Proceedings of the 2022 21st IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom 2022), pp. 383-390, IEEE, 2022, https://doi.org/10.1109/TrustCom56396.2022.00060

"Comments Matter and The More The Better!": Improving Rumor Detection with User Comments

Yang Xu*, Jie Guo*‡, Weidong Qiu*, Zheng Huang*, Enes Altuncu† and Shujun Li†‡
*Shanghai Jiao Tong University, China
{xuyang2018, guojie, qiuwd, huang-zheng}@sjtu.edu.cn
†University of Kent, UK
{ea483, S.J.Li}@kent.ac.uk
‡Corresponding co-authors.

***Abstract*—While many online platforms bring great benefits to their users by allowing user-generated content, they have also facilitated generation and spreading of harmful content such as rumors. Researcher have proposed different rumor detection methods based on features extracted from the original post and/or associated comments, but how comments affect the performance of such methods remains largely less understood. In this paper, we first propose a new BERT-based rumor detection method that can outperform other state-of-the-art methods, and then used it to study the role of comments in rumor detection. Our proposed method concatenates the original post and associated comments to form a single long text, which is then segmented into shorter chunks more suitable for BERT-based vectorization. Features extracted from all trunks are fed into a classifier based on an LSTM network or a transformer layer for the classification task. The experimental results on the PHEME and Ma-Weibo datasets proved the superior performance of our method. We conducted additional experiments on different settings of our proposed method to study different aspects of the role comments play in the rumor detection task. These additional experiments led to some very interesting findings, including the surprising result that fixed-length segmentation is better than natural segmentation, and the observation that including more comments can help improve the rumor detector's performance. Some of these findings have profound operational implications for online platforms, e.g., commentators can contribute to rumor detection positively so online platforms can leverage the crowd intelligence to detect online rumors more effectively without applying over-strict content consensus policies.**

***Keywords*—Rumor Detection, Social Media, BERT, Transformer, Comments**

I. Introduction

With the rapid development of the Internet technology, online services facilitating user-generated content (UGC) have helped online users to obtain information and exchange opinions more effectively. Despite their great values to their users, the rapid expansion of online services supporting UGC has brought in new problems such as wide spread of mis-, dis- and mal-information. For example, since the outbreak of the COVID-19 pandemic in early 2020, many Internet users have participated in creating and/or spreading misinformation or rumors related to COVID-19 on different online platforms, leading to unwanted consequences such as failures of COVID-19 interventions and even deaths of people who chose to believe in misinformation or rumors rather than follow official guidelines from authorities and health experts [1].

In order to effectively detect rumors, most researchers chose to focus on text content on microblog platforms such as Twitter and Sina Weibo [2]. In addition to rumor detection based on analyzing text in the original post, some researchers also considered including text in comments associated with the original post into the text analysis and rumor detection process. However, the role of such associated comments in the rumor detection task has been much less studied and therefore remains less understood. To fill these gaps, we propose a new rumor detection method considering both the original post and associated comments, and conducted a series of experiments to investigate different aspects of the role such comments play in the rumor detection task.

We use the text classification models RoBERT (Recurrence over BERT) and ToBERT (Transformer over BERT) proposed by Pappagari et al. [3] to automatically extract useful semantic features from the original post and the associated comments. Our method works as follows: 1) it first preprocesses the input posts by concatenating texts in the original post and the associated comments into a single long text; 2) it truncates the long text into smaller sequential trunks and obtains features for each trunk using a fine-tuned BERT (Bidirectional Encoder Representations from Transformers) model, and finally 3) it feeds the features to either an LSTM (Long Short-Term Memory) network or a transformer layer for the rumor classification task. Our experiments on PHEME and Ma-Weibo, two public rumor detection datasets representing the two most spoken languages – English and Chinese – and two of the largest web platforms – Twitter and Sina Weibo, showed that our proposed model outperformed other state-of-the-art methods. We also conducted additional experiments on different settings of the proposed methods to study different aspects of the role comments may play in the rumor detection task, including three different ways of using comments, three different ways to segment the text into trunks, and applications of the method to different subsets of posts with different comments numbers.

Our work's main contributions can be summarized as follows.

1) We propose a new rumor detection method combining BERT and an LSTM- or Transform-based classifier, which works with automatic features extracted from concatenated-and-then-segmented texts of the original

post and the associated comments.
2) We more systematically considered how comments can be incorporated in the rumor detection task and designed experiments to investigate different aspects of their role.
3) The experimental results showed that our proposed method significantly outperformed other state-of-the-art rumor detection methods, and consistently across the two datasets we used.
4) Results of our additional experiments revealed new insights about the role comments play in rumor detection, some of which have very interesting operational implications for online platforms.

II. Related Work

Several different definitions of rumor exist in the research literature. Still, the majority of the literature defines rumors as "unverified and instrumentally relevant information statements in circulation" [4]. In this article, we follow this commonly used definition to consider rumor as an unverified piece of information at the time of posting. Such unverified information may later turn out to be true or false, or remain unverified.

The term 'rumor detection' also deserves some clarification as it is often confused with another highly related term 'rumor verification'. The goal of rumor detection is to determine whether a post is 'rumor' or 'non-rumor' based on relevant information posted by users online, while 'rumor verification' is the task of determining the veracity of a suspected rumor (true, false or unverified) [2]. In this paper, we focus on rumor detection, rather than rumor verification.

Rumor detection methods can be classified into three categories: content-based, user-based and propagation-based methods [2]. We only discuss content-based methods in this article since our method mainly focuses on the text contents of the blogs and associated comments.

Content-based rumor detection methods mainly rely on analyzing the content of the text, and they are normally designed for analysis of relatively long texts. Many researchers proposed to use machine learning for content-based rumor detection [5], [6]. Most earlier rumor detection methods are based on feature engineering, i.e., manually defined features, so they are harder to generalize. In recent years, the development of deep learning has provided new methods to address shortcomings of traditional machine learning based methods. Li et al. [7] proposed a multigraph neural network framework by associating posts containing the same high-frequency words to facilitate the feature cross-topic propagation, and captured the attribute information of the post context more flexibly. Lin et al. [8] proposed a novel rumor detection method based on a hierarchical recurrent convolutional neural network and a bidirectional GRU (Gated Recurrent Unit) network with attention mechanism, which could integrate contextual features and learn the time period information. One limitation of content-based methods is that they are more suitable for long texts: machine learning-based methods often require a sufficient long text to extract the required features for classification, and deep learning-based methods require even longer texts especially for training purposes. Therefore, such methods will have problems processing short texts on many online platforms such as microblogging websites.

Comments associated with a post can include rich semantic information about if the post contains rumor, e.g., confrimatory or disapproval stance of some commentators and challenges to the rumor in the original post. Therefore, some researchers have looked at the use of comments for rumor detection. Ma et al. [9] applied RNN (Recurrent Neural Network) for learning the hidden representations that capture the rich semantic information in both the original post and the associated comments. Lv et al. [10] considered adding comment sentiment to rumor detection as a new feature, and combined an CNN (Convolutional Neural Network) with an LSTM network for the classification task. These rumor detection methods considering comments are based mainly on models like CNNs and RNNs, which have been shown to perform worse than more recently developed pre-trained natural language models represented by Transformers and BERT [11]. Therefore, some researches have explored the use of BERT for improving the performance of rumor detection methods. Rao et al. [12] proposed a new ensemble model that adopts two level-grained attention-masked BERT (LGAM-BERT) models as the base encoders and takes comments as important auxiliary features for rumor detection. Although some previous work has considered comments as part of the input, we did not see any past studies that investigated different aspects of the role comments play on rumor detection. For example, they did not discuss whether how the improvement correlates with the number of comments and different ways of combining the original post and the associated comments.

III. Proposed Rumor Detection Method with Comments

A. Problem Definition

Rumor detection in online platforms supporting user comments can be formulated as a binary classification problem, which is defined as determining if a particular user-generated 'event' (i.e., a thread of posts) is rumor or not [2]. The detector tries to identify rumors in a set of events $E = \{e_1, e_2, \ldots, e_n\}$, where $e_i = \{m, r_1, r_2, \ldots\}$ represents a single event, i.e., a number of online posts including the original post m and a number of follow-up comments of other users represented by $r_1, r_2, \ldots$. The detector produces a set of prediction labels, each belongs to a set of label categories $L = \{l_1, l_2\}$, where l_1 represents rumor and l_2 represents non-rumor.

B. Overall Architecture

Our proposed rumor detection method works as follows. It first preprocesses the input event by concatenating the original post and the associated comments into a single piece of long text. Then, it segments the long text into smaller sequential trunks, and applies a fine-tuned BERT model to automatically obtain semantic features from each trunk. Features from all trunks are combined and fed to either an LSTM network or a transformer layer for the rumor classification task. For the

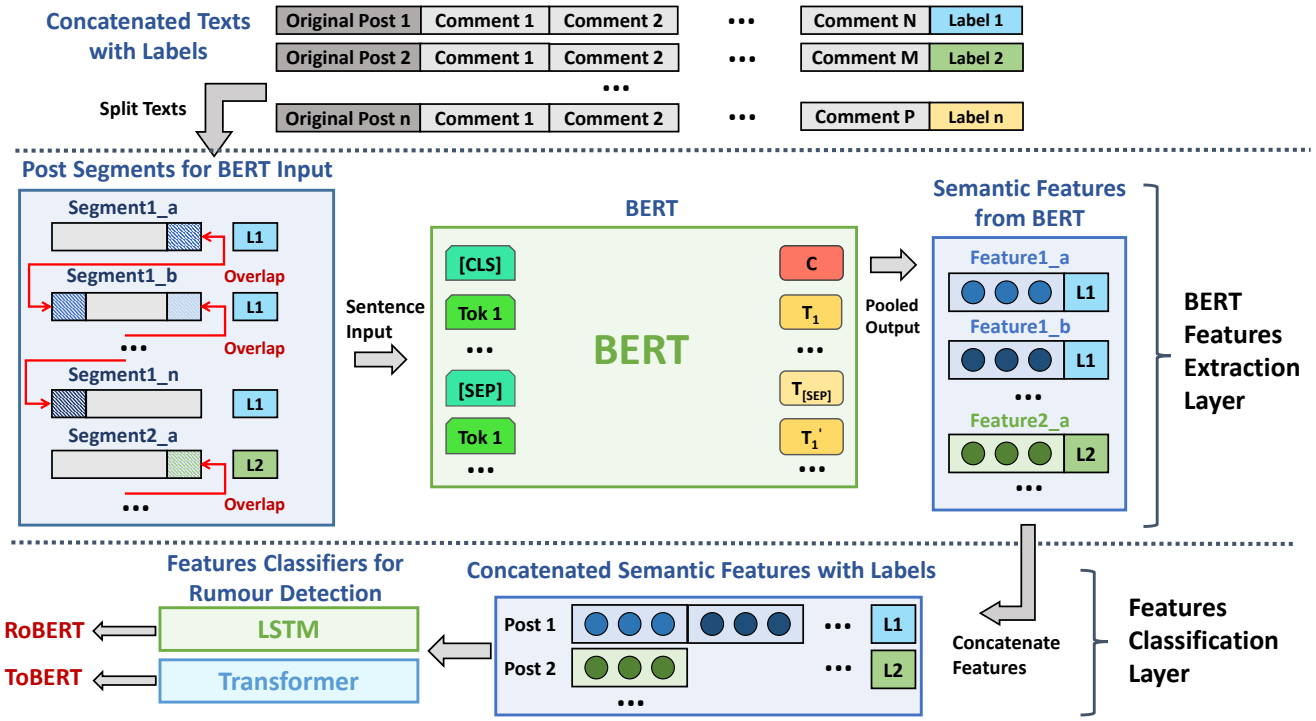

Fig. 1. The overall structure of our proposed rumor detection model

LSTM- and transformer-based classifiers, we use RoBERT and ToBERT [3], respectively.

As shown in the Figure 1, our model is mainly composed of two parts: a **BERT-based feature extraction layer** and a **classification layer**. We explain these two layers with greater details below.

C. BERT-based Feature Extraction Layer

Pre-trained language models featured by BERT have achieved leading results in many NLP problems [11]. Based on the transformer network architecture, BERT uses self-attention, feed-forward layers, residual connections and layer normalization as the main building blocks, and can learn a wealth of semantic knowledge after large-scale text training, and its knowledge can be transferred to many downstream NLP tasks.

Our proposed rumor detection method can fully take the performance advantages of the BERT model in mainstream social media texts. In our task, we obtain the two types of feature representations we need from the BERT classification model. They are the **pooled output** of the last transformer block and the posterior probability after the fully connected layer and the softmax layer.

Although BERT is very suitable for processing relatively short text sequences, the Transformer structure has a limitation on the length of the input text, which also limits the applicability of the BERT model for long text classification. Since the input of our rumor detector is a merged long text of the original post and the associated comments, in order to better extract the feature information between the comment and the text, we choose to divide the text reasonably and split it into shorter trunks more suitable for BERT processing. These trunks can be segmented following the natural boundary between the original post and comments, and between comments, but can also be done by applying a fixed-length trunk size with or without overlaps, where the overlaps may help capture some correlation between consecutive posts. These short text trunks form our final data for BERT processing. We used a short text sequence set to fine-tune and train the pre-trained BERT model, then applied the fine-tuned BERT model to extract the features of these short text trunks, which are then sent to the classification layer for classification.

D. Classification Layer

The classification layer first splices the features of the short text in the order of the original long text. It should be noted that since the overlap between the texts is considered when the text is segmented, the final long text feature vector representation is also naturally included the contextual semantic relationship between text sequences. These feature vectors are finally sent to the classification network for training and predicting. In the final classification network, we can either choose LSTM or Transformer, which corresponds to the RoBERT and ToBERT models respectively.

For RoBERT, since our network's input is already in the form of word vectors, no additional embedding layer is

needed. We directly fed the spliced features into a small LSTM network, and use two fully connected layers. Finally a softmax activation function is applied for the final classification prediction.

For ToBERT, the transformer structure itself can capture the word vector connection between long distances in the sequence, in order to take full advantage of its capability, we add a transformer layer to form ToBERT on the basis of the original RoBERT, and introduced self-attention mechanism in the transformer block in order to better capture the connection between word vectors.

IV. Experiments

A. Datasets Used

Since our work focuses on rumor detection rather than rumor verification, we choose to use two widely used rumor detection datasets, PHEME [13] and Ma-Weibo [9]. The PHEME dataset was collected in 2016, and it contains a collection of tweets from Twitter. The Ma-Weibo dataset was also collected in 2016, and it contains a number of posts collected from the Sina Weibo Community Management Center, an online portal where users can report suspected rumor posts to the platform. Both datasets contain original posts, associated comments, and some other meta-information.

Table I shows the statistics of our datasets. The proportion distribution of text lengths for each dataset is shown in Figure 2. From the distribution graph of the data length, we can see that compared to the English PHEME dataset, the lengths of the Chinese Ma-Weibo dataset's texts are much longer. Obviously, due to the limitation of the length of the input text, the traditional BERT text classification model is more difficult to process the long text in our dataset, so we will use the ideas proposed in this article to train and verify these datasets.

TABLE I
Dataset statistics of the PHEME and Ma-Weibo datasets

Statistics	PHEME	Ma-Weibo
Events #	5802	4664
Rumors #	1972	2313
Non-rumors #	3830	2351
Users #	49345	2746818
Microblogs #	103212	3805656
Median # of Comments / Event	13	117
Max # of Comments / Event	340	44763
Median Length of Concatenated Text	160	2002
Max Length of Concatenated Text	4561	736005

B. Datasets Preprocessing and Reorganizing

We first performed data preprocessing on the original data in the public dataset. Since our model aims to explore the impact of comment semantics on rumor detection, we concatenated the original text of the post with all accompanying comments to form an ultra-long text. The semantic information of this long text is richer than that of the original blog post. We cleaned the original comment data and deleted some meaningless texts (such as 'Repost Weibo') to reduce the interference of these text data on the accuracy of rumor detection.

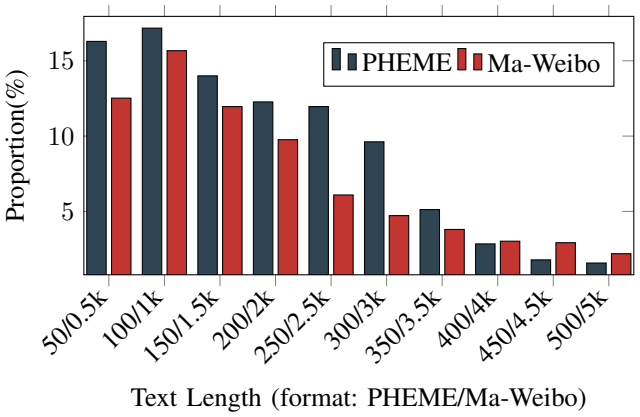

Fig. 2. Some statistics of the PHEME and Ma-Weibo datasets

In the BERT-based feature extraction layer, we divide the long text into shorter texts suitable for BERT model training through reasonable segmentation methods. We considered three segmentation methods discussed below.

The first segmentation method is "**Natural Segmentation**". As its name suggests, this method splits long texts according to their natural boundaries. Specifically, we divide the long text into "Original Post", "Comment 1", etc. This method produces non-overlapping trunks by definition.

The other two segmentation methods we adopted divide the long text into shorter chunks with a fixed number of words. Based on whether there is an overlap between each text segment, we used two separate methods, "**Fixed-length without Overlap**" and "**Fixed-length with Overlap**". In the latter method, we chose to have an overlap of a number of words between adjacent trunks, in order to preserve semantic connections between adjacent trunks.

The "Fixed-length with Overlap" method is our default choice when splitting long texts in the following experiments, and it was also the segmentation method described in Section III and in Figure 1. All these mentioned methods are discussed in Section V-B to compare the differences in performance.

C. Experimental Setup and Details

The running environment used in this experiment is as follows: Inter(R) Xeon(R) CPU E7-4830 v3 @ 3.10GHz, with the operating system as Ubuntu 18.04.5 LTS and the GPU as Tesla M40. We chose Python 3.7.10 as our main programming language, with PyTorch 1.6.0 and TensorFlow 2.0.0 as our deep learning structure.

In the selection of the pre-trained BERT model, we chose the basic bert-base-uncased pre-trained model [11] for the English dataset, which is the pre-trained model on English language using a Masked Language Modeling (MLM) objective. For the Chinese dataset, we chose the chinese-bert-wwm-ext

model [14] as the pre-trained model, which is the Chinese pre-trained BERT with Whole Word Masking (WWM). Using a pre-trained model can significantly speed up the convergence process during finetuning. Furthermore, since the pre-trained model is obtained from massive textual data, it can potentially achieve a better generalization effect. In the process of fine-tuning and training the BERT model, we chose AdamW as our optimizer. The initial learning rate was set to 2e-5, and epsilon for Adam was set to 1e-8.

As mentioned before, we chose either LSTM or transformer as our final classifier. We used the Adam optimizer in the LSTM network and the transformer to minimize the sparse categorical cross-entropy loss. The initial learning rate was set to 0.001. We also used 'ReduceLROnPlateau' method to accelerate the training and update the learning rate, learning rate was reduced by a factor of 0.95 if validation loss did not decrease for 3-epochs. In all tests, we used accuracy, precison, recall and F1-score as the evaluation metrics, and chose to use the best model on the validation set to predict the results of the test set.

D. Other Methods Compared

We compared our proposed model with the following baseline methods:

- **DTC model**: A decision tree classifier model uses four categories of hand-crafted features from posts to detect rumors [6].
- **Naive Bayes**: Based on naive Bayes theorem, the naive Bayes classifier learns features extracted from posts [15].
- **SVM-BOW**: A baseline SVM model represents text contents of the posts by using bag-of-words and n-gram features [16].
- **Random Forest**: A random forest classifier uses text features to aggregate many binary decision trees for rumor classification [17].
- **CNN**: A CNN-based model for obtaining the representation of each post with multiple filter sizes [18].
- **BiLSTM**: A bidirectional RNN-based tweet model with conditional LSTM encoding that considers the bidirectional context between the target and the post [19].
- **GRU-RNN**: A RNN-based model uses TF-IDF method to calculate the text representation of each time period, and uses the double-layer GRU model for training, with two hidden layers for capturing higher-level feature interactions [9].
- **RCNN-FAN**: A recurrent convolution neural network with an attention mechanism which contains the event feature vector to learn the time period information [8].
- **Ma-RvNN**: A tree-structured recursive neural network which learns discriminative features from microblog posts by following their non-sequential propagation structure [20].
- **STANKER**: An ensemble model which adopts two LGAM-BERT models as base encoders and extracted microblog comments features to perform rumor detection [12].
- **BERT**: A rumor detection method simplified from our proposed method with the fine-tuned BERT model alone.

Table II shows the comparison results, with accuracy, precision, recall and F1-score as our metrics. It should be noted that due to absence of the original source code and the lack of full implementation details, we could not reproduce the results of "Sentimental CNN-LSTM" [10] and "PostCom2DR" [21], two other state-of-the-art rumor detection methods leveraging both the original post and associated comments. Comparing the performance figures reported in [10], [21] with ours in Table II, we believe that our proposed model could also outperform these two methods.

V. Results

A. Overall Results

From Table II, we can conclude that almost all the deep learning models had better performance than models based on more traditional machine learning methods. The BERT model has achieved the accuracy of 94.832% and 97.093% on two datasets, which has been the best results in all the baseline models. Moreover, Table II shows our proposed method (either the one based on RoBERT or ToBERT) achieved the best detection performance compared with other methods on both datasets.

Specifically, ToBERT obtained the highest accuracy (96.287% and 98.128%) and F1-score (94.670% and 98.093%) on both datasets. Our methods performed better on the Ma-Weibo dataset than on the PHEME dataset. This may be explained by the fact that the Ma-Weibo dataset is larger than the PHEME dataset in terms of the number of comments and overall length so the machine learning model can more effectively trained.

B. The Effect of Including Comments

In order to verify the effect of our application of comment semantic information to rumor detection, we have performed a comparative experiment. For the two datasets, we divided the data into three categories: the original post only, the associated comments only, and both the original post and the associated comments. For this work, we chose RoBERT and ToBERT as the experimental models. The remaining experimental conditions are not changed so as to verify the effectiveness of comment semantic information for rumor detection. The experimental results are shown in Table III.

From Table III, we can observe that the best performance on both datasets was achieved when both the original post and associated comments were used, which gives direct evidence on the effectiveness of our proposed method – adding comments can indeed help improve the performance of the rumor detector.

At the same time, we also noticed that compared with using comments only, the accuracy of rumor detection using the original post only is slightly higher. We believe that this result could be explained by the following two reasons. Firstly, the rumor labels in both datasets were annotated in terms of the main post, rather than the associated comments. Secondly, this

TABLE II
RUMOR DETECTION RESULTS ON THE PHEME AND MA-WEIBO DATASETS

Methods	PHEME Dataset				Ma-Weibo Dataset			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
DTC [6]	0.74505	0.72004	0.72004	0.72004	0.80064	0.84006	0.79640	0.81764
Naive Bayes [15]	0.76744	0.78823	0.76744	0.77195	0.89674	0.88611	0.87484	0.88044
SVM-BOW [16]	0.77347	0.75133	0.75436	0.75276	0.86602	0.88065	0.86602	0.87327
Random Forest [17]	0.80448	0.80946	0.74996	0.76576	0.82208	0.86109	0.81801	0.83899
CNN [18]	0.83444	0.83180	0.83444	0.83312	0.80246	0.80645	0.80246	0.80445
BiLSTM [19]	0.86908	0.86835	0.86908	0.86871	0.85357	0.85723	0.86357	0.86039
GRU-RNN [9]	0.87696	0.88558	0.86696	0.87617	0.90710	0.92685	0.89710	0.91183
RCNN-FAN [8]	0.91515	0.90258	0.91515	0.90882	0.95491	0.95533	0.95491	0.95512
Ma-RvNN [20]	0.94120	0.94300	0.92140	0.93920	0.94810	0.94840	0.94950	0.94810
STANKER [12]	0.95120	0.95030	0.93785	0.94056	0.97470	0.96755	0.97460	0.97106
BERT	0.94832	0.92923	0.93730	0.93325	0.97107	0.97093	0.95143	0.96108
RoBERT	0.96301	0.94521	**0.94694**	0.94607	0.98075	**0.97252**	0.98785	0.98011
ToBERT	**0.96287**	**0.95118**	0.94283	**0.94670**	**0.98128**	0.97185	**0.99022**	**0.98093**

TABLE III
RUMOR DETECTION RESULTS WITH DIFFERENT SETTINGS OF USING THE ORIGINAL POST AND THE ASSOCIATED COMMENTS

	RoBERT				ToBERT			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
Results on the PHEME dataset								
Comments Only	0.93395	0.91711	0.88645	0.90152	0.91926	0.88470	0.88831	0.88650
Original Post Only	0.95483	0.93316	0.93030	0.93173	0.95464	0.92134	0.94179	0.93145
Original Post & Comments	**0.96301**	**0.94521**	**0.94694**	**0.94607**	**0.96387**	**0.95118**	**0.94283**	**0.94670**
Results on the Ma-Weibo dataset								
Comments Only	0.95561	0.95722	0.95389	0.95555	0.95294	0.94845	0.95819	0.95330
Original Post Only	0.97594	0.97030	0.98492	0.97756	0.97861	0.96757	0.98815	0.97814
Original Post & Comments	**0.98075**	**0.97252**	**0.98785**	**0.98011**	**0.98128**	**0.97185**	**0.99022**	**0.98033**

result may be explained by the fact that the original text of the post contains more direct semantic information about the rumor than the associated comments.

C. The Effect of the Number of Comments

We also further classified the two datasets according to the number of comments, and tested them under the RoBERT and ToBERT models. In order to ensure that the experiment is not affected by the size of the dataset, we divided each of the datasets into three sub-groups according to the distributions of the number of comments. We try to keep the number of events in each group as close as possible. The English dataset is divided into three groups: "0-7 comments", "7-18 comments" and "19 comments or more", each name indicates the range of comment number in this group. Similarly, the Chinese dataset is also divided into "0-70 comments", "71-224 comments" and "225 comments or more". Table IV shows the results of the experiment.

From the data in Table IV, we can conclude that with the increase in the number of comments, the accuracy of rumor detection is also steadily improving, which also verifies the effectiveness of comment semantic information for rumor detection. More comments mean that there will be more semantic information for the BERT model and the follow-up classification to extract, which also brings an improvement in detection accuracy.

D. Impact of the Segmentation Method

We also discussed the impact of the long text segmentation method on the performance of rumor detection. Here, we discussed two different segmentation categories with three methods previously mentioned:"Natural Segmentation", "Fixed-length without Overlap" and "Fixed-length with Overlap". For the last method, we used the trunk size of 200 words, and three different settings of adjacent trunk overlaps: 25, 50 and 75 words. We used RoBERT and ToBERT on the both dataset to test these segmentation methods. The detection results are shown in Table V.

From the results, it can be concluded that "Fixed-length with Overlap" achieved the best performance among the three methods, while the "Natural Segmentation" had the worst performance, which indicated that both "fixed length" and "overlap" process could boost the detection performance. Besides, we also found that, instead of using an overlap size of 25 or 75, fixed-length with an overlap size of 50 had the highest accuracy, which indicate that an optimum overlap size exists to achieve the best balance between information in each post and semantic correlation between adjacent posts. During the experiments, we also noticed both "Fixed-length with Overlap" and "Fixed-length" were also much higher in time efficiency than "Natural Segmentation". We believe that there are three main reasons for such detection results:

TABLE IV
RUMOR DETECTION RESULTS WITH DIFFERENT NUMBER OF COMMENTS

# Comments	RoBERT Accuracy	RoBERT Precision	RoBERT Recall	RoBERT F1-score	ToBERT Accuracy	ToBERT Precision	ToBERT Recall	ToBERT F1-score
Results on the PHEME dataset								
0-7	0.90821	0.87078	0.87102	0.87090	0.91484	0.88180	0.87703	0.87941
8-18	0.93988	0.89173	0.91679	0.90409	0.94125	0.89748	0.90765	0.90254
19-	**0.95103**	**0.92780**	**0.93549**	**0.93163**	**0.95232**	**0.92547**	**0.93108**	**0.92827**
Results on the Ma-Weibo dataset								
0-70	0.91720	0.88554	0.93046	0.90744	0.90720	0.91642	0.90961	0.91300
71-224	0.95200	0.94915	0.94915	0.94915	0.96000	0.96552	0.94915	0.95726
225-	**0.97100**	**0.97203**	**0.97404**	**0.97303**	**0.97426**	**0.96933**	**0.97208**	**0.97070**

TABLE V
RUMOR DETECTION RESULTS WITH DIFFERENT SEGMENTATION METHODS

Segmentation Methods	RoBERT Accuracy	RoBERT Precision	RoBERT Recall	RoBERT F1-score	ToBERT Accuracy	ToBERT Precision	ToBERT Recall	ToBERT F1-score
Results on the PHEME dataset								
Natural Segmentaion	0.94538	0.90665	0.94088	0.92345	0.94366	0.90821	0.93459	0.92121
Fixed-length (200) without Overlap	0.95312	0.92831	0.93439	0.93134	0.95312	0.92930	0.93306	0.93112
Fixed-length (200) with Overlap (25)	0.96014	0.94272	0.94260	0.94266	0.96018	0.94302	0.94156	0.94229
Fixed-length (200) with Overlap (50)	**0.96301**	**0.94521**	**0.94694**	**0.94607**	**0.96387**	**0.95118**	**0.94283**	**0.94670**
Fixed-length (200) with Overlap (75)	0.95922	0.94065	0.94088	0.94075	0.95978	0.94080	0.94120	0.94092
Results on the Ma-Weibo dataset								
Natural Segmentation	0.95882	0.94072	0.97660	0.95832	0.96203	0.94649	0.97782	0.96190
Fixed-length without Overlap	0.97465	0.96598	0.97974	0.97485	0.97063	0.96829	0.97949	0.97408
Fixed-length (200) with Overlap (25)	0.97814	0.97026	0.98420	0.97652	0.97802	0.97056	0.98214	0.97780
Fixed-length (200) with Overlap (50)	**0.98075**	**0.97252**	**0.98785**	**0.98011**	**0.98128**	**0.97185**	**0.99022**	**0.98033**
Fixed-length (200) with Overlap (75)	0.97786	0.96958	0.98026	0.97574	0.97652	0.97004	0.97926	0.97686

- First, the "Natural Segmentation" method divides the text into shorter and smaller sequences, so that BERT could only extract a small number of features for learning, resulting in lower detection accuracy.
- Second, segmentation method without overlap could not retain the semantic connection between adjacent texts. However, this semantic connection should not be overlooked since comments attached to the post may also reply to comments themselves.
- Third, compared to the "Fixed-length" and "Fixed-length with Overlap" methods, the number of the segments generated by "Natural Segmentation" is roughly 10 times that of the former, which also greatly increases the space and time burden of training.

VI. FURTHER DISCUSSIONS

There can be multiple possible explanations to the fact that adding more user comments of an original post can improve the performance of the rumor detector. One likely reason is that any rumor can attract commentators who would challenge its veracity and therefore leave comments to debunk the rumor, which then can provide valuable information for a rumor detector to better distinguish rumors from non-rumors. We manually sampled some rumor posts that were correctly detected only after adding comments, and found that most of these comments contained a lot of questioning text, much more than those associated with a non-rumor post. In other words, online users who are willing to actively "police" online rumors based on their knowledge or judgment become a very useful source of crowd intelligence. This implies that online platforms can try to encourage active online discussions without over-policing them and participants of such discussions, and can provide more intelligent tools (such as tools for detecting rumors and other harmful content) by leveraging the active discussions between online users. This will allow identification of online users who regularly create and/or spread rumors and also those who often helped debunk rumors, so different interventions can be applied to them to foster a more helathy online environment without over-policing it. Engaging online users who are willing to help can also help achieve early detection of rumors before they spread more widely, especially for rumors that are too new to be detected by fully automated rumor detectors. There are many ways such engagement can be implemented, e.g., identifying online users whose comments helped debunk confirmed rumors and pushing potential rumors to them for comments to help improve the decision making of an automated rumor detector.

The finding on fixed-length segmentation with overlap outperformed natural segmentation was a surprise to us. We predicted the opposite result because natural segmentation clearly can better maintain the semantic boundary between

different posts (including the original post and comments). This result gives yet another piece of evidence that simple heuristics may not always work for AI tasks so we need to explore more possible settings and parameters.

As all research work, our work has a number of limitations. First, the results we obtained in this paper should be further validated on data collected from more platforms, and probably at a larger scale. Second, like many other researchers did, we followed a relatively simpler model of rumor detection, where each thread is labeled for a single rumor. In more real-world settings, the situation can be more complicated, e.g., each post can include multiple rumors and a comment can introduce one or more new rumors. Third, our finding on "the more comments, the better detection performance" also implies that early detection of rumors remain a future research challenge, since in many applications we would like to limit the spread of rumor as much as possible so that the number of comments available is often limited. Finally but not the least, we will also explore the more complicated research problems of multi-modal (i.e., based on textual and non-textual data such as digital images and videos) or cross-platform rumor detection with user comments as auxiliary features.

VII. Conclusions

This paper shows that user comments are a very useful source of information for rumor detection. Leveraging a richer set of features extracted from the concatenated longer text including both the original post and associated comments, a new rumor detection method is proposed, whose performance was shown significantly better than other state-of-the-art methods. Further experiments with the new rumor detection method also led to interesting new findings and further evidence on the role of comments in rumor detection, including the fact that having access to more comments can help improve the performance of the rumor detector. Findings reported in this paper indicate that online commentators can be a good source of crowd intelligence for detecting rumors.

References

[1] M. S. Islam, T. Sarkar, S. H. Khan, A.-H. M. Kamal, S. M. M. Hasan, A. Kabir, D. Yeasmin, M. A. Islam, K. I. A. Chowdhury, K. S. Anwar, A. A. Chughtai, and H. Seale, "COVID-19-related infodemic and its impact on public health: A global social media analysis," *The American Journal of Tropical Medicine and Hygiene*, vol. 103, no. 4, pp. 1621–1629, 2020.

[2] Q. Li, Q. Zhang, L. Si, and Y. Liu, "Rumor detection on social media: Datasets, methods and opportunities," in *Proceedings of the 2nd Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*. ACL, 2019, pp. 66–75.

[3] R. Pappagari, P. Zelasko, J. Villalba, Y. Carmiel, and N. Dehak, "Hierarchical transformers for long document classification," in *Proceedings of the 2019 IEEE Automatic Speech Recognition and Understanding Workshop*. IEEE, 2019, pp. 838–844.

[4] N. Difonzo and P. Bordia, "Rumor, gossip and urban legends," *Diogenes*, vol. 54, pp. 19–35, 2007.

[5] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea, "Automatic detection of fake news," pp. 3391–3401, 2017. [Online]. Available: https://aclanthology.org/C18-1287

[6] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on Twitter," in *Proceedings of the 20th International Conference on World Wide Web*. ACM, 2011, pp. 675–684.

[7] C. Li, H. Peng, J. Li, L. Sun, L. Lyu, L. Wang, P. S. Yu, and L. He, "Joint stance and rumor detection in hierarchical heterogeneous graph," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[8] X. Lin, X. Liao, T. Xu, W. Pian, and K.-F. Wong, "Rumor detection with hierarchical recurrent convolutional neural network," in *Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9–14, 2019, Proceedings, Part II*. Springer, 2019, pp. 338–348.

[9] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K.-F. Wong, and M. Cha, "Detecting rumors from microblogs with recurrent neural networks," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI-16)*. IJCAI, 2016, pp. 3818–3824. [Online]. Available: https://www.ijcai.org/Proceedings/16/Papers/537.pdf

[10] S. Lv, H. Zhang, H. He, and B. Chen, "Microblog rumor detection based on comment sentiment and CNN-LSTM," in *Artificial Intelligence in China*. Springer, 2020, pp. 148–156.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," pp. 4171–4186, 2019.

[12] D. Rao, X. Miao, Z. Jiang, and R. Li, "STANKER: Stacking network based on level-grained attention-masked BERT for rumor detection on social media," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 3347–3363. [Online]. Available: https://aclanthology.org/2021.emnlp-main.269

[13] A. Zubiaga, G. Wong Sak Hoi, M. Liakata, and R. Procter, "PHEME dataset of rumours and non-rumours," 2016. [Online]. Available: https://doi.org/10.6084/m9.figshare.4010619.v1

[14] Y. Cui, W. Che, T. Liu, B. Qin, S. Wang, and G. Hu, "Revisiting pre-trained models for Chinese natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*. ACL, 2020, pp. 657–668.

[15] R. Dayani, N. Chhabra, T. Kadian, and R. Kaushal, "Rumor detection in Twitter: An analysis in retrospect," in *Proceedings of the 2015 IEEE International Conference on Advanced Networks and Telecommuncations Systems*. IEEE, 2015.

[16] J. Ma, W. Gao, and K.-F. Wong, "Rumor detection on Twitter with tree-structured recursive neural networks," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, vol. 1. ACL, 2018, pp. 1980–1989.

[17] H. Bingol and B. Alatas, "Rumor detection in social media using machine learning methods," in *Proceedings of the 2019 1st International Informatics and Software Engineering Conference*. IEEE, 2019.

[18] Y.-C. Chen, Z.-Y. Liu, and H.-Y. Kao, "IKM at SemEval-2017 task 8: Convolutional neural networks for stance detection and rumor verification," in *Proceedings of the 11th International Workshop on Semantic Evaluation*. ACL, 2017, pp. 465–469.

[19] I. Augenstein, T. Rocktäschel, A. Vlachos, and K. Bontcheva, "Stance detection with bidirectional conditional encoding," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. ACL, 2016, pp. 876–885.

[20] J. Ma, W. Gao, S. Joty, and K.-F. Wong, "An attention-based rumor detection model with tree-structured recursive neural networks," *ACM Transactions on Intelligent Systems and Technology*, vol. 11, no. 4, pp. 42:1–42:28, 2020.

[21] Y. Yang, Y. Wang, L. Wang, and J. Meng, "PostCom2DR: Utilizing information from post and comments to detect rumors," *Expert Systems with Applications*, vol. 189, pp. 116 071:1–116 071:13, 2022.