A Stochastic Attention CNN Model for Rumor Stance Classification

NA BAI, ZHIXIAO WANG, AND FANRONG MENG

School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China
Mine Digitization Engineering Research Center of Ministry of Education of the People’s Republic of China, Xuzhou 221116, China
Corresponding authors: Zhixiao Wang (zhxwang@cumt.edu.cn) and Fanrong Meng (mengfr@cumt.edu.cn)

This work was supported in part by the National Key Research and Development Program under Grant 2016YFC060908, and in part by the National Natural Science Foundation of China under Grant 61876186.

ABSTRACT Rumor stance classification is a task for identifying different stances about specific social media posts, and it is considered as an important step to prevent rumors from spreading. However, most previous studies for the rumor stance classification, which focus on content features of posts, ignore the textual information granularity and reading (or writing) habits of the public. In social networks, people may have different stances towards unverified posts with different reading habits. Based on this observation, we propose a Stochastic Attention Convolutional Neural Net (SACNN) under different textual information granularity to catch different habits of the public for the rumor stance classification task. Specifically, being treated as view-windows of readers, convolutional kernels in a SACNN contain trainable convolutional kernels and stochastic untrainable convolutional kernels, and different sizes of the stochastic convolutional kernels are used to simulate casual online reading habits and to extract course-grained features such as phrase features. After the convolutional layer, fine-grained features such as keywords can be extracted by a pooling layer. The experiments show that the average accuracy and F1 score of our proposed model are respectively 0.26% and 0.46% higher than the state-of-the-art results on the rumor stance classification task on PHEME dataset.

INDEX TERMS Convolutional neural net, stochastic attention mechanism, information granularity, stochastic attention convolutional neural net.

I. INTRODUCTION

The popularity of social network makes rumors spread quickly, and the unverified rumors may cause anxiety and panic [1], [2]. Usually, it is difficult for the public to distinguish verified information from rumors [3]. Therefore, the rumor stance classification is considered as a crucial component in debunking false rumors. Specifically, the rumor stance classification task aims to classify replies of a post to supporting, denying, querying and commenting. In order to efficiently finish this task, researchers set out to develop a rumor stance classification system that identifies the attitude of Twitter users towards the truthfulness [4]. An effective rumor stance classification system can be useful for limiting the diffusion of rumor information which brings anxiety and panic to individuals, communities and society [5].

In the past few years, there is a growing interest in studying rumor stance classification task in social media [6]–[9], but conventional methods do not consider the temporal correlation between replies. In order to extract textual features and temporal correlation between replies [10]–[13], Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) and the attention mechanism are proposed to finish this classification task [14], [15]. However, these models neglect the reading habits of the public. Most people read posts quickly and express their stances according to the key information they pay attention to or the keywords that they inadvertently see. This reading habit may lead that different people could have different or even opposite stances on the same post. Moreover, posts on different stances may only differ in some keywords.

Our work is motivated by the following observations: Firstly, different readers have different reading habits, and based on different reading habits and occupations, readers may focus on different key information. Therefore, for a
broadcasted post, any word or phrase might become the key information for different readers to express their stances, and the traditional attention mechanism cannot completely reflect such habits. Therefore, we propose a stochastic attention mechanism to simulate this observation. For a traditional CNN, its classification can be expressed as a process of constructing a conditional probability $P(y|x)$, and this process is equivalent to making a smooth nonlinear fitting and to extract features such as key phases or keywords for classification. However, if some readers have different reading and writing habits from the public’s habits, using the traditional mapping $P(y|x)$ may result in misclassification. The stochastic attention mechanism gives stochastic initialized weights to different keywords, and almost any keyword is likely to be given a high activation probability based on the stochastic initialized weights, and the keywords that are classification related can be retained by the full connection layer. Secondly, readers may express their stances about a post based on a key phrase or a single keyword, and therefore, in order to capture the key information, the information transmission process in the proposed model should be a textual information graining process from coarse to fine granularity. Therefore, we use a feature extraction process of the CNN to simulate the information graining process, which uses a convolutional layer to extract phrase features and uses a pooling layer to extract keywords. Lastly, previous work utilizes word2vec to realize word embedding process of texts, but each local context window in the word2vec model is trained independently without using global information. In order to make full use of the global statistical information of corpus, we use GloVe for word embedding [16].

Based on the above observations, we propose a Stochastic Attention Convolutional Neural Net (SACNN), which contains a CNN model and a stochastic attention mechanism. The main contributions of our paper are the follows:

Our main contributions can be summarized as follows:

1) We propose a stochastic attention mechanism to simulate different reading habits. The stochastic attention mechanism contains stochastic convolutional kernels and a stochastic pooling layer. The stochastic convolutional kernels simulate the stochastic attention for phrases, and the stochastic pooling operation simulates the stochastic attention for keywords.

2) The proposed stochastic attention mechanism can simulate different readers’ reading habits by selecting the keywords related to classification. Different readers may use different keywords, and the stochastic attention mechanism is able to give stochastic initialized weights to different keywords. Therefore, based on this random convolution kernel, almost any keyword is likely to be given a high activation probability, and then the model can filter out the keywords that are not classification related through the full connection layer.

3) We propose a SACNN to simulate a graining process of text information, and the proposed SACNN uses a convolutional layer to extract coarse-grained features such as phrase features and uses a pooling layer to extract fine-grained features such as keywords. Moreover, we use GloVe for word embedding. The experiments verify the effectiveness of the proposed stochastic attention mechanism.

4) In this paper, convolutional kernels are used to extract key information. As different readers tend to use key information of different lengths, this paper explores the influence of stochastic convolutional sizes on rumor stance classification results through experiments. Furthermore, this paper discusses the influence of the ratio between stochastic pooling and max pooling on the rumor detection task.

The remainder of this paper is organized as follows: Section 2 introduces the related works about the rumor stance classification task, including stance classification based on single tweet and stance classification based on sequence. Section 3 details the proposed model. In Section 4, experimental results prove the feasibilities of the proposed model. The last section gives the conclusion and outlook.

II. RELATED WORK

In the past few years, lots of research about rumor stance classification has been done, and the research can be divided into two stages: in the first stage, an independent post is used for text analysis and replies of the post are neglected. In the second stage, some sequential models are used to learn the conversational structure of both the source post and its replies to finish the rumor stance classification task:

A. STANCE CLASSIFICATION BASED ON SINGLE TWEET

The stance classification task is proposed by Mendoza (2010) et al. [6], 14 posts are used in their experiments, and half of data is marked as true and the other half is marked as false. Their experiments are finished by manual annotation and classification. Subsequently, Qazvinian et al. [7] try to solve the textual classification problem by using machine learning algorithms, which can be treated as the first work for the automatic stance classification in blogs [7]. Hamidian, Diab [8], and Zeng et al. [9] apply supervised learning algorithms for the stance classification task, such as like J48, logistic regression, naive Bayes, random forest, and so on [8], [9].

B. STANCE CLASSIFICATION BASED ON SEQUENCE

When the Convolutional Neural Network is introduced into the rumor stance classification task, it does not consider the sequential relation between the replies. Actually, CNN processes the text information in a parallel way. By introducing self attention mechanism, CNN can achieve satisfactory results for the rumor stance classification [17]–[26]. In addition, some scholars consider the sequential relation between replies. According to the structural relation between the source and the replies, rumor detection can be processed in the form of tree, or in the form of linked list [27]–[32].
Some scholars combine the rumor position classification task with the rumor veracity classification task by using the idea of multi-task learning [25], [33]. Recently, sequential stance classification attracts more and more interest. Elena Kochkina et al. win the SemEval 2017 RumorEval competition by using sequential stance classification algorithms (SemEval 2017 Task 8, Subtask A) [10]. They propose a LSTM-based sequential model, which models the conversational structure of posts to addresses the task of rumor stance classification. Another work that has utilized sequential structure is done by Arkaitz Zubiaza et al. They propose an algorithm that makes use of the sequence in tree-structured conversation threads in Twitter [11]. To incorporate temporal ordering of posts, Michal Lukasik et al. use Hawkes Processes and Gaussian Processes for modelling information diffusion for stance classification [12], [13]. Compared with the non-sequential methods, significant improvements have been achieved in their experiments. Chen et al. [14] utilize a Convolutional Neural Network (CNN) for short text classification by using multiple filters for SemEval-2017 Task 8 [14]. It is found that the convolutional windows and word embedding methods obtain an improvement on the experimental results. However, a single convolutional operation only extracts the features of a single post, which makes the final classification results unsatisfactory.

C. RELATED TERMINOLOGIES STOCHASTIC ATTENTION MECHANISM

Using stochastic convolutional kernels and stochastic pooling to simulate the reading and writing habits of different readers. stochastic convolutional kernel: A kind of convolutional kernel which is untrainable, the convolutional kernels can be initialized randomly. stochastic pooling: Different from max pooling, stochastic pooling randomly select a unit as the activation value from its pooling kernel.

According to the viewpoint of this paper, these models neglect reading habits of the public. Most people read posts quickly and express their stances according to the key information they pay attention to or the keywords that they inadvertently see. This habit may lead that different people could have different or even opposite stances on the same post. Moreover, comments from different stances may only differ in some keywords. Therefore, this paper proposes a Stochastic Attention Convolutional Neural Net (SACNN) to alleviate this question.

III. THE PROPOSED MODEL

In this paper, our proposed SACNN tries to classify stances from the viewpoints of the stochastic attention mechanism and a graining process of text. CNNs have natural advantages in representing granularity information in text classification: The convolutional kernels can be treated as view-windows of readers, and different sizes of kernels mean different visual fields. CNNs extract phrase features based on convolutional kernels, and these features can be regarded as coarse-grained features. After convolutional layers, fine-grained features such as keywords can be obtained by pooling layers. Therefore, the convolution and pooling operations can be treated as a graining process of text information. The conventional attention mechanism introduces sub neural nets into the CNN in dimensions of channels and kernels, and then assigns different weights to channels (or kernels) in training process. Although this attention mechanism is effective in image processing, it brings a risk of over-fitting for text data. In order to simulate the reading habits, this paper proposes a stochastic attention mechanism to the CNN model. According to the habits of online reading, people always read posts rapidly and always have different concerns for the same event, therefore, different readers may have different or even opposite views on the same post.

A. STOCHASTIC ATTENTION MECHANISM

Attention mechanism plays an important role in sequential learning tasks [26], [27]. For example, people may support rumors at beginning, but as more and more evidence, pictures, and URLs are provided, deny posts will gradually increase. However, the traditional attention mechanism cannot entirely reflect different reading habits of the public. In this paper, we propose a stochastic attention mechanism to the CNN model under different information granularity, which simulates different online reading habits. For coarse-grained convolutional layers, convolutional kernels are divided into two parts: trainable convolutional kernels and untrainable stochastic convolutional kernels. We use a couple of untrainable stochastic convolutional kernels in different sizes to simulate the stochastic attention to different phrases, and different initialized kernels will extract different weighted phrase-features. Meanwhile we use a couple of trainable convolutional kernels to extract general text features. According to the theory of Extreme Learning Machines (ELMs) and Broad Learning Systems (BLSs), the classification model will be convergent when the number of stochastic hidden units is large enough [34], [35]. In fine-grained features extraction, the pooling layers are also divided into two parts, one is a max pooling layer, and the other one is a stochastic pooling layer. The stochastic pooling layer is used to simulate the stochastic attention on key words.

B. A STOCHASTIC ATTENTION CNN MODEL

In our model, the training process is divided into the following steps. Firstly, for text data, we use the word embedding method to build input matrices, and we call this word embedding process a word embedding layer, which is used as an input of a Stochastic Attention CNN (SACNN). Then, we realize a stochastic attention mapping from coarse granularity to fine granularity based on the proposed SACNN. The traversal of the convolutional kernels on a post can be regarded as a reading process with attention windows in different sizes. A reader may read posts unconsciously and use some vocabulary to express or imply his or her views unintentionally. In order to simulate this reading habits, we introduce the idea of stochastic attention to convolutional...
and pooling layers, and the convolutional kernels are divided into two parts, one of them are untrainable kernels which are initialized randomly, the other of them are trained using conventional BP algorithm. We initialize the two groups of convolutional kernels in sizes of 3, 4, 5, respectively. In pooling layers, the max pooling method is used to extract the most important fine-grained features. In other words, it is extracting keywords from phrases. Subsequently, when keywords are not obvious, we use the stochastic pooling method for features extraction. After the pooling layer, a full-connection layer is used for classification, and the dropout method is used to alleviate the over-fitting problem. The structure of a SACNN is shown in Fig 1:

In Fig 1, the dotted lines denote stochastic initialized weights which are untrainable, and the solid lines are common weights which can be trained using the BP algorithm. The convolutional layers in a SACNN can be expressed as follow:

$$Y = \left( \sum_{i=1}^{N} W^{(i)} x + b^{(i)} \right) + \left( \sum_{j=1}^{\tilde{N}} \tilde{W}^{(j)} x + \tilde{b}^{(j)} \right)$$  \quad (1)$$

where, $W^{(i)}$ denotes the i-th convolutional kernel, which can be optimized by the BP algorithm, and $\tilde{W}^{(j)}$ denotes the j-th randomly initialized convolutional kernel, which does not need to be optimized during the training process. After the convolution operations, a sigmoid active function is added to Formula (1):

$$Y_{\text{conv}} = \text{sigmoid} \left( Y_{\text{norm}} + Y_{\text{stochastic}} \right)$$  \quad (2)$$

After convolutional layers, course-grained features such as keywords are extracted by the stochastic attention method and the conventional convolutional kernels, and then fine-grained features such as keywords are extracted by the pooling layers. The pooled features can be expressed as Formula (3):

$$Y_{\text{pool}} = \alpha P_{\text{max}} (Y_{\text{conv}}) + (1 - \alpha) P_{\text{stochastic}} (Y_{\text{conv}})$$  \quad (3)$$

where, $P_{\text{max}}$ denotes the max-pooling operation, $P_{\text{stochastic}}$ denotes the stochastic-pooling operation, and $\alpha$ is a coefficient. The features extracted by pooling layers are passed to a full connection layer to finish the stance classification task.

With the increasing of hidden convolutional features, when the number of stochastic convolutional kernels is large enough, the CNN will converge to a satisfactory Local Optimal Solution based on the theory of ELM and Stochastic Gradient Descents. In ELMs, the Local Optimal Solution can be obtained following the Formula (4) and Formula (5):

$$\sum_{i=1}^{\hat{N}} f_i(Y_{\text{conv}}) \beta_i = H \hat{\beta} = T \quad (4)$$

$$\hat{\beta} = H^\dagger T \quad (5)$$

where, $\hat{\beta}$ is the Local Optimal Solution and $H^\dagger$ is the generalized inverse of the hidden layer output matrix $H$. When the number of hidden units in ELMs is large enough, the ELM model can obtain a satisfactory result.

The Local Optimal Solution of the proposed SACNN model can be obtained in the same idea of ELMs based on the Stochastic Gradient Descent method.

The algorithm of the proposed SACNN can be summarized as follows

**Algorithm 1 Minibatch Training of the Proposed SACNN**

for number of training iterations do:

1. Sample minibatch of $N$ data samples $X_{\text{train}}$ from training dataset.
2. Calculate the activations of trainable convolutional layer $h_1$.
3. Calculate the activations of untrainable stochastic convolutional layer $h_2$.
4. The output $h$ of the convolutional layer is concatenation of $h_1$ and $h_2$.
5. After the convolutional layer, $h$ is passed to the pooling layer.
6. The pooling layer contains a stochastic pooling part and a max pooling part, the output of the pooling layer is denoted as $p$.
7. The output of the pooling layer $p$ is passed to the full connection layer.
8. The output of the SACNN is calculated using the output of the full connection layer.
9. Calculate the gradients and update the parameters.

end for

**IV. EXPERIMENTS**

In our experiments, we use the PHEME rumor dataset which provides tweet-level annotations of stances, and this is a
dataset collected and annotated within the journalism use case of the PHEME FP7 project. These rumors are associated with 9 different breaking news. The PHEME dataset is created for the task of social media rumors analysis, and contains Twitter conversations which are initiated by a source tweet. In these datasets, there are 8 events in English labelled supporting, denying, agreed, disagreed, appeal for more information (querying) or commenting. Because these annotations are different from our final purposes, Zubiaga et al. convert the labels to support, deny, query, comment. The dataset contains 330 conversational threads (297 in English, and 33 in German), with a folder for each thread. Table 1 gives the numbers and proportions of posts from different stances of 8 events [11]. Events ebola-essen, putinmissing and prince-toronto have only 199 tweets.

Details can be found in Reference [36]. We directly use the labels of Reference [36] in our experiments. Notably, in any event, the stance data is slightly skewed towards the comment category, and the proportion of comment tweets is near 65%.

### A. EVALUATION MEASURES

Accuracy is often treated as a suitable evaluating indicator to estimate a classifier on a multi-classification task. As mentioned above, our datasets are imbalance and the stance data is slightly skewed towards the comment category, and the proportion of comment tweets is near 65%.

#### A.1. PRECISION AND RECALL

Precision is defined as:

\[
\text{Precision} = \frac{\sum_{k=1}^{c} tp_k}{\sum_{k=1}^{c} tp_k + \sum_{k=1}^{c} fp_k}
\]

Recall is defined as:

\[
\text{Recall} = \frac{\sum_{k=1}^{c} tp_k}{\sum_{k=1}^{c} tp_k + \sum_{k=1}^{c} fn_k}
\]

#### A.2. F1-SCORE

The F1-score is the harmonic mean of precision and recall, which is defined as:

\[
F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}
\]

#### A.3. MICRO- AND MACRO-AVERAGED F1-SCORES

\[
\text{F1}_{\text{micro}} = \frac{\sum_{k=1}^{c} \text{Precision}_k \cdot \text{Recall}_k}{\sum_{k=1}^{c} \text{Recall}_k}
\]

\[
\text{F1}_{\text{macro}} = \frac{\sum_{k=1}^{c} \text{Recall}_k}{c}
\]

where, \(tp_k\) (true positives) refer to the number of instances which are correctly classified in class \(k\), \(fp_k\) is the number of instances which are incorrectly classified in class \(k\), and \(fn_k\) is the number of instances that belong to class \(k\) but were not classified as such.

### B. BENCHMARK MODELS

#### B.1. MAJORITY VOTE

A classifier based on the training label distribution.

#### B.2. SVM

Support Vector Machines with the cost coefficient selected via nested cross-validation.

#### B.3. CRF

Conditional Random Field over temporally ordered sequences using both text and neighbouring label features. The model is trained using \(l_2\) penalized log-likelihood where the regularisation parameters are chosen using cross-validation [11].

#### B.4. GP

Gaussian Processes in supervised settings where a multitask learning kernel has been used to learn correlations across different rumors. In this paper, we use a single task kernel result [12].

#### B.5. HP

Hawkes Processes for classifying sequences of temporal textual data, which exploit both temporal and textual information [13].

#### B.6. LSTM

A simple Long Short-Term Memory -based classifier which takes the sequence of words of the tweet as its input.

#### B.7. TURING

A branch-LSTM, a neural network architecture that uses layers of LSTM units to process the whole branch of tweets, thus incorporating structural information of the conversation. Turing used in this paper is the name of the winner of SemEval 2017 Task 8.A [10].

### C. RESULTS

The PHEME dataset contains eight different topics. We compare our test results with different methods, such as Turing [10] (the winner of SemEval 2017 Task 8.A), CRF [11], GP [12], HP [13], and a simple LSTM-based classifier. As Table 2 shows, our model achieves the state-of-art evaluating indicators.

Moreover, we compare the Accuracy and \(F1_{macro}\) with some baseline methods, such as Majority Vote, SVM and models above. References [12], [13] try to use temporal information in training by Hawkes process and Gaussian processes of tweet generation. As can be seen in Table 2, the SACNN model is effective to extract features in the task.

### TABLE 1. Distribution of categories for the eight events in the dataset PHEME.

| Event             | Supporting | Denying | Querying | Commenting | Total |
|-------------------|------------|---------|----------|------------|-------|
| charliehebdo      | 239 (22.0%)| 58 (5.0%)| 53 (4.0%)| 721 (67.0%)| 1071  |
| ebola-essen       | 6 (17.0%)  | 6(17.0%) | 1 (2.0%) | 21 (61.0%) | 34    |
| ferguson          | 176 (16.0%)| 91(8.0%) | 99 (9.0%)| 718 (66.0%)| 1084  |
| Germanwings-crash | 69 (24.0%)  | 11(3.0%) | 28 (9.0%)| 173 (61.0%)| 281   |
| ottawashooting    | 161 (20.0%)| 76(9.0%) | 63 (8.0%)| 477 (61.0%)| 777   |
| prince-toronto    | 21 (20.0%)  | 7 (6.0%) | 11 (10.0%)| 64 (62.0%) | 103   |
| putinmissing      | 18 (29.0%)  | 6 (9.0%) | 5 (8.0%) | 33 (53.0%) | 62    |
| sydneysiege       | 220 (19.0%) | 89(8.0%) | 98 (8.0%)| 700 (63.0%)| 1107  |
| Total             | 910 (20.1%) | 344(7.6%) | 358(7.9%) | 2907(64.3%)| 4519  |
TABLE 2. Test results of our proposed model and other different methods.

| Model | Accuracy | \(F_{\text{macro}}\) |
|-------|----------|----------------------|
| LSTM  | 78.1     | 42.1                 |
| Turing| 78.4     | 43.4                 |
| CRF   | NA       | 44                   |
| HP    | 69.43    | 30.83                |
| GP    | 65.57    | 40.41                |
| Ours  | 78.68    | 44.47                |

FIGURE 2. Comparison of accuracy results of 4 different events.

of stance classification. We do not consider the sequential relation between posts, and our experiments verify that the sequential relation may not be such important for the rumor stance classification task. Based on the stochastic attention mechanism, the SACNN model extracts most classification related features and key information for the rumor stance classification task.

Since some methods use only four topics in training data, our experiment also includes these events. We list our experimental results in Table 3 for comparison. In Table 3, the TCNN is the Text CNN model, which has only a trainable convolutional layer and a max pooling layer. The PTCNN is the Text CNN model which adds the stochastic pooling to the pooling layer. As we can see in Table 3, our model achieves competitive results with the commonly used machine learning models. After the stochastic convolutional layer and the trainable convolutional layer, the coarse grained information can be obtained. The fined grained information is extracted by the stochastic pooling layer and max pooling layer. The SACNN is trained by BP algorithm, and the experiments verify that the simple SACNN model is effective for the rumor stance classification task.

In order to show the accuracy comparison intuitively, we show the Accuracy indicator of the 4 events in Table 3 and Fig 2. As we can see in Fig 2, our model obtains the best results in Accuracy indicator.

In order to show the performance of our model in each event, we display all the results in Table 4. The Accuracy is often deemed as an indicator of classifiers on a classification task. However, as mentioned above, for PHEME dataset, the samples are highly imbalanced, and stance posts are biased towards comments. As we can see in Table 4, other indicators are not particularly good when accuracy is high. Our algorithm performs well in events putinmissing, prince-toronto and ebola-essien.

Lastly, we discuss the influence of hyper-parameters in our proposed model. Because we use stochastic convolutional kernels and a stochastic pooling layer, we discuss the effects of different sizes of convolutional kernels and different hyper-parameter alpha in our experiments. Table 5 shows the experimental evaluating indicators on eight events based on different sizes of stochastic convolutional kernels.

In theory, a group of stochastic convolutional kernels in different sizes can better reflect the stochastic attention mechanism than a group of stochastic convolutional kernels in a single size. As we can see in Table 5, the size of stochastic convolutional kernels can affect the final experimental indices obviously. The combination of stochastic convolutional kernels in different sizes yields better results than convolutional kernels in a single size. When the size of convolutional kernels is too large, the model extracts much noise, which corresponds to an overlong phrase, so the SACNN model cannot extract effective phrase features. When the size of convolutional kernels is too small, the SACNN model cannot extract enough phrases to build the phrase features. In our experiments, a group of stochastic convolutional kernels in sizes of [3, 4, 5] achieves the best result.

We also discuss the influence of hyper-parameter alpha, in this experiment, we use a group of stochastic convolutional kernels in sizes of [3, 4, 5] and a group of trainable convolutional kernels in the same sizes. Alpha is selected from \([0, 0.2, 0.5, 0.8, 1]\), and the experimental results are shown in Table 6 and Fig 3.

In theory, the keywords extracted from phrases should contain both stochastic attentions and max attentions. As we can see in Table 6 and Fig 3, when the alpha = 0, the pooling layer in the proposed model is a max pooling layer, it means that the keywords most likely to be noticed in phrases are extracted. When the alpha = 1, the pooling layer is a stochastic-pooling layer, which means that the keyword in a phrase is selected randomly. The experimental results show that when the keywords are taken from both max pooling...
and stochastic pooling, the proposed model achieves the best results, and this conclusion coincides with our theory.

V. CONCLUSION AND FUTURE SCOPE
In this paper, we propose a SACNN model to finish the rumor stance classification task on PHEME dataset. In training, we increase the number of convolutional features generated by the stochastic attention mechanism to improve the training quality of the CNN according to the theory of BLSs and ELMs. Each evaluating indicator on the PHEME dataset is calculated separately according to each event, and the final result is the micro-average of eight events. For

| Method      | Sydney Siege | Charlie Hebdo | Ferguson | Ottawa |
|-------------|--------------|---------------|----------|---------|
| Majority    | 63.43        | 20.15         | 66.86    | 20.04   | 61.51   | 19.04   |
| SVM         | 67.26        | 35.11         | 66.86    | 20.04   | 64.58   | 35.39   |
| CRF         | 67.44        | 40.12         | 67.35    | 28.11   | 64.58   | 33.07   |
| HP          | 68.59        | 32.56         | 68.44    | 25.99   | 67.77   | 32.29   |
| GP          | 65.04        | 44.09         | 64.31    | 32.90   | 62.28   | 42.41   |
| ACNN-LSTM   | 73.15        | 40.81         | 72.14    | 36.17   | 74.93   | 43.08   |
| TCNN        | 65.23        | 41.03         | 69.14    | 43.49   | 62.50   | 39.31   | 68.75   | 43.24   |
| PTCNN       | 73.05        | 44.22         | 70.31    | 44.22   | 63.28   | 39.80   | 69.14   | 43.49   |
| Ours        | **73.83**    | **46.43**     | **82.78**| 40.33   | 70.26   | 34.21   | **75.97**| 36.00   |

### TABLE 5. Results of using different window size.

| Window size | Accuracy | Precision_{macro} | Recall_{macro} | F1_{macro} |
|-------------|----------|--------------------|---------------|-----------|
| 2           | 76.32    | 43.08              | 34.4          | 37.9      |
| 3           | 75.28    | 41.32              | 31.92         | 35.55     |
| 4           | 76.67    | 43.59              | 35.08         | 38.44     |
| 5           | 75.77    | 39.73              | 31.5          | 34.82     |
| 3, 4        | 76.89    | 40.35              | 34.78         | 37.36     |
| 4, 5        | 77.88    | 43.9               | 39.15         | 41.39     |
| 2, 3, 4     | 76.1     | 39.03              | 31.87         | 34.75     |
| 3, 4, 5     | **78.68**| **49.77**          | **40.69**     | **44.47** |
| 4, 4, 5     | 73.33    | 37.46              | 29.62         | 32.79     |

### TABLE 6. Results of using different parameter alpha.

| Accuracy  | 0     | 0.2   | 0.5   | 0.8   | 1    |
|-----------|------|-------|-------|-------|-----|
| Precision_{macro} | 73.91 | 75.63 | **78.68** | 76.76 | 74.4 |
| Recall_{macro}    | 35.95 | 37.8  | **49.77** | 39.16 | 34.97 |
| F1_{macro}        | 29.59 | 30.57 | **40.69** | 31.87 | 28.88 |
| 32.13 | 33.47 | **44.47** | 34.38 | 31.3 |

VOLUME 8, 2020

80777
the Accuracy indicator, our model achieves the state-of-art results. Data preprocessing is important in the training process. In our work, the GloVe is applied for word embedding. According to different data characteristics, better word embedding technology might be considered in our future research. Combining multi-mode features, including text characteristics and other features such as user information may bring some improvement to the classification task on unbalanced datasets, and this will be our feature research as well. The rumor stance classification task is important for the rumor detection task and the veracity classification task. Effectively combine the rumor stance classification with other related tasks in a multi-task frame will also be our future work.

REFERENCES
[1] B. Fang, Y. Jia, Y. Han, S. Li, and B. Zhou, “A survey of social network and information dissemination analysis,” Chin. Sci. Bull., vol. 59, no. 32, pp. 4163–4172, Nov. 2014.
[2] A. Zubiaga, A. Aker, and K. Bontcheva, “Detection and resolution of rumours in social media: A survey,” ACM Comput. Surv., vol. 51, no. 2, pp. 1–36, 2017.
[3] P. Tolmie, R. Procter, D. W. Randall, M. Rounecefield, C. Burger, G. W. S. Hoi, A. Zubiaga, and M. Liakata, “Supporting the use of user generated content in journalistic practice,” in Proc. CHI Conf. Hum. Factors Comput. Syst., May 2017, pp. 3632–3644.
[4] A. Zubiaga, M. Liakata, R. Procter, G. Wong Sak Hoi, and P. Tolmie, “Analysing how people orient to and spread rumours in social media by looking at conversational threads,” PLoS ONE, vol. 11, no. 3, 2016, Art. no. e0150989.
[5] E. Wahyu Pamungkas, V. Basile, and V. Patti, “Stance classification and information dissemination analysis,” in Proc. Conf. Empirical Methods Natu- ral Lang. Process. (EMNLP), 2014, pp. 1532–1543. [Online]. Available: https://www.aclweb.org/anthology/D14-1124.
[6] W.-F. Chen and L.-W. Ku, “UTCNN: A deep learning model of stance classification on social media text,” 2016, arXiv:1611.03599. [Online]. Available: http://arxiv.org/abs/1611.03599
[7] G. Rajendran, P. Poornachandran, and B. Chitturi, “Deep learning model on stance classification,” in Proc. Int. Conf. Adv. Comput. Commun. Inform., Sep. 2017, pp. 2407–2409.
[8] L. H. Son, A. Kumar, S. R. Sangwan, A. Arora, A. Nayyar, M. Abdel-Basset, “Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network,” IEEE Access, vol. 7, pp. 23319–23328, 2019.
[9] A. Kumar, S. R. Sangwan, and A. Nayyar, “Rumour veracity detection on twitter using particle swarm optimized shallow classifiers,” Multimedia Tools Appl., vol. 78, no. 17, pp. 24083–24101, 2019.
[10] F. S. Saeed, A. Paul, P. Karthigaikumar, and A. Nayyar, “Convolutional neural network based early fire detection,” Multimedia Tools Appl., vol. 79, pp. 9083–9099, Jun. 2019.
[11] J. Alrubi, A. Nayyar, and A. Kumar, “Machine learning from theory to algorithms: An overview,” J. Phys., Conf. Ser., vol. 1142, no. 1, 2018, Art. no. 012012.
[12] K. Xuan and R. Xia, “Rumor stance classification via machine learning with text, user and propagation features,” in Proc. IEEE Int. Conf. Data Mining Workshops, Nov. 2019, pp. 560–566.
[13] L. Poddar, W. Hsu, and M. L. Lee, “Predicting stances in Twitter conversations for detecting veracity of rumors: A neural approach,” in Proc. IEEE Int. Conf. Tools Artif. Intel., Nov. 2018, pp. 65–72.
[14] P. Wei, N. Xu, and W. Mao, “Modeling conversation structure and temporal dynamics for jointly predicting rumor stance and veracity,” in Proc. EMNLP, 2019, pp. 4789–4800.
[15] A. P. B. Veyseh, J. Ebrahimi, D. Lowd, and D. Dou, “A temporal attentional model for rumor stance classification,” in Proc. ACM Conf. Inf. Knowl. Manag., 2017, pp. 2335–2338.
[16] V. Vaswani, A. P. B. Veyseh, J. Ebrahimi, D. Lowd, and D. Dou, “A temporal attentional model for rumor stance classification,” in Proc. ACM Conf. Inf. Knowl. Manag., 2017, pp. 2335–2338.
[17] T. Chen, X. Li, H. Yin, and J. Zhang, “Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection,” in Proc. Paciﬁc–Asia Conf. Knowl. Discovery Data Mining, 2018, pp. 40–52.
[18] A. Zubiaga, M. Liakata, and R. Procter, “Exploiting context for rumour stance detection in social media,” in Proc. Int. Conf. Social Inform. Cham, Switzerland: Springer, 2017, pp. 109–123.
[19] E. Kochkina, M. Liakata, and I. Augenstein, “Turing at SemEval-2017 task 8: Sequential approach to rumour stance classiﬁcation with branch-LSTM,” 2017, arXiv:1704.07221. [Online]. Available: http://arxiv.org/abs/1704.07221
[20] I. Augenstein, T. Rocktäschel, A. Vlachos, and K. Bontcheva, “Stance detection with bidirectional conditional encoding,” 2016, arXiv:1606.05464. [Online]. Available: http://arxiv.org/abs/1606.05464
[21] A. Zubiaga, E. Kochkina, M. Liakata, R. Procter, and M. Lukasik, “Stance classiﬁcation in rumours as a sequential task exploiting the tree structure of social media conversations,” 2016, arXiv:1609.09028. [Online]. Available: http://arxiv.org/abs/1609.09028
[22] C. Raffel and D. P. W. Ellis, “Feed-forward networks with attention can solve some long-term memory problems,” 2015, arXiv:1512.08756. [Online]. Available: http://arxiv.org/abs/1512.08756
[23] E. Kochkina, M. Liakata, and A. Zubiaga, “All-in-one: Multi-task learning for rumor veriﬁcation,” 2018, arXiv:1806.03713. [Online]. Available: http://arxiv.org/abs/1806.03713
[24] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, “Extreme learning machine for regression and multiclass classiﬁcation,” IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 42, no. 2, pp. 513–529, Apr. 2012.
[25] C. P. Chen and Z. Liu, “Broad learning system: An effective and efficient incremental learning system without the need for deep architecture,” IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 1, pp. 10–24, Jan. 2018.
[26] A. Zubiaga, M. Liakata, R. Procter, K. Bontcheva, and P. Tolmie, “Crowd-sourcing the annotation of rumours from social media,” in Proc. 24th Int. Conf. World Wide Web (WWW Companion), 2015, pp. 347–353.