太原理工大學
TAIYUAN UNIVERSITY OF TECHNOLOGY

STANKER: Stacking Network based on Level-grained Attention-masked BERT for Rumor Detection on Social Media

Dongning Rao, Xin Miao, Zhihua Jiang , Ran Li
School of Computer, Guangdong University of Technology, Guangzhou 510006, P. R. China
Department of Computer Science, Jinan University, Guangzhou 510632, P. R. China

EMNLP 2021
Code: https://github.com/fip-lab/STANKER.

Zhuomin Chen
2022.03.27

Introduction

Dataset: rumor detection datasets in Chinese companies with comments are rare.

BERT, ensembles of multiple BERT models: a big ensemble size makes the fine-tuning computationally expensive, for the training time and the inference time increase linearly with the ensemble size.

The attention mechanism: a few studies indicated that not all attention is necessary-----partial attention can be pruned or masked depending on specific tasks, because BERT learns different features at different levels.

The input length limitation: social media posts often have comments whose total length exceeds the input-length limitation, demanding pre-processing like truncation. Although Longformer was proposed recently to tackle long input sequences, excessive attention interactions may degrade the overall performance.

Methodology

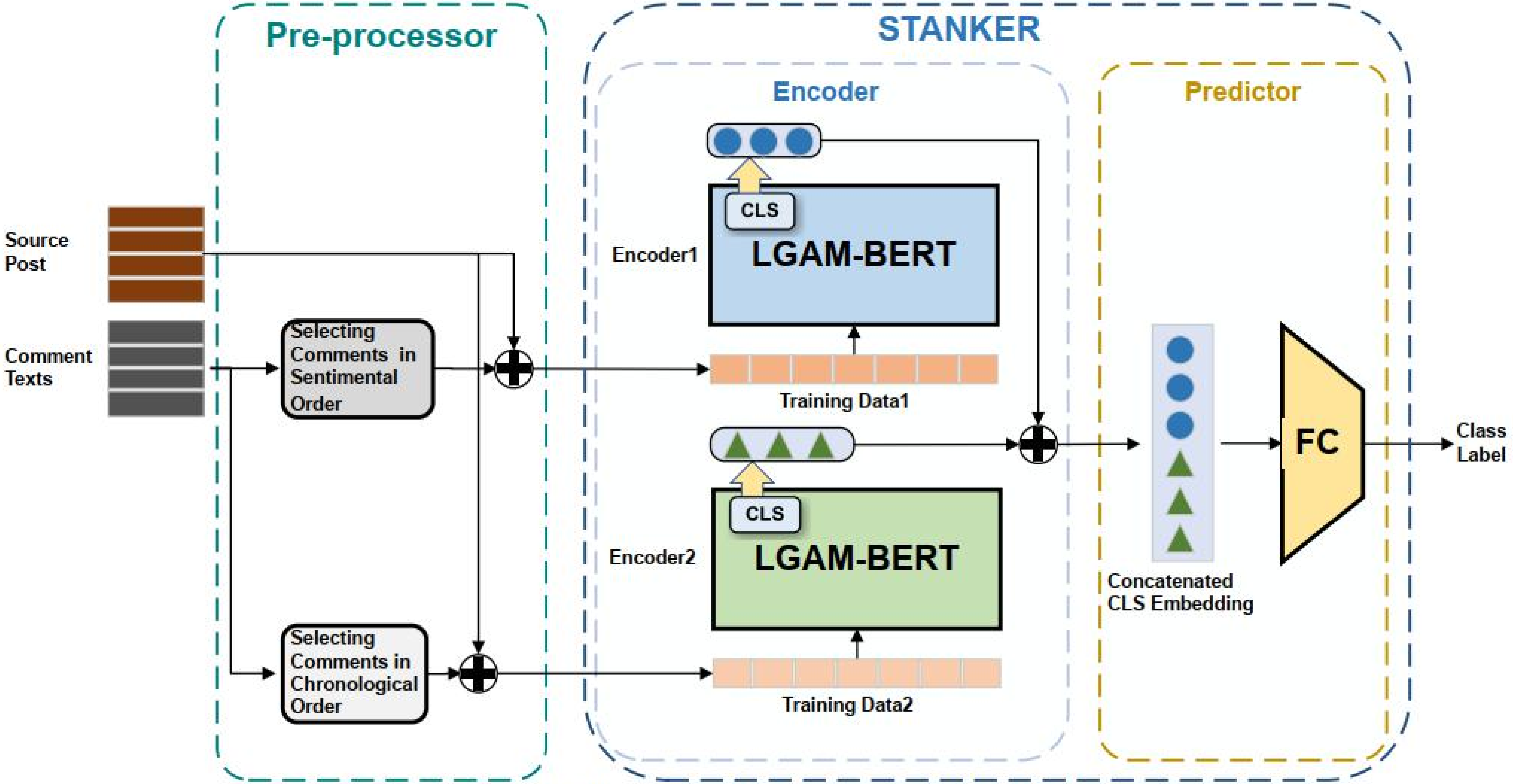


Figure 1: The overall structure of *STANKER*

Methodology

a set of source posts $S = \{s_1, s_2, ..., s_{|S|}\}$

Each $s_i \in S$ is a short text

A word (in English) or character (in Chinese) sequence $< w_1^i, w_2^i, ...w_{l_i}^i >$ given l_i as the length of s_i

Each $s_i \in S$ is associated with a set of comment texts $C_i = \{c_1^i, c_2^i, ...c_{|C_i|}^i\}$

each $c_j^i \in C_i$ is a word or character sequence

the dataset $D = \{d_1, d_2, ..., d_{|D|}\}$

each $d_i \in D$ is a tuple $\{s_i, C_i, y_i\}$

Comment Selection

1、sort comments according to their replying time and prioritize comments that respond earlier.
2、calculate sentiment scores of comments and select those with high scores.

adopt a sentiment dictionary $Dict$ to score all comments

if a word w is in $Dict$, then $score_w$ is a pre-defined score; otherwise, it is set to be 0.

Given a comment c, its sentiment score $score_c$ is an average on $score_w$ for all $w \in c$.

DBSCAN algorithm

Before:

哎！...娱乐真讽刺。[SEP]谣言[SEP]...无语 [SEP] 无语 [SEP] 无语 [SEP] 抄袭 [SEP] 恶心 [SEP] 恶心 [SEP]...

Ah!...entertainment industry is really ironic. [SEP]Rumor [SEP]...Speechless [SEP] Speechless [SEP] Speechless [SEP] Plagiarism [SEP] Gross [SEP] Gross [SEP]...

After:

哎！...娱乐真讽刺。[SEP]谣言[SEP]...无语 [SEP] 抄袭 [SEP] 恶心 [SEP]...

Ah!...entertainment industry is really ironic. [SEP]Rumor [SEP]...Speechless [SEP] Plagiarism [SEP] Gross [SEP]...

Methodology

a source post s_i $=< w_1^i, w_2^i, ...w_{l_i}^i >$

chronological-comment set $CCS_i = \{c_1^i, c_2^i, ...c_{|CCS_i|}^i\}$

$E_{[s_i;CCS_i]}$

$\mathbf{L}_i = [\mathbf{l}_1^i; \mathbf{l}_2^i; ......\mathbf{l}_m^i] \in \mathbb{R}^{m*d}$

$$\mathbf{L}_i = LGAM - BERT(E_{[s_i;CCS_i]}) \quad (1)$$

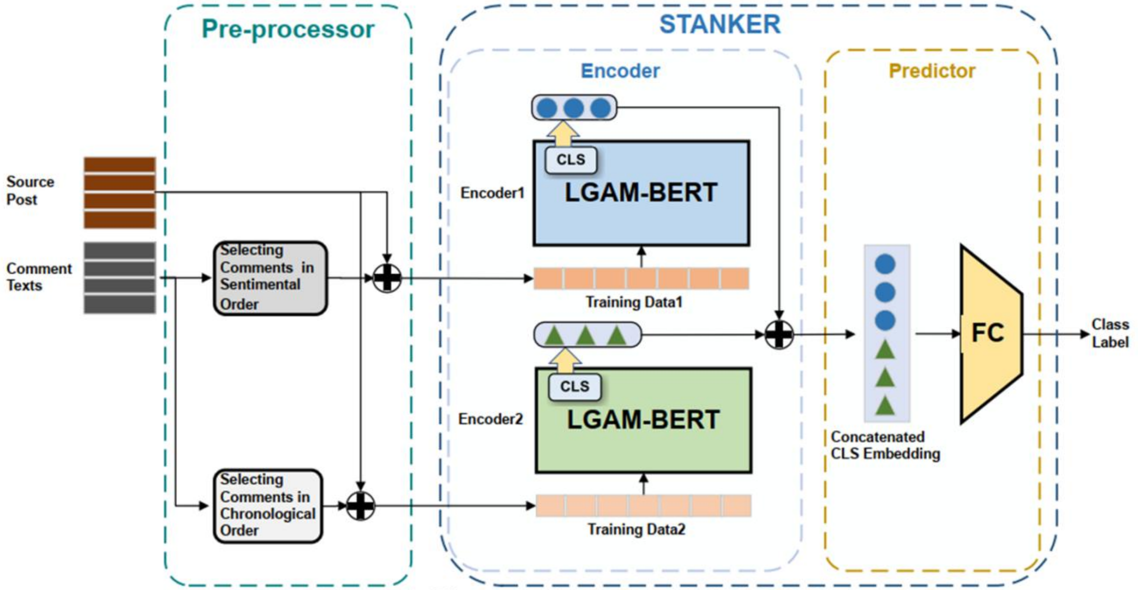


Figure 1: The overall structure of *STANKER*

sentimental- comment set $SCS_i = \{c_1^i, c_2^i, ...c_{|SCS_i|}^i\}$

$\mathbf{R}_i = [\mathbf{r}_1^i; \mathbf{r}_2^i; ......\mathbf{r}_m^i] \in \mathbb{R}^{m*d}$

$$\mathbf{R}_i = LGAM - BERT(E_{[s_i;SCS_i]}) \quad (2)$$

$$\mathbf{PR}_i = concate(\mathbf{L}_i[0], \mathbf{R}_i[0]) \quad (3)$$

feed $\mathbf{PR}_i$ to a fully-connected network and output the prediction via softmaxing

LGAM-BERT

The standard attention mechanism:

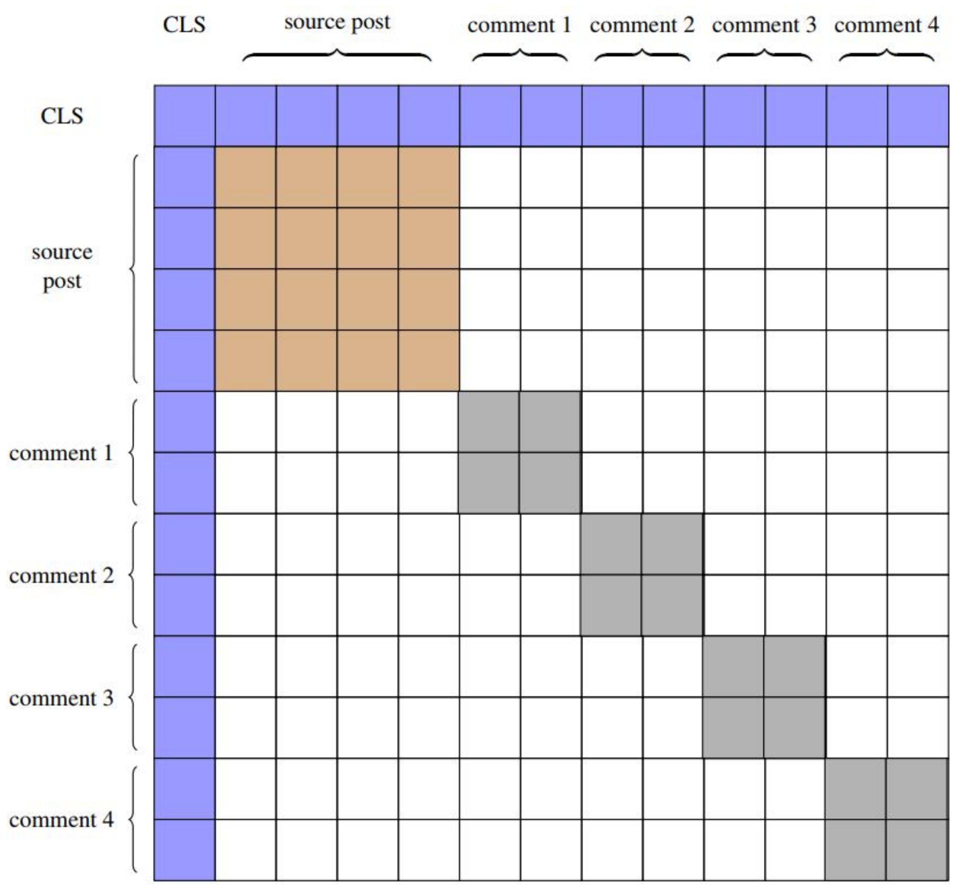
$$A(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d}})V \qquad (4)$$

define a visible matrix M of tokens:

$$M_{ij} = \begin{cases} 0 & Q_i \ominus K_j \\ -\infty & Q_i \oslash K_j \end{cases} \qquad (5)$$

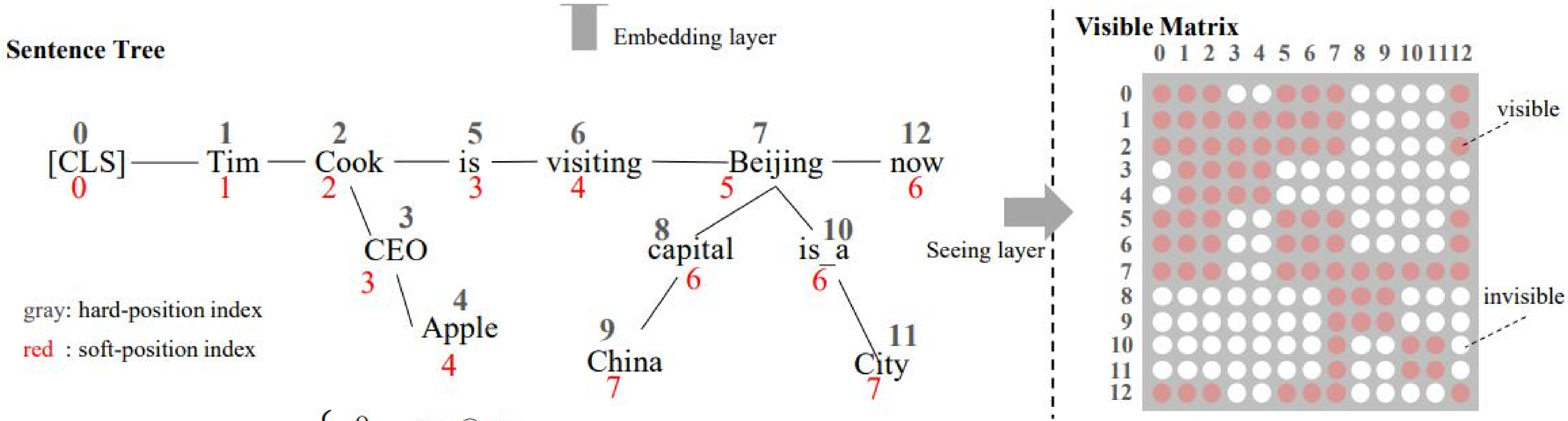
$\ominus$ means that Q_i and K_j are injected from the same sentence

$\oslash$ means that Q_i and K_j are injected from different sentences



Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020.
K-bert: Enabling language representation with knowledge graph. In AAAI.

Visible Matrix



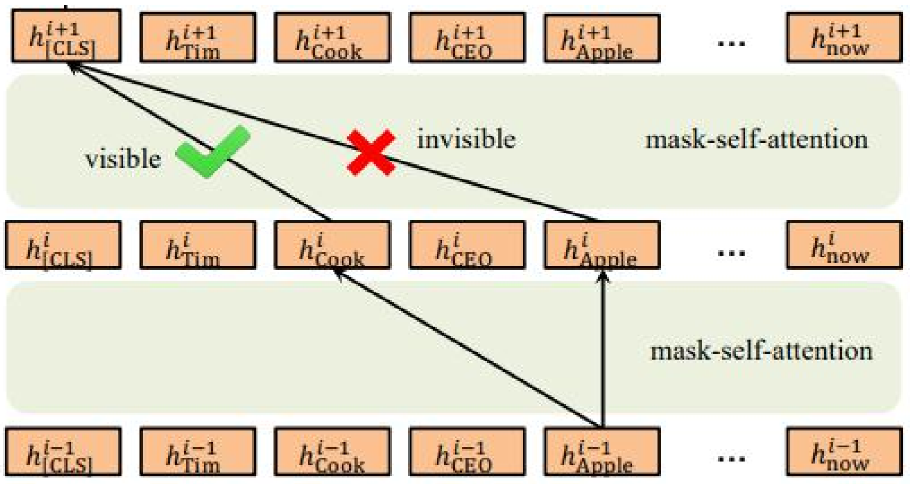
$$M_{ij} = \begin{cases} 0 & w_i \ominus w_j \\ -\infty & w_i \oslash w_j \end{cases} \qquad (3)$$

where, $w_i \ominus w_j$ indicates that w_i and w_j are in the same branch, while $w_i \oslash w_j$ are not. i and j are the hard-position index.

Mask-Self-Attention: $Q^{i+1}, K^{i+1}, V^{i+1} = h^i W_q, h^i W_k, h^i W_v, \qquad (4)$

$$S^{i+1} = softmax\left(\frac{Q^{i+1} K^{i+1^\top} + M}{\sqrt{d_k}}\right), \qquad (5)$$

$$h^{i+1} = S^{i+1} V^{i+1}, \qquad (6)$$



Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. **K-bert: Enabling language representation with knowledge graph**. In AAAI.

LGAM-BERT

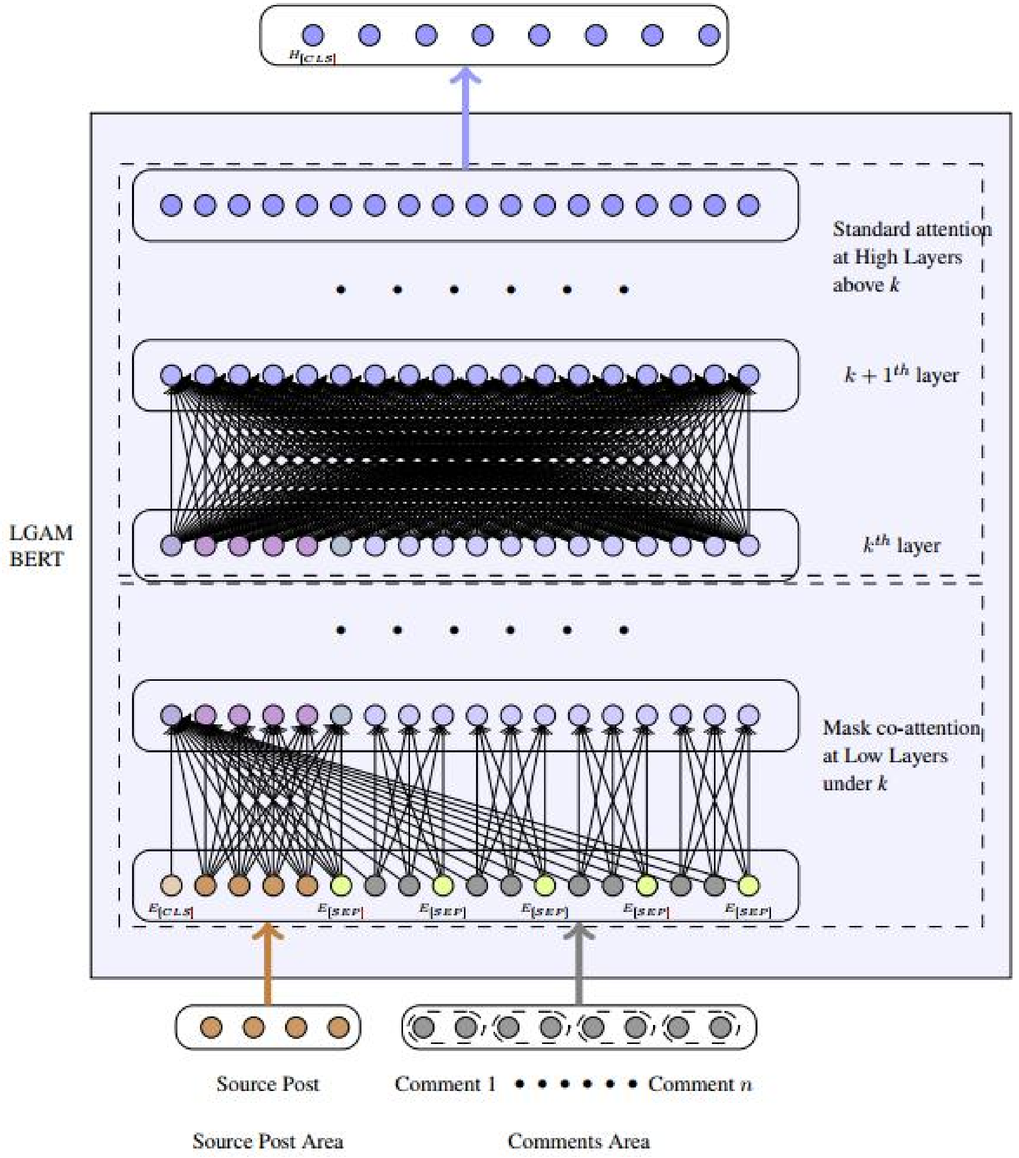
Attention-Mask:

$$AM(Q,K,V) = softmax(\frac{QK^T + M}{\sqrt{d}})V \quad (6)$$

$H^0 = E_{[s;CS]}$ is the embedding of the input sequence

$$H^i = \begin{cases} AM(W_Q^i H^{i-1}, W_K^i H^{i-1}, W_V^i H^{i-1}), \\ \qquad 1 \le i \le k \\ A(W_Q^i H^{i-1}, W_K^i H^{i-1}, W_V^i H^{i-1}), \\ \qquad k < i \le n \end{cases} \quad (7)$$

the H^n is **L** in Formula (1) or **R** in Formula (2)



Experiments

Statistic[1]	Ma-Weibo	Weibo20	Twitter15	Twitter16
# of post	4664	6068	742	412
# of true	2351	3034	370	205
# of false	2313	3034	372	207
Avg. len. of post	105	88	19	19
Avg. # of cmt.	804	62	22	16
Avg. len. of cmt. set	8484	1359[2]	242	202

Experiments

Method[1]	Ma-Weibo				Weibo20				Twitter15				Twitter16			
	F1	Rec	Pre	Acc	F1	Rec	Pre	Acc	F1	Rec	Pre	Acc	F1	Rec	Pre	Acc
Traditional ML:																
SVM-TS	0.8827	0.8858	0.9150	0.8846	0.8914	0.8943	0.9242	0.8932	0.7372	0.7387	0.7437	0.7385	0.7589	0.7638	0.7901	0.7646
Graph-structured:																
Ma-RvNN	0.9481	0.9484	0.9495	0.9481	0.9419	0.9459	0.9379	0.9431	0.9412	**0.9730**	0.9114	0.9392	0.9302	**0.9756**	0.8889	0.9268
CNN	0.9515	0.9520	0.9515	0.9510	0.9322	0.9334	0.9314	0.9331	0.8756	0.9103	0.8559	0.8721	0.9233	0.9408	0.9142	0.9214
Bi-GCN	0.9612	0.9613	0.9616	0.9612	0.9047	0.9098	0.9112	0.9112	0.9596	0.9595	0.9599	0.9596	0.9514	0.9514	0.9519	0.9515
GCAN	-	-	-	-	-	-	-	-	0.8250	0.8295	0.8257	0.8767	0.7593	0.7632	0.7594	0.9084
Transformer-based:																
BERT	0.9603	0.9598	0.9634	0.9603	0.9613	0.9616	0.9611	0.9621	0.9343	0.9397	0.9364	0.9367	0.9291	0.9274	0.9304	0.9320
RoBERTa	0.9603	0.9605	0.9603	0.9603	0.9611	0.9611	0.9612	0.9611	0.9352	0.9354	0.9368	0.9353	0.9367	0.9371	0.9400	0.9369
Longformer	0.8998	0.8999	0.9108	0.9084	0.9557	0.9558	0.9571	0.9561	0.9056	0.9056	0.9069	0.9057	0.9075	0.9076	0.9110	0.9078
PLAN	0.9208	0.9271	0.9159	0.9226	0.9246	0.9231	0.9275	0.9256	0.9278	0.9133	0.9510	0.9213	0.9431	0.9508	0.9336	0.9423
Ensemble models:																
Wu-Stacking	0.9347	0.9352	0.9391	0.9348	0.9378	0.9379	0.9398	0.9379	0.9285	0.9285	0.9297	0.9286	0.9247	0.9246	0.9261	0.9248
Bagging-BERT(2)	0.9667	0.9668	0.9667	0.9667	0.965	0.9651	0.9671	0.9651	0.9649	0.9649	0.9661	0.9650	0.9489	0.9488	0.9531	0.9490
Geng-Ensemble	0.9565	0.9567	0.9560	0.9560	0.9541	0.9532	0.9544	0.9534	0.9506	0.9528	0.9503	0.9512	0.9523	0.9537	0.9512	0.9518
STANKER(best)	**0.9747**	**0.9746**	**0.9746**	**0.9745**	**0.9716**	**0.9716**	**0.9719**	**0.9717**	**0.9715**	0.971	**0.9723**	**0.9717**	**0.9632**	0.962	**0.9651**	**0.9635**

Experiments

	Ma-Weibo		Weibo20		Twitter15		Twitter16	
model[1]	S[2]	C	S	C	S	C	S	C
BERT_0	0.9348		0.9385		0.9340		0.9247	
BERT_1	**0.9653**	**0.9648**	**0.9628**	**0.9665**	**0.9582**	**0.9447**	**0.9393**	**0.9393**
BERT_2	0.9601	0.9603	0.9601	0.9621	0.9514	0.9367	0.9272	0.9320
BERT_3	0.9554	0.9593	0.9586	0.9618	0.9514	0.9368	0.9271	0.9318

[1] "BERT_0": a single BERT, given only source posts; "BERT_1": a single BERT, equipped with the LGAM strategy; "BERT_2": a single BERT, not equipped with LGAM (viz. w/o LGAM); "BERT_3": a single BERT, not equipped with LGAM and DBSCAN (viz. w/o LGAM+DBSCAN).

[2] "S": only use sentimental comments as auxiliary data; "C": only use chronological comments as auxiliary data.

Table 4: Ablation study on BERT.

model[1]	Ma-Weibo	Weibo20	Twitter15	Twitter16
STANKER (best)	**0.9745**	**0.9717**	**0.9717**	**0.9635**
STANKER w/o LGAM	0.9684	0.9672	0.9649	**0.9635**
STANKER w/o C	0.9695	0.9669	0.9683	0.9562
STANKER w/o S	0.9691	0.9683	0.9635	0.9489
STANKER w/o C+S	0.945	0.9457	0.9491	0.9489
STANKER w/o *[CLS]*	0.9714	0.9696	0.9656	0.9564

[1] "w/o": without. "LGAM": level-grained attention mask. On two LGAM-BERT models, "w/o C": only use sentimental comments. "w/o S": only use chronological comments. "w/o C+S": only use source posts. "w/o *[CLS]*": use binary classification results instead of *[CLS]* vectors.

Table 5: Ablation study on *STANKER*.

Experiments

k[1]	Ma-Weibo		Weibo20		Twitter15		Twitter16	
	S	C	S	C	S	C	S	C
0	0.9601	0.9603	0.9601	0.9621	0.9514	0.9367	0.9272	0.9320
1	0.9575	0.9603	0.9624	0.9626	0.9406	0.9447	0.9344	0.9198
2	0.9601	0.9575	0.9596	0.9634	0.9406	0.9434	0.9345	0.9368
3	0.9612	0.9620	0.9576	0.9629	0.9474	0.9366	**0.9416**	0.9296
4	0.9625	0.9631	0.9609	0.9578	0.9420	0.9460	0.9246	0.9272
5	0.9582	0.9610	0.9550	0.9629	0.9407	0.9420	0.9343	0.9341
6	0.9630	0.9597	0.9619	0.9647	0.9474	0.9379	0.9222	0.9367
7	0.9646	0.9618	0.9618	0.9634	0.9512	0.9380	0.9319	0.9175
8	0.9644	0.9629	0.9618	0.9623	0.9539	**0.9474**	0.9318	0.9344
9	0.9623	0.9597	0.9600	0.9608	0.9472	0.9420	0.9197	0.9127
10	**0.9653**	**0.9648**	**0.9628**	**0.9665**	**0.9582**	0.9447	0.9393	**0.9393**
11	0.9618	0.9644	0.9621	0.9659	0.9407	0.9326	0.9199	0.9368
12	0.9610	0.9601	0.9614	0.9636	0.9487	0.9393	0.9249	0.9343

[1] "k=0" means "w/o LGAM".

Table 6: Ablation study on the splitting layer.

Experiments

	Ma-Weibo	Weibo20	Twitter15	Twitter16	Total
SVM-TS	0.25	0.33	0.08	0.05	0.71
Ma-RvNN	40	50	5	4	99
CNN	10	12.5	1.67	1.25	25.42
Bi-GCN	6	7	0.67	0.5	14.17
BERT	2.5	3.33	0.5	0.33	6.66
RoBERTa	2.5	3.33	0.5	0.33	6.66
Longformer	7.5	6	0.5	0.33	14.33
PLAN	3.33	4.17	0.83	0.67	9
Wu-Stacking	2.08	2.5	0.67	0.42	5.67
Bagging BERT(2)	5	6.67	1	0.67	13.34
Geng-Ensemble	15	17.5	3.75	2.5	38.75
STANKER(best)	5.17	6.83	1.12	0.75	13.87

Table 7: Training time (hours) of compared methods.

Experiments

	Xu's	**TsingHua**	**NTUSD**	
Weibo20	**0.9628**	0.9601	0.9612	
Ma-Weibo	**0.9653**	0.9554	0.9605	
	EmoLex	**SentiStrength**	**Bing Liu's**	**HowNet**
Twitter15	**0.9582**	0.9474	0.9339	0.9474
Twitter16	**0.9393**	0.9344	0.9344	0.9247

Table 8: Using different sentiment dictionaries.