A Graph Convolutional Network with Adaptive Graph Generation and Channel Selection for Event Detection

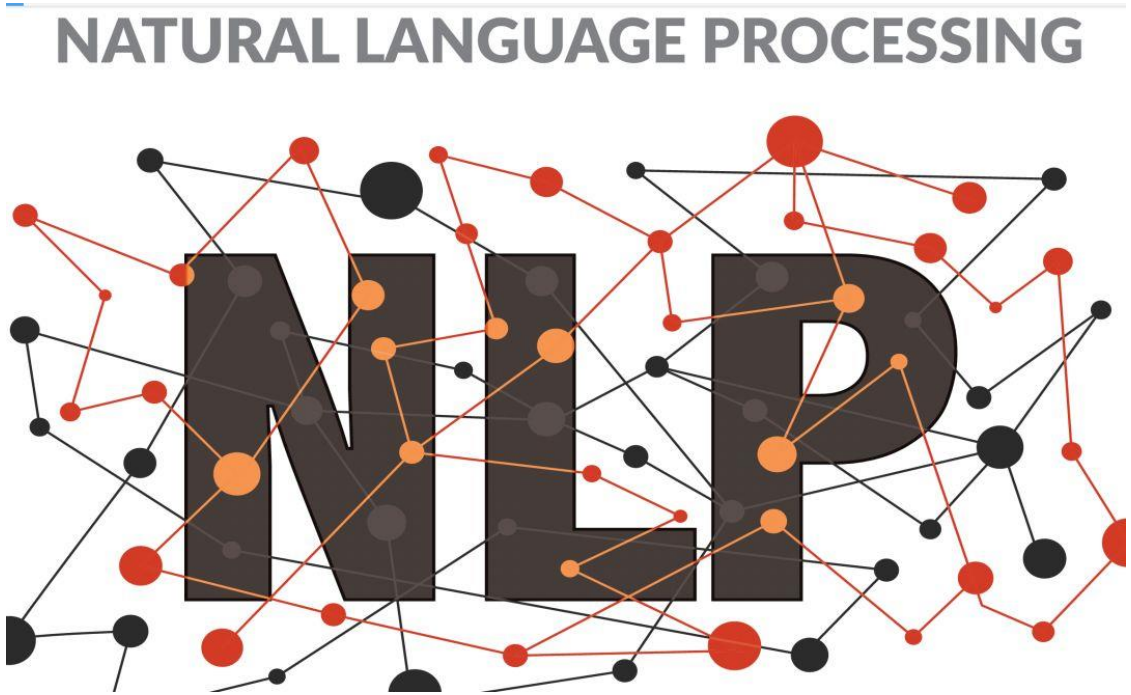
Zhipeng Xie,[1] **Yumin Tu** [1]

[1] School of Computer Science
Shanghai Key Laboratory of Data Science
Fudan University, Shanghai, China
xiezp@fudan.edu.cn

2022.07.10 • ChongQing

2022_AAAI

Reported by Yidan Liu



1.Introduction

2.Method

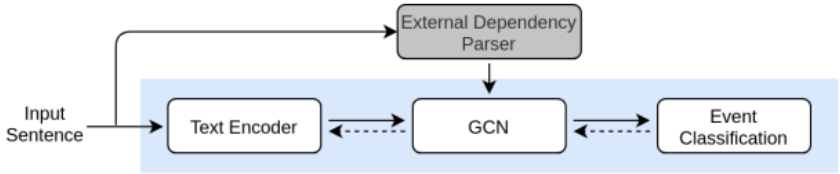
3.Experiments

Introduction

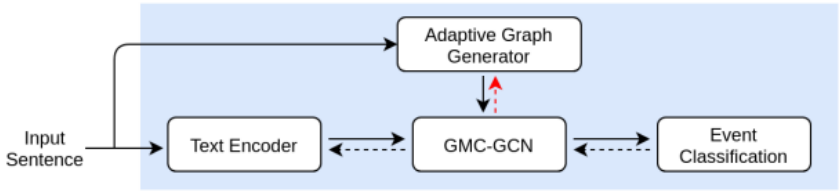
- Existing works rely heavily on a fixed syntactic parse tree structure from an external parser.

- In addition, the information content extracted for aggregation is determined simply by the (syntactic) edge direction or type but irrespective of what semantics the vertices have, which is somewhat rigid.

Figure 1: An example sentence of event type *Life:Be-Born*.



(a) Traditional GCN-based ED methods



(b) The proposed AGGED method

Figure 2: Modular structures of GCN-based ED methods, where the solid arrows denote the forward pass of information, and the dashed arrows denotes the backpropagation of gradients.

Method

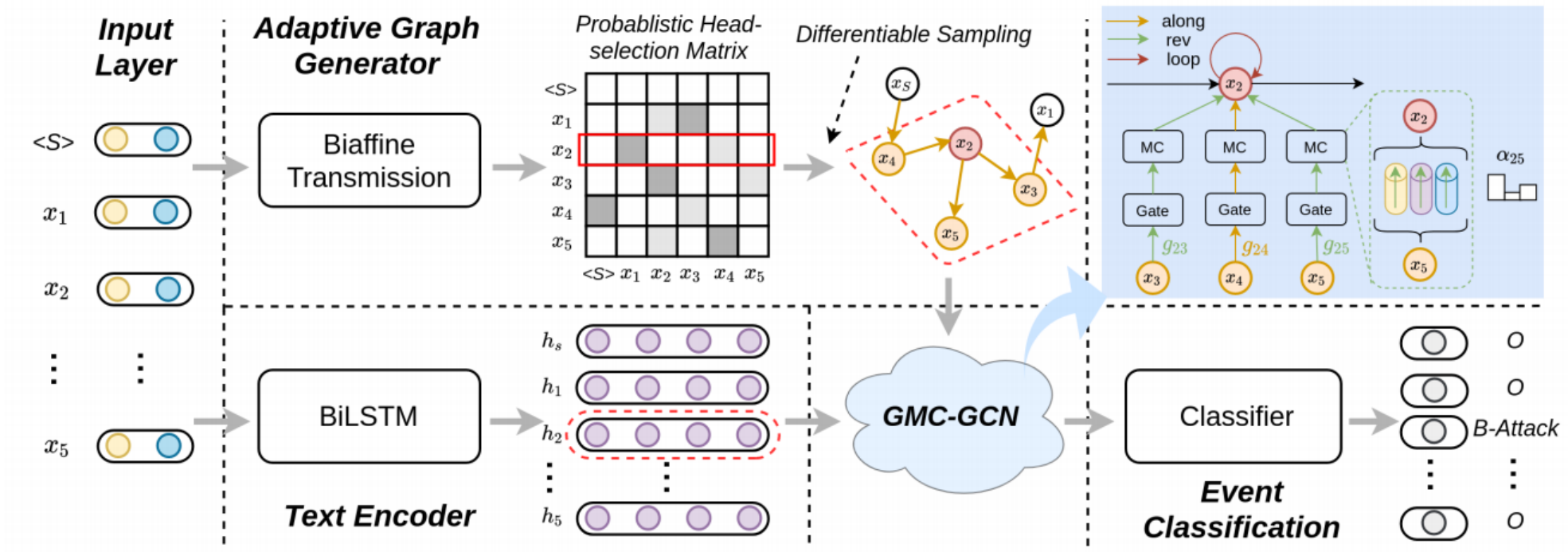
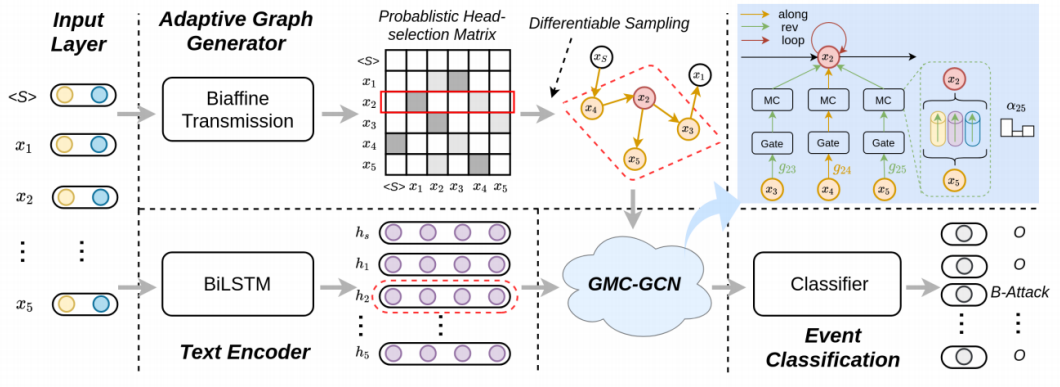


Figure 3: Model architecture. After the input layer, the initial vector representation of input sentence is fed into two modules: one is the adaptive graph generator that outputs a discrete graph structure based on Deep Biaffine Attention and Gumbel-Softmax sampling; the other is the text encoder implemented by a BiLSTM network that generates node embeddings (or equivalently, the contextualized embeddings of words). In turn, both the graph structure and the node embeddings flow into the proposed GMC-GCN module for aggregating information from one-hop neighborhoods. Finally, the updated representations are used for event classification as sequence labeling.

Method

$$S = [w_1, w_2, ..., w_n] \qquad \mathbf{x}_i = [\mathbf{w}_i; \mathbf{p}_i]$$

$$[\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n]$$



Deep biaffine attention for PHS matrix

Next, two multi-layer perceptrons(MLPs) are used to obtain the representations of word w_i being the head or the dependent in any dependency relation:

$$\mathbf{h}_i^{\text{head}} = \mathbf{MLP}_{\text{arc}}^{\text{head}}(\mathbf{r}_i) \tag{1}$$

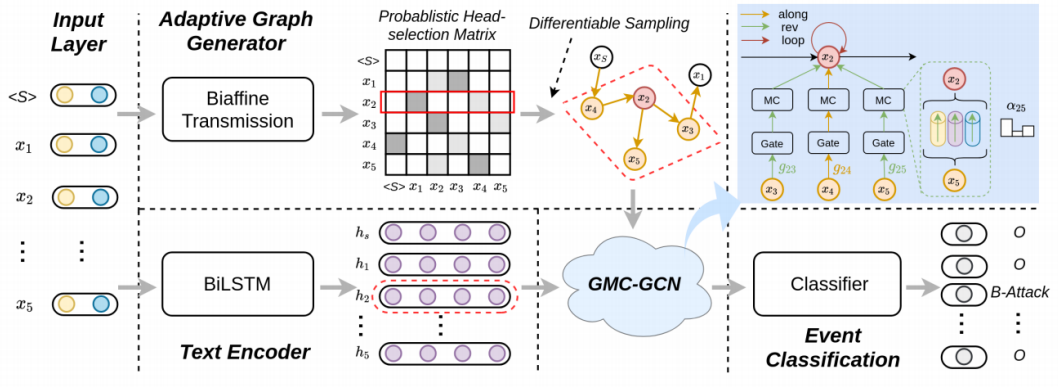
$$\mathbf{h}_i^{\text{dep}} = \mathbf{MLP}_{\text{arc}}^{\text{dep}}(\mathbf{r}_i) \tag{2}$$

$$\mathbf{s}_i = \left(\mathbf{H}^{\text{head}}\right)^{\top} \mathbf{U}_{\text{arc}} \mathbf{h}_i^{\text{dep}} + \left(\mathbf{H}^{\text{head}}\right)^{\top} \mathbf{v}_{\text{arc}} \tag{3}$$

$$p_{i,j} = \frac{\exp\left(s_{i,j}\right)}{\sum_{k=1}^{n} \mathbb{I}_{[k \neq i]} \exp\left(s_{i,k}\right)} \tag{4}$$

where $\mathbb{I}_{[k \neq i]} \in \{0, 1\}$ is an indicator function that equals 1 only if $k \neq i$ and $p_{i,j}$ represents the probability of word w_j being the head of word w_i.

Method



ST-Gumbel-Softmax trick for sparse-graph sampling

$$\mathbf{a}_i = \text{one-hot}\left(\arg\max_{k \neq i}[\log p_{i,k} + g_{i,k}]\right) \tag{5}$$

$$^{1}g_{i,k} = -\log(-\log(u_{i,k})) \text{ and } u_{i,k} \sim \text{Uniform}(0,1)$$

$$\hat{a}_{i,k} = \frac{\exp((\log p_{i,k} + g_{i,k})/\tau)}{\sum_{k'=1}^{n} \mathbb{I}_{[k' \neq i]} \exp((\log p_{i,k'} + g_{i,k'})/\tau)} \tag{6}$$

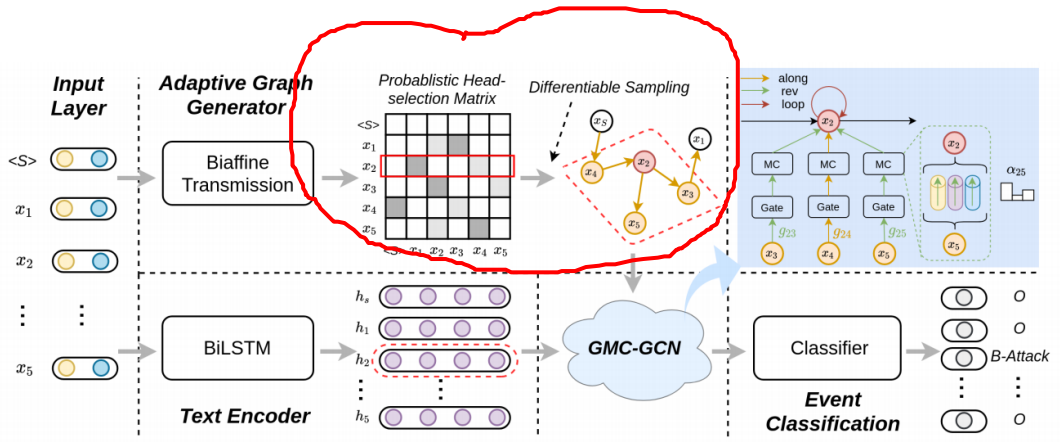
Vanilla GCN for event detection

$$\gamma_{i,j} = \gamma(w_i, w_j) = \begin{cases} 0 & \text{if } i = j \text{ (self-loop)} \\ 1 & \text{if } (w_i, w_j) \in \mathcal{E} \text{ (along)} \\ 2 & \text{if } (w_j, w_i) \in \mathcal{E} \text{ (rev)} \end{cases} \tag{7}$$

$$\mathbf{h}_i^{\text{conv}} = f\left(\sum_{(w_i,w_j) \in \mathcal{E}^*} \text{IC}_{\gamma_{i,j}}(\mathbf{h}_j)\right) \tag{8}$$

$$\text{IC}_c(\mathbf{h}_j) = \mathbf{W}_c \mathbf{h}_j + \mathbf{b}_c \tag{9}$$

Method



Gated-Multi-Channel GCN

$$\boldsymbol{\alpha}_{i,j} = \mathrm{softmax}\left(\mathbf{h}_i^\top \mathbf{U}_s^{\gamma_{i,j}} \mathbf{h}_j + (\mathbf{h}_i \oplus \mathbf{h}_j)^\top \mathbf{M}_s^{\gamma_{i,j}} + \mathbf{b}_s^{\gamma_{i,j}}\right) \tag{10}$$

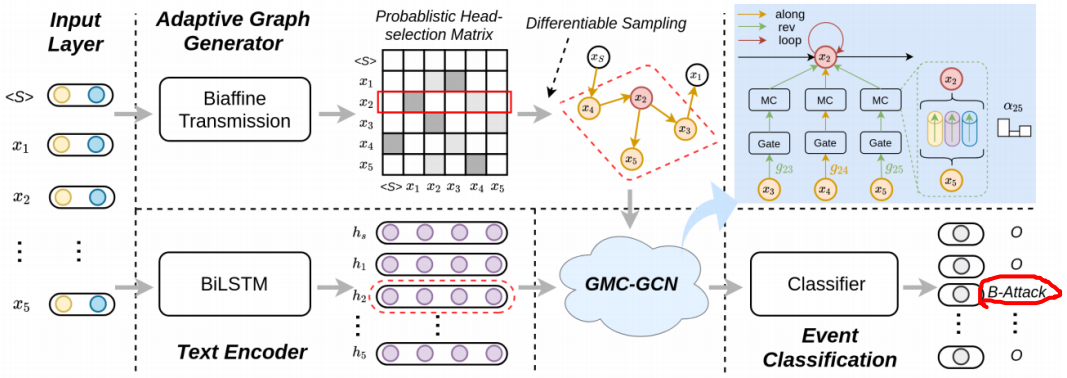
where $\boldsymbol{\alpha}_{i,j} \in \mathbb{R}^C$ represents the attention distribution over the C information channels, $\mathbf{U}_s^1, \mathbf{U}_s^2 \in \mathbb{R}^{d \times C \times d}$, $\mathbf{M}_s^1, \mathbf{M}_s^2 \in \mathbb{R}^{(2d) \times C}$ and $\mathbf{b}_s^1, \mathbf{b}_s^2 \in \mathbb{R}^C$ are parameters.

$$\mathbf{h}_i^{\mathrm{conv}} = f\left(\mathrm{IC}_0(\mathbf{h}_i) + \sum_{\substack{(w_i,w_j)\in\mathcal{E}^* \\ j\neq i}} \sum_{c=1}^{C} \alpha_{i,j,c} \cdot \mathrm{IC}_c(\mathbf{h}_j)\right) \tag{11}$$

$$g_{i,j} = \sigma\left((\mathbf{h}_i \oplus \mathbf{h}_j)^\top \mathbf{v}_g + b_g\right) \tag{12}$$

$$\mathbf{h}_i^{\mathrm{conv}} = f\left(\mathrm{IC}_0(\mathbf{h}_i) + \sum_{\substack{(w_i,w_j)\in\mathcal{E}^* \\ j\neq i}} g_{i,j} \sum_{c=1}^{C} \alpha_{i,j,c} \cdot \mathrm{IC}_c(\mathbf{h}_j)\right) \tag{13}$$

Method



Event Classification

$$\mathbf{y}_i = \text{softmax}\left(\mathbf{W}_o \mathbf{h}_i^{\text{conv}} + \mathbf{b}_o\right) \tag{14}$$

where $\mathbf{y}_i \in \mathbb{R}^{2N_e+1}$, and N_e denotes the number of event types.

Training

$$L = -\sum_{S \in \mathcal{D}} \sum_{i=1}^{n_S} \log y_{i,t_i} \tag{15}$$

Experiment

Model	P	R	F1
GCN-ED (Nguyen and Grishman 2018)	77.9	68.8	73.1
JMEE (Liu, Luo, and Huang 2018)	76.3	71.3	73.7
MOGANED (Yan et al. 2019)	**79.5**	72.3	75.7
EE-GCN (Cui et al. 2020)	76.7	78.6	77.6
DMBERT (Wang et al. 2019)	77.9	72.5	75.1
SS-VQ-VAE (Huang and Ji 2020)	75.7	77.8	76.7
GatedGCN (Lai, Nguyen, and Nguyen 2020)	78.8	76.3	77.6
EKD (Tong et al. 2020)	79.1	78.0	78.6
Our AGGED Method	77.8	**82.2**	**79.9**

Table 1: Performance comparison with existing event detection methods.

Experiment

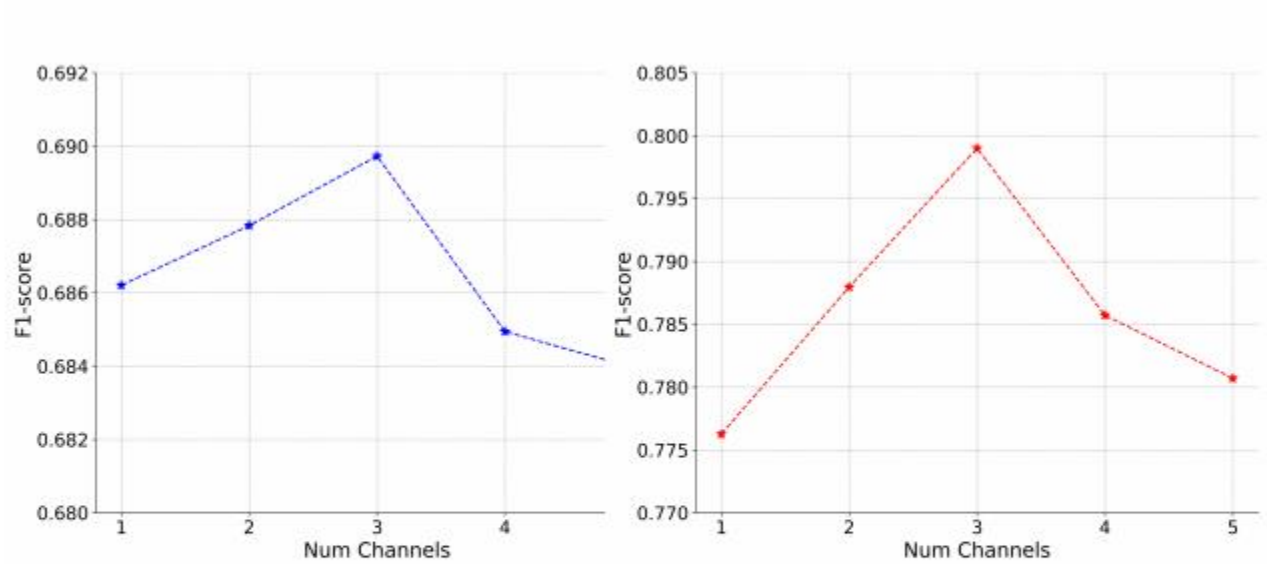
Model	P	R	F1
Full AGGED Model	77.8	82.2	79.9
–MC	77.9	81.0	79.4
–IG	77.5	80.0	78.7
–MC & IG	77.8	80.3	79.0
–AG	76.8	79.9	78.3

Table 2: Ablation study on ACE test set.

	P	R	F1
without burn-in	75.7	80.3	77.9
with burn-in	77.8	82.2	79.9

Table 3: Performance of our method with or without the burn-in phase for the adaptive graph generation module

Experiment



(a) On validation data (b) On test data

Figure 4: Event dection performance with respect to the number of information channels

Experiment

	Sentence	Prediction	Ground-Truth
Dependency Parser	Mirjana Markovic, the power behind the scenes during Milosevic's 13-year reign, is *accused of illegally* **providing** their grandson's *nanny* with a state-owned luxury apartment in Belgrade in *2000*.	✗ None	Transfer-Ownership
Adaptive Generator	Mirjana Markovic, the power behind the scenes during Milosevic's 13-year reign, is accused *of illegally* **providing** their grandson's nanny with a state-owned luxury *apartment* in *Belgrade* in *2000*.	✓ Transfer-Ownership	

Table 4: A case study of the graph structures from the dependency parser and the adaptive graph generator, where the trigger word "*providing*" is marked in red color, and its adjacent words in the graph structures are marked in blue color.

Thank you!