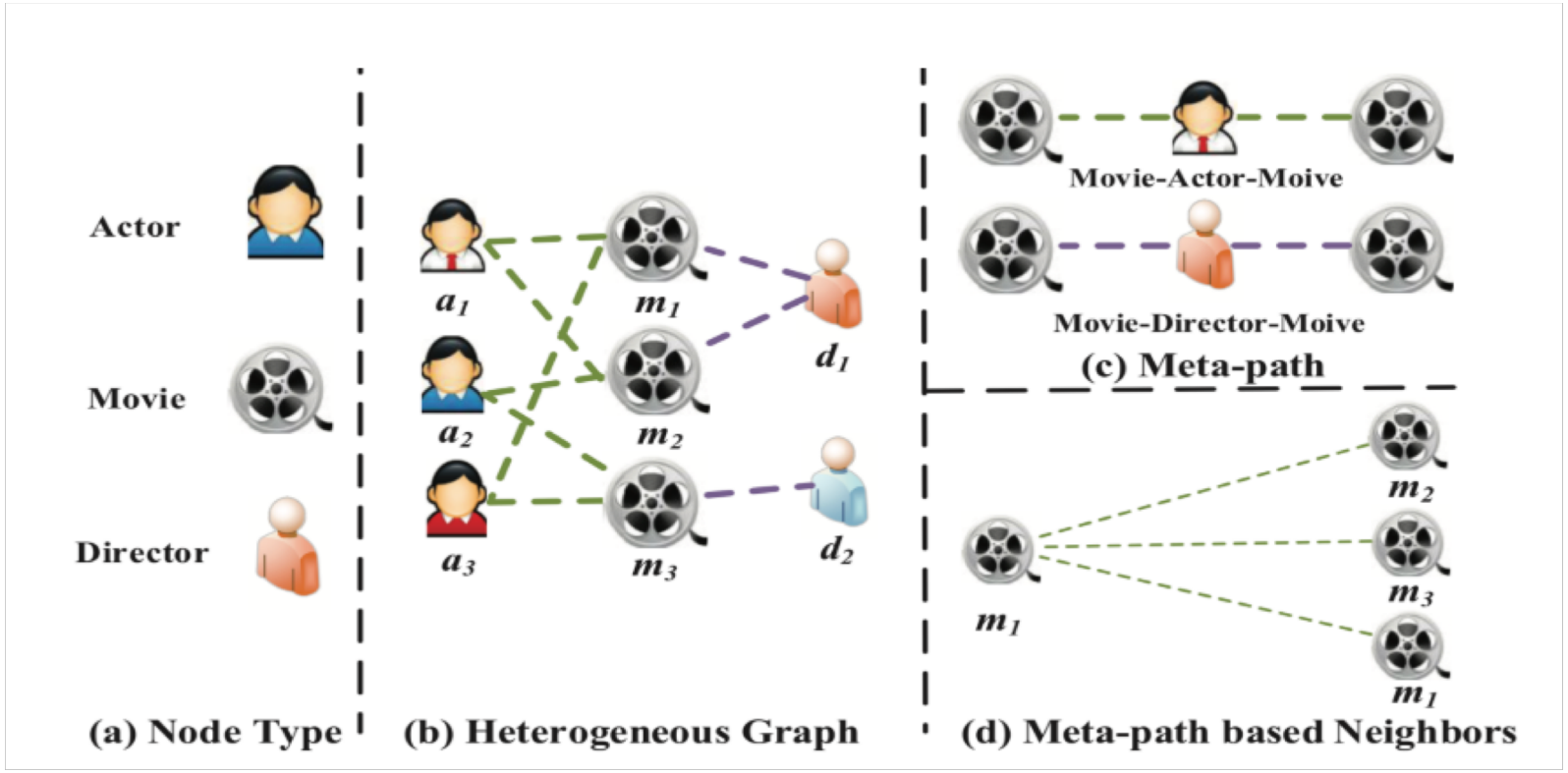
Heterogeneous Graph Attention Network

Xiao Wang, Houye Ji
Beijing University of Posts and Telecommunications
Beijing, China
{xiaowang,jhy1993}@bupt.edu.cn

Chuan Shi*, Bai Wang
Beijing University of Posts and Telecommunications
Beijing, China
{shichuan,wangbai}@bupt.edu.cn

Peng Cui, P. Yu
Tsinghua University
Beijing, China
{cuip,psyu}@tsinghua.edu.cn

Yanfang Ye
West Virginia University
WV, USA
yanfang.ye@mail.wvu.edu



Meta-path $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} A_{l+1}$

Eg： · Movie-Actor-Movie (MAM), · Movie-Director-Movie (MDM).

Meta-path Neighbors · the meta-path based neighbors of node i are defined as the set of nodes which connect with node i via meta-path Φ.

Tasks

Dataset	Relations(A-B)	Number of A	Number of B	Number of A-B	Feature	Training	Validation	Test	Meta-paths
DBLP	Paper-Author	14328	4057	19645	334	800	400	2857	*APA*
	Paper-Conf	14328	20	14328					*APCPA*
	Paper-Term	14327	8789	88420					*APTPA*
IMDB	Movie-Actor	4780	5841	14340	1232	300	300	2687	*MAM*
	Movie-Director	4780	2269	4780					*MDM*
ACM	Paper-Author	3025	5835	9744	1830	600	300	2125	*PAP*
	Paper-Subject	3025	56	3025					*PSP*

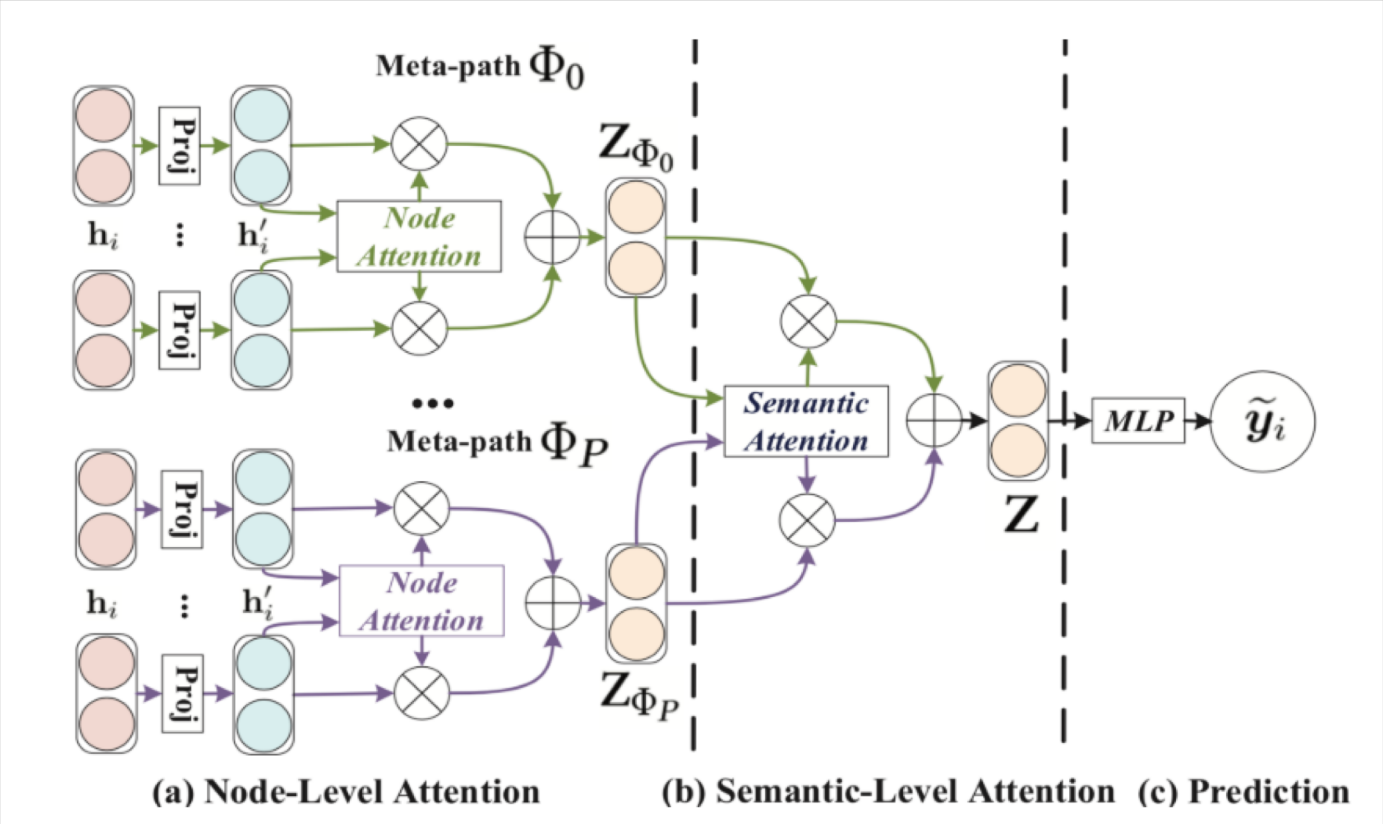
DBLP: The authors are divided into four areas: **database**, **data mining**, **machine learning**, **information retrieval.**

Movie: The movies are divided into three classes: **Action, Comedy, Drama**

ACM: The papers are divided into three areas: **Database**, **Wireless Communication**, **Data Mining**

Motivation: two-level attention, node-level && semantic-level(meta-path level)

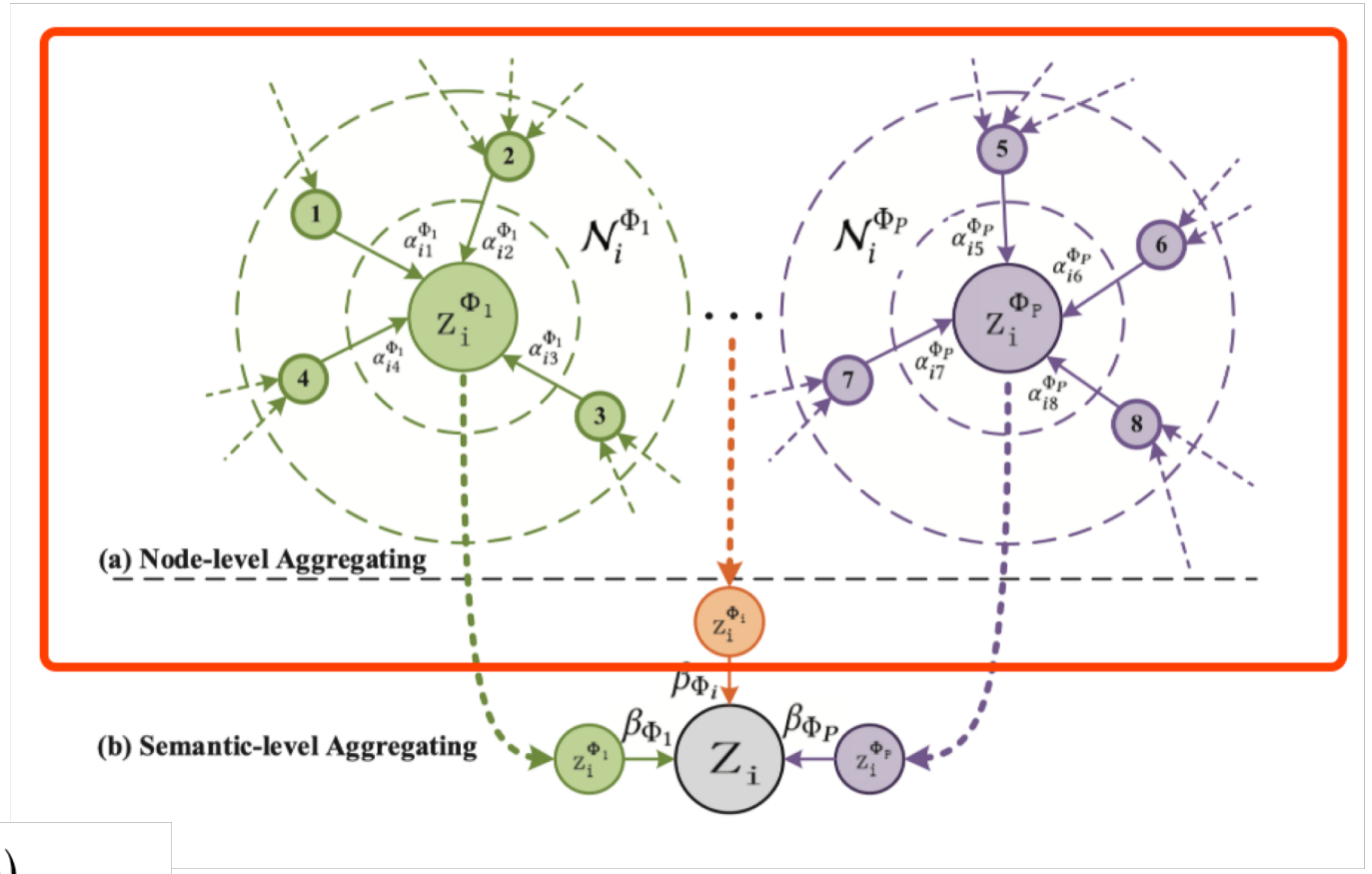
Framework


Meta-path Φ_0
Proj
Proj
$\mathbf{h}_i$
$\mathbf{h}'_i$
Node Attention
Z_{Φ_0}
Meta-path Φ_P
Proj
Proj
$\mathbf{h}_i$
$\mathbf{h}'_i$
Node Attention
Z_{Φ_P}
Semantic Attention
Z
MLP
$\tilde{y}_i$
(a) Node-Level Attention
(b) Semantic-Level Attention
(c) Prediction

Node-level Attention

$$\mathbf{h}_i' = \mathbf{M}_{\phi_i} \cdot \mathbf{h}_i$$

$$e_{ij}^{\Phi} = att_{node}(\mathbf{h}_i', \mathbf{h}_j'; \Phi).$$



$$\alpha_{ij}^{\Phi} = softmax_j(e_{ij}^{\Phi}) = \frac{\exp\left(\sigma(\mathbf{a}_{\Phi}^{\mathrm{T}} \cdot [\mathbf{h}_i' \| \mathbf{h}_j'])\right)}{\sum_{k \in \mathcal{N}_i^{\Phi}} \exp\left(\sigma(\mathbf{a}_{\Phi}^{\mathrm{T}} \cdot [\mathbf{h}_i' \| \mathbf{h}_k'])\right)},$$

$$\mathbf{z}_i^{\Phi} = \sigma\left(\sum_{j \in \mathcal{N}_i^{\Phi}} \alpha_{ij}^{\Phi} \cdot \mathbf{h}_j'\right).$$

$$\mathbf{z}_i^{\Phi} = \Big\|_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}_i^{\Phi}} \alpha_{ij}^{\Phi} \cdot \mathbf{h}_j'\right).$$

Multi-head attention，head number=K

Semantic-level Attention

$$(\beta_{\Phi_0}, \beta_{\Phi_1}, \ldots, \beta_{\Phi_P}) = att_{sem}(\mathbf{Z}_{\Phi_0}, \mathbf{Z}_{\Phi_1}, \ldots, \mathbf{Z}_{\Phi_P}).$$

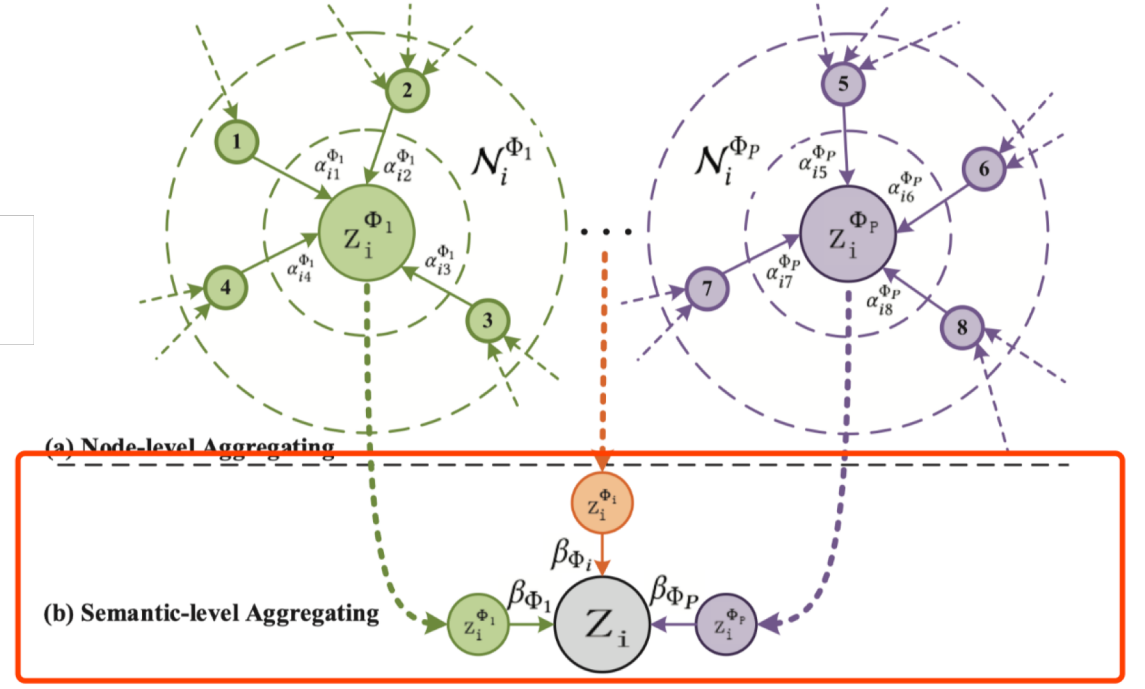
$$w_{\Phi_i} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \mathbf{q}^{\mathrm{T}} \cdot \tanh(\mathbf{W} \cdot \mathbf{z}_i^{\Phi} + \mathbf{b}),$$

$$\beta_{\Phi_i} = \frac{\exp(w_{\Phi_i})}{\sum_{i=1}^{P} \exp(w_{\Phi_i})},$$

$$\mathbf{Z} = \sum_{i=1}^{P} \beta_{\Phi_i} \cdot \mathbf{Z}_{\Phi_i}.$$

$$L = -\sum_{l \in \mathcal{Y}_L} \mathrm{Y}^l \ln(\mathrm{C} \cdot \mathrm{Z}^l),$$

Loss function



Experiments-classification

Datasets	Metrics	Training	DeepWalk	ESim	metapath2vec	HERec	GCN	GAT	HAN_{nd}	HAN_{sem}	HAN
ACM	Macro-F1	20%	77.25	77.32	65.09	66.17	86.81	86.23	88.15	89.04	**89.40**
		40%	80.47	80.12	69.93	70.89	87.68	87.04	88.41	89.41	**89.79**
		60%	82.55	82.44	71.47	72.38	88.10	87.56	87.91	**90.00**	89.51
		80%	84.17	83.00	73.81	73.92	88.29	87.33	88.48	90.17	**90.63**
	Micro-F1	20%	76.92	76.89	65.00	66.03	86.77	86.01	87.99	88.85	**89.22**
		40%	79.99	79.70	69.75	70.73	87.64	86.79	88.31	89.27	**89.64**
		60%	82.11	82.02	71.29	72.24	88.12	87.40	87.68	**89.85**	89.33
		80%	83.88	82.89	73.69	73.84	88.35	87.11	88.26	89.95	**90.54**
DBLP	Macro-F1	20%	77.43	91.64	90.16	91.68	90.79	90.97	91.17	92.03	**92.24**
		40%	81.02	92.04	90.82	92.16	91.48	91.20	91.46	92.08	**92.40**
		60%	83.67	92.44	91.32	92.80	91.89	90.80	91.78	92.38	**92.80**
		80%	84.81	92.53	91.89	92.34	92.38	91.73	91.80	92.53	**93.08**
	Micro-F1	20%	79.37	92.73	91.53	92.69	91.71	91.96	92.05	92.99	**93.11**
		40%	82.73	93.07	92.03	93.18	92.31	92.16	92.38	93.00	**93.30**
		60%	85.27	93.39	92.48	93.70	92.62	91.84	92.69	93.31	**93.70**
		80%	86.26	93.44	92.80	93.27	93.09	92.55	92.69	93.29	**93.99**
IMDB	Macro-F1	20%	40.72	32.10	41.16	41.65	45.73	49.44	49.78	**50.87**	50.00
		40%	45.19	31.94	44.22	43.86	48.01	50.64	52.11	50.85	**52.71**
		60%	48.13	31.68	45.11	46.27	49.15	51.90	51.73	52.09	**54.24**
		80%	50.35	32.06	45.15	47.64	51.81	52.99	52.66	51.60	**54.38**
	Micro-F1	20%	46.38	35.28	45.65	45.81	49.78	55.28	54.17	55.01	**55.73**
		40%	49.99	35.47	48.24	47.59	51.71	55.91	56.39	55.15	**57.97**
		60%	52.21	35.64	49.09	49.88	52.29	56.44	56.09	56.66	**58.32**
		80%	54.33	35.59	48.81	50.99	54.61	56.97	56.38	56.49	**58.51**

Experiments-clustering

Table 4: Qantitative results (%) on the node clustering task.

Datasets	Metrics	DeepWalk	ESim	metapath2vec	HERec	GCN	GAT	HAN_{nd}	HAN_{sem}	HAN
ACM	NMI	41.61	39.14	21.22	40.70	51.40	57.29	60.99	61.05	**61.56**
	ARI	35.10	34.32	21.00	37.13	53.01	60.43	61.48	59.45	**64.39**
DBLP	NMI	76.53	66.32	74.30	76.73	75.01	71.50	75.30	77.31	**79.12**
	ARI	81.35	68.31	78.50	80.98	80.49	77.26	81.46	83.46	**84.76**
IMDB	NMI	1.45	0.55	1.20	1.20	5.45	8.45	9.16	10.31	**10.87**
	ARI	2.15	0.10	1.70	1.65	4.40	7.46	7.98	9.51	**10.01**

Graph Convolutional Networks for Text Classification

Liang Yao, Chengsheng Mao, Yuan Luo*
Northwestern University
Chicago IL 60611
{liang.yao, chengsheng.mao, yuan.luo}@northwestern.edu

Graph Convolutional Networks（GCN）

· A GCN is a multilayer neural network that operates directly on a graph and induces embedding vectors of nodes based on properties of their neighborhoods.

Consider a graph $G = (V, E)$

$X \in \mathbb{R}^{n \times m}$. feature matrix, n is the node number, m is the dimension of the feature vectors

$A \in \mathbb{R}^{n \times n}$ · . adjacency matrix, · diagonal elements of A are set to 1 because of self-loops

$D \in \mathbb{R}^{n \times n}$. · degree matrix D, where $D_{ii} = \sum_j A_{ij}$

· For a **one-layer GCN**, · the new k-dimensional node feature matrix $L^{(1)} \in \mathbb{R}^{n \times k}$

$$L^{(1)} = \rho(\tilde{A}XW_0)$$

Where $\tilde{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ is the normalized symmetric adjacency matrix, $W_0 \in \mathbb{R}^{m \times k}$ is a weight matrix

incorporate higher order neighborhoods information by stacking **multiple GCN layers**

$$L^{(j+1)} = \rho(\tilde{A}L^{(j)}W_j)$$

Text Graph Convolutional Networks (Text GCN)

$$G = (V, E)$$

$$L^{(1)} = \rho(\tilde{A}XW_0)$$

$$\tilde{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$$

- $\boldsymbol{V}$: documents (corpus) plus unique words (vocabulary)

- $\boldsymbol{X}$: feature matrix $X = I$ as an identity matrix which means every word or document is represented as a one-hot vector as the input to Text GCN

- $\boldsymbol{A}$:

$$A_{ij} = \begin{cases} \text{PMI}(i,j) & i,j \text{ are words, PMI}(i,j) > 0 \\ \text{TF-IDF}_{ij} & i \text{ is document, } j \text{ is word} \\ 1 & i = j \\ 0 & \text{otherwise} \end{cases}$$

$$\text{PMI}(i,j) = \log \frac{p(i,j)}{p(i)p(j)}$$

$$p(i,j) = \frac{\#W(i,j)}{\#W}$$

$$p(i) = \frac{\#W(i)}{\#W}$$

- $\boldsymbol{D}$: $D_{ii} = \sum_j A_{ij}$, other = 0

· After building the text graph, feed the graph into a simple two layer GCN

$$Z = \text{softmax}(\tilde{A}\,\text{ReLU}(\tilde{A}XW_0)W_1)$$

representation

classifier

· The loss function is defined as the cross-entropy error over all labeled documents:

$$\mathcal{L} = -\sum_{d \in \mathcal{Y}_D} \sum_{f=1}^{F} Y_{df} \ln Z_{df}$$

Experiments

Model	20NG	R8	R52	Ohsumed	MR
TF-IDF + LR	0.8319 ± 0.0000	0.9374 ± 0.0000	0.8695 ± 0.0000	0.5466 ± 0.0000	0.7459 ± 0.0000
CNN-rand	0.7693 ± 0.0061	0.9402 ± 0.0057	0.8537 ± 0.0047	0.4387 ± 0.0100	0.7498 ± 0.0070
CNN-non-static	0.8215 ± 0.0052	0.9571 ± 0.0052	0.8759 ± 0.0048	0.5844 ± 0.0106	**0.7775 ± 0.0072**
LSTM	0.6571 ± 0.0152	0.9368 ± 0.0082	0.8554 ± 0.0113	0.4113 ± 0.0117	0.7506 ± 0.0044
LSTM (pretrain)	0.7543 ± 0.0172	0.9609 ± 0.0019	0.9048 ± 0.0086	0.5110 ± 0.0150	0.7733 ± 0.0089
Bi-LSTM	0.7318 ± 0.0185	0.9631 ± 0.0033	0.9054 ± 0.0091	0.4927 ± 0.0107	0.7768 ± 0.0086
PV-DBOW	0.7436 ± 0.0018	0.8587 ± 0.0010	0.7829 ± 0.0011	0.4665 ± 0.0019	0.6109 ± 0.0010
PV-DM	0.5114 ± 0.0022	0.5207 ± 0.0004	0.4492 ± 0.0005	0.2950 ± 0.0007	0.5947 ± 0.0038
PTE	0.7674 ± 0.0029	0.9669 ± 0.0013	0.9071 ± 0.0014	0.5358 ± 0.0029	0.7023 ± 0.0036
fastText	0.7938 ± 0.0030	0.9613 ± 0.0021	0.9281 ± 0.0009	0.5770 ± 0.0049	0.7514 ± 0.0020
fastText (bigrams)	0.7967 ± 0.0029	0.9474 ± 0.0011	0.9099 ± 0.0005	0.5569 ± 0.0039	0.7624 ± 0.0012
SWEM	0.8516 ± 0.0029	0.9532 ± 0.0026	0.9294 ± 0.0024	0.6312 ± 0.0055	0.7665 ± 0.0063
LEAM	0.8191 ± 0.0024	0.9331 ± 0.0024	0.9184 ± 0.0023	0.5858 ± 0.0079	0.7695 ± 0.0045
Graph-CNN-C	0.8142 ± 0.0032	0.9699 ± 0.0012	0.9275 ± 0.0022	0.6386 ± 0.0053	0.7722 ± 0.0027
Graph-CNN-S	–	0.9680 ± 0.0020	0.9274 ± 0.0024	0.6282 ± 0.0037	0.7699 ± 0.0014
Graph-CNN-F	–	0.9689 ± 0.0006	0.9320 ± 0.0004	0.6304 ± 0.0077	0.7674 ± 0.0021
Text GCN	**0.8634 ± 0.0009**	**0.9707 ± 0.0010**	**0.9356 ± 0.0018**	**0.6836 ± 0.0056**	0.7674 ± 0.0020

Effects of the Size of Labeled Data

- **achieve satisfactory results in text classification, even with limited labeled data**

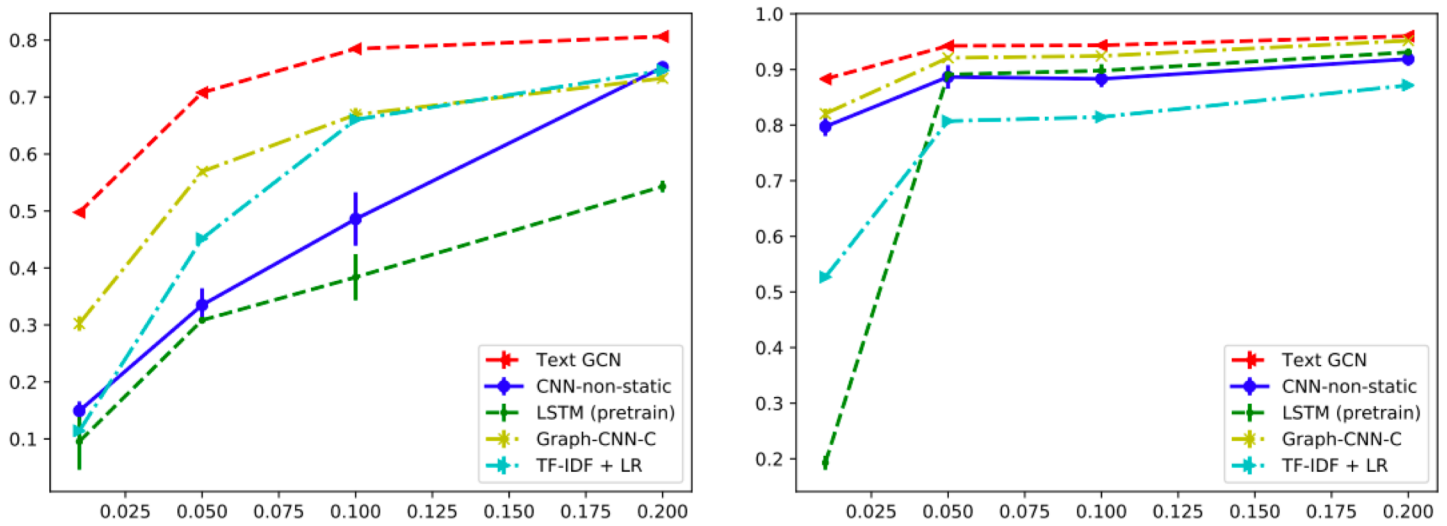


Figure 4: Test accuracy by varying training data proportions.

Document Visualization

- **learn predictive word and document embeddings**

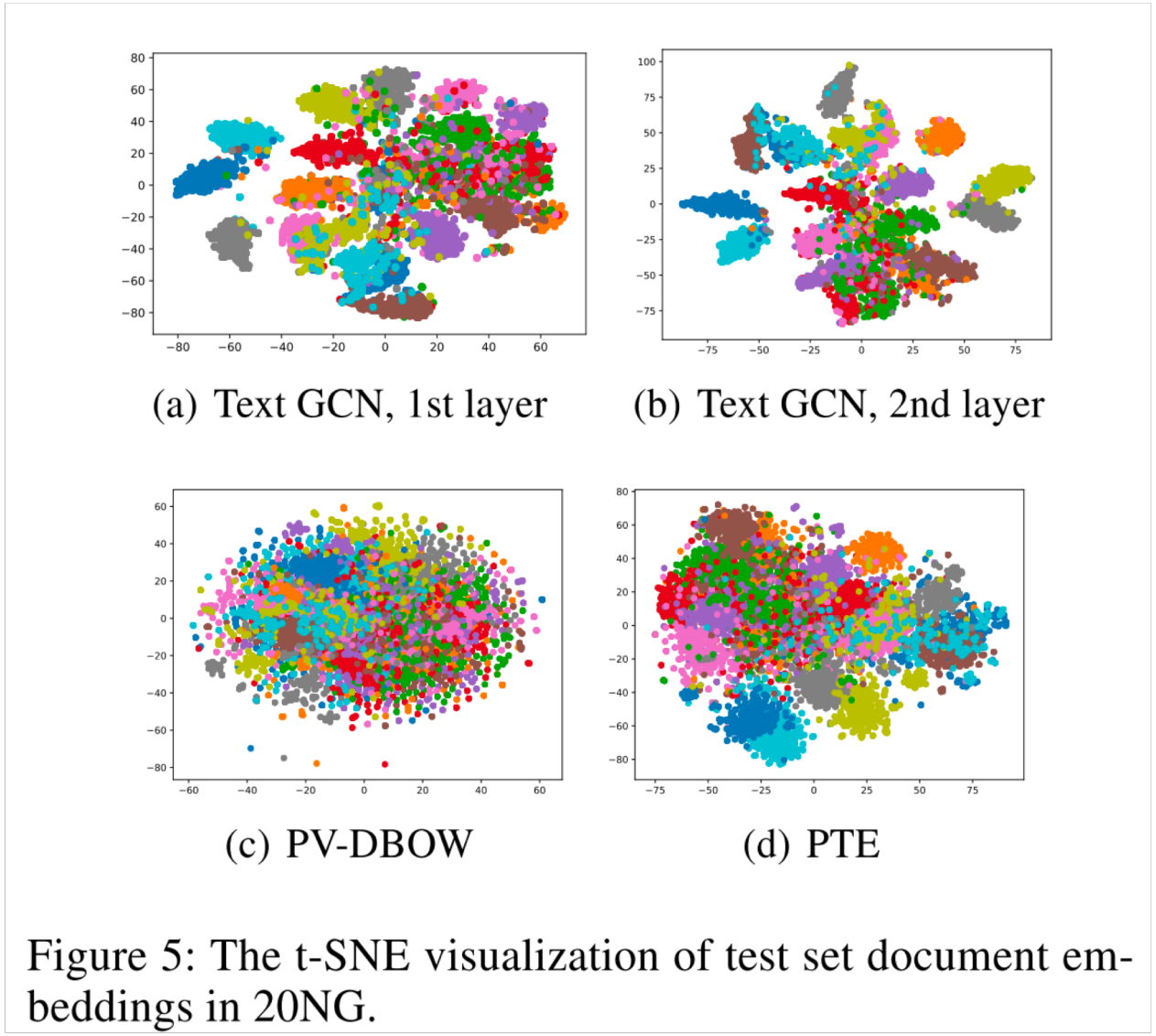
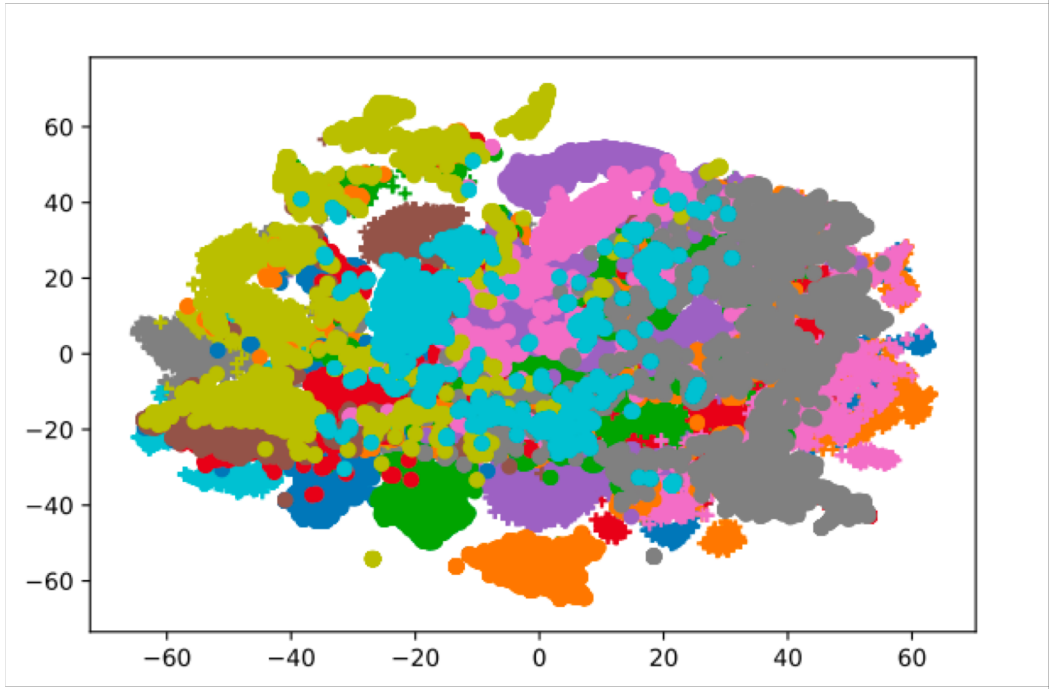


Figure 5: The t-SNE visualization of test set document embeddings in 20NG.

Word Visualization

- We set the dimension with the highest value as a word's label.



comp.graphics	sci.space	sci.med	rec.autos
jpeg	space	candida	car
graphics	orbit	geb	cars
image	shuttle	disease	v12
gif	launch	patients	callison
3d	moon	yeast	engine
images	prb	msg	toyota
rayshade	spacecraft	vitamin	nissan
polygon	solar	syndrome	v8
pov	mission	infection	mustang
viewer	alaska	gordon	eliot

总结

1. 图结构的表示对于少量数据集的半监督训练效果也不错

2. 异质图结构能够综合利用多个方面的信息，可以考虑在一张图里构建来源不同的节点信息，比如对话里面同时构建历史信息和回复之间的关系。