Semi-supervised Stance Detection of Tweets Via Distant Network Supervision

[1]Subhabrata Dutta, [2]Samiya Caur, [3]Soumen Chakrabarti, [2]Tanmoy Chakraborty
[1]Jadavpur University, India; [2] IIIT-Delhi, India; [3] IIT Bombay, India

（WSDM-2022)

code : https://github.com/Annon-arch/SANDS

Reported by Zhaoze Gao

1. Introduction
2. Approach
3. Experiments

gesis
Leibniz-Institut für Sozialwissenschaften

Forschungszentrum · Research Center
L3S

Introduction

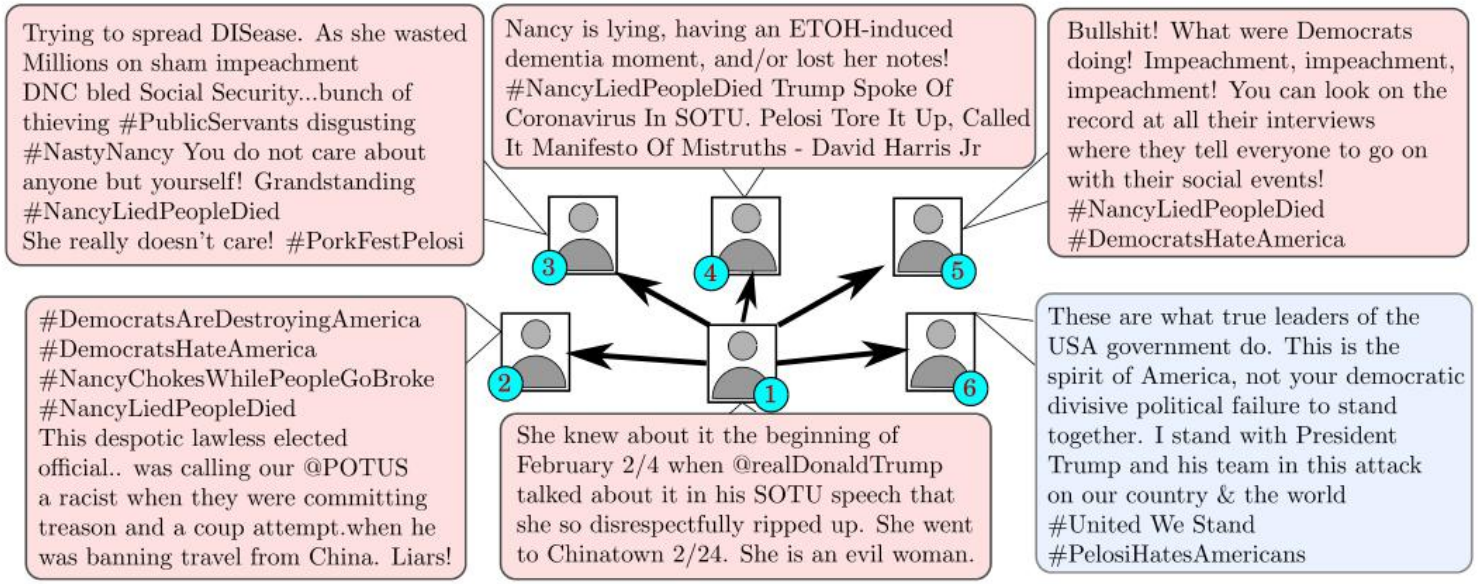

Figure 1: Stance homophily in Twitter. User 1 follows users 2, 3, 4, 5, and 6. All these users carry similar opinion, with user 6's tweet showing support to the Republicans while the rest are anti-Democrat. While the tweet posted by user 1 does not link any entity related to Republican or Democrats directly to some polarity words (thereby making the stance classification difficult), a classification framework with the knowledge of the rest of the tweets can break the ambiguity.

Approach

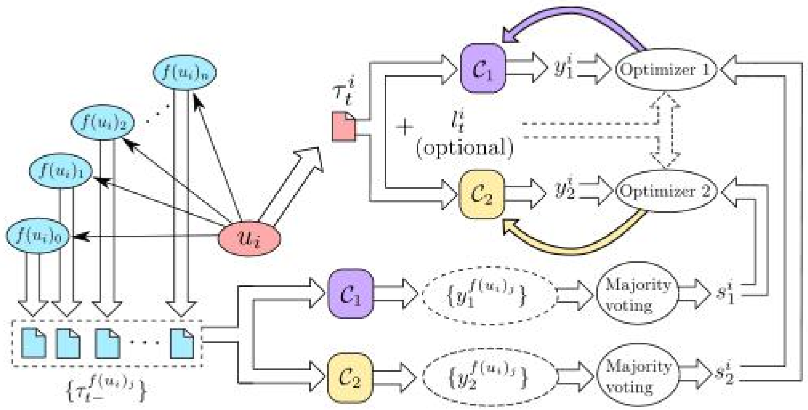

Figure 2: Training process of SANDS. Classifiers C_1 and C_2 independently predict the stance for a tweet from user u_i. In parallel, the classifiers generate pseudo-label sets $\{y_1^{f(u_i)_j}\}$ and $\{y_1^{f(u_i)_j}\}$, respectively from the recent tweets by followees of u_i. After majority voting, lables s_1^i and s_2^i are selected from pseudo-labels. C_1 is optimized using the label generated by C_2, and vice versa. Additionally, annotated labels l_t^i (if present) of the tweet are also used to compute loss and optimize the classifiers.

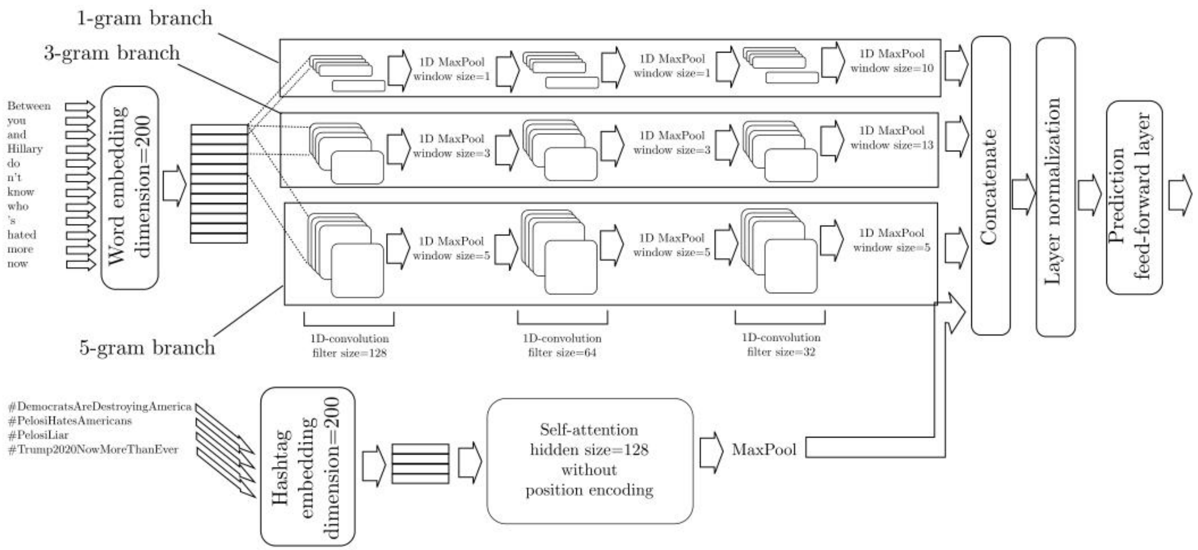

Figure 7: Fully expanded view of the convolutional model

Approach

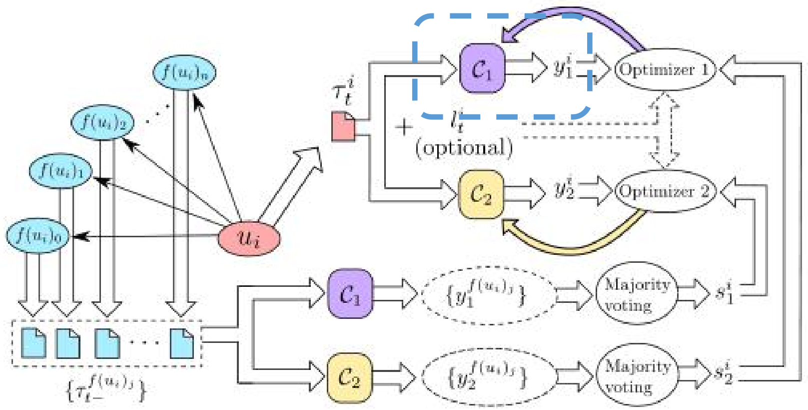

Figure 2: Training process of SANDS. Classifiers C_1 and C_2 independently predict the stance for a tweet from user u_i. In parallel, the classifiers generate pseudo-label sets $\{y_1^{f(u_i)_j}\}$ and $\{y_1^{f(u_i)_j}\}$, respectively from the recent tweets by followees of u_i. After majority voting, lables s_1^i and s_2^i are selected from pseudo-labels. C_1 is optimized using the label generated by C_2, and vice versa. Additionally, annotated labels l_t^i (if present) of the tweet are also used to compute loss and optimize the classifiers.

$$f(u_i) := \{u_j | u_i \rightarrow u_j\}$$

$$\mathcal{D} := \{\tau_t^i | 0 \leq t \leq T\}$$

$$\mathcal{D}_s = \{(\tau_t^i, l_t^i) | \tau_t^i \in \mathcal{D}, l_t^i \in \{0,1\}^{|\mathcal{L}|}\},$$

$$\mathbf{X}^{\mathbf{H}} = \{x_i^H\}$$

$$q_i = \mathrm{ReLU}(\mathbf{W}_q x_i^H + \mathbf{B}_q);\ k_i = \mathrm{ReLU}(\mathbf{W}_k x_i^H + \mathbf{B}_k);\ v_i = \mathrm{ReLU}(\mathbf{W}_v x_i^H + \mathbf{B}_v)$$

$$\alpha_{ij} = \frac{\exp(q_i^\top k_j)}{\sum_j \exp(q_i^\top k_j)}; \qquad \hat{x}_i^H = \sum_j \alpha_{ij} v_j$$

Finally, we apply max-pooling over $\{\hat{x}_i^H\}$ to compute the combined representation of the hashtags, $\mathbf{Z}^H$.

$$\mathbf{h}_{i+1} = \mathrm{Maxpool}(\mathrm{ReLU}(\mathrm{Conv}(\mathbf{h}_i))).$$

$$\mathbf{Z}_1 = \mathrm{LayerNorm}([\mathbf{Z}^H : \mathbf{Z}^W_{\mathrm{conv}-1} : \mathbf{Z}^W_{\mathrm{conv}-3} : \mathbf{Z}^W_{\mathrm{conv}-5}])$$

$$y_1 = \mathrm{Softmax}(\mathbf{W}_{p1}\mathbf{Z}_1 + \mathbf{B}_{p1})$$

Approach

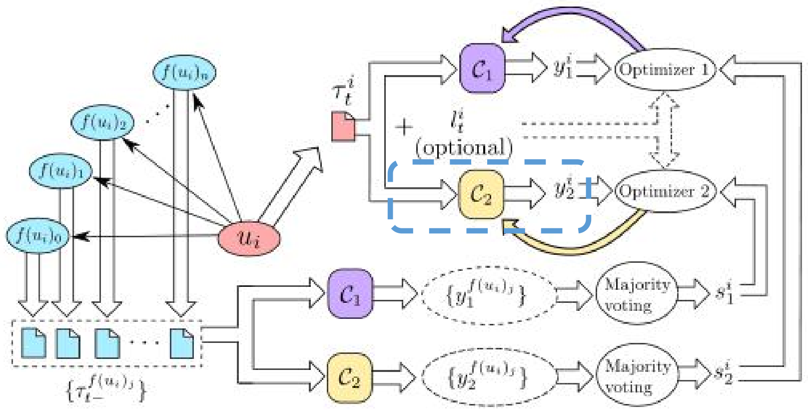

Figure 2: Training process of SANDS. Classifiers C_1 and C_2 independently predict the stance for a tweet from user u_i. In parallel, the classifiers generate pseudo-label sets $\{y_1^{f(u_i)_j}\}$ and $\{y_1^{f(u_i)_j}\}$, respectively from the recent tweets by followees of u_i. After majority voting, lables s_1^i and s_2^i are selected from pseudo-labels. C_1 is optimized using the label generated by C_2, and vice versa. Additionally, annotated labels l_t^i (if present) of the tweet are also used to compute loss and optimize the classifiers.

$$\mathbf{Z}_2 = \mathrm{LayerNorm}([\mathbf{Z}^H : \mathbf{Z}_{\mathrm{blstm}}])$$
$$y_2 = \mathrm{Softmax}(\mathbf{W}_{p2}\mathbf{Z}_2 + \mathbf{B}_{p2})$$

$$\mathcal{J}_1^s = -\omega_s(l_t^i) \sum_{j=1}^{|\mathcal{L}|} \mu_j \log v_{1,j}; \quad \mathcal{J}_2^s = -\omega_s(l_t^i) \sum_{j=1}^{|\mathcal{L}|} \mu_j \log v_{2,j}$$

$$\omega_s(l_t^i) = \log \frac{|\mathcal{D}_s|}{|\{l_{t'}^j | l_{t'}^j = l_t^i, \tau_{t'}^j \neq \tau_t^i \forall \{\tau_{t'}^j, l_{t'}^j\} \in \mathcal{D}_s\}| + \epsilon}$$

Approach

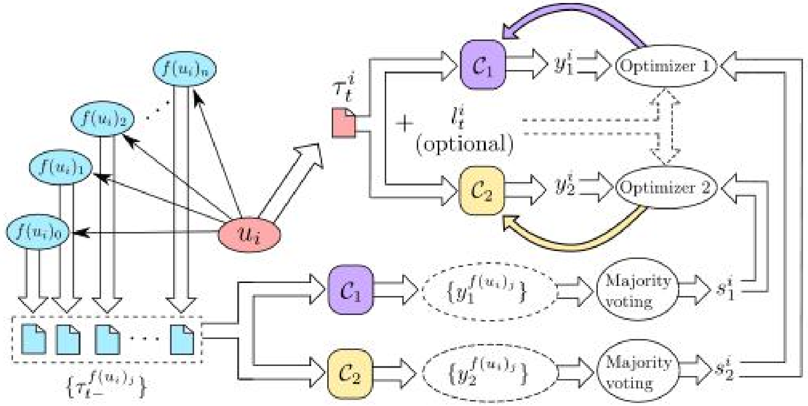

Figure 2: Training process of SANDS. Classifiers C_1 and C_2 independently predict the stance for a tweet from user u_i. In parallel, the classifiers generate pseudo-label sets $\{y_1^{f(u_i)_j}\}$ and $\{y_1^{f(u_i)_j}\}$, respectively from the recent tweets by followees of u_i. After majority voting, lables s_1^i and s_2^i are selected from pseudo-labels. C_1 is optimized using the label generated by C_2, and vice versa. Additionally, annotated labels l_t^i (if present) of the tweet are also used to compute loss and optimize the classifiers.

$$y_1^{f(u_i)_j} = C_1(\tau_{t-}^{f(u_i)_j}|\theta_1);\quad y_2^{f(u_i)_j} = C_2(\tau_{t-}^{f(u_i)_j}|\theta_2)$$

$$I_1^{f(u_i)} = \{\text{argmax}(y_1^{f(u_i)_j})\}_{j=1}^{|f(u_i)|}$$

$$C_1^{f(u_i)} = \{\text{freq}(k, I_1^{f(u_i)})\}_{k=0}^{|\mathcal{L}|-1}$$

$$s_1^i = \text{one-hot}(\text{argmax}(C_1^{f(u_i)}))$$

$$\omega_{u,1}^{\mathrm{B}}(s_1^i) = \log \frac{|\mathrm{B}|}{\text{freq}(s_1^i, \mathcal{S}_1) + \epsilon}$$

$$\mathcal{J}_1^u = -\omega_u(s_2^i)\sum_{j=1}^{|\mathcal{L}|}\sigma_{2,j}\log \nu_{1,j} - \omega_s(l_t^i)\sum_{j=1}^{|\mathcal{L}|}\mu_j \log \nu_{1,j}$$

$$\mathcal{J}_2^u = -\omega_u(s_1^i)\sum_{j=1}^{|\mathcal{L}|}\sigma_{1,j}\log \nu_{2,j} - \omega_s(l_t^i)\sum_{j=1}^{|\mathcal{L}|}\mu_j \log \nu_{2,j}$$

Experiments

Classes	Train-500	Train-1000	Train-1500	Test
StanceUS				
Pro-Dem	320	654	981	1543
Anti-Dem	9	22	32	46
Pro-Rep	133	253	381	576
Anti-Rep	13	17	29	46
Other	25	54	77	111
Total	500	1000	1500	2322
StanceIN				
Pro-BJP	67	149	208	360
Anti-BJP	136	275	425	680
Pro-INC	24	60	83	158
Anti-INC	2	3	6	15
Pro-AAP	35	61	99	142
Anti-AAP	52	101	163	210
Other	184	351	516	1120
Total	500	1000	1500	2685

Table 1: Class-wise sample distribution in different training and testing splits of the annotated datasets.

Model	StanceUS			StanceIN		
	$\|\mathcal{D}_s\|$ =0.5K	$\|\mathcal{D}_s\|$ =1K	$\|\mathcal{D}_s\|$ =1.5K	$\|\mathcal{D}_s\|$ =0.5K	$\|\mathcal{D}_s\|$ =1K	$\|\mathcal{D}_s\|$ =1.5K
SiamNet	0.39	0.43	0.42	0.12	0.14	0.13
BICE	0.27	0.30	0.33	0.16	0.17	0.23
TAN	0.38	0.46	0.45	0.14	0.14	0.17
SVM	0.37	0.37	0.45	0.13	0.13	0.16
BERT	0.39	0.50	0.51	0.17	0.17	0.21
ConvNet	0.37	0.43	0.45	0.35	0.40	0.41
BLSTM	0.35	0.43	0.44	0.31	0.39	0.38
LS-SVM	0.39	0.42	0.44	0.18	0.19	0.18
ST-ConvNet	0.13	0.15	0.16	0.10	0.11	0.11
ST-BLSTM	0.13	0.16	0.19	0.09	0.12	0.11
UST	0.35	0.42	0.41	0.12	0.16	0.16
GCN-ConvNet	0.41	0.45	0.47	0.33	0.35	0.40
GCN-BLSTM	0.39	0.42	0.46	0.36	0.41	0.42
SANDS/Net.(C_1)	0.32	0.41	0.42	0.10	0.12	0.15
SANDS/Net.(C_2)	0.36	0.46	0.46	0.28	0.31	0.37
SANDS/Cont.(C_1)	0.41	0.47	0.49	0.36	0.41	0.43
SANDS/Cont.(C_2)	0.47	0.51	0.53	0.38	0.44	0.45
SANDS(C_1)	**0.46**	**0.47**	**0.49**	**0.37**	**0.42**	**0.45**
SANDS(C_2)	**0.49**	**0.53**	**0.55**	**0.42**	**0.45**	**0.47**

Table 2: F1 scores of all models with different sizes of labeled training data on StanceUS and StanceIN.

Experiments

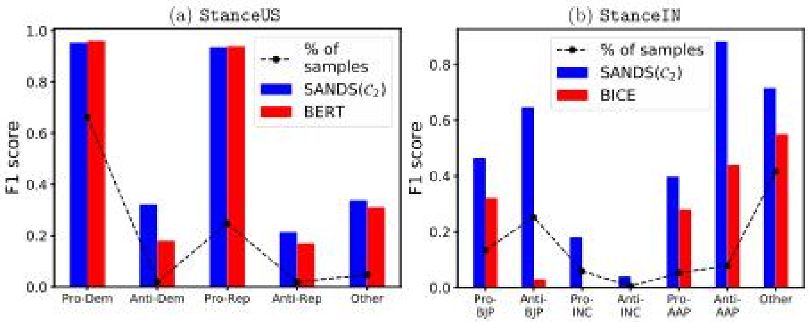

Figure 3: Comparison of class-wise F1 scores between SANDS(C_2) and best supervised baselines – (a) BERT in US data and (b) BICE in India data.

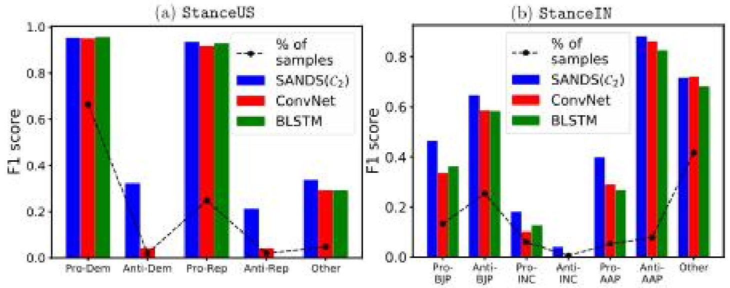

Figure 4: Comparison of class-wise F1 scores of SANDS(C_2) with the supervised counterparts, ConvNet and BLSTM on (a) StanceUS and (b) StanceIN. All of these frameworks use 1500 labelled data instances for training. SANDS provides better prediction performance on samples from minority classes compared to the supervised ConvNet and BLSTM.

like **Pro-INC** or **Anti-INC**, BICE did not predict even a single sample on the test data.

Experiments

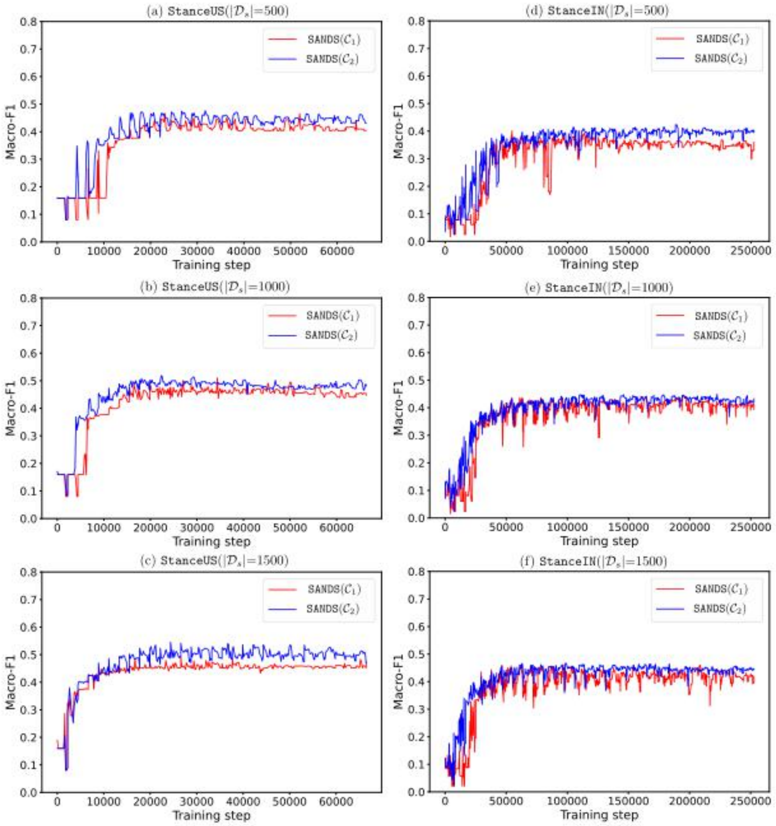

Figure 5: Macro-F1 scores of SANDS(C_1) and SANDS(C_2) on test set as training progress on different training splits for StanceUS and StanceIN. In each case, SANDS provides stable convergence.

Classifier pair	StanceUS		StanceIN	
(C_1, C_2)	m-F1 C_1	m-F1 C_2	m-F1 C_1	m-F1 C_2
Conv, bi-LSTM	0.49	0.55	0.45	0.47
Conv, Conv	0.48	0.48	0.43	0.43
bi-LSTM, bi-LSTM	0.52	0.52	0.45	0.45
Conv, BERT	0.50	0.52	0.44	0.45
BERT, bi-LSTM	0.53	0.53	0.42	0.45

Table 3: Macro-F1 scores of different classifier-pairs with SANDS. The size of labelled dataset used for all these pairs is 1500.

Experiments

% unlabeled data used	StanceUS		StanceIN	
	m-F1 C_1	m-F1 C_2	m-F1 C_1	m-F1 C_2
100	0.49	0.55	0.45	0.47
80	0.49	0.54	0.44	0.46
50	0.46	0.51	0.42	0.43
30	0.45	0.48	0.42	0.41
10	0.45	0.45	0.41	0.39

Table 4: Macro-F1 scores of SANDS with different amount of unlabeled data used in semi-supervised phase. The size of labeled dataset used is 1500.

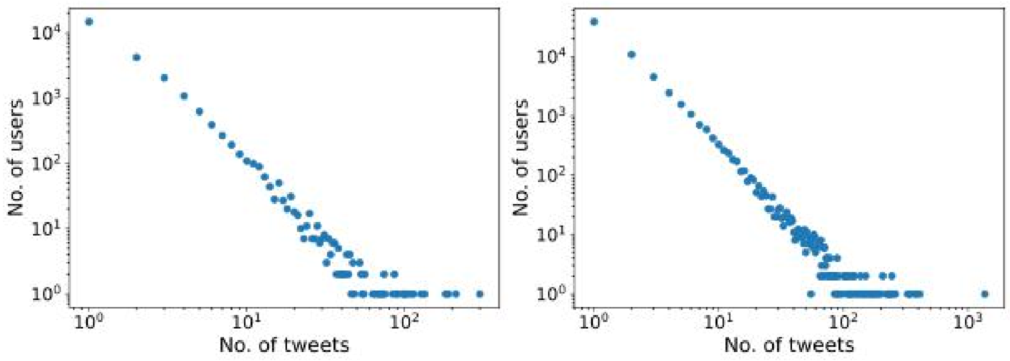

Figure 6: Log-log distribution of tweets posted by users in StanceUS (left) and StanceIN (right).

Thank you！