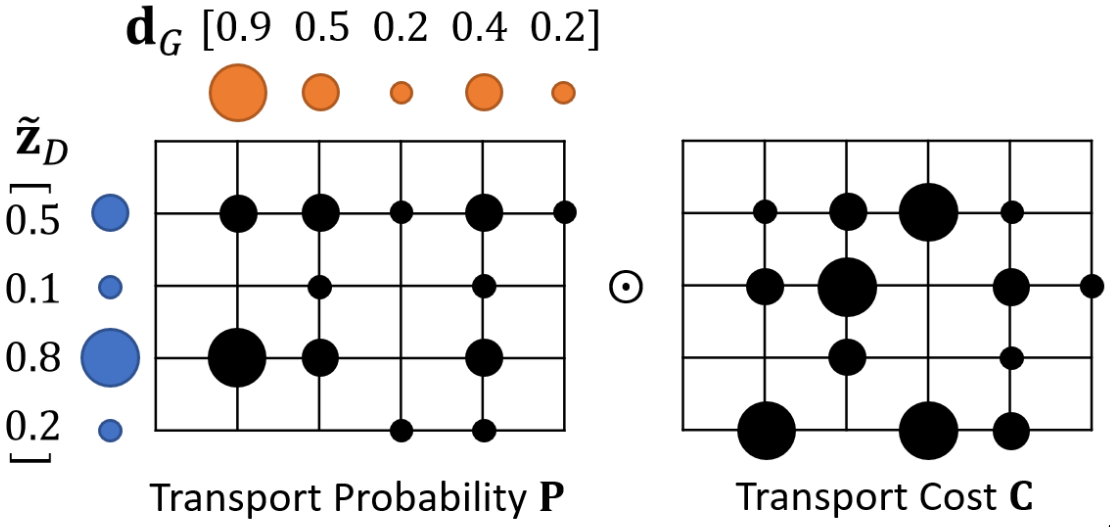
- 4. Neighbor generation with Optimal Transport

$$\min \sum_{j \in \mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j)$$

Optimal Transport distance



- 5. Design of Optimal Transport



$$= \min_{\mathbf{P}} \sum_{i=1}^{K} \sum_{j=1}^{|\mathcal{V}|} p_{ij} c_{ij} = \mu(\tilde{\mathbf{z}}_D, \mathbf{d}_G)$$

Optimal Transport distance

Trainable topic embedding

Cost is defined by $c_{ij} = 1 - \cos(\mathbf{g}_i, \mathbf{e}_j)$

Experiments

- Datasets

Name	#Documents	#Links	Vocabulary	#Labels
DS	1,703	3,234	3,134	9
ML	3,087	8,573	3,040	7
PL	2,597	7,754	3,106	9
Aminer	42,564	40,269	4,094	10
Web	445,657	565,502	10,015	N.A.

Experiments

- Text classification

	Model	Micro F1 score			
		DS	ML	PL	Aminer
Topic Models	ProdLDA	51.4±1.1	67.0±0.6	51.8±0.6	40.5±0.0
	DVAE	54.7±2.2	68.9±0.6	55.7±1.3	66.1±0.5
	ETM	42.2±2.4	53.9±1.8	45.0±2.1	53.2±0.7
Graph Embedding	Adjacent-Encoder	58.8±1.2	72.8±0.6	60.0±1.7	59.5±0.2
	LANTM	56.8±2.4	71.8±1.0	62.6±1.3	N.A.
	VGAE	39.8±2.0	56.6±1.7	47.6±3.4	64.7±0.5
	PGCL	62.9±1.2	74.9±1.2	64.7±1.1	69.7±0.4
Our model	DBN	**66.2±1.4**	78.4±0.8	67.3±0.5	71.2±0.4
	D^2BN	65.8±1.6	**81.1±1.2**	**71.3±0.7**	**72.3±0.3**

Hyperbolic Graph Topic Modeling Network with Continuously Updated Topic Tree

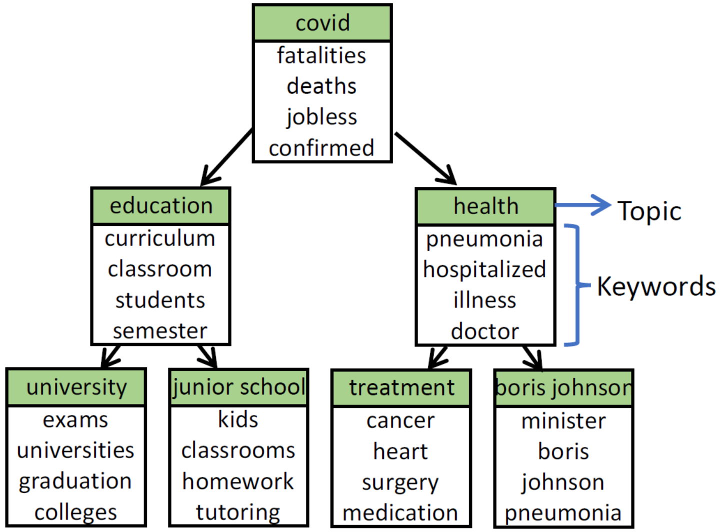
Delvin Ce Zhang[1], Rex Ying[1], and Hady W. Lauw[2]

[1]Yale University, [2]Singapore Management University

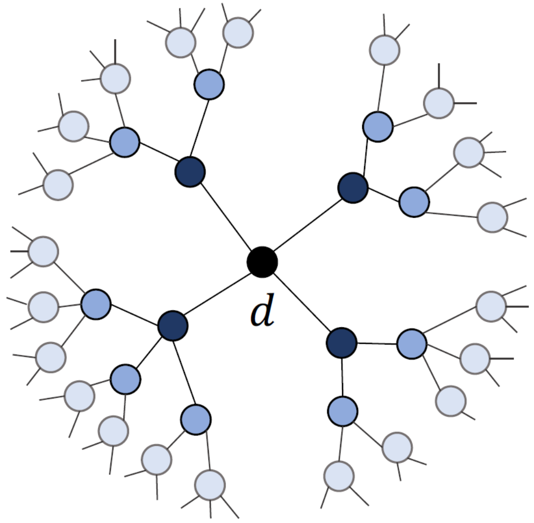
ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD-23)

Motivation

- **Hierarchical topics $\mathcal{D}$**
  - Some articles report global COVID situation, while others focus on specific event.
  - → Text hierarchy $\mathcal{D}$
- **Hierarchical edges $\mathcal{E}$**
  - A breaking news article is traced by following articles reporting subsequent events.
  - → Graph hierarchy $\mathcal{E}$



(a) Text hierarchy

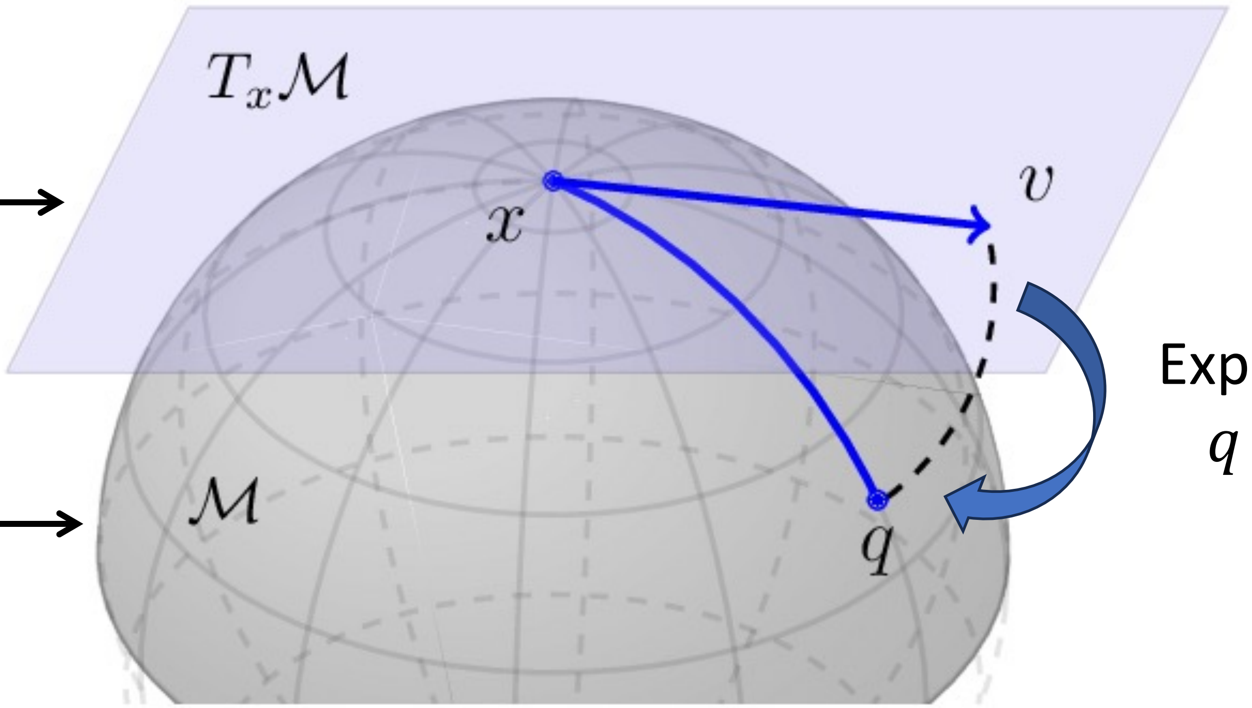


(b) Graph hierarchy

Introduction to Hyperbolic Space

Tangent space at x
(Euclidean space)

Hyperbolic space

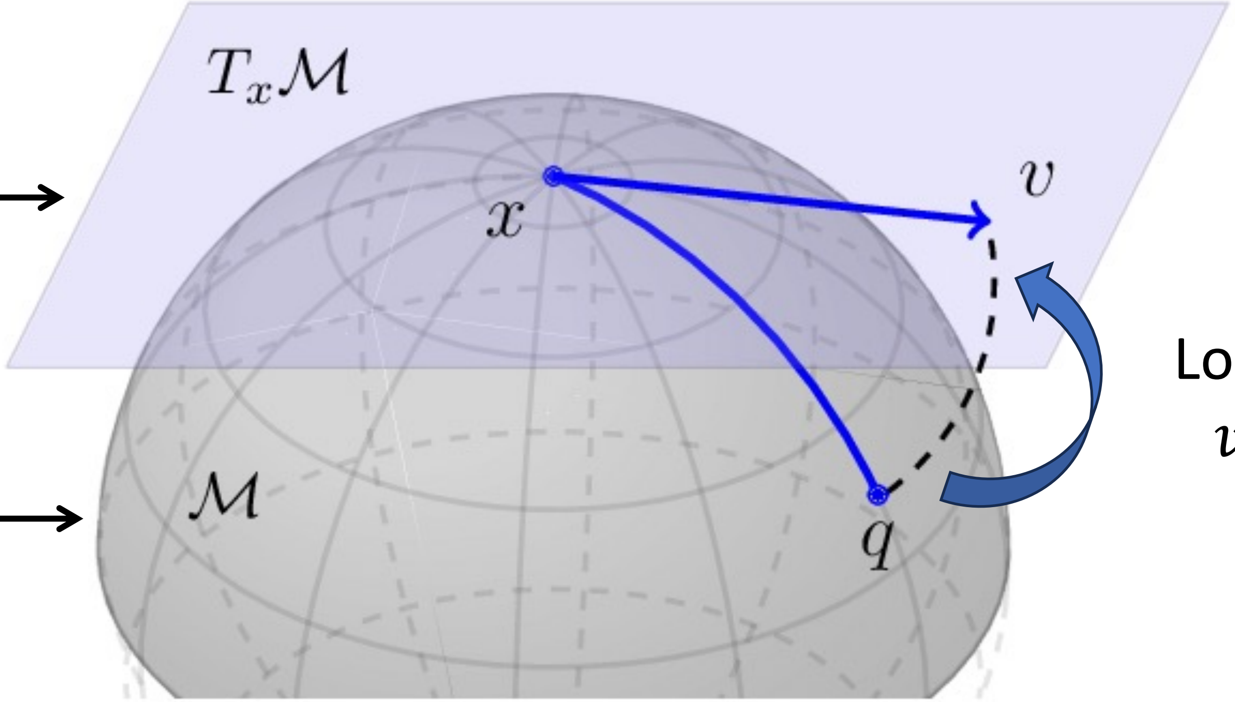


Exponential map
$q = \exp_x(v)$

Introduction to Hyperbolic Space

Tangent space at x
(Euclidean space)

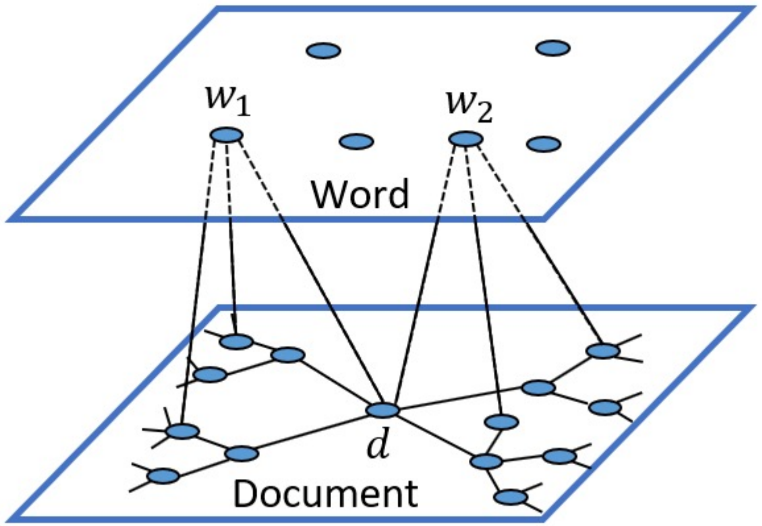
Hyperbolic space



Logrithmic map
$v = \log_x(q)$

Model Architecture

- **Hyperbolic Graph Encoder (for graph hierarchy $\mathcal{E}$)**
- 1. Given a graph $\{\mathcal{D}, \mathcal{E}\}$, we construct a two-layer graph for documents and words.



Model Architecture

- **Hyperbolic Graph Encoder (for graph hierarchy $\mathcal{E}$)**
- 2. Intra-layer encoding
  - A. Hyperbolic linear transformation

$$\tilde{\mathbf{z}}'^{(l)}_d = \exp^c_{\mathbf{0}}(\mathbf{W}^{(l)} \log^c_{\mathbf{0}}(\mathbf{z}^{(l-1)}_d))$$

  - B. Hyperbolic neighbor attention

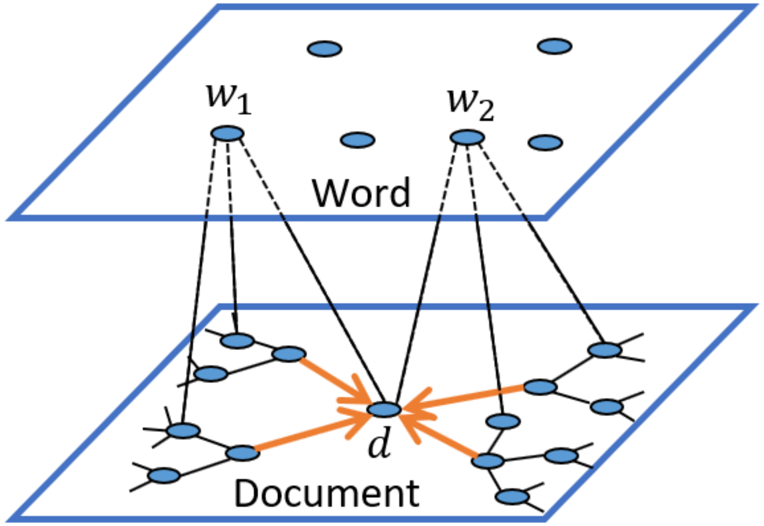
$$a_{ij} = \mathrm{softmax}\left(\sigma(\boldsymbol{\beta}^{(l)\top}[\log^c_{\mathbf{0}}(\tilde{\mathbf{z}}^{(l)}_{d_i})||\log^c_{\mathbf{0}}(\tilde{\mathbf{z}}^{(l)}_{d_j})])\right) \text{ where } d_j \in \mathcal{N}(i)$$

  - C. Hyperbolic aggregation

$$\mathbf{z}^{(l)}_{d_i} = f^{c,c'}_{\mathrm{act}}\left(\exp^c_{\mathbf{0}}\left(\frac{1}{2}(\log^c_{\mathbf{0}}(\tilde{\mathbf{z}}^{(l)}_{d_i}) + \sum_{d_j \in \mathcal{N}(i)} a_{ij} \log^c_{\mathbf{0}}(\tilde{\mathbf{z}}^{(l)}_{d_j}))\right)\right)$$

  - Summarizing above three steps, we have

$$z_{intra} = HGNN\left(d, N_{intra}(d)\right)$$



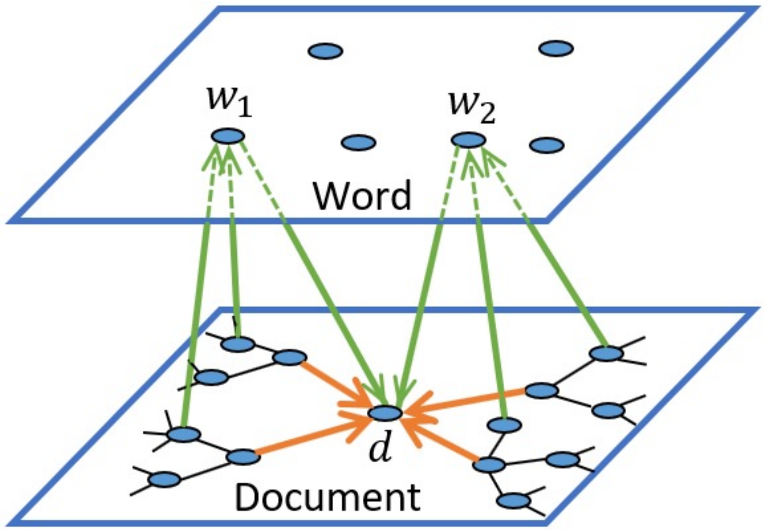
Model Architecture

- **Hyperbolic Graph Encoder (for graph hierarchy $\mathcal{E}$)**
- 3. Cross-layer encoding

$$\boldsymbol{z}_{\boldsymbol{cross}} = HGNN\big(d, N_{cross}(d)\big)$$

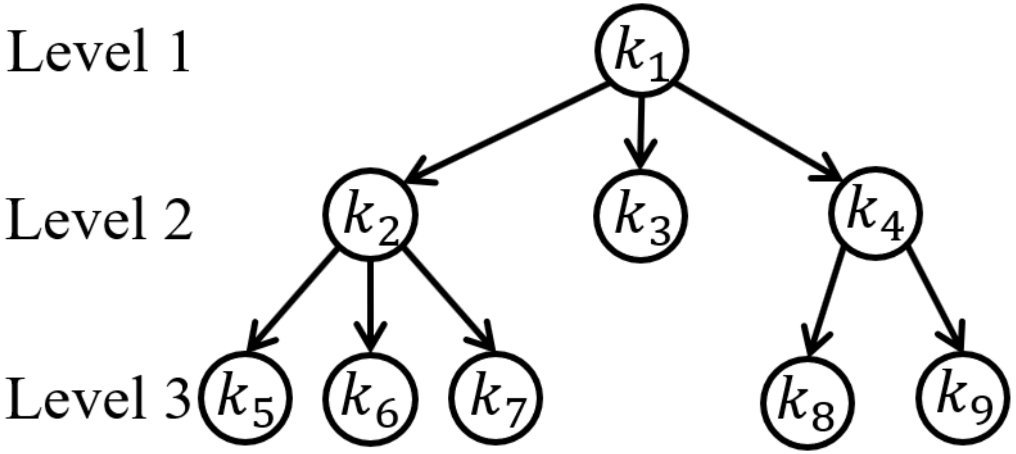
- 4. Hyperbolic mean pooling

$$z_d = \exp\left(\frac{1}{2} \times (\log(\boldsymbol{z}_{\boldsymbol{intra}}) + \log(\boldsymbol{z}_{\boldsymbol{cross}}))\right)$$



Model Architecture

- **Tree-Structured Decoder (for text hierarchy $\mathcal{D}$)**
- 1. Initialize a latent topic tree

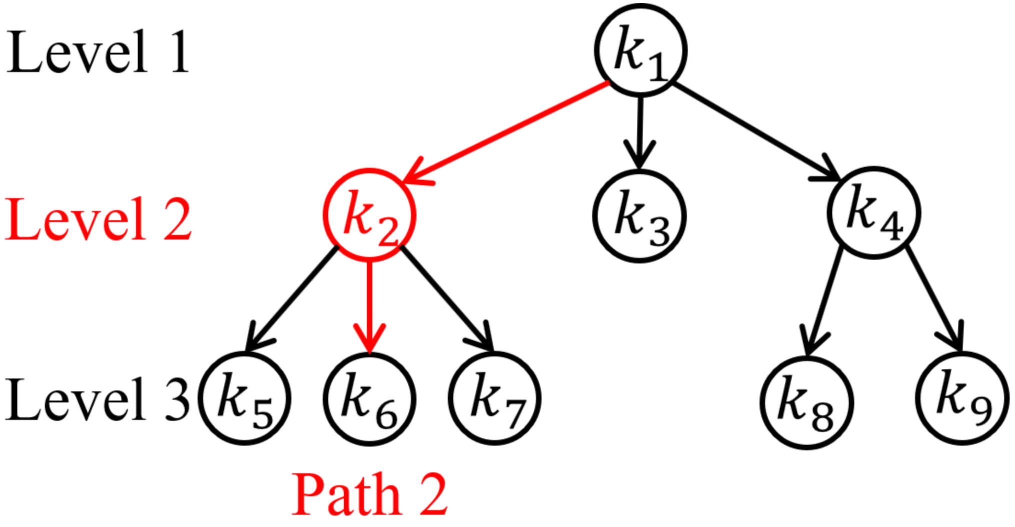


Model Architecture

- **Tree-Structured Decoder (for text hierarchy $\mathcal{D}$)**
- 1. Initialize a latent topic tree
- 2. For doc d, evaluate path distribution

$$p(k_1 \rightarrow k_2 \rightarrow k_6) = p(k_6|k_2)p(k_2|k_1)p(k_1)$$

$$p(k_2|k_1) = \frac{\left(1 + \mathrm{dist}\left(z_d, t_{k_2}\right)\right)^{-1}}{\sum_{k'=k_2,k_3,k_4}\left(1 + \mathrm{dist}(z_d, t_{k'})\right)^{-1}}$$



- 3. Evaluate level distribution

$$p(\text{level } s) = \frac{(1 + h(s)^2)^{-1}}{\sum_{s'=1,2,3}(1 + h(s')^2)^{-1}} \quad \text{where} \quad h(s)^2 = \min\left\{\mathrm{dist}\left(z_d, t_{k_s}\right)^2 \middle| k_s \in \text{level } s\right\}$$

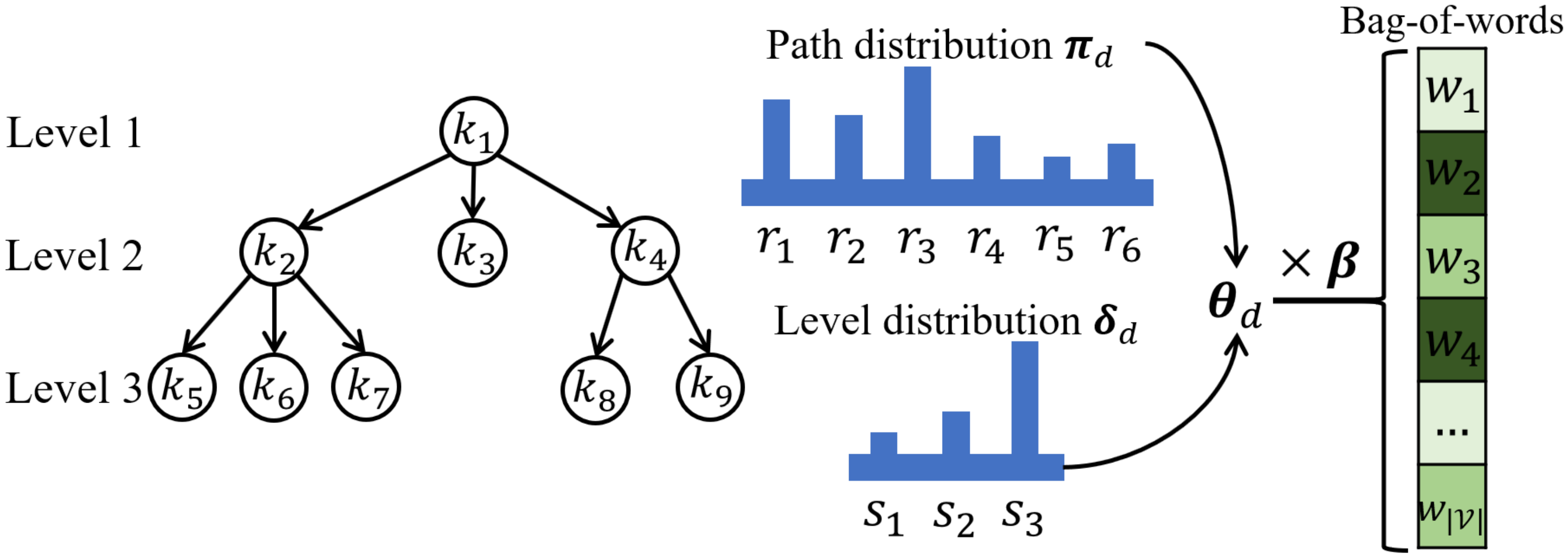
Model Architecture

- **Tree-Structured Decoder (for text hierarchy $\mathcal{D}$)**
- 4. Topic distribution

$$p(k_2) = p(s = 2)\times\big(p(\text{path } 1) + p(\text{path } 2) + p(\text{path } 3)\big)$$

$$\theta_d = [p(k_1), p(k_2), \dots, p(k_9)]$$

- 5. Decoding with cross-entropy loss $\hat{d} = \mathrm{softmax}(\beta\theta_d)$



Experiments

- Datasets

Name	#Documents	#Links	Vocabulary	#Labels
ML	3,087	8,573	2,885	7
PL	2,597	7,754	3,106	9
COVID	1,500	5,706	5,083	5
Aminer	114,741	265,345	10,018	10
Web	445,657	565,502	10,015	N.A.

Experiments

- Text classification

	Model	Micro F1 score			
		ML	PL	COVID	Aminer
HGNN	HGCN	82.6±1.3	70.3±1.0	86.0±0.5	67.6±1.0
	LGCN	73.0±2.2	62.8±2.6	60.5±5.0	N.A.
Text Classification	TextGCN	78.3±0.7	67.5±0.7	83.7±0.5	N.A.
	HyperGAT	80.0±0.4	65.8±2.5	84.3±1.2	**74.2±1.5**
	HINT	69.5±1.1	55.4±2.3	85.7±1.5	66.5±0.5
Our model	HGTM	**83.8±0.5**	**72.2±1.4**	**86.3±1.7**	70.0±0.3

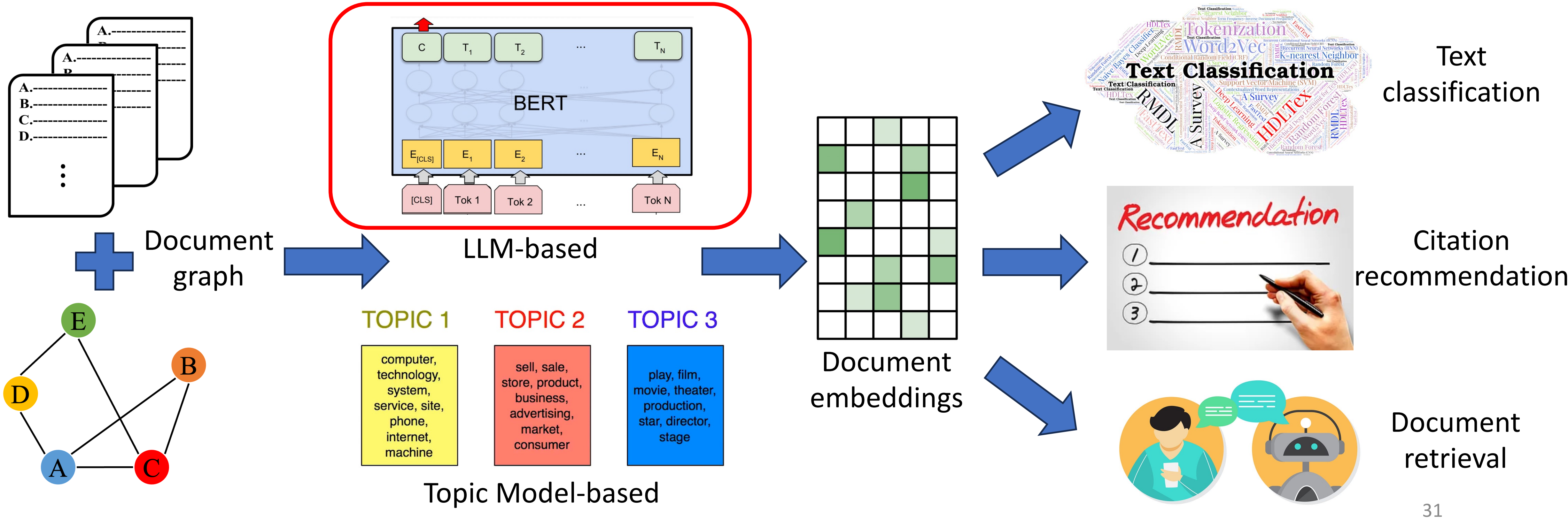
Experiments

- Link prediction

	Model	ML	PL	COVID	Aminer	Web
HGNN	HGCN	89.7±0.4	90.3±0.3	94.8±0.3	94.2±0.1	90.5±0.2
	LGCN	89.2±0.4	90.8±0.5	93.4±0.5	N.A.	N.A.
Text Classification	TextGCN	76.5±0.5	68.2±0.4	87.1±0.4	N.A.	N.A.
	HyperGAT	82.0±0.8	77.5±1.0	87.1±0.4	90.0±0.0	N.A.
	HINT	71.7±1.4	69.7±1.4	86.6±0.2	89.8±0.1	N.A.
Our model	HGTM	**89.9±0.8**	**91.3±0.3**	**95.7±0.2**	**95.9±0.2**	**91.3±0.1**

Limitations and Future Work

- Designing LLM-based models for document graph representation learning;
- Explainability: which subset of topics are informative for predictions.



Thank You

Delvin Ce Zhang

Yale University