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

Dongning Rao[1], **Xin Miao**[1], **Zhihua Jiang**[2]*, **Ran Li**[1]
[1]School of Computer, Guangdong University of Technology, Guangzhou 510006, P. R. China
[2] Department of Computer Science, Jinan University, Guangzhou 510632, P. R. China
raodn@gdut.edu.cn, miaox@mail2.gdut.edu.cn, tjiangzhh@jnu.edu.cn,
2111905160@mail2.gdut.edu.cn

Abstract

Rumor detection on social media puts pre-trained language models (LMs), such as BERT, and auxiliary features, such as comments, into use. However, on the one hand, rumor detection datasets in Chinese companies with comments are rare; on the other hand, intensive interaction of attention on Transformer-based models like BERT may hinder performance improvement. To alleviate these problems, we build a new Chinese microblog dataset named **Weibo20**[1] by collecting posts and associated comments from Sina Weibo and propose a new ensemble named *STANKER* (*Stacking neTwork bAsed-on atteNtion-masKed BERT*). *STANKER* adopts two level-grained attention-masked BERT (LGAM-BERT) models as base encoders. Unlike the original BERT, our new LGAM-BERT model takes comments as important auxiliary features and masks co-attention between posts and comments on lower-layers. Experiments on Weibo20 and three existing social media datasets showed that *STANKER* outperformed all compared models, especially beating the old state-of-the-art on Weibo dataset.

1 Introduction

Social media like Sina Weibo is an indispensable part of life, while rumors have severe consequences in political decision-making or manipulating public opinions (Lazer et al., 2018). Therefore, evil-doers can create and spread rumors on social media conveniently on a massive scale at a low cost (Ma et al., 2020), which provokes a text classification task called rumor detection (Li et al., 2019).

For text classification tasks, including rumor detection, transformer-based models like Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018), which achieved impressive results, have a significant performance variation when fine-tuned on small datasets (Risch and Krestel, 2020). Thus researchers proposed ensembles of multiple BERT models (Risch and Krestel, 2020; Fajcik et al., 2019; Liu et al., 2019a) to provide more robust predictions, but a big ensemble size makes the fine-tuning computationally expensive, for the training time and the inference time increase linearly with the ensemble size.

Moreover, the attention mechanism, which is a key of Transformer (Vaswani et al., 2017), makes the computational complexity scale quadratically with the input sequence length. It facilitates complex models to learn the contextual representation of a word via attending to other words. Encouragingly, a few studies indicated that not all attention is necessary (Gordon et al., 2020): partial attention can be pruned (Gordon et al., 2020; Michel et al., 2019) or masked (Liu et al., 2020; Yang et al., 2019) depending on specific tasks, because BERT learns different features at different levels.

Apart from the computation cost, the input length limitation is another obstacle in using pre-trained language models (LMs) to detect rumors. The content of social media posts is text shorter than 140 words with rich auxiliary features, e.g., comments and user profiles. Among these features, comments are semantically relevant to a source post and support or deny the original claim (Wei et al., 2019; Bian et al., 2020). However, social media posts often have comments whose total length exceeds the input-length limitation of LMs, demanding pre-processing like truncation. Unfortunately, as the classical pre-processing for inputting long texts into LMs, truncation discards valuable information in the truncated part[2]. Meanwhile, although

*Corresponding author.

[1]Our data and source code are available at: https://github.com/fip-lab/STANKER. All data collection was conducted following the policies of the host institutions' ethics board.

[2]For instance, for the Chinese microblog dataset Ma-Weibo, each post has 804 comments, and each comment set contains 8484 tokens on average; by contrast, an input sequence must be shorter than 512 tokens on BERT.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 3347–3363
November 7–11, 2021.

Longformer (Beltagy et al., 2020) was proposed recently to tackle long input sequences, excessive attention interactions may degrade the overall performance.

To alleviate these problems, we propose the *STANKER (Stacking neTwork bAsed-on atteNtion-masKed BERT)*, which adopts two level-grained attention-masked BERT models as base encoders, stacked with a final dense prediction layer with softmax activation that maps the 768-dimensional vectors to two outputs for this binary classification. Contributions of this paper:

- The only recent five-year Chinese social media rumor detection dataset, Weibo20, is built.
- We devise a new variate of BERT with linguistic noises targeted layer-grained technique, Level-Grained Attention-Masked BERT (viz. LGAM-BERT), which masks insignificant co-attention between source posts and comments on lower-layers.
- To make full use of comments, we select relative influential comments according to chronological order and a sentimental intensity ranking, thus producing two different training sets for base learners in the ensemble. Differences in training sets lead to the diversity of base learners, which contributes to the efficiency of the ensemble network.

Experimental results on four datasets show that *STANKER* outperforms existing methods. *STANKER* is the best approach for Weibo20, and its accuracy on Ma-Weibo (Ma et al., 2016) is higher than the old SOTA, Ma-RvNN (Ma et al., 2020). Furthermore, *STANKER* is the best of all compared methods on Twitter15 and Twitter16. Unlike previous ensemble models (Risch and Krestel, 2020; Liu et al., 2019a), the training cost of the *STANKER* is low due to the minimal ensemble size.

2 Related Work

2.1 Rumor Detection

A rumor is a statement whose authenticity is certified to be false or unverified (Difonzo and Bordia, 2007). Considering the tremendous number of Twitter and Weibo users, even a little promotion of the rumor detecting accuracy is precious. Rumor detection, framed as text classification tasks, can be cracked by either traditional machine learning approaches (Vicario et al., 2019; Gravanis et al., 2019) or deep neural networks (Meel and Vishwakarma, 2020), and comments or replies, as auxiliary features, are widely used.

Recent deep-learning based studies include: Wang embedded source posts and comments with sentimental features and then inputted them into a two-layer Gated Recurrent Unit (GRU) network (Wang and Guo, 2020); Kumar applied a tree LSTM to predict rumors with tree-structured replies (Kumar and Carley, 2019); Bian fed posts and replies into a Graph Convolution Network (GCN) to take advantage of propagation features, and later extended GCN to be Bi-directional GCN (viz. Bi-GCN) to explore the structures of wide dispersion on rumor detection (Bian et al., 2020); Zhang encoded replies in a temporal order through an LSTM component (Zhang et al., 2019); Riedel profited from the cosine similarity of news content and comments while setting a threshold of similarity to filter those irrelevant comments (Riedel et al., 2017); Lu put user profiles into GCNs to extract propagation features (Lu and Li, 2020).

Discouragingly, the only available dataset for Chinese social media rumor detection, Ma-Weibo (Ma et al., 2015), was collected five years ago, unlike similar tasks whose unique datasets are recently proposed (Wang et al., 2021; Mathew et al., 2021; Ana-Cristina et al., 2021).

2.2 Ensemble Strategy

Ensemble strategy can achieve better performance than a single model; also, the diversity of base learners is crucial (Zhou, 2012). All three types of ensembling algorithms, which are bagging, boosting, and stacking, improve performance, while recent studies standing on the shoulder of BERT further showed their advantages.

Bagging. Risch proposed an ensemble of multiple fine-tuned BERT models based on bagging and found that the F1-score drastically increased when ensembling up to 15 models, but the returns diminished for more models (Risch and Krestel, 2020). **Boosting**. Sharma recognized question entailment using two Sci-BERT models, stacked with a gradient boosting classifier (Sharma and Roychowdhury, 2019). Huang integrated multi-class boosting into BERT and used Transformer as the base classifier to choose more challenging training sets to fine-tune NLP tasks (Huang et al., 2020). **Stacking**. Stacking algorithms were proposed to accelerate BERT training via transferring knowledge (Gong et al., 2019; Yang et al., 2020). Liu proposed an architecture by blending 25 BERT models (Liu et al., 2019a). Wu combined feature engineering and an ensemble stacked with SVM,

Random Forest, and Naive Bayes (Wu et al., 2020).

2.3 Attention Mechanism

The self-attention mechanism is central and indispensable to SOTA Transformer models, including BERT (Vaswani et al., 2017), but not all attention is necessary: Gong found that in most layers, the self-attention distribution will concentrate locally around its position and the start-of-sentence token (Gong et al., 2019); Jawahar showed that BERT captures a rich hierarchy of linguistic information, with surface features (e.g., the presence of words in the sentence) in lower layers, syntactic features (e.g., the sensitivity to word order) in middle layers and semantic features (e.g., the tense) in higher layers (Jawahar et al., 2019).

Thus, attention masking or pruning methods have been proposed: 1) Liu introduced a visible matrix to limit the attention area of each token in their knowledge-enabled language representation model (K-BERT) (Liu et al., 2020). 2) Yang trained the permutation language model with two-stream attention: content stream attention, which is the same as the standard self-attention, and query stream attention, which does not have access information about the content (Yang et al., 2019). 3) Beltagy proposed the Longformer with an attention mechanism that is a drop-in replacement for the standard self-attention and scales linearly with the sequence length (Beltagy et al., 2020). 4) Gordon found that low levels of weight pruning do not affect pre-training loss or transfer to downstream tasks at all (Gordon et al., 2020).

3 Problem Statement

Let $S = \{s_1, s_2, ..., s_{|S|}\}$ be a set of source posts. Each $s_i \in S$ is a short text composed of 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 (viz. replies) $C_i = \{c_1^i, c_2^i, ...c_{|C_i|}^i\}$. Like s_i, each $c_j^i \in C_i$ is a word or character sequence. Each s_i is also associated with a binary label $y_i \in \{0, 1\}$ to represent its truthfulness, where $y_i = 1$ indicates s_i is a rumor and $y_i = 0$ means s_i is not.

Suppose the dataset is symbolized as $D = \{d_1, d_2, ..., d_{|D|}\}$ where each $d_i \in D$ is a tuple $\{s_i, C_i, y_i\}$. Given d_i, our goal is to predict the truthfulness y_i of source post s_i, i.e., binary classification. Due to the nature of social media, we regard s_i as primary data and C_i as auxiliary data.

4 The *STANKER*

4.1 Overall Structure

The overall structure of *STANKER* is shown in Figure 1. We select relatively valuable comments in the pre-processing according to chronological order and a sentimental intensity ranking (see Section 4.4), thus producing two different training sets for base learners. In training, we devise two Level-Grained Attention-Masked BERT (LGAM-BERT) models as base learners, which mask co-attention between source posts and comments on low-layers of BERT (see Section 4.3). Since the first token *[CLS]* summarizes the information from input tokens using a global attention mechanism, we extract the embedding representation of *[CLS]* (viz. a 768-dimensional vector) in the last layer of two LGAM-BERT models and concatenate them. The final prediction layer is a dense network with softmax activation that maps the concatenated vector to two outputs for this binary classification.

4.2 Stacking Ensemble

The basic idea of stacking is multi-stage training (Wu et al., 2020). In *STANKER*, the stacking ensemble strategy uses the pre-processed training data to train primary learners at the first stage and then combines their final representations to form a meta data set for training the meta learner at the second stage. The benefit of this stacking strategy is two-fold. On the one hand, BERT is a strong classifier, so integrating it or its variants as primary learners will provide a start-up ensemble with high accuracy. On the other hand, extracting the embedding representation of *[CLS]*, instead of the binary prediction result, will train the meta learner in a high-dimensional feature space.

4.3 LGAM-BERT

The detailed design of LGAM-BERT is shown in Figure 2. An attention function can be formulated as querying a dictionary with key-value pairs. The Transformer is a stack of multiple self-attention blocks (Vaswani et al., 2017). Inspired by masking *self-attention* (Liu et al., 2020; Yang et al., 2019), we present a new mask strategy that masks **co-attention** at low-levels of BERT. The co-attention concept was first proposed by (Lu et al., 2016)[3], indicating the attention between question texts and

[3]Lu also used co-attention to indicate the connection between source tweets and re-tweet users (Lu and Li, 2020).

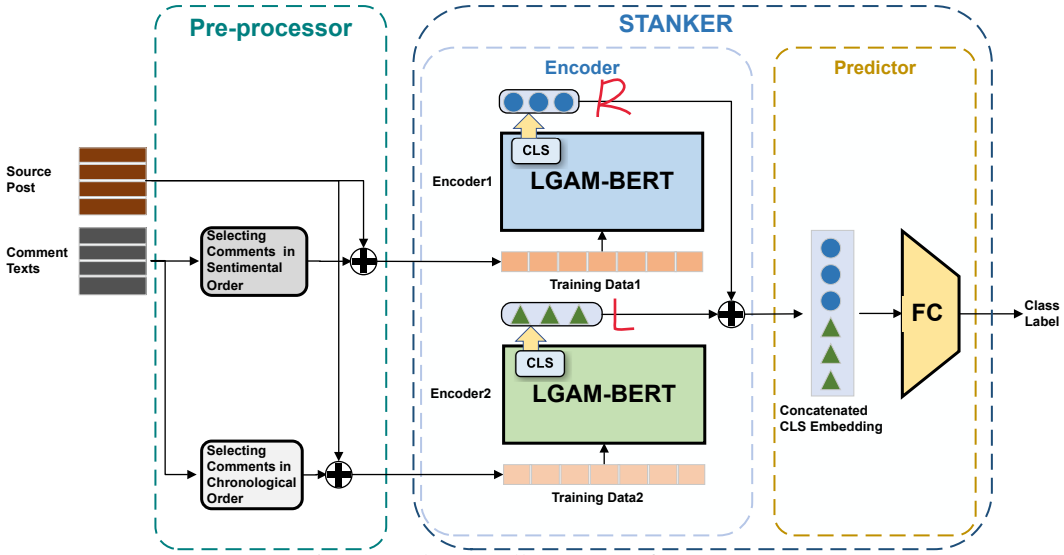


Figure 1: The overall structure of *STANKER*

answer texts in Question-Answer (QA) tasks. Similarly, given a sentence set separated by *[SEP]*, we suggest that *self-attention* attends words in the same sentence, while *co-attention* attends words in different sentences. See Figure 5 for such an example. After pre-processing, a comment-rich sentence set, where the source-post sentence and all comment sentences are separated by *[SEP]*, is inputted to LGAM-BERT. Precisely, for a pre-defined splitting layer k, we mask co-attention from the bottom layer to the k^{th} layer but calculate the standard attention from the $k+1^{th}$ layer to the top layer. The k is a super-parameter learned in the training process.

For our problem, since source posts and comments are not coherent texts, BERT may suffer from linguistic noise, via learning basic features (e.g., surface and syntactic features) from nearby texts on lower-layers (Jawahar et al., 2019). The LGAM strategy is novel. It masks co-attention between posts and comments on whole levels (viz. level-grained), while previous strategies only consider some local areas from the single-level aspect (Liu et al., 2020; Yang et al., 2019; Beltagy et al., 2020).

To support this, we conducted an interesting experiment to illustrate attention distance level-by-level on BERT. We calculated the accumulated distance between a token and its top 10 most-attended tokens, visualizing with the heat-maps. We found that tokens prefer to attend nearer words at low levels on BERT, while more distant words at high levels. From the view of the attention mechanism, erroneous predictions occur when the predictor attends inappropriate words. This phenomenon provides some expandability to our LGAM strategy. See Figure 7 and 8 in Appendix B for the details.

4.4 Comment Selection

The input layer first appends relevant comments to it for a given source post, transforming the original sentence into a comment-rich sentence set. When the length of the sentence set exceeds the input limitation, we select comments using some strategies instead of simple truncation[4]. On the one hand, we sort comments according to their replying time and prioritize comments that respond earlier. On the other hand, we calculate sentiment scores of comments and select those with high scores.

Formally, we adopt a sentiment dictionary *Dict* to score all comments (Rao et al., 2021). 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 $w \in c$. Then, we sort all comments according to sentiment scores and pick up the top ones until exceeding the input-length limitation.

Besides, we find that there exist highly similar comments, especially on Weibo datasets, are a waste of the tight input space. Therefore, we use the DBSCAN algorithm (Ester et al., 1996), a density-based spatial clustering algorithm, to reduce redundancy before selection. DBSCAN can remove similar comments and repeated words, making comments more compact. Figure 4 in Appendix A is an example.

[4]BERT, RoBERTa, Longformer, and Bagging-BERT adopt simple truncation when inputted data exceeds their length limitations.

4.5 Formal Description

Given a source post $s_i =< w_1^i, w_2^i, ...w_{l_i}^i >$, along with its chronological-comment set $CCS_i = \{c_1^i, c_2^i, ...c_{|CCS_i|}^i\}$, we suppose that the embedding[5] of s_i appended by CCS_i is $E_{[s_i;CCS_i]}$. Then, the post representation learned by a LGAM-BERT is $\mathbf{L}_i = [\mathbf{l}_1^i; \mathbf{l}_2^i; .....\mathbf{l}_m^i] \in \mathbb{R}^{m*d}$, where d is the dimensionality of hidden-state embedding.

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

Similarly, given s_i, along with its sentimental-comment set $SCS_i = \{c_1^i, c_2^i, ...c_{|SCS_i|}^i\}$, the post representation learned by a LGAM-BERT is $\mathbf{R}_i = [\mathbf{r}_1^i; \mathbf{r}_2^i; .....\mathbf{r}_m^i] \in \mathbb{R}^{m*d}$.

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

Then, we extract the first element of $\mathbf{L}_i$ and $\mathbf{R}_i$ respectively, which is the embedding of *[CLS]* in the last layer. The contextual post representation $\mathbf{PR}_i$ derived by:

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

Finally, we feed $\mathbf{PR}_i$ to a fully-connected network (FCN) and output the prediction via softmaxing.

The standard attention mechanism (Vaswani et al., 2017) is defined as:

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

where Q is a query vector, K is a key vector, and V is a value vector.

Inspired by mask-self-attention (Liu et al., 2020), we define a visible matrix M of tokens:

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

where $\ominus$ means that Q_i and K_j are injected from the same sentence and $\oslash$ means that Q_i and K_j are injected from different sentences. Figure 6 in Appendix A gives an example of the visible matrix. All co-attention is masked except for *[CLS]*, which sees every token and summarizes the global information. Thus, attention-mask (viz. AM) can be:

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

This equation sets an attention to be zero by adding the dot product sum and a negative value.

Next, level-grained attention-mask can be derived as follows. Suppose that there are n layers on BERT, and H^i is the output representation of the i^{th} layer $(1 \leq i \leq n)$ and $H^0 = E_{[s;CS]}$ is the embedding of the input sequence, given s is a source post and CS is its comment set. Let k be the number of the splitting layer shown in Figure 2, H^i can be derived by:

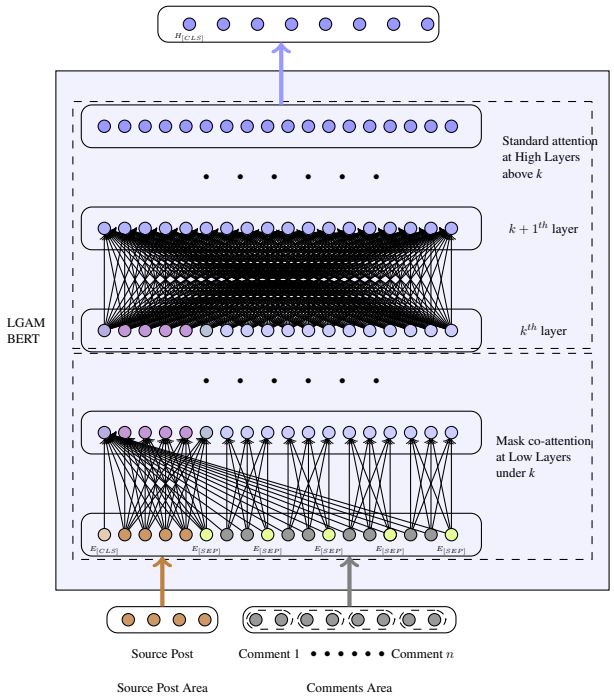


Figure 2: LGAM-BERT

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

where A is the standard attention function in Formula (4) and AM is the attention-mask function in Formula (6). Finally, the H^n is $\mathbf{L}$ in Formula (1) or $\mathbf{R}$ in Formula (2).

5 Experiments

5.1 Datasets

The experiments were conducted on four datasets (Weibo20, Ma-Weibo, Twitter15, and Twitter16). Weibo20 is constructed by ourselves, while the other three are widely used in the research line of rumor detection. Table 1 displays the basic statistics. Considering the average length of items, we allow at most 128 tokens for the post area and 384 tokens for the comment area on two Weibo datasets, and 64 tokens for the post area, and 312 tokens for the comment area on two Twitter datasets.

- **Weibo20** (ours). We collected 6068 Chinese posts published on Sina Weibo[6] in the last five years (i.e., 2016-2020), along with comments. We obtained user information and comments via Weibo API[7].

[5] It is the concatenation of word embedding and position embedding following the original BERT.

[6] https://service.account.weibo.com/?type=5&status=0

[7] https://open.weibo.com/wiki/API

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

[1] "#" means "number", "Avg." means "average", "len." means "length" and "cmt." means "comment". The length is the total number of tokens. A token is a word in an English sentence or a character in a Chinese sentence.
[2] Recently, Sina added a restriction on the length of collected data via API (viz. at most 200 comments per post). Therefore, the average length of comment sets on Weibo20 is much smaller than that on Ma-Weibo.

Table 1: Statistic of datasets.

The *annotation* process of Weibo20 is as follows. First, we collected 4411 rumors with their corresponding comments from the official Sina Weibo community management center[8], which gives a factual basis to testify against each rumor. Then, after data cleaning, which excludes redundant rumors and rumors without comments, only 3034 rumors were left. To balance the corpus, while assuming posts on trending topics that Weibo officially recommends are facts, we collected 3034 recommended posts with their corresponding comments as negative samples (viz. non-rumors). Further, we tried our best to balance the number of rumors and non-rumors on all 15 topics. The topic distribution is shown in Table 9 in Appendix A.

- **Ma-Weibo** (Ma et al., 2016). *Ma et al.* collected 4664 Chinese posts published on Sina Weibo before 2016, accompanied by user-profiles and comments.
- **Twitter15** and **Twitter16**. We also experimented on two Twitter datasets (Ma et al., 2017). We choose only "true" and "fake" labels as the ground truth. Since the original data does not contain comments, we obtained user information and comments via Twitter API.

5.2 Experimental Setting

We implemented LGAM-BERT based on pre-trained BERT-base[9]. The machine learning platform employed in the experiments is TensorFlow 1.14 with Python 3.6.7. Exerting a Xeon E5-2680(v2) CPU and an RTX 2080/3090 ti GPU, *STANKER* ran fast on Ubuntu 18.04.4 LTS.

The training process of *STANKER* has two stages. In the first stage, we fine-tune the two LGAM-BERT models, given a dataset. In the second stage, we freeze all the parameters of LGAM-BERT and learn the parameters of the final prediction layer. The learning rate was set to 2e-5 on all datasets. We ran eight epochs on Weibo datasets and 20 epochs on Twitter datasets. We adopted the tokenizer (Che et al., 2020), a Chinese sentiment dictionary (Xu et al., 2008) on Weibo datasets and an English sentiment dictionary (Mohammad and Turney, 2013) on Twitter datasets.

5.3 Compared Methods

We compared *STANKER* with 12 competitive methods on four datasets. These methods can be divided into four categories, as shown in Table 3. We ran the source code of all compared methods, except for GCAN[10]. We used the same setting presented in the original papers for a fair comparison. Apart from source post data, auxiliary data used by each method in our experiments is shown in Table 2.

- SVM-TS (Ma et al., 2015). A SVM based method.
- Ma-RvNN (Ma et al., 2020). They proposed a tree-structured model based on Recursive Neural Network (RvNN). This paper declared the recent SOTA on Ma-Weibo.
- CNN (Tu et al., 2021). A CNN-based model with joint text and propagation structure learning.
- Bi-GCN (Bian et al., 2020). The Bi-Directional Graph Convolution Network (Bi-GCN) is a new technique that beat five compared models, including SVM, CNN, and RvNN.
- GCAN (Lu and Li, 2020). The Graph-aware Co-Attention Network (GCAN) presented co-attention to connect source tweets and the corresponding re-tweet users' sequences for fake news detection.
- BERT (Devlin et al., 2018). BERT is a multi-layer bidirectional Transformer encoder. We experimented on BERT-base (L=12, H=768, A=12, Total Parameters=110M).
- RoBERTa (Liu et al., 2019b). Liu tested important BERT design choices and training strategies to present a more robust variant of BERT.
- Longformer (Beltagy et al., 2020). Beltagy presented a combination of local windowed attention and task-motivated global attention, making it easy to process long sequences.
- PLAN (Khoo et al., 2020). A post-level attention model which learns long-distance interactions between posts by Transformer.
- Wu-Stacking (Wu et al., 2020). Wu combined a stacking ensemble fused with feature engineering.
- Bagging-BERT(2). We re-produced the idea (Risch and Krestel, 2020) via bagging two original BERT models and randomly selecting comments.

[8] https://service.account.weibo.com/?type=5&status=0
[9] https://github.com/google-research/bert
[10] GCAN did neither release a complete version of source code in the provided link https://github.com/l852888/GCAN, nor give any result on Chinese microblog datasets in their original paper.

Methods	Auxiliary Data[1]	Length Limit
SVM-TS	19 content/user features	-
Ma-RvNN	comments[2]	-
CNN	comments	-
Bi-GCN	comments	-
GCAN	10 user features	-
BERT	C comments	512
RoBERTa	C comments	512
Longformer	C comments	4096
PLAN	C comments	-
Wu-Stacking	10 content/user features	-
Bagging-BERT(2)	R comments	512
Geng-Ensemble	C comments	-
STANKER	C & S comments	512

[1] "C" means "chronological", "S" means "sentimental", and "R" means "random". The length limit is the allowed largest number of input tokens. "-" means "no limit".
[2] Ma-RvNN, CNN, and Bi-GCN use comment contents and propagation paths built by reply-user orders.

Table 2: Input description of compared methods.

- Geng-Ensemble (Geng et al., 2019). An ensemble network is composed of three RNN-based learners, aggregating results by majority voting.
- *STANKER* (ours). We presented our best model in the experiments by probing important design choices of *STANKER*.

5.4 Primary Results

Table 3 shows primary experimental results of all compared methods on four datasets. We reported the average result on 5-fold cross-validation. The best model of *STANKER* has the following design choices: using chronological comments and sentimental comments, utilizing attention mask via setting the best splitting layer k, and stacking with the final FCN composed of 128 hidden units. The ablation study in Section 5.5 and 5.6 will further explain the contribution of design choices.

Preliminary conclusions are:

- Among all tested methods, *STANKER* achieved the highest classification accuracy and the F1 score on four datasets.
- Both BERT and RoBERTa are SOTA on general text classification tasks. However, compared with BERT, *STANKER* gained an up to 1.4% accuracy improvement on Weibo datasets and 3.5% accuracy improvement on Twitter datasets.
- Both PLAN and Longformer are good at processing long sequences. However, *STANKER* performed better than any of them, which indicates that using all comments is not the best option.
- Graph-structured models include Ma-RvNN, CNN, Bi-GCN, and GCAN. Ma-RvNN, the recent SOTA on Ma-Weibo, uses tree structures for propagation paths. CNN jointly learns text and propagation structure representation. Bi-GCN trains graph convolution networks. GCAN proposes graph-aware co-attention networks. Bi-GCN performed best among these four models; however, STANKER was superior to all of them.
- We compared *STANKER* with related ensemble models proposed in the recent two years. Both *STANKER* and Bagging-BERT(2) performed better than Wu-Stacking and Geng-Ensemble, which indicates the advantage of integrating BERT models. Further, *STANKER* performed better than Bagging-BERT(2), which indicates the advantage of taking our LGAM-BERT models.

5.5 Ablation Study

There were two experiment sets in the ablation study. We tested the contribution of design choices of *STANKER* in two modes: a single-model mode and an ensemble mode. We reported the average accuracy on each dataset.

- In the single-model mode, we designed the "BERT_N" models, where N = 0, 1, 2, 3. We used the training subset that contained only one kind of comment: sentimental(S) or chronological(C). As shown in Table 4, "BERT_1" performed best, which reveals that the LGAM strategy is effective even for a single model. Besides, the result of "BERT_3" showed that the DBSCAN algorithm is more effective on Weibo datasets than on Twitter datasets and more useful for sentimental comments than for chronological comments.
- In the ensemble mode, we utilized two LGAM-BERT models and tested the performance of *STANKER* by removing a separate component or their combinations. As shown in Table 5, there were three findings. First, the overall performance degraded most when running "*STANKER* w/o C+S", which revealed the importance of comments as auxiliary data. Take the Weibo20 dataset as an example. Given only source posts, a *STANKER* model only achieved an accuracy of 0.9457. However, added by C+S comments, this model got a much higher accuracy of 0.9672. Second, the performance of "*STANKER* w/o LGAM" was second to last, which indicated the LGAM strategy contributed more to *STANKER* than other components. Third, both "*STANKER* w/o S" and "*STANKER* w/o C" degraded, which indicated that adopting diverse comments is more effective.

5.6 Attention Mask Strategy Analysis

In this experiment, we tested the super-parameter k, the splitting layer on LGAM-BERT shown in Figure 2. Thanks to the implementation on BERT-base, we tested all values of k (viz. from 0 to 12), attempting to find out an “oracle” value. Experimental

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**

[1] We ran source code of all compared methods, except for GCAN, whose result was cited from the original paper.

Table 3: Results of compared methods on four datasets.

	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*.

results on four datasets were shown in Table 6. We found that, even though there was some volatility, the accuracy increased when setting a big value to k; however, the returns diminished for bigger and bigger values. Particularly, when $k = 10$, we got the highest accuracy in six-eighth cases. Therefore, we found an approximate "oracle" value, i.e., $k = 10$. As a result, we set $k = 10$ whenever adopting the LGAM strategy in *STANKER*.

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.

	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.

5.7 Training Efficiency

In this part, we reported the training time of all compared methods. As shown in Table 7, as an ensemble model, the training cost of *STANKER* was low. It spent a little more time than Bagging-BERT(2) due to the pre-processing. However, our model got up to 0.8% improvement on Weibo datasets and 1.4% on Twitter datasets over Bagging-BERT(2). Also, *STANKER* ran faster than most non-ensemble models, e.g., Longformer.

5.8 Early Detection

The earlier a model can detect rumors, the more practical it is (Gao et al., 2020). Therefore, we conducted experiments for early detection. We collected comments every five minutes (viz. a checkpoint) and fed them to each detection model. Figure 3 showed that, as comments accumulated over time, our model was the earliest to reach a maximum classification accuracy. This result reveals the early-detection ability of *STANKER*.

5.9 Sentiment Dictionaries

Finally, we reported the results of using different sentiment dictionaries on *STANKER*. In total, we tested three Chinese dictionaries (Xu's lexicon (Xu et al., 2008), TsingHua lexicon (Li and Sun, 2007), and NTUSD (Ku and Chen, 2007)) and four En-

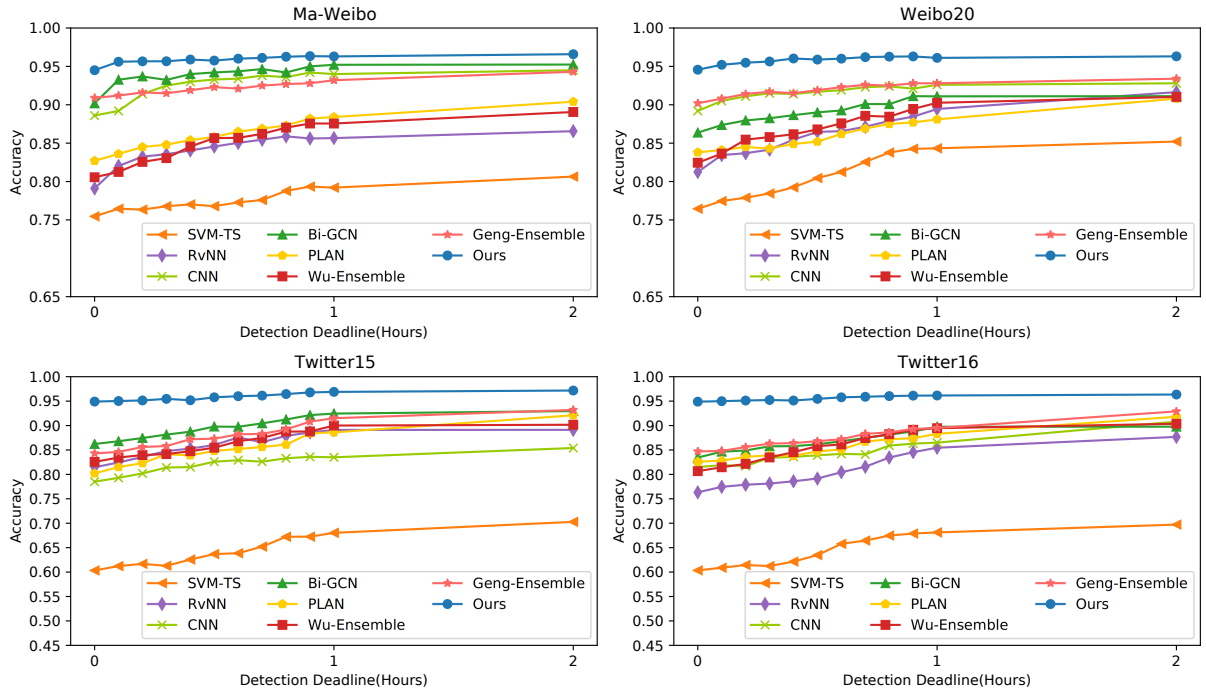


Figure 3: Early detection.

	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.

glish dictionaries (EmoLex (Mohammad and Turney, 2013), SentiStrength[11], Bing Liu's lexicon (Hu and Liu, 2004), and HowNet lexicon (Zhu et al., 2006). These dictionaries have different sizes and sentiment levels. For polarity-only dictionaries (e.g., Bing Liu's lexicon), we set the sentiment value of a positive word to be 1 and that of a negative word to be -1. Further, with the same sentiment score, a shorter sentence has higher sentimental intensity. The accuracy scores were reported in Table 8. The experimental findings demonstrated non-significant improvement when using different sentiment dictionaries. However, Xu's lexicon and EmoLex performed best, respectively.

6 Conclusion

Rumor control is one of the principal tasks of the Cybersecurity & Infrastructure Security Agency (CISA)[12]. Even 1% of the number of rumors posted or forwarded by the 521 million active Sina Weibo users will be a big event. For rumor detection, existing ensemble models did not realize their full potential. To alleviate this, we build a new Weibo dataset and propose a new ensemble model which achieved the best results on all tested datasets.

The novelty of our method does not rely on the overall architecture but on its novel proposal of LGAM-BERT models with comments to the original post as auxiliary data. We model co-attention between source posts and comments and propose a strategy that masks co-attention on lower layers of BERT. Unlike previous studies, we employ the masking strategy on the whole attention layer instead of on random text spans. Although the impact of each used component is not significant, a convincing set of experiments shows *STANKER* has superior performance when compared to numerous other SOTA methods on four different datasets. Our future work includes considering more features as auxiliary data, e.g., user profiles, and testing the LGAM strategy on more NLP tasks, e.g., dialog generation or text summarization.

Acknowledgements

This paper is supported by Guangdong Basic and Applied Basic Research Foundation, China (Grant No. 2021A1515012556).

[11] http://sentistrength.wlv.ac.uk/

[12] https://www.cisa.gov/rumorcontrol

References

Rogoz Ana-Cristina, Gaman Mihaela, and Ionescu. Radu Tudor. 2021. Saroco: Detecting satire in a novel romanian corpus of news articles. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Tian Bian, Xi Xiao, Tingyang Xu, Peilin Zhao, Wenbing Huang, Yu Rong, and Junzhou Huang. 2020. Rumor detection on social media with bi-directional graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 549–556.

Wanxiang Che, Yunlong Feng, Libo Qin, and Ting Liu. 2020. N-ltp: A open-source neural chinese language technology platform with pretrained models. *arXiv preprint arXiv:2009.11616*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

N. Difonzo and P. Bordia. 2007. Rumor psychology: Social and organizational approaches. *american psychological association*.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA*, pages 226–231.

Martin Fajcik, Pavel Smrz, and Lukás Burget. 2019. BUT-FIT at semeval-2019 task 7: Determining the rumour stance with pre-trained deep bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2019, Minneapolis, MN, USA, June 6-7, 2019*, pages 1097–1104.

Jie Gao, Sooji Han, Xingyi Song, and Fabio Ciravegna. 2020. RP-DNN: A tweet level propagation context based deep neural networks for early rumor detection in social media. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 6094–6105.

Yue Geng, Zheng Lin, Peng Fu, and Weiping Wang. 2019. Rumor detection on social media: A multi-view model using self-attention mechanism. In *International Conference on Computational Science*, pages 339–352. Springer.

Linyuan Gong, Di He, Zhuohan Li, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Efficient training of BERT by progressively stacking. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 2337–2346.

Mitchell A Gordon, Kevin Duh, and Nicholas Andrews. 2020. Compressing bert: Studying the effects of weight pruning on transfer learning. *arXiv preprint arXiv:2002.08307*.

Georgios Gravanis, Athena Vakali, Konstantinos Diamantaras, and Panagiotis Karadais. 2019. Behind the cues: A benchmarking study for fake news detection. *Expert Systems with Applications*, 128:201–213.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004*, pages 168–177.

Tongwen Huang, Qingyun She, and Junlin Zhang. 2020. Boostingbert: Integrating multi-class boosting into BERT for NLP tasks. *CoRR*, abs/2009.05959.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Ling Min Serena Khoo, Hai Leong Chieu, Zhong Qian, and Jing Jiang. 2020. Interpretable rumor detection in microblogs by attending to user interactions. *arXiv preprint arXiv:2001.10667*.

Lunwei Ku and Hsin-Hsi Chen. 2007. Mining opinions from the web: Beyond relevance retrieval. *J. Assoc. Inf. Sci. Technol.*, 58(12):1838–1850.

Sumeet Kumar and Kathleen M Carley. 2019. Tree lstms with convolution units to predict stance and rumor veracity in social media conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5047–5058.

David M. J. Lazer, Matthew A. Baum, Yochai Benkler, Adam J. Berinsky, Kelly M. Greenhill, Filippo Menczer, Miriam J. Metzger, Brendan Nyhan, Gordon Pennycook, David Rothschild, Michael Schudson, Steven A. Sloman, Cass R. Sunstein, Emily A. Thorson, Duncan J. Watts, and Jonathan L. Zittrain. 2018. The science of fake news. *Science*, 359(6380):1094–1096.

Jun Li and Maosong Sun. 2007. Experimental study on sentiment classification of chinese review using machine learning techniques. pages 393 – 400.

Quanzhi Li, Qiong Zhang, Luo Si, and Yingchi Liu. 2019. Rumor detection on social media: Datasets, methods and opportunities. *CoRR*, abs/1911.07199.

Shuaipeng Liu, Shuo Liu, and Lei Ren. 2019a. Trust or suspect? an empirical ensemble framework for fake news classification. In *WSDM Cup 2019 Fake News Classification Challenge. Online proceeding.*

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*, pages 2901–2908.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical question-image co-attention for visual question answering. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 289–297.

Yiju Lu and Chengte Li. 2020. Gcan: Graph-aware co-attention networks for explainable fake news detection on social media. *arXiv preprint arXiv:2004.11648*.

Jing Ma, Wei Gao, Shafiq R. Joty, and Kam-Fai Wong. 2020. An attention-based rumor detection model with tree-structured recursive neural networks. *ACM Trans. Intell. Syst. Technol.*, 11(4):42:1–42:28.

Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016. Detecting rumors from microblogs with recurrent neural networks.

Jing Ma, Wei Gao, Zhongyu Wei, Yueming Lu, and Kam-Fai Wong. 2015. Detect rumors using time series of social context information on microblogging websites. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*, pages 1751–1754.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2017. Detect rumors in microblog posts using propagation structure via kernel learning. Association for Computational Linguistics.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2021. Hatexplain: A benchmark dataset for explainable hate speech detection. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, New York, NY, USA, February 2-9, 2021*, volume abs/2103.02548. AAAI Press.

Priyanka Meel and Dinesh Kumar Vishwakarma. 2020. Fake news, rumor, information pollution in social media and web: A contemporary survey of state-of-the-arts, challenges and opportunities. *Expert Systems with Applications*, 153:112986.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, pages 14014–14024.

Saif Mohammad and Peter D. Turney. 2013. Crowdsourcing a word-emotion association lexicon. *Comput. Intell.*, 29(3):436–465.

Dongning Rao, Sihong Huang, Zhihua Jiang, Ganesh Gopal Deverajan, and Rizwan Patan. 2021. A dual deep neural network with phrase structure and attention mechanism for sentiment analysis. *Neural Comput. Appl.*, 33(17):11297–11308.

Benjamin Riedel, Isabelle Augenstein, Georgios P Spithourakis, and Sebastian Riedel. 2017. A simple but tough-to-beat baseline for the fake news challenge stance detection task. *arXiv preprint arXiv:1707.03264*.

Julian Risch and Ralf Krestel. 2020. Bagging BERT models for robust aggression identification. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, TRAC@LREC 2020, Marseille, France, May 2020*, pages 55–61.

Prakhar Sharma and Sumegh Roychowdhury. 2019. IIT-KGP at MEDIQA 2019: Recognizing question entailment using sci-bert stacked with a gradient boosting classifier. In *Proceedings of the 18th BioNLP Workshop and Shared Task, BioNLP@ACL 2019, Florence, Italy, August 1, 2019*, pages 471–477.

Kefei Tu, Chen Chen, Chunyan Hou, Jing Yuan, Jundong Li, and Xiaojie Yuan. 2021. Rumor2vec: a rumor detection framework with joint text and propagation structure representation learning. *Information Sciences*, 560:137–151.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Michela Del Vicario, Walter Quattrociocchi, Antonio Scala, and Fabiana Zollo. 2019. Polarization and fake news: Early warning of potential misinformation targets. *ACM Transactions on the Web (TWEB)*, 13(2):1–22.

Xiaoyang Wang, C. Li, Jianqiao Zhao, and Dong Yu. 2021. Naturalconv: A chinese dialogue dataset towards multi-turn topic-driven conversation. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, New York, NY, USA, February 2-9, 2021*, volume abs/2103.02548. AAAI Press.

Zhihong Wang and Yi Guo. 2020. Rumor events detection enhanced by encoding sentimental information into time series division and word representations. *Neurocomputing*.

Penghui Wei, Nan Xu, and Wenji Mao. 2019. Modeling conversation structure and temporal dynamics for jointly predicting rumor stance and veracity. *arXiv preprint arXiv:1909.08211*.

Yu Wu, Yan Zeng, Jie Yang, and Zhenni Zhao. 2020. Weibo rumor recognition based on communication and stacking ensemble learning. *Discrete Dynamics in Nature and Society, vol. 2020, Article ID 9352153, 12 pages. DOI:https://doi.org/10.1155/2020/9352153*, 2020.

L. Xu, Hongfei Lin, Y. Pan, H. Ren, and J. Chen. 2008. Constructing the affective lexicon ontology. *Journal of the China Society for Scientific and Technical Information*, 27:180–185.

Cheng Yang, Shengnan Wang, Chao Yang, Yuechuan Li, Ru He, and Jingqiao Zhang. 2020. Progressively stacking 2.0: A multi-stage layerwise training method for BERT training speedup. *CoRR*, abs/2011.13635.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 5754–5764.

Qiang Zhang, Aldo Lipani, Shangsong Liang, and Emine Yilmaz. 2019. Reply-aided detection of misinformation via bayesian deep learning. In *The World Wide Web Conference*, pages 2333–2343.

Zhihua Zhou. 2012. *Ensemble Methods: Foundations and Algorithms (1st ed.)*. Chapman and Hall/CRC. https://doi.org/10.1201/b12207.

Yanlan Zhu, Jin Min, Yaqian Zhou, Xuanjing Huang, and Lide Wu. 2006. Semantic orientation computing based on hownet. *Journal of Chinese Information Processing*, 20(1):16–22.

Appendices

A Examples

Figure 4 shows an example of how the DBSCAN algorithm removes repeated words. Before precessing, redundant words exit. E.g., three "Speechless" and two "Gross" (circled in red). After precessing, only one copy is kept (circled in green).

Figure 5 shows an example of the self-attention and the co-attention, given a source post sentence and two comment sentences separated by *[SEP]*. In Figure 5, brown lines indicate the self-attention inside the source post sentence; gray lines signal the self-attention inside a comment sentence; blue lines highlight the co-attention between the source post sentence and a comment sentence.

Figure 6 shows an example of the visible matrix for masking co-attention, given a source post sentence and four comment sentences. The blank areas indicate the invisible areas. All co-attention is masked, except for that of the *[CLS]*. We keep all co-attention of *[CLS]* because it has to see each token to summarize the global information.

B Attention Study

We conducted an interesting experiment to illustrate attention distance. We calculated the accumulated distance between a token and its top 10 most-attended tokens, visualizing with the heat-maps. Figure 7 and Figure 8 show the heat-maps on Ma-Weibo and Weibo20 as an example. Each figure has two branches: the chronological branch (viz. using the training sub-set containing only chronological comments) and the sentimental branch (viz. using the training sub-set containing only sentimental comments). Given a token t, let $t_1, ...t_{10}$ be its top 10 most-attended tokens. We use a function called *Distance* to return the distance between two tokens in an input sequence. Then, the average attention-distance sum (*ADS*) is defined as follows:

$$ADS = \frac{\sum_{i=1}^{n}\sum_{j=1}^{10} Distance(t_i, t_{i_j})}{n} \quad (8)$$

where n is the total number of tokens on a dataset.

We list all *ADS* values layer by layer with the growth of training depth, as shown in Figure 7 and Figure 8. We set every 50 steps as a checkpoint to illustrate training depth. Larger *ADS* values indicate higher attention weights. Further, the deeper the color is, the farther the attention distance is. This phenomenon reveals that tokens prefer to attend nearer words at low levels on BERT, while more distant words at high levels. This test provides some expandability to our LGAM strategy.

Further, Table 10 lists the top 10 most-attended tokens for each dataset. The list provides evidential words for the prediction and some guidance for the saliency analysis. Another interesting finding is the differences between the word clouds of four datasets (see Figure 9), which adjusts the necessity of building an updated social media rumor detection dataset.

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]...

Figure 4: A DBSCAN example.

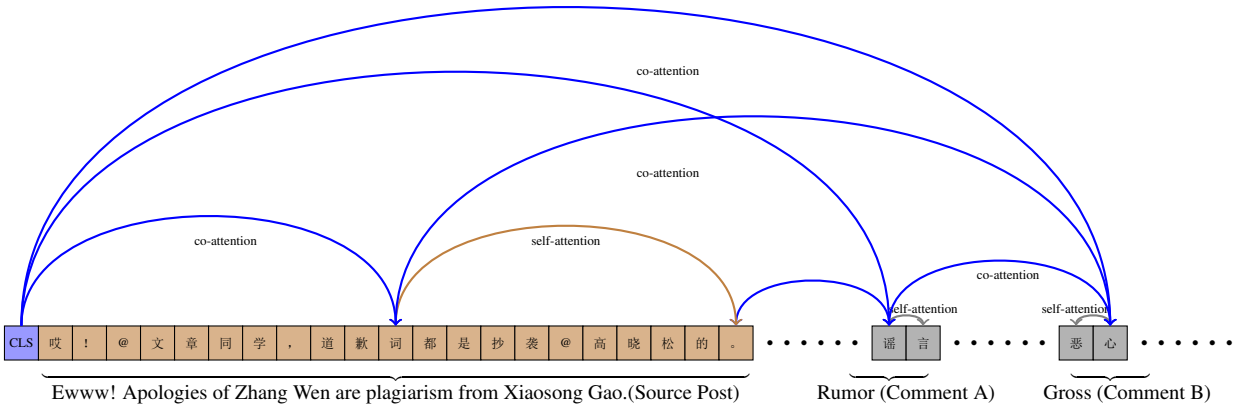


Figure 5: A self-attention and co-attention example

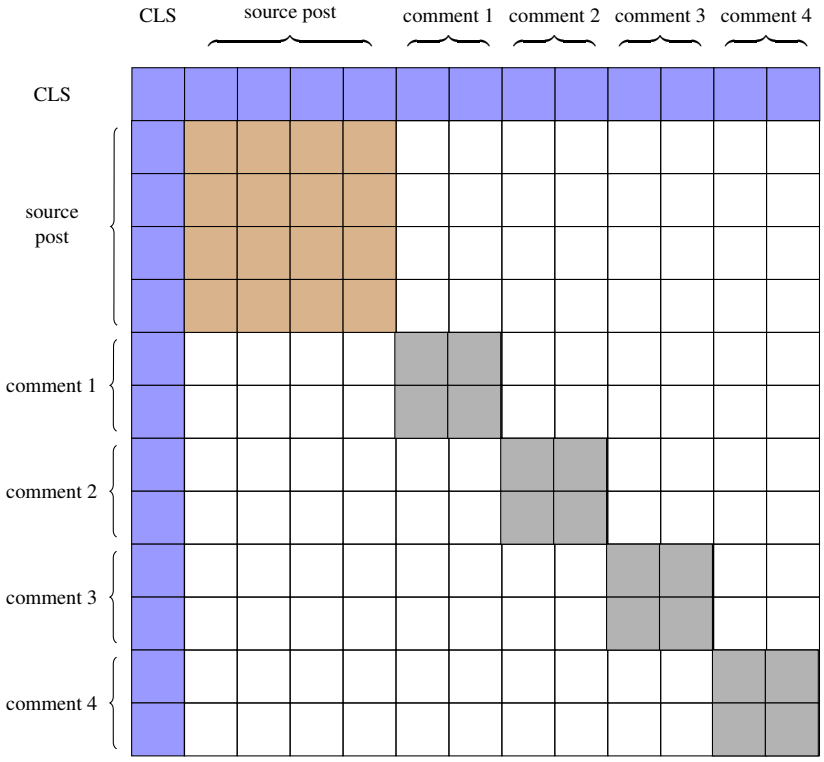
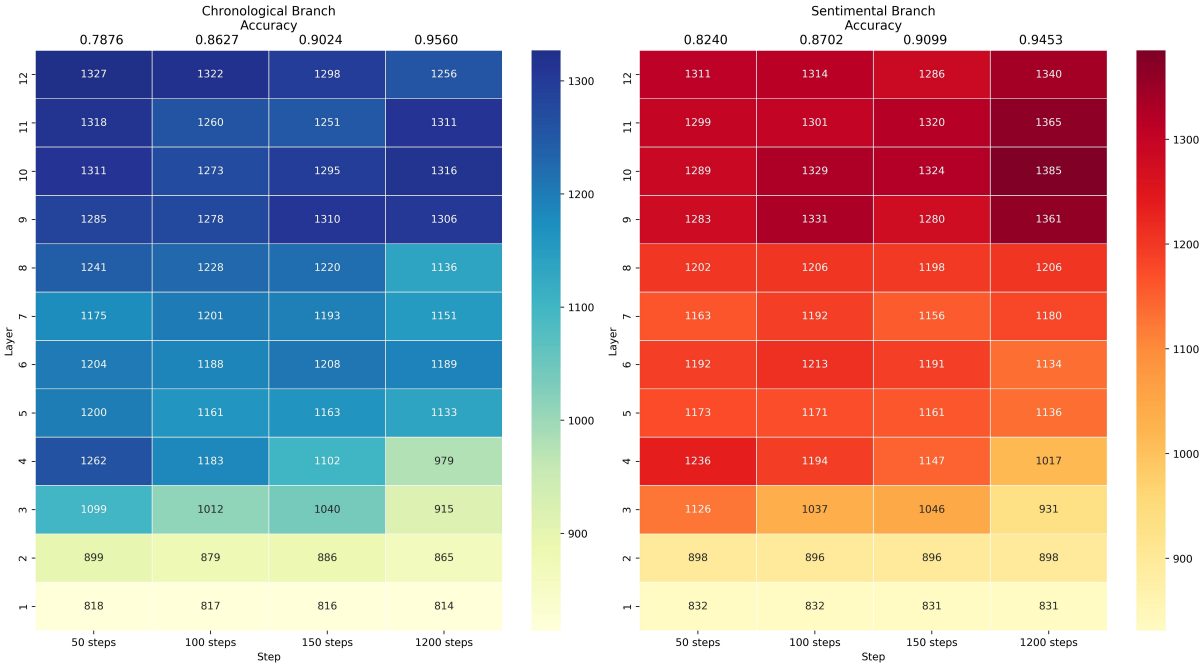


Figure 6: Visible matrix for attention mask on LGAM-BERT (the blank area indicates invisibility).

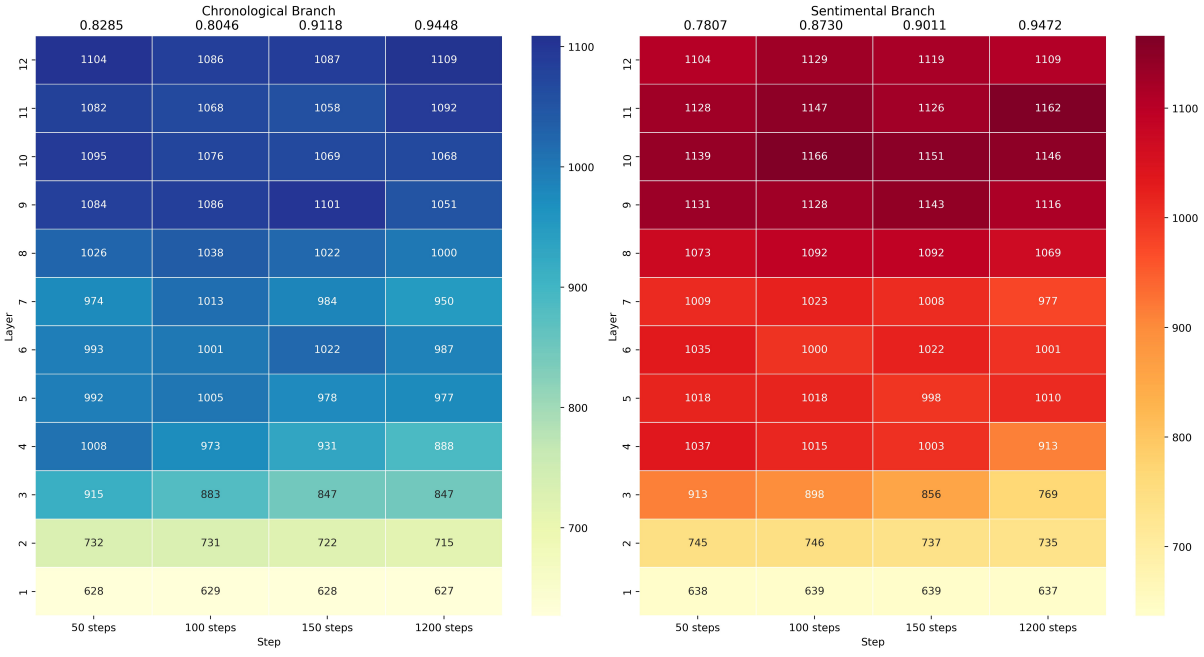
	Social	International	Food	Technology	School	Parenting	Sports	Finance	Health	Science	Traveling	Military	History	Estate	Celebrity	Total
Non-rumor	1887	321	261	11	185	6	18	22	139	68	12	35	17	2	50	3034
Rumor	1651	419	368	18	188	4	19	23	154	34	11	46	11	4	84	3034

Table 9: Topic distribution of Weibo20.



(a) Chronological Branch (b) Sentimental Branch

Figure 7: Attention heatmap for Ma-Weibo

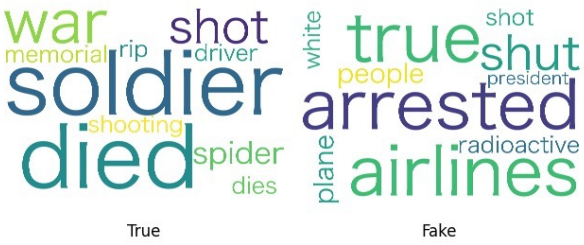


(a) Chronological Branch (b) Sentimental Branch

Figure 8: Attention heatmap for Weibo20

(a) Word Cloud of Ma-Weibo

(b) Word Cloud of Weibo20



(c) Word Cloud of Twitter15

(d) Word Cloud of Twitter16

Figure 9: Word Clouds of four datasets

	Ma-Weibo	Weibo20	Twitter15	Twitter16
TRUE	笑(laugh)	爱(love)	soldier	house
	喵(meow)	喜(delight)	died	rainbow
	花(flower)	很(very)	war	hostages
	爱(love)	好(good)	shot	police
	眼(eye)	吃(eat)	spider	white
	傻(foolish)	悲(sad)	memorial	colors
	赞(praise)	错(fault)	dies	cafe
	喜(delight)	人(person)	shooting	pope
	人(person)	老(old)	rip	soldier
	美(pretty)	笑(laugh)	driver	shooting
FAKE	假(fake)	造(make)	arrested	refugees
	求(beg)	假(fake)	true	jobs
	造(make)	事(affair)	airlines	shooting
	怒(anger)	人(person)	shut	father
	请(please)	州(state)	people	poll
	惊(shock)	死(death)	plane	biological
	恐(fear)	南(south)	radioactive	employees
	骗(cheat)	北(north)	shot	mass
	疑(doubt)	爆(burst)	white	terrorist
	人(person)	黑(black)	president	true

Table 10: Top-10 attended words on datasets.