Detect Rumors in Microblog Posts for Low-Resource Domains via Adversarial Contrastive Learning

Hongzhan Lin[1,2], Jing Ma[2,*], Liangliang Chen[1], Zhiwei Yang[3], Mingfei Cheng[1], Guang Chen[1,*]
[1]Beijing University of Posts and Telecommunications, Beijing, China
[2]Hong Kong Baptist University, Hong Kong SAR, China
[3]Jilin University, Changchun, China
{linhongzhan,outside,mingfeicheng,chenguang}@bupt.edu.cn
majing@comp.hkbu.edu.hk, yangzw18@mails.jlu.edu.cn

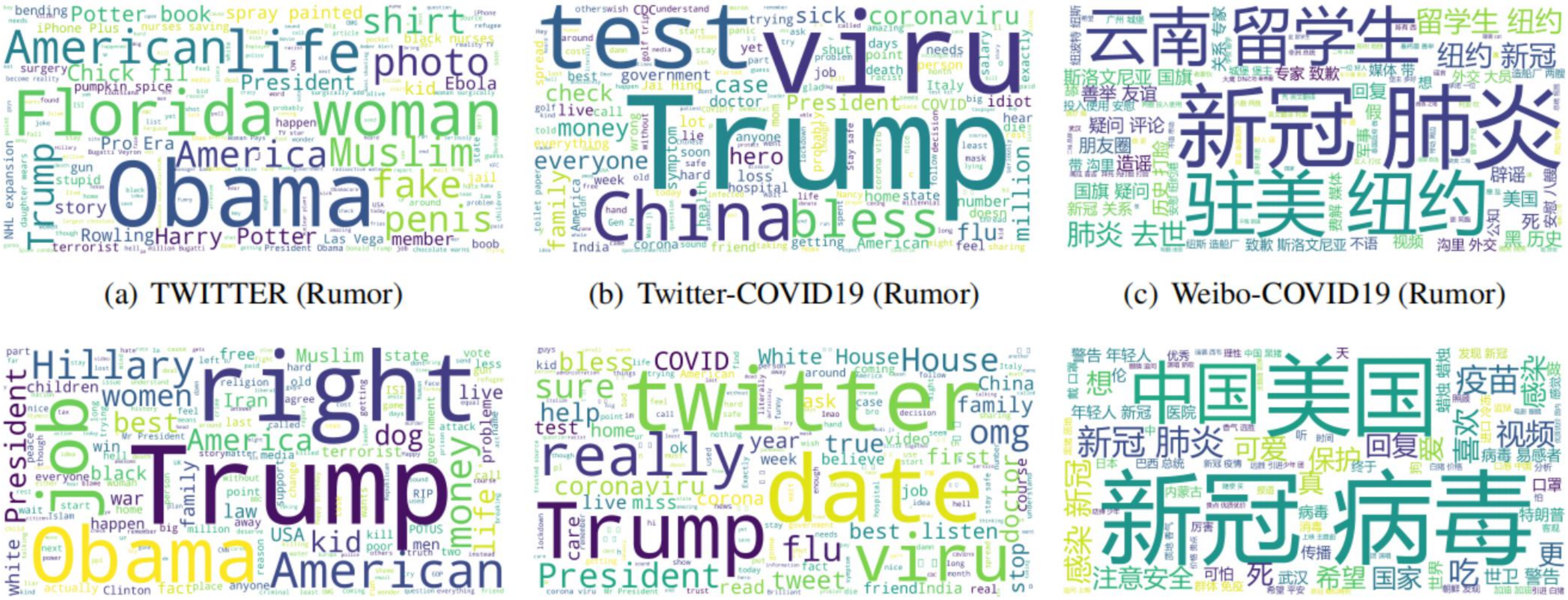
NAACL2022

code: https://github.com/DanielLin97/ACLR4RUMOR-NAACL2022

2022. 5. 22

Reported by Xiaoke Li

Introduction

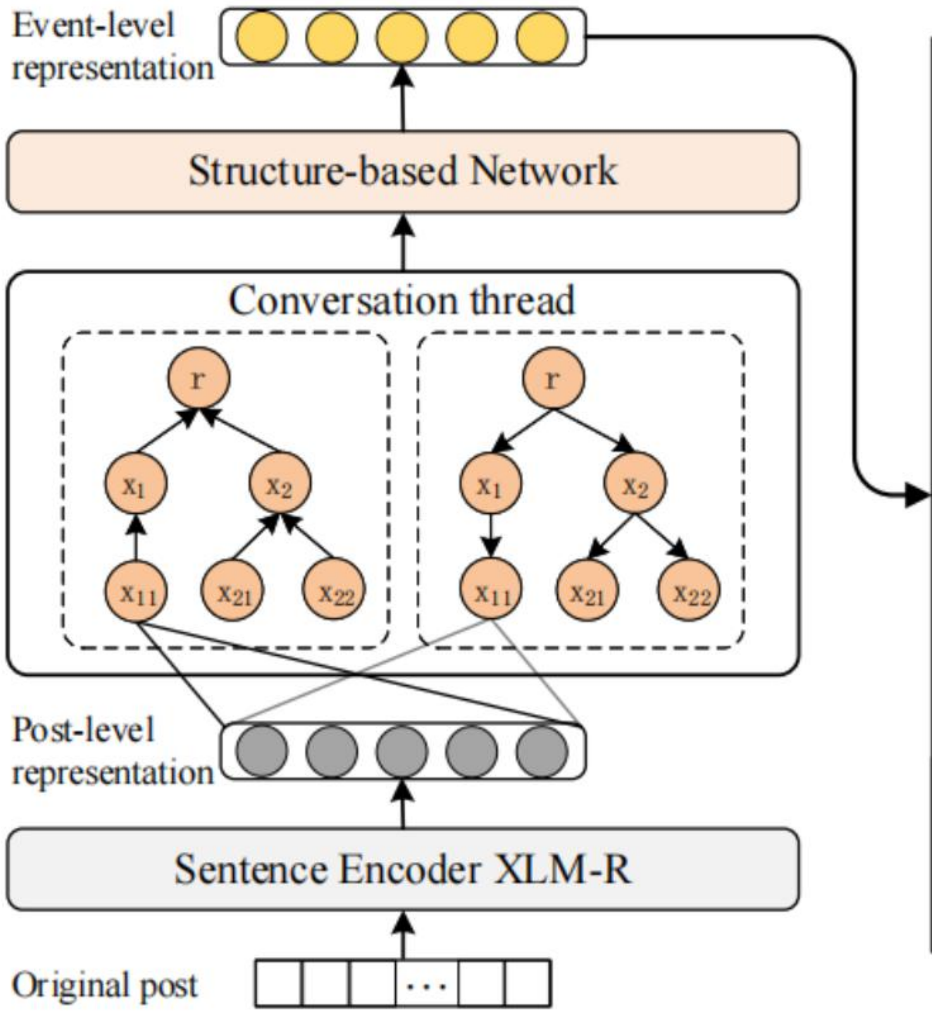


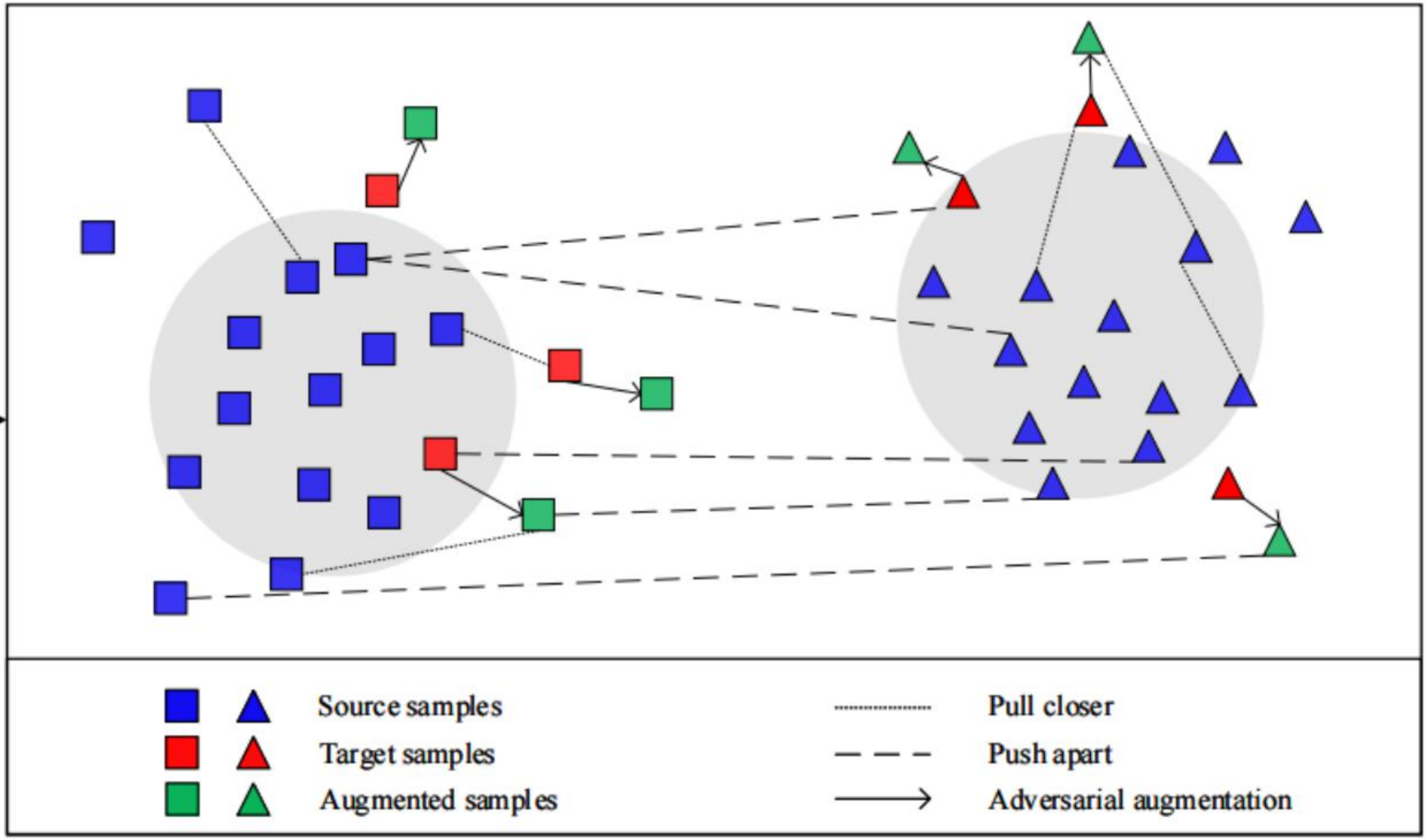
(a) TWITTER (Rumor) (b) Twitter-COVID19 (Rumor) (c) Weibo-COVID19 (Rumor)

(d) TWITTER (Non-rumor) (e) Twitter-COVID19 (Non-rumor) (f) Weibo-COVID19 (Non-rumor)

Figure 1: Word clouds of rumor and non-rumor data generated from TWITTER, Twitter-COVID19, and Weibo-COVID19 datasets, where the size of terms corresponds to the word frequency. Both TWITTER and Twitter-COVID19 are presented in English while Weibo-COVID19 in Chinese.

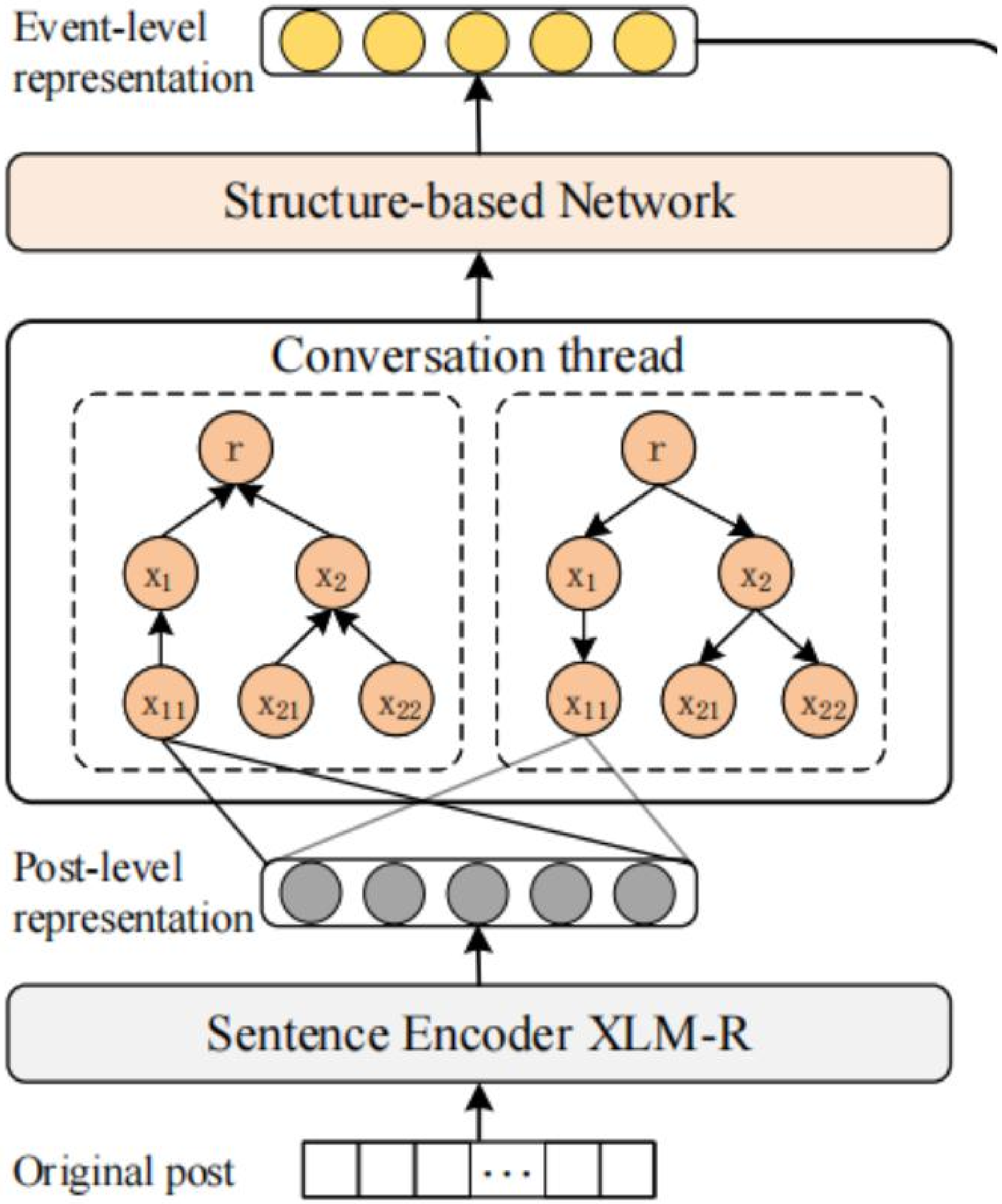
Method





Adversarial Contrastive Training Paradigm

Method



$$\mathcal{D}_s = \{C_1^s, C_2^s, \cdots, C_M^s\} C^s = (y, c, \mathcal{T}(c))$$

$$\mathcal{T}(c) = \{c, x_1^s, x_2^s, \cdots, x_{|C|}^s\}$$

$$\mathcal{D}_t = \{C_1^t, C_2^t, \cdots, C_N^t\}, \text{ where } N (N \ll M)$$

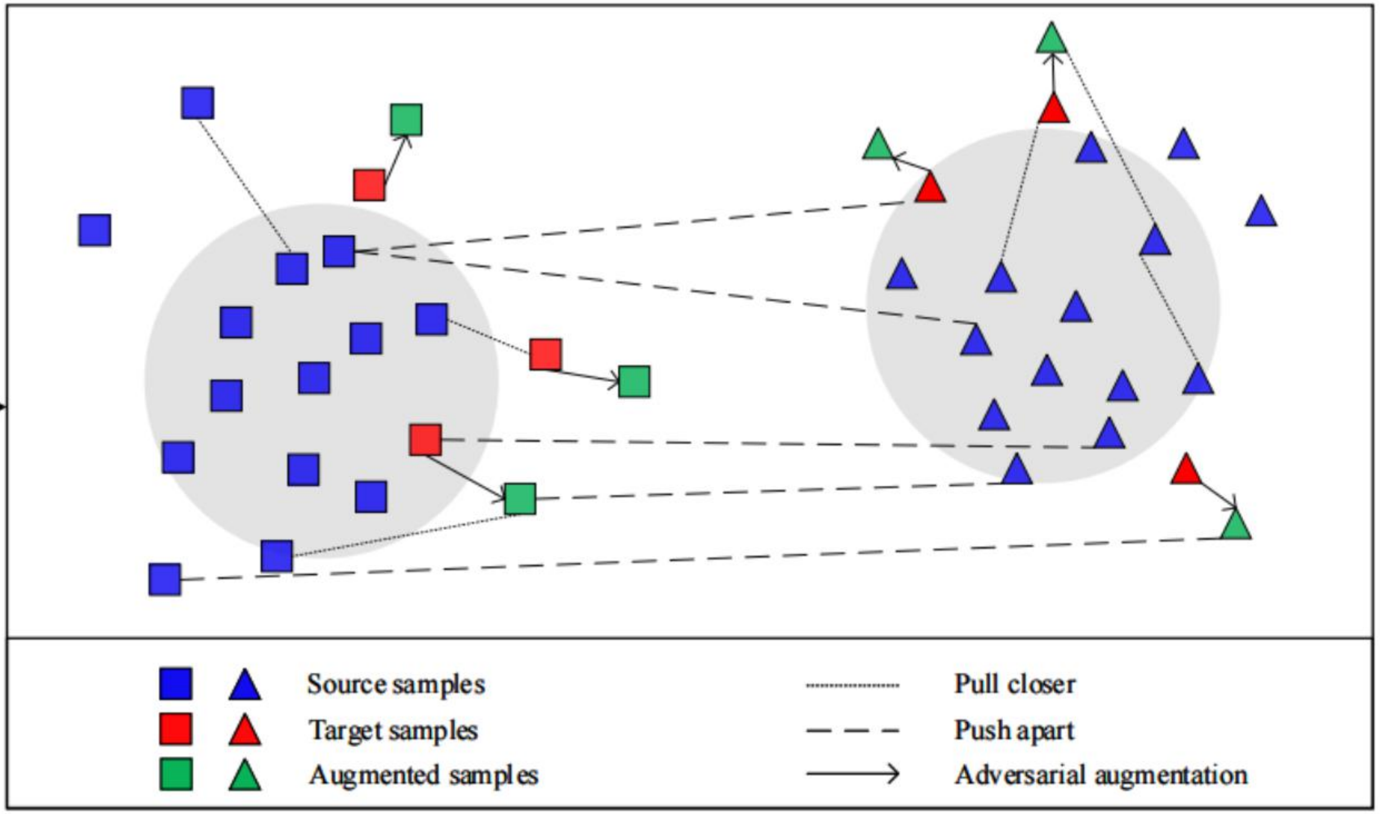
$$\bar{x} = \textit{XLM-R}(\mathbf{x}) \quad (1)$$

$$X^* = [\bar{x}_0^*, \bar{x}_1^*, \bar{x}_2^*, \ldots, \bar{x}_{|X^*|-1}^*]^\top; * \in \{s, t\}$$

$$H^{(l+1)} = ReLU(\hat{\mathbf{A}} \cdot H^{(l)} \cdot W^{(l)}) \quad (2)$$

$$o = \textit{mean-pooling}([H_{TD}; H_{BU}]) \quad (3)$$

Method



Adversarial Contrastive Training Paradigm

$$\mathcal{L}_{CE}^{s} = -\frac{1}{N^s}\sum_{i=1}^{N^s} log(p_i) \quad (4)$$

$$\mathcal{L}_{SCL}^{s} = -\frac{1}{N^s}\sum_{i=1}^{N^s}\frac{1}{N_{y_i^s}-1}\sum_{j=1}^{N^s}\mathbb{1}_{[i\neq j]}\mathbb{1}_{[y_i^s=y_j^s]} log\frac{exp(\mathrm{sim}(o_i^s,o_j^s)/\tau)}{\sum_{k=1}^{N^s}\mathbb{1}_{[i\neq k]}exp(\mathrm{sim}(o_i^s,o_k^s)/\tau)} \quad (5)$$

$$\mathcal{L}_{SCL}^{t} = -\frac{1}{N^t}\sum_{i=1}^{N^t}\frac{1}{N_{y_i^t}}\sum_{j=1}^{N^s}\mathbb{1}_{[y_i^t=y_j^s]} log\frac{exp(\mathrm{sim}(o_i^t,o_j^s)/\tau)}{\sum_{k=1}^{N^s}exp(\mathrm{sim}(o_i^t,o_k^s)/\tau)} \quad (6)$$

$$\tilde{o}_{noise}^{t} = \epsilon\frac{g}{||g||}; \text{where } \ g = \nabla_{o^t}\mathcal{L}_{CE}^{t} \quad (7)$$

$$\mathcal{L}^{*} = (1-\alpha)\mathcal{L}_{CE}^{*} + \alpha\mathcal{L}_{SCL}^{*}; * \in \{s,t\} \quad (8)$$

Experiments

Target (Source)	Weibo-COVID19 (TWITTER)				Twitter-COVID19 (WEIBO)			
Model	Acc.	Mac-F_1	Rumor F_1	Non-rumor F_1	Acc.	Mac-F_1	Rumor F_1	Non-rumor F_1
CNN	0.445	0.402	0.476	0.328	0.498	0.389	0.528	0.249
RNN	0.463	0.414	0.498	0.329	0.510	0.388	0.533	0.243
RvNN	0.514	0.482	0.538	0.426	0.540	0.391	0.534	0.247
PLAN	0.532	0.496	0.578	0.414	0.573	0.423	0.549	0.298
BiGCN	0.569	0.508	0.586	0.429	0.616	0.415	0.577	0.252
DANN-RvNN	0.583	0.498	0.591	0.405	0.577	0.482	0.648	0.317
DANN-PLAN	0.601	0.507	0.606	0.409	0.593	0.471	0.574	0.369
DANN-BiGCN	0.629	0.561	0.616	0.506	0.618	0.510	0.676	0.344
ACLR-RvNN	0.778	0.716	0.843	0.589	0.653	0.616	0.710	0.521
ACLR-PLAN	0.824	0.769	0.842	0.696	0.709	0.648	0.752	0.544
ACLR-BiGCN	**0.873**	**0.861**	**0.896**	**0.827**	**0.765**	**0.686**	**0.766**	**0.605**

Table 1: Rumor detection results on the target test datasets.

Experiments

Model	Weibo-COVID19		Twitter-COVID19	
	Acc.	Mac-F_1	Acc.	Mac-F_1
BiGCN(T)	0.569	0.508	0.616	0.415
BiGCN(S)	0.578	0.463	0.611	0.425
BiGCN(S,T)	0.693	0.472	0.617	0.471
DANN-BiGCN	0.629	0.561	0.618	0.510
CLR-BiGCN	0.844	0.804	0.719	0.618
ACLR-BiGCN	0.873	0.861	0.765	0.686

Table 2: Ablation studies on our proposed model.

Experiments

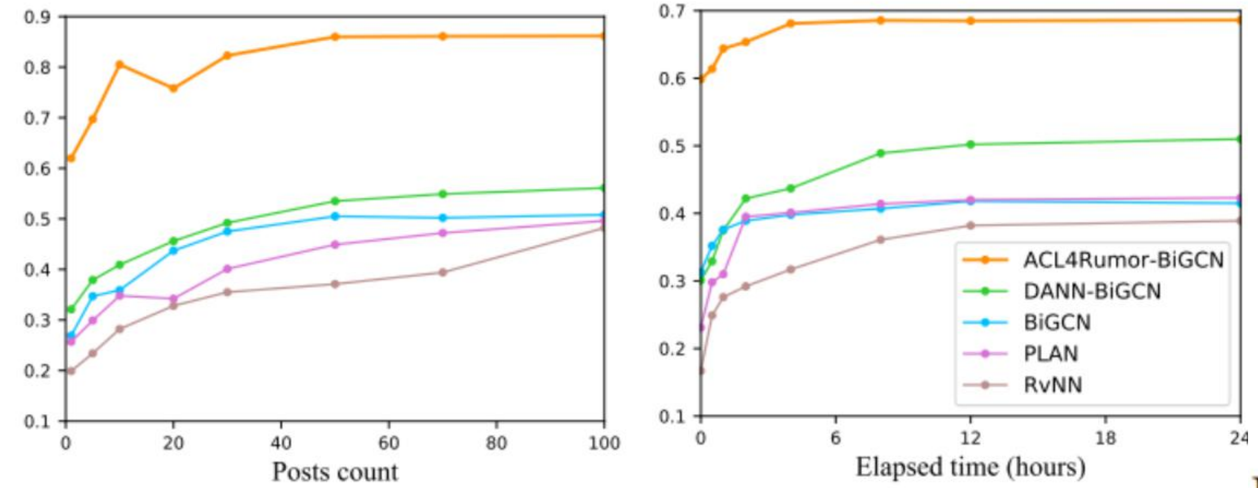


Figure 3: Early detection performance at different checkpoints of posts count (or elapsed time) on Weibo-COVID19 (left) and Twitter-COVID19 (right) datasets.

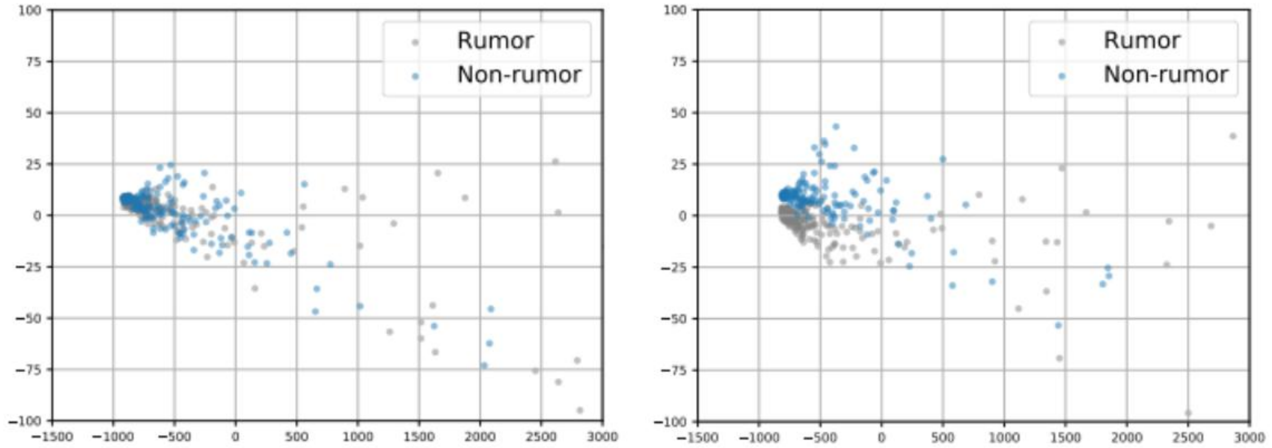


Figure 4: Visualization of target event-level representation distribution.

Thanks