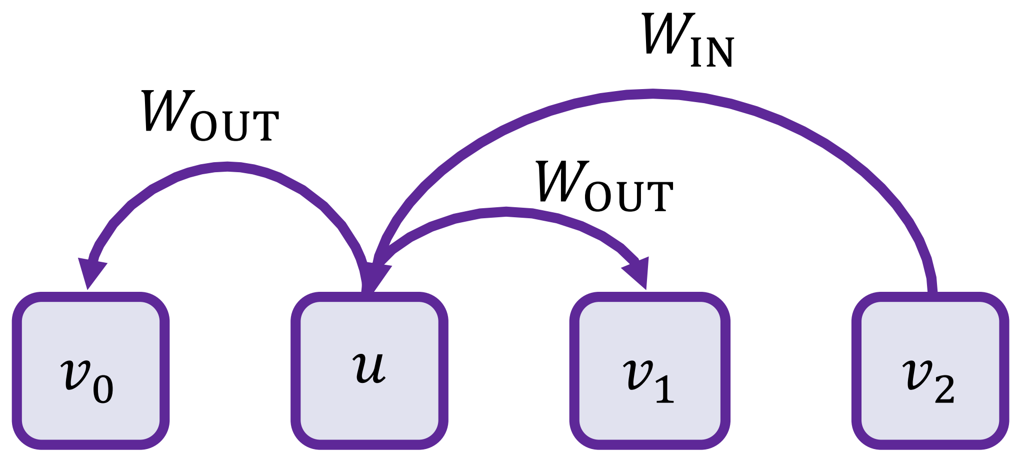
图卷积神经网络应用

目录

- 图卷积神经网络编程实现
- 应用：机器翻译
- 应用：关系分类

图卷积神经网络编程实现

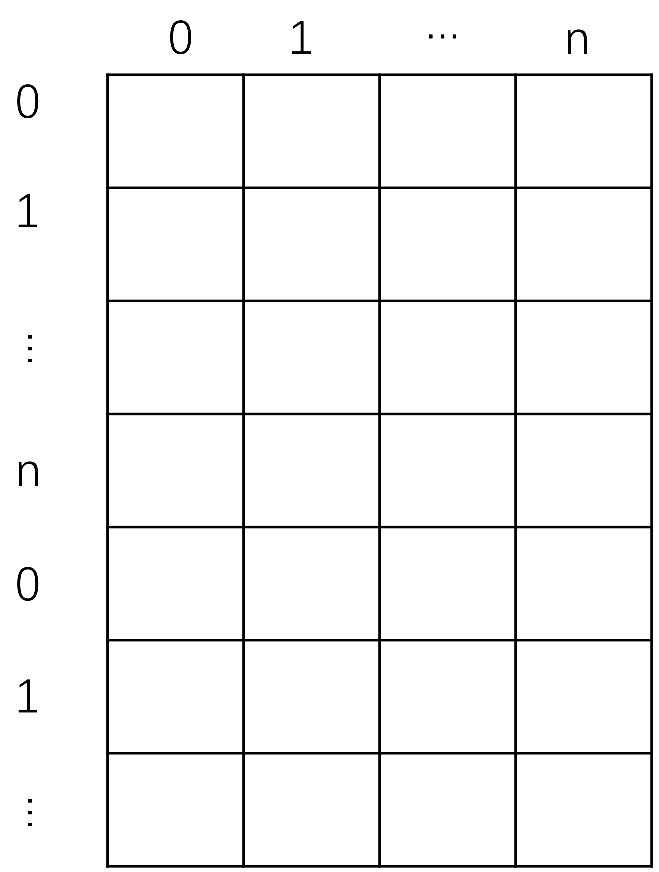


核心公式： $$\mathbf{h}_v^{(j+1)} = \rho\Big(\sum_{u\in\mathcal{N}(v)} \mathbf{W}_{\mathrm{dir}(u,v)}^{(j)}\mathbf{h}_u^{(j)} + \mathbf{b}_{\mathrm{dir}(u,v)}^{(j)}\Big), \qquad \mathbf{W}_{dir(u,v)} \in \{\mathbf{W}_{\mathrm{IN}}, \mathbf{W}_{\mathrm{OUT}}\}$$

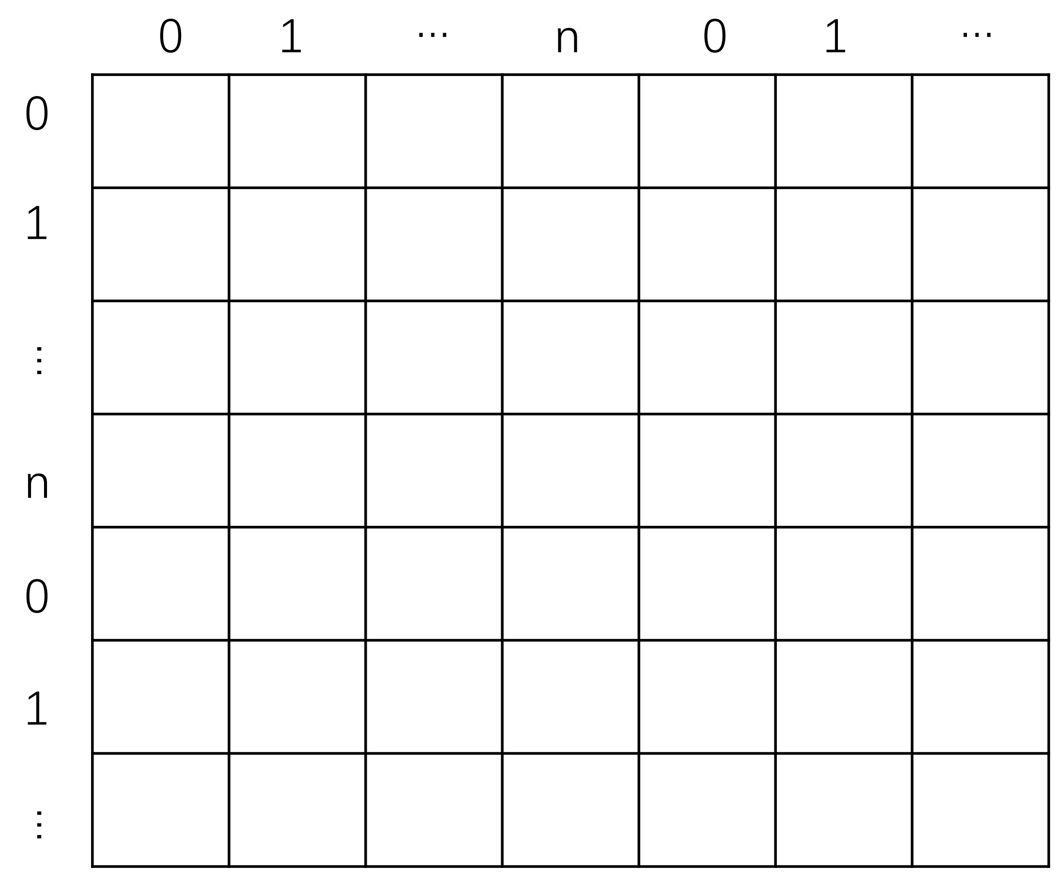
图卷积神经网络编程实现

构建图：

方式一: (batch_size * n, n)



方式二: (batch_size * n, batch_size * n)



图卷积神经网络编程实现

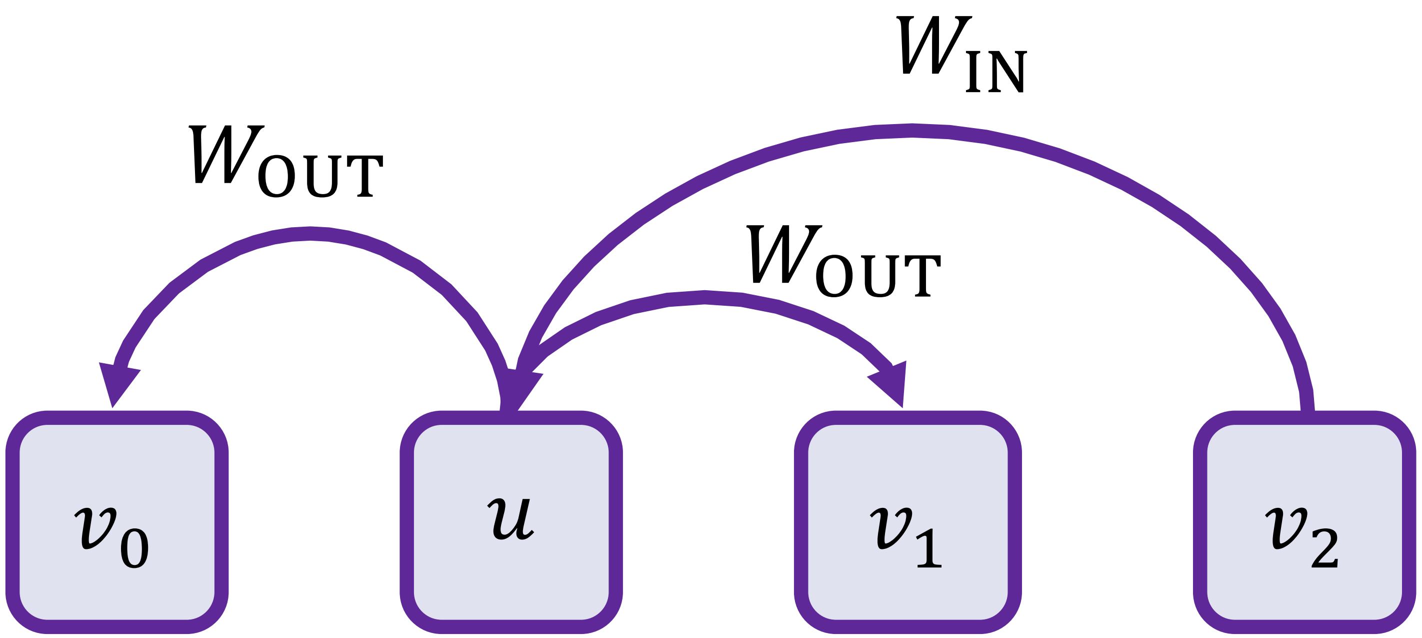
$$\begin{aligned}\mathbf{h}_u^{(j+1)} &= \rho\Big(\sum_{u\in\mathcal{N}(v)} W^{(j)}_{\mathrm{dir}(u,v)}\mathbf{h}_u^{(j)} + \mathbf{b}^{(j)}_{\mathrm{dir}(u,v)}\Big) \\ &= \rho\Big(\sum_{u\in\mathcal{N}(v)} \mathrm{Linear}(\mathbf{h}_u^{(j)})\Big) \\ &= \rho(A_{u,:}\cdot \mathrm{Linear}(H^{(j)})) \in \mathbb{R}^d\end{aligned}$$

（邻接矩阵中u对应的行）

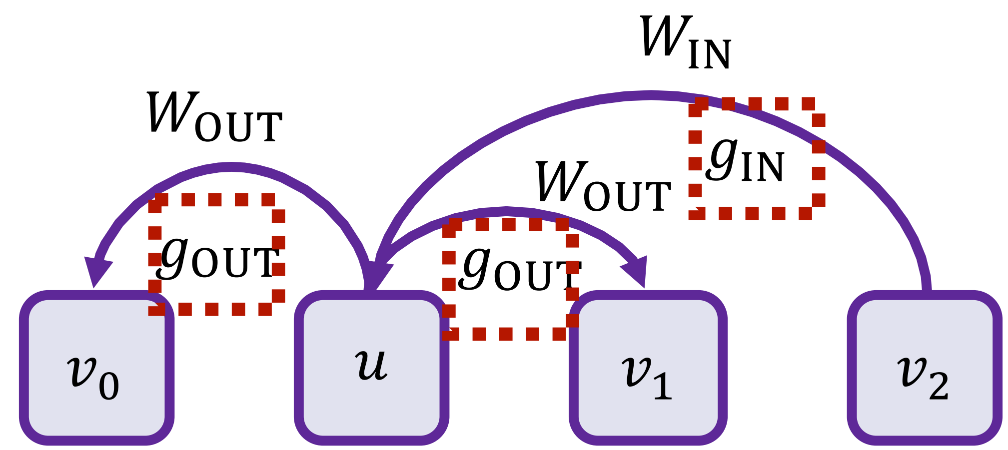
$$H^{(j)} = [\mathbf{h_0}^{(j)}\ \mathbf{h_1}^{(j)}\ \cdots\ \mathbf{h_u}^{(j)}\ \cdots\ \mathbf{h_{n-1}}^{(j)}]^{\mathrm{T}} \in \mathbb{R}^{n\times d}$$

$$H^{(j+1)} = \rho(A\mathrm{Linear}(H^{(j)}))$$

$$H^{(j+1)} = \rho(A_{\mathrm{IN}}\mathrm{Linear}(H^{(j)}) + A_{\mathrm{OUT}}\mathrm{Linear}(H^{(j)}))$$



图卷积神经网络编程实现



核心公式：
$$\mathbf{h}_v^{(j+1)} = \rho\Big(\sum_{u \in \mathcal{N}(v)} \mathbf{W}^{(j)}_{\mathrm{dir}(u,v)} \mathbf{h}_u^{(j)} + \mathbf{b}^{(j)}_{\mathrm{dir}(u,v)}\Big), \qquad \mathbf{W}_{dir(u,v)} \in \{\mathbf{W}_{\mathrm{IN}}, \mathbf{W}_{\mathrm{OUT}}\}$$

核心公式：
（加入门控机制）
$$\mathbf{h}_v^{(j+1)} = \rho\Big(\sum_{u \in \mathcal{N}(v)} \boxed{g_{u,v}^{(j)}}\big(\mathbf{W}^{(j)}_{\mathrm{dir}(u,v)} \mathbf{h}_u + \mathbf{b}^{(j)}_{\mathrm{dir}(u,v)}\big)\Big) \in \mathbb{R}^d$$

$$g_{u,v}^{(j)} = \sigma\Big(\mathbf{h}_u^{(j)} \cdot \hat{\mathbf{w}}^{(j)}_{\mathrm{dir}(u,v)} + \hat{b}^{(j)}_{\mathrm{dir(u,v)}}\Big) \in \mathbb{R}$$

图卷积神经网络编程实现

（加入门控机制）

$$\mathbf{h}_v^{(j+1)} = \rho\Big(\sum_{u\in\mathcal{N}(v)} g_{u,v}^{(j)}\big(\mathbf{W}_{\mathrm{dir}(u,v)}^{(j)}\mathbf{h}_u + \mathbf{b}_{\mathrm{dir}(u,v)}^{(j)}\big)\Big) \in \mathbb{R}^d$$

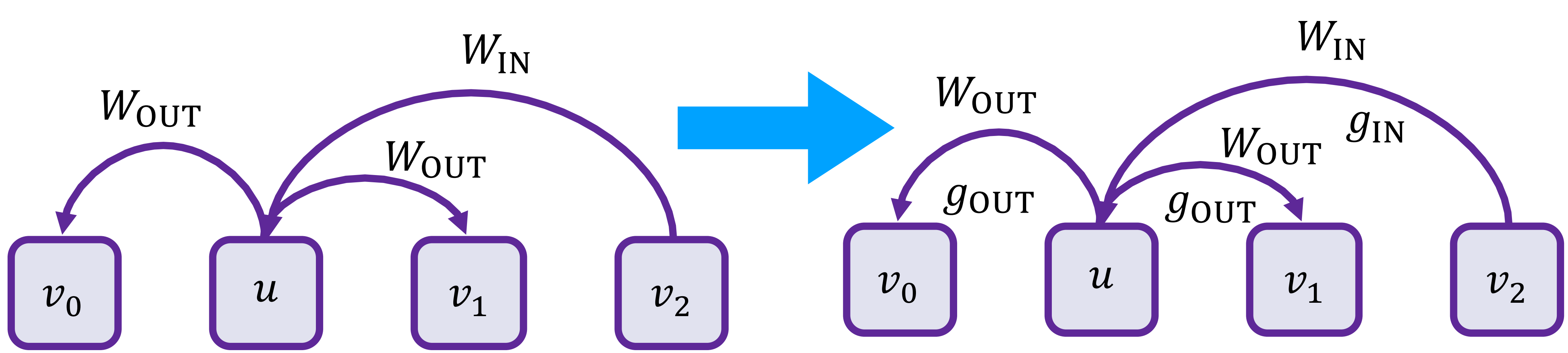
$$g_{u,v}^{(j)} = \sigma\Big(\mathbf{h}_u^{(j)} \cdot \hat{\mathbf{w}}_{\mathrm{dir}(u,v)}^{(j)} + \hat{b}_{\mathrm{dir(u,v)}}^{(j)}\Big) \in \mathbb{R}$$

$$H^{(j+1)} = \rho(\mathrm{Gate}(A)\mathrm{Linear}(H^{(j)}))$$

（邻接矩阵中u对应的行）

$$H^{(j)} = [\mathbf{h_0}^{(j)}\ \mathbf{h_1}^{(j)}\ \cdots\ \mathbf{h_u}^{(j)}\ \cdots\ \mathbf{h_{n-1}}^{(j)}]^{\mathrm{T}} \in \mathbb{R}^{n\times d}$$

$$H^{(j+1)} = \rho(\mathrm{Gate}(A_{\mathrm{IN}})\mathrm{Linear}(H^{(j)}) + \mathrm{Gate}(A_{\mathrm{OUT}})\mathrm{Linear}(H^{(j)}))$$



图卷积神经网络编程实现

GCN一般不超过两层：

$$H^{(j+1)} = \rho(A\text{Linear}(H^{(j)})) \Longrightarrow H^{(j+1)} = \rho(AH^{(j)}W) \Longrightarrow H^{(j+1)} = AH^{(j)}W$$

$$\Longrightarrow \quad \begin{aligned} H^{(n)} &= A^n H^{(0)} W^n \\ &= A^n H^{(0)} W^n \\ &= [\boxed{A^n\mathbf{f_0}}\; A^n\mathbf{f_1} \;\cdots\; A^n\mathbf{f_{d-1}}]W^n \end{aligned}$$

$$H^{(j)} = [\mathbf{h_0}^{(j)} \;\cdots\; \mathbf{h_u}^{(j)} \;\cdots\; \mathbf{h_{n-1}}^{(j)}]^{\mathrm{T}} \in \mathbb{R}^{n\times d}$$

$$= \begin{bmatrix} -\mathbf{h_0}^{(j)\mathrm{T}}- \\ \vdots \\ -\mathbf{h_u}^{(j)\mathrm{T}}- \\ \vdots \\ -\mathbf{h_{n-1}}^{(j)\mathrm{T}}- \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \mathbf{f_0} & \mathbf{f_1} & \cdots & \mathbf{f_{d-1}} \\ | & | & & | \end{bmatrix},$$

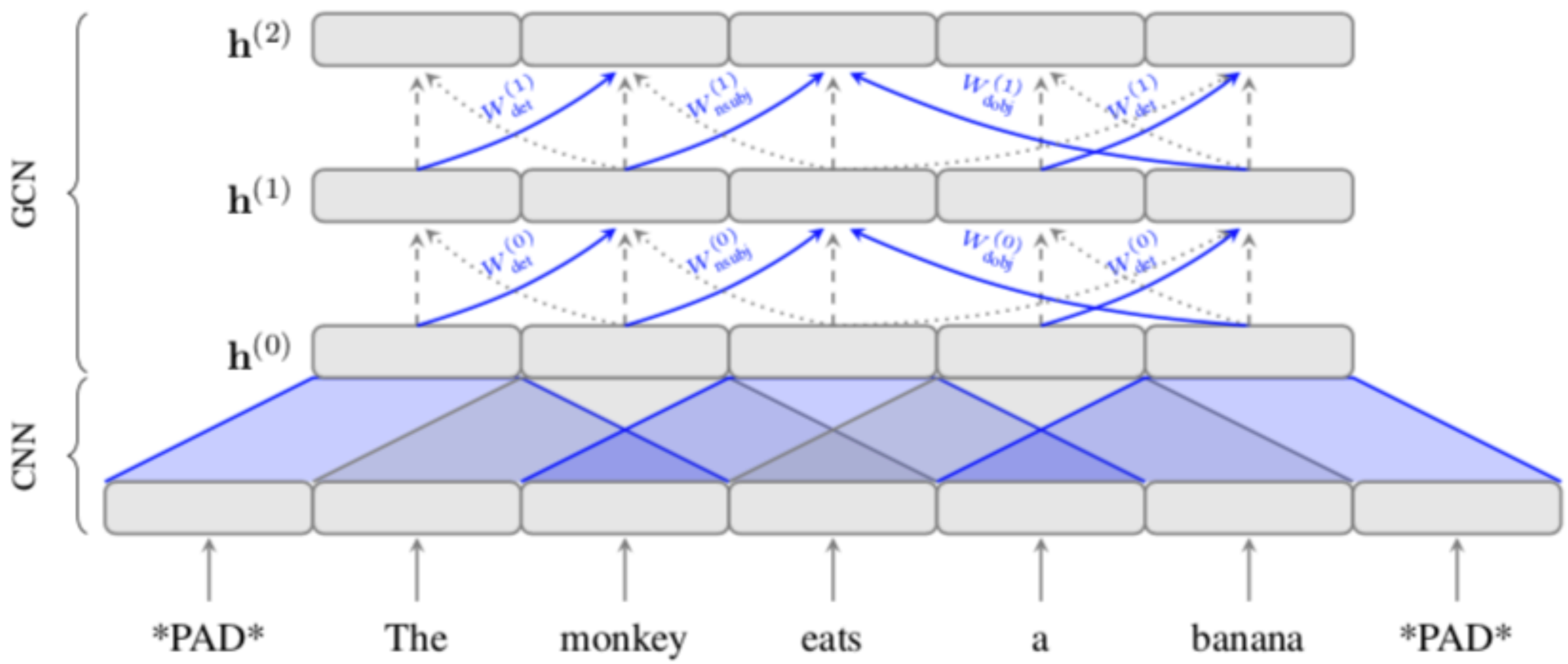
$\mathbf{f_i} \in \mathbb{R}^n$：所有节点的第$i$维特征

$$\Longrightarrow \quad A^n\mathbf{f_0}^{(n)} = A^n\mathbf{f_1}^{(n)} = \cdots = A^n\mathbf{f_{n-1}}^{(n)}, \; n \gg 0$$

∴ 各维特征无法区分

Li, Q., Han, Z., & Wu, X. M. (2018, April). Deeper insights into graph convolutional networks for semi-supervised learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

应用：机器翻译



	Train	Val.	Test
English-German	226822	2169	2999
English-German (full)	4500966	2169	2999
English-Czech	181112	2656	2999

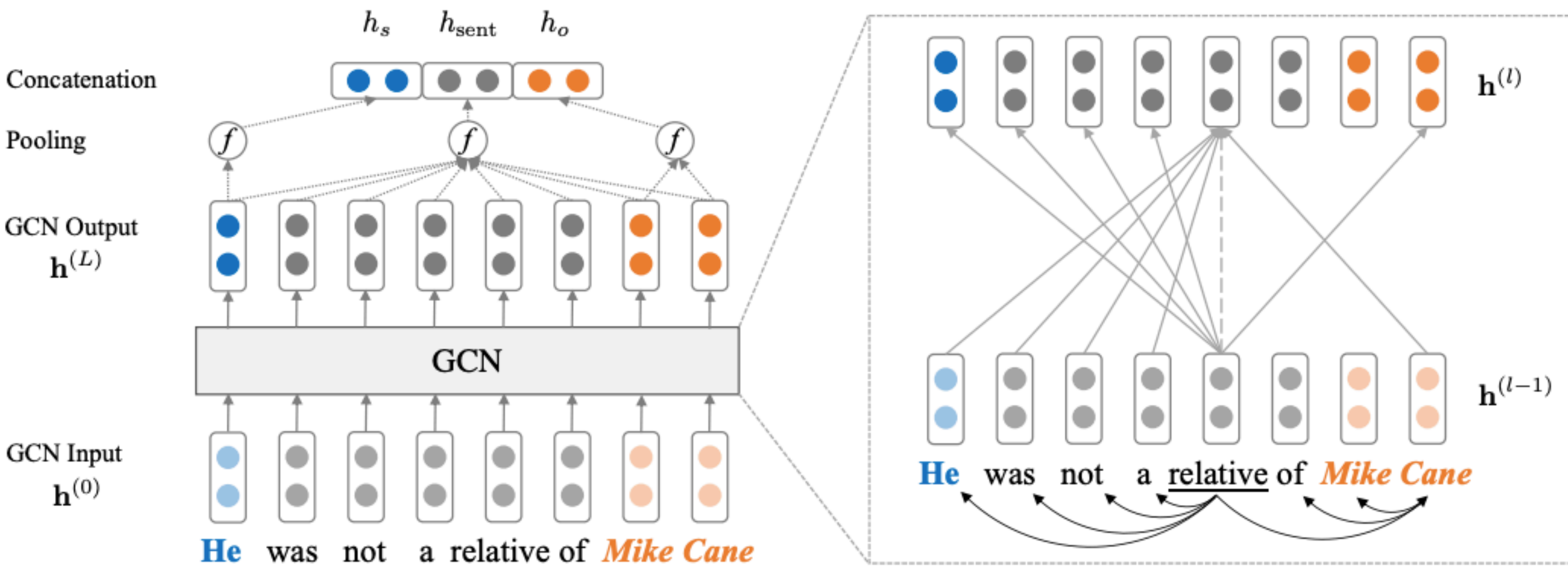
Table 1: The number of sentences in our data sets.

	Kendall	BLEU_1	BLEU_4
BoW	0.3352	40.6	9.5
+ GCN	0.3520	44.9	12.2
CNN	0.3601	42.8	12.6
+ GCN	0.3777	44.7	13.7
BiRNN	0.3984	45.2	14.9
+ GCN	0.4089	47.5	16.1
BiRNN (full)	0.5440	53.0	23.3
+ GCN	0.5555	54.6	23.9

Table 3: Test results for English-German.

Joost Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, Khalil Sima'an: **Graph Convolutional Encoders for Syntax-aware Neural Machine Translation.** EMNLP 2017: 1957-1967

应用：关系分类



TACRED：

System	P	R	F_1
LR† (Zhang+2017)	<u>73.5</u>	49.9	59.4
SDP-LSTM† (Xu+2015b)	66.3	52.7	58.7
Tree-LSTM‡ (Tai+2015)	66.0	59.2	62.4
PA-LSTM† (Zhang+2017)	65.7	<u>64.5</u>	65.1
GCN	69.8	59.0	64.0
C-GCN	69.9	63.3	<u>66.4</u>*
GCN + PA-LSTM	71.7	63.0	67.1*
C-GCN + PA-LSTM	71.3	**65.4**	**68.2***

SemEval：

System	*with-m*	*mask-m*
SVM† (Rink+2010)	82.2	–
SDP-LSTM† (Xu+2015b)	83.7	–
SPTree† (Miwa+2016)	84.4	–
PA-LSTM‡ (Zhang+2017)	82.7	75.3
Our Model (C-GCN)	**84.8***	**76.5***

Yuhao Zhang, Peng Qi, Christopher D. Manning:
Graph Convolution over Pruned Dependency Trees Improves Relation Extraction. EMNLP 2018: 2205-2215