Zero-Shot Rumor Detection with Propagation Structure via Prompt Learning

Hongzhan Lin[♠*], Pengyao Yi[◇*], Jing Ma[♠†], Haiyun Jiang[†],
Ziyang Luo[♠], Shuming Shi, Ruifang Liu[◇]

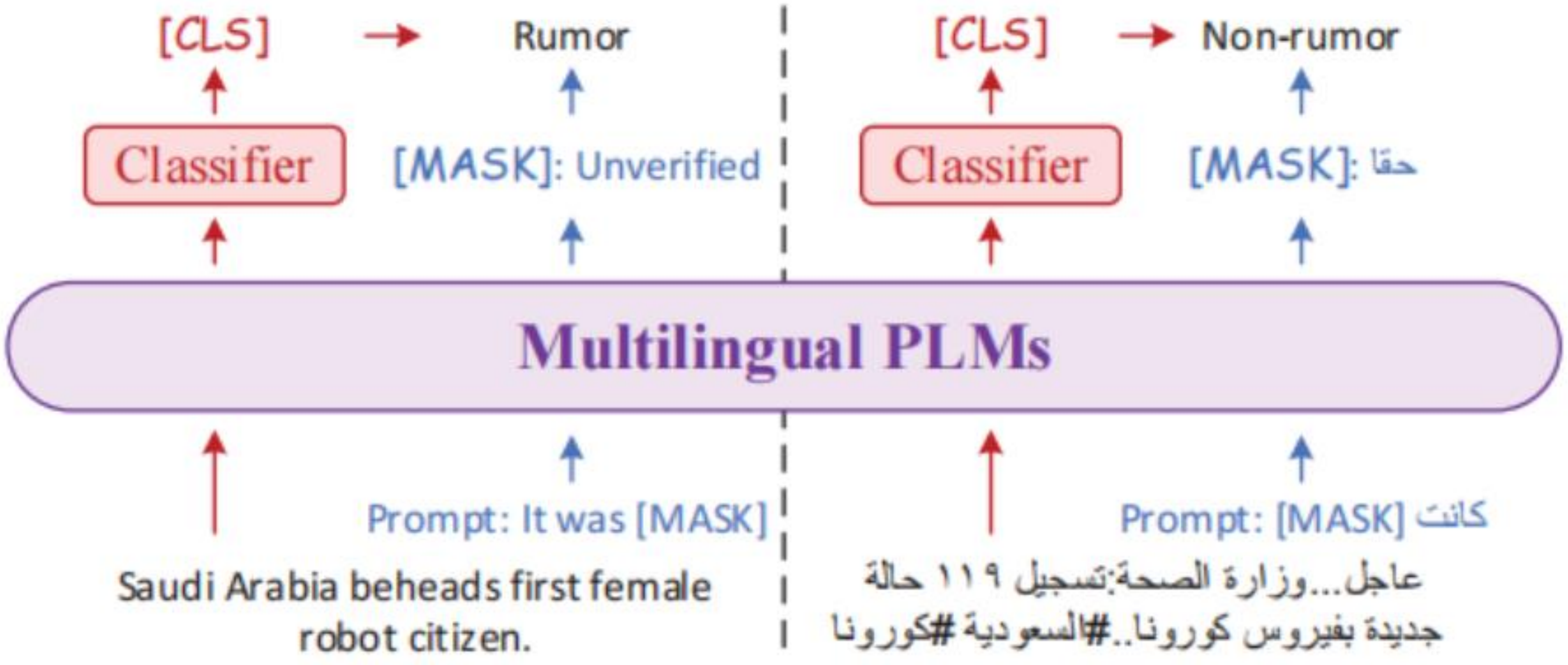
[♠]Hong Kong Baptist University
[◇]Beijing University of Posts and Telecommunications
{cshzlin, majing, cszyluo}@comp.hkbu.edu.hk, jianglu@fudan.edu.cn, {yi.py, lrf}@bupt.edu.cn

AAAI2023

code: https://github.com/PengyaoYi/zeroRumor AAAI

Reported by Xiaoke Li

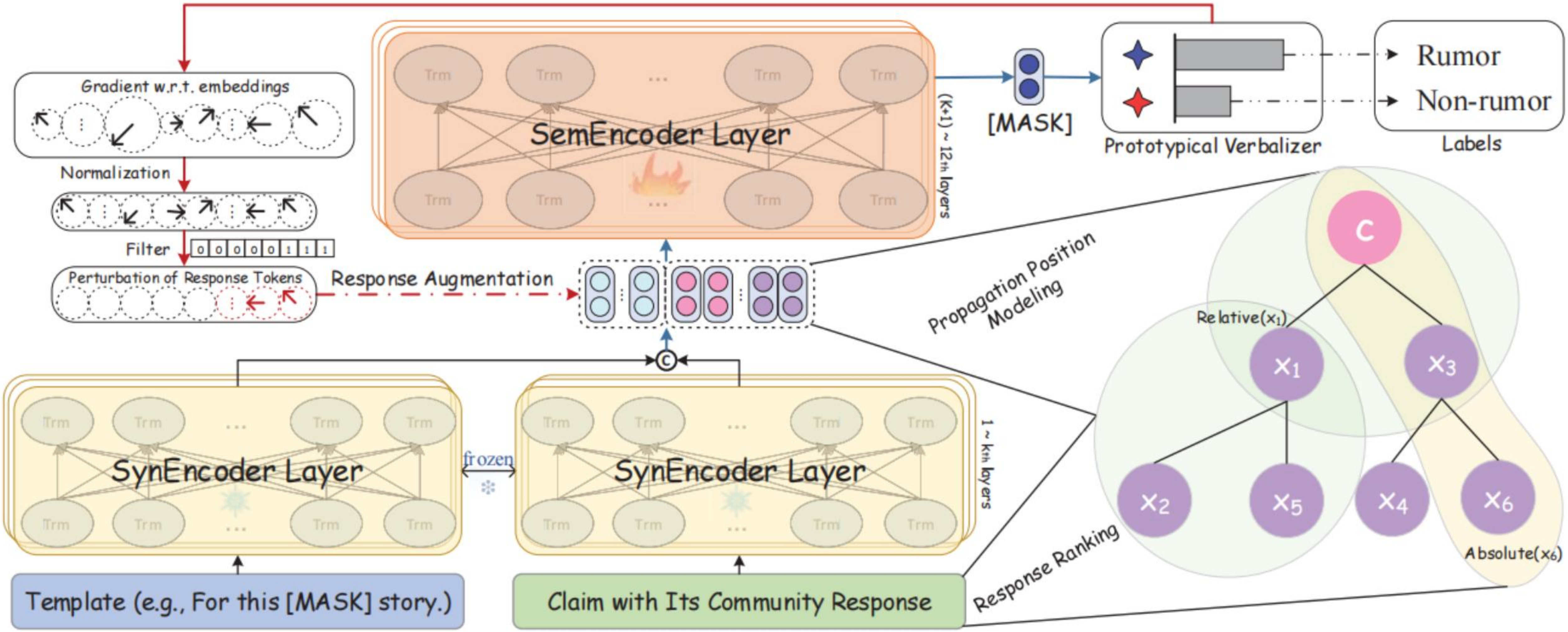
Introduction



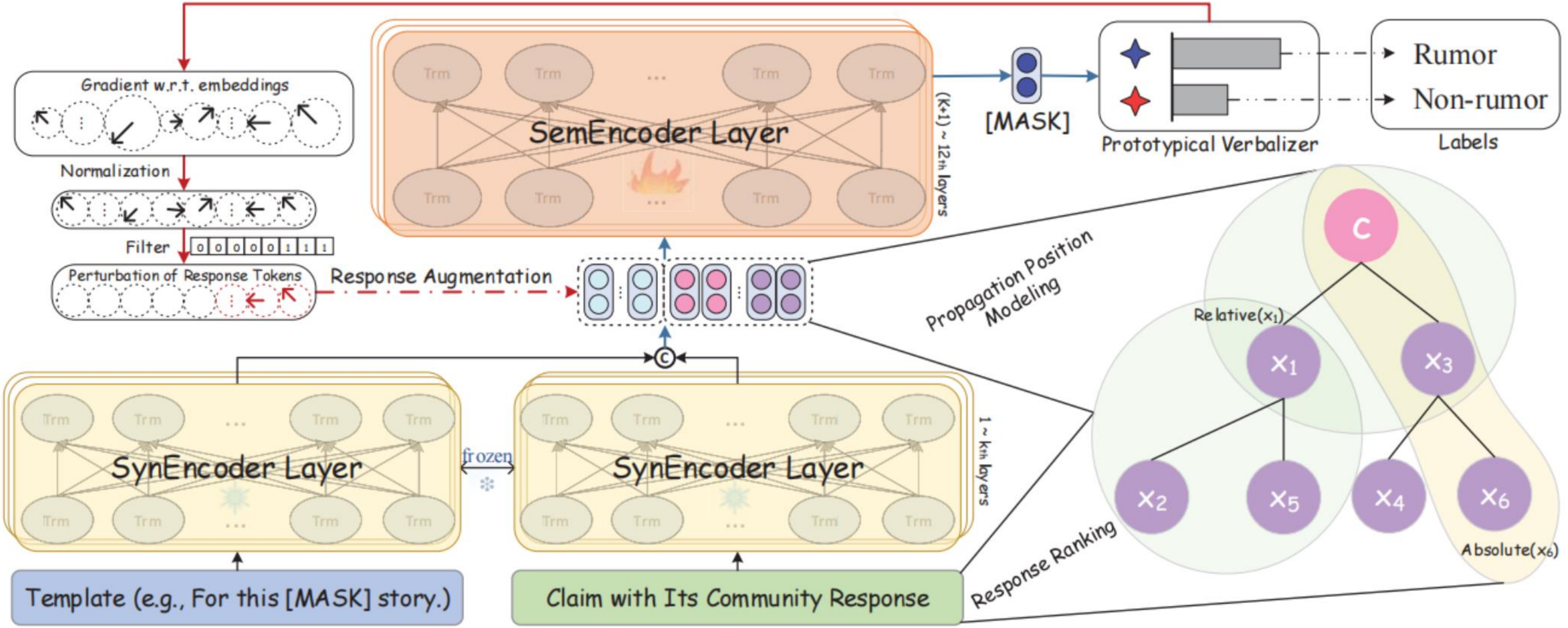
a) Training on ***Politics*** domain in ***English*** b) Testing on ***COVID-19*** domain in ***Arabic***

Figure 1: Illustration between the task-specific fine-tuning and the prompt learning paradigms for solving ZRD task.

Method


Gradient w.r.t. embeddings
Normalization
Filter
0 0 0 0 0 1 1 1
Perturbation of Response Tokens
Response Augmentation
SemEncoder Layer
Trm
(k+1) ~ 12th layers
[MASK]
Prototypical Verbalizer
Rumor
Non-rumor
Labels
Propagation Position Modeling
Relative(x1)
Absolute(x6)
Response Ranking
C
x1
x2
x3
x4
x5
x6
SynEncoder Layer
frozen
SynEncoder Layer
1 ~ kth layers
Template (e.g., For this [MASK] story.)
Claim with Its Community Response

Method



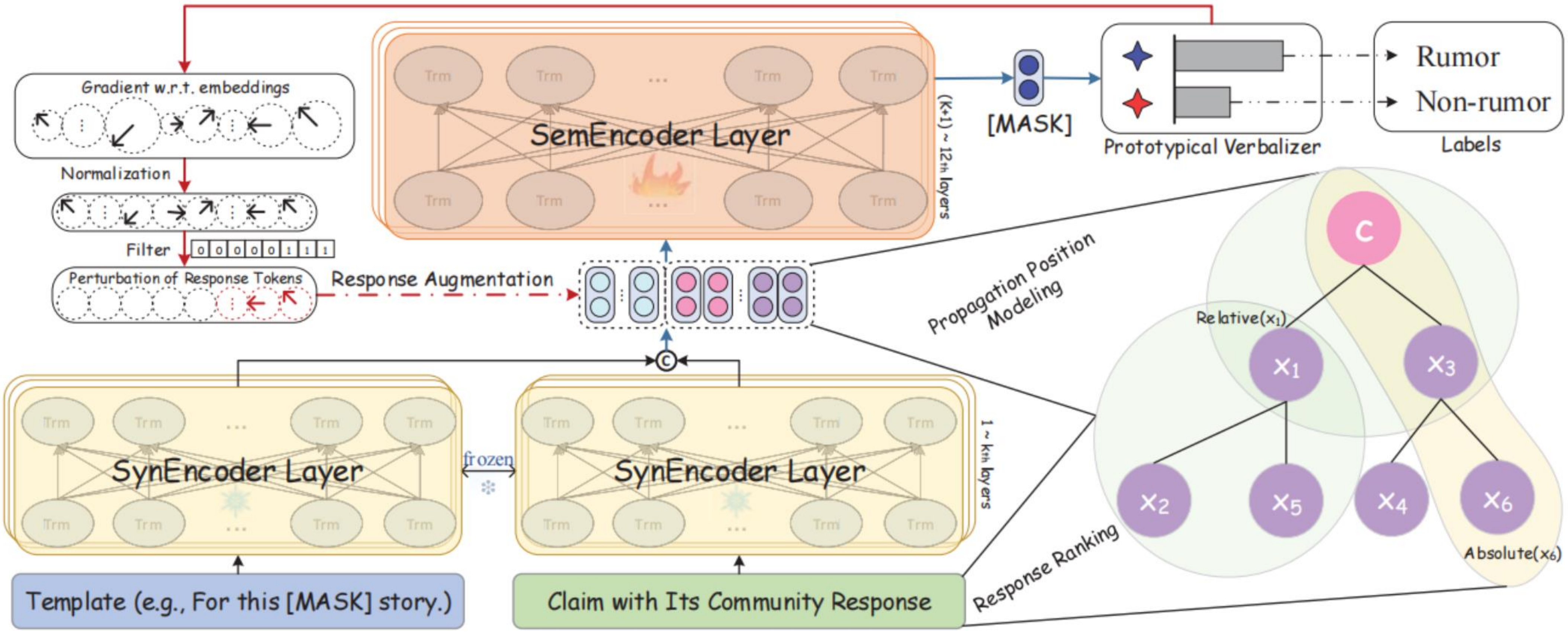
$$\mathcal{D}_s = \{C_1^s, C_2^s, \cdots, C_M^s\} \quad \mathcal{D}_t = \{C_1^t, C_2^t, \cdots, C_N^t\}$$

$$C^s = (y, c, \mathcal{T}(c)) \quad C^t = (c', \mathcal{T}(c'))$$

$$\mathcal{T}(c) = [x_1^s, x_2^s, \cdots, x_m^s]$$

$$\mathcal{P}(y|\hat{c}) = g(\mathcal{P}([MASK] = v|\hat{c})|v \in \mathcal{V}_y) \qquad (1)$$

Method



$$X_p = \text{SynEncoder}(p) \quad (2)$$

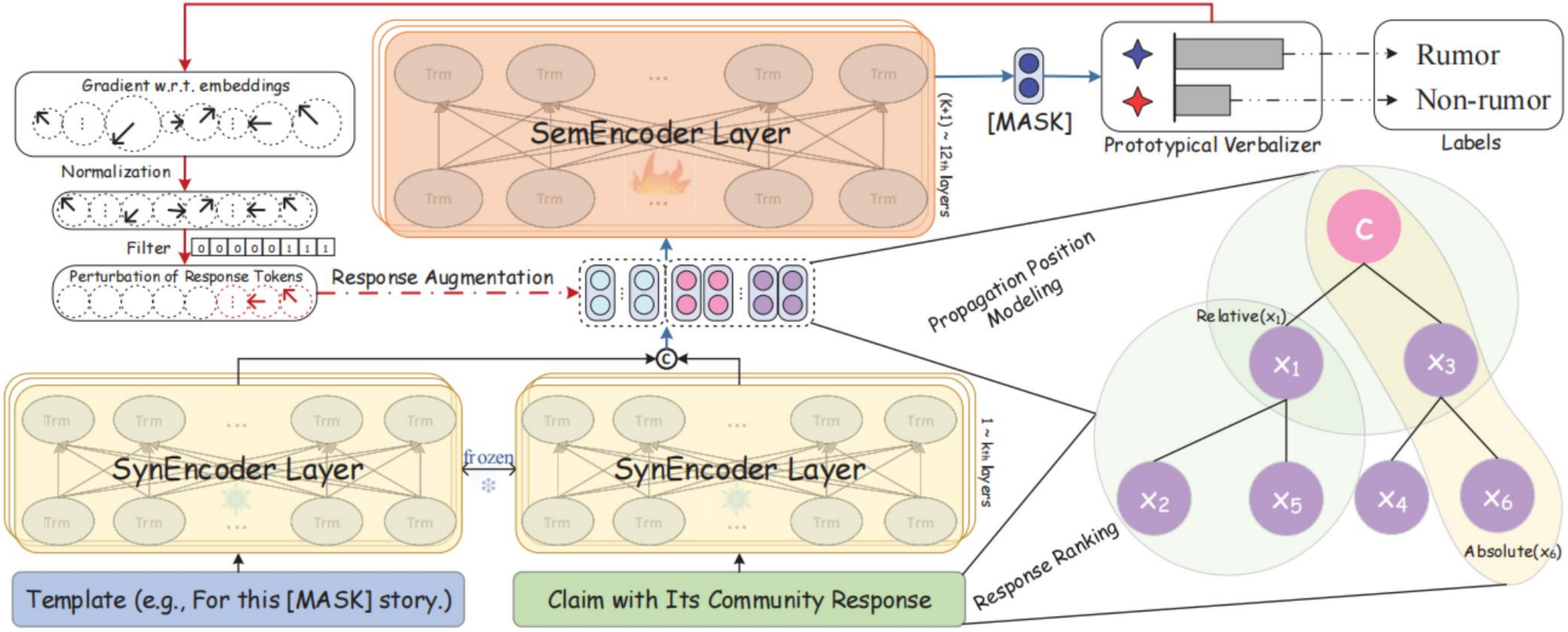
where $X_p \in \mathbb{R}^{|p| \times d}$ is the template embeddings and d is the dimension of the output state of SynEncoder.

$$X_{cr} = \text{SynEncoder}([c, \mathcal{T}(c)]) \quad (3)$$

$$H = \text{SemEncoder}([X_p, X_{cr}]) \quad (4)$$

$$abs_{pro}(q) = \text{distance}_{tree}(x_i, c) \quad (5)$$

Method

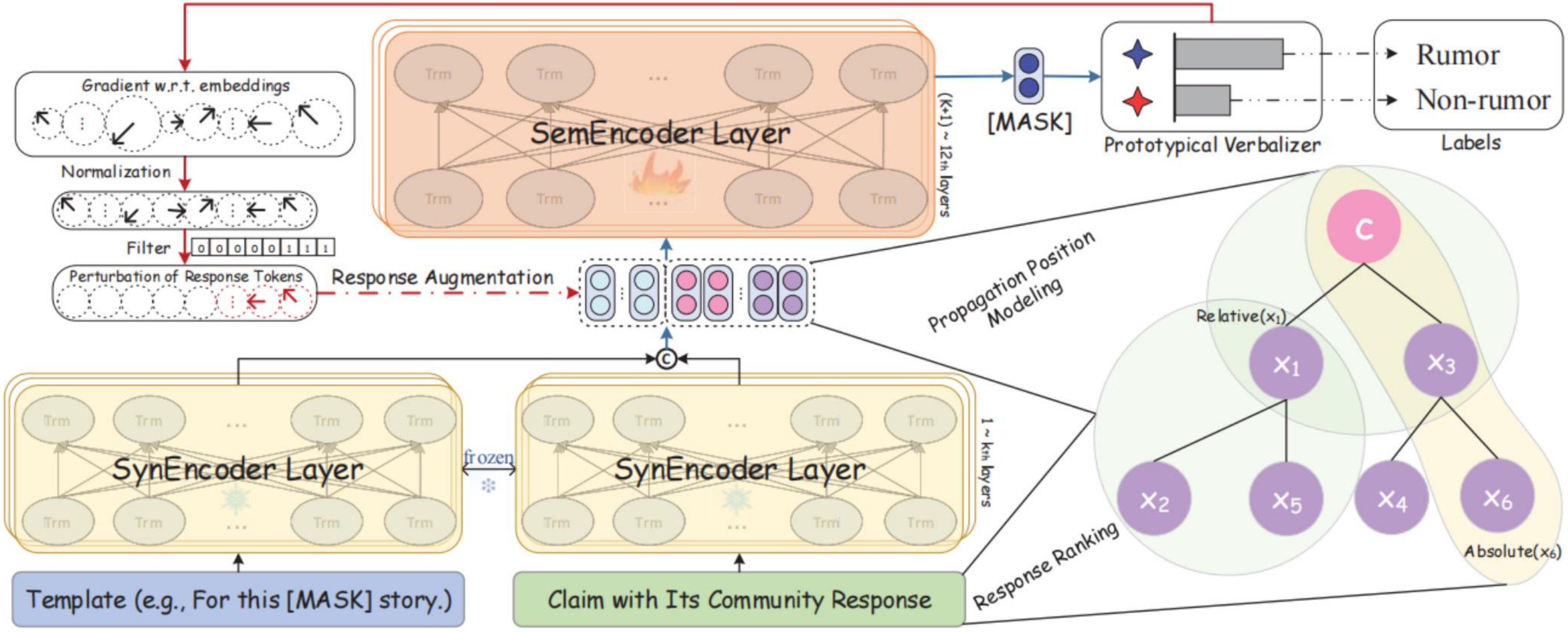


Given the *[MASK]* token representation H_i^m of a training example C_i, we minimize a prototypical loss that is the negative log-likelihood:

$$\mathcal{L}_{proto} = -log \frac{e^{\mathcal{S}(H_i^m, l_y)}}{\sum_{y'} e^{\mathcal{S}(H_i^m, l_{y'})}} \quad (6)$$

$$\mathcal{L}_{con} = -\frac{1}{B_{y_i}-1} \sum_j \mathbb{1}_{[i \neq j]} \mathbb{1}_{[y_i = y_j]} \, log \frac{e^{\mathcal{S}(H_i^m, H_j^m)}}{\sum_{j'} \mathbb{1}_{[i \neq j']} e^{\mathcal{S}(H_i^m, H_{j'}^m)}} \quad (7)$$

Method



$$\mathcal{L} = \alpha \mathcal{L}_{proto} + (1-\alpha)\mathcal{L}_{con} \quad (8)$$

Experiments

Dataset	Source		Target		
	TWITTER	WEIBO	Twitter-COVID19	Weibo-COVID19	CatAr-COVID19
# of events	1154	4649	400	399	1699
# of tree nodes	60409	1956449	406185	26687	168276
# of non-rumors	579	2336	148	146	976
# of rumors	575	2313	252	253	723
Avg. time/tree	389 Hours	1007 Hours	2497 Hours	248 Hours	858 Hours
Avg. depth/tree	11.67	49.85	143.03	4.31	13.26
Language	English	Chinese	English	Chinese	Cantonese&Arabic
Domain	Open	Open	COVID-19	COVID-19	COVID-19

Table 1: Statistics of Datasets.

Experiments

Source	TWITTER								WEIBO							
Target	Weibo-COVID19				CatAr-COVID19				Twitter-COVID19				CatAr-COVID19			
Model	Acc.	Mac-F_1	Rumor F_1	Non-rumor F_1	Acc.	Mac-F_1	Rumor F_1	Non-rumor F_1	Acc.	Mac-F_1	Rumor F_1	Non-rumor F_1	Acc.	Mac-F_1	Rumor F_1	Non-rumor F_1
Vanilla-Finetune	0.623	0.585	0.711	0.459	0.518	0.402	0.583	0.220	0.603	0.602	0.619	0.585	0.481	0.481	0.479	0.474
Translate-Finetune	0.639	0.567	0.745	0.388	0.523	0.457	0.637	0.277	0.634	0.574	0.653	0.495	0.505	0.512	0.528	0.496
Contrast-Finetune	0.656	0.582	0.759	0.405	0.584	0.458	0.720	0.196	0.653	0.644	0.699	0.590	0.562	0.561	0.571	0.551
Adapter	0.644	0.600	0.737	0.463	0.558	0.438	0.665	0.211	0.652	0.612	0.736	0.487	0.548	0.556	0.605	0.508
Parallel-Adapter	0.651	0.598	0.730	0.467	0.567	0.450	0.701	0.198	0.667	0.653	0.731	0.574	0.579	0.585	0.636	0.534
Source-Prompt	0.664	0.648	0.722	0.574	0.589	0.564	0.460	0.669	0.670	0.616	0.760	0.472	0.599	0.565	0.688	0.441
Translate-Prompt	0.650	0.489	0.776	0.201	0.573	0.568	0.519	0.617	0.674	0.651	0.740	0.562	0.604	0.542	0.374	0.711
Soft-Prompt	0.652	0.574	0.756	0.392	0.590	0.565	0.446	0.683	0.685	0.652	0.758	0.546	0.609	0.575	0.518	0.633
RPL-Cho	0.713	0.675	0.786	0.563	0.613	0.581	0.455	0.707	0.715	0.689	0.778	0.601	0.634	0.633	**0.616**	0.650
RPL-Inv	0.728	0.666	**0.810**	0.521	0.601	0.592	0.473	**0.711**	**0.733**	0.710	0.788	0.632	0.647	0.640	0.586	0.693
RPL-Dep	0.732	0.689	0.805	0.574	**0.640**	**0.619**	0.530	0.708	0.723	**0.711**	0.771	**0.650**	0.657	0.636	0.547	**0.724**
RPL-Bre	**0.745**	**0.719**	0.804	**0.634**	0.631	0.617	**0.544**	0.689	0.727	0.697	**0.793**	0.601	**0.672**	**0.664**	0.614	0.714

Table 2: Rumor detection results on the target test datasets.

Experiments

Source	TWITTER		WEIBO	
Model	Acc.	Mac-F_1	Acc.	Mac-F_1
RPL-Bre	0.631	0.617	0.672	0.664
RPL-Bre w/o RR	0.605	0.598	0.613	0.611
RPL-Bre w/o APP	0.622	0.607	0.626	0.624
RPL-Bre w/o RPP	0.610	0.601	0.633	0.632
RPL-Bre w/o ViRA	0.626	0.612	0.644	0.634
RPL-Bre w/o HPE	0.571	0.451	0.581	0.433
RPL-Bre w/o PV	0.592	0.589	0.621	0.617

Table 3: Ablation studies on our proposed model.

Experiments

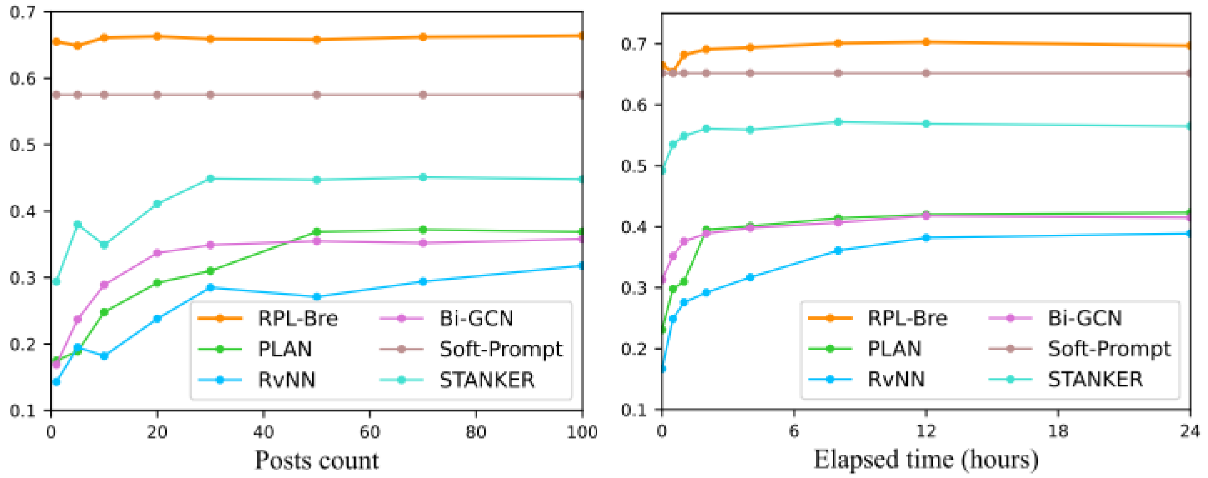


Figure 3: Early detection performance at different checkpoints of posts count (or elapsed time) on CatAr-COVID19 (left) and Twitter-COVID19 (right) datasets.

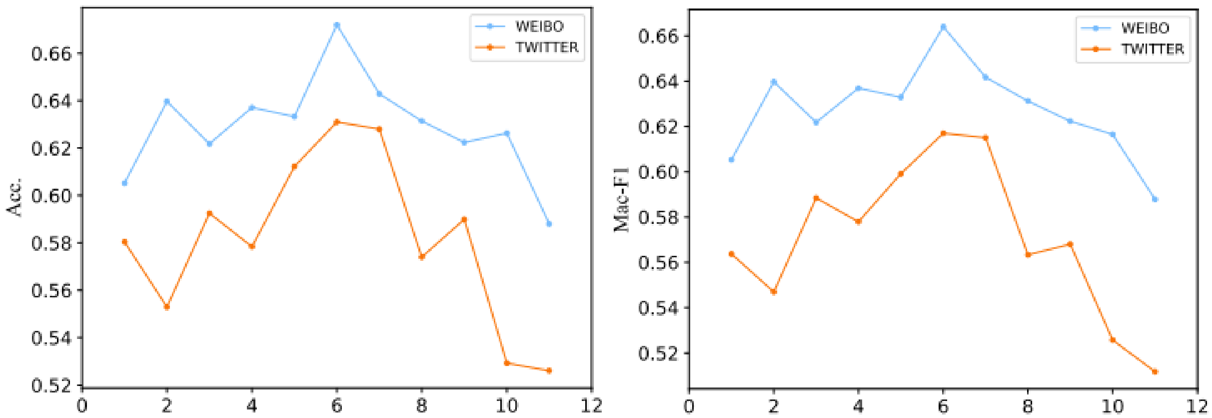


Figure 4: Effect of the layer number k of SynEncoder with Accuracy (left) and Macro F1 (right).

Thanks