scientific reports

Check for updates

OPEN

A microblog content credibility evaluation model based on collaborative key points

Ling Xing✉, Jinglong Yao, Honghai Wu & Huahong Ma

The spread of false content on microblogging platforms has created information security threats for users and platforms alike. The confusion caused by false content complicates feature selection during credibility evaluation. To solve this problem, a collaborative key point-based content credibility evaluation model, CECKP, is proposed in this paper. The model obtains the key points of the microblog text from the word level to the sentence level, then evaluates the credibility according to the semantics of the key points. In addition, a rumor lexicon constructed collaboratively during word-level coding strengthens the semantics of related words and solves the feature selection problem when using deep learning methods for content credibility evaluation. Experimental results show that, compared with the Att-BiLSTM model, the F1 score of the proposed model increases by 3.83% and 3.8% when the evaluation results are true and false respectively. The proposed model accordingly improves the performance of content credibility evaluation based on optimized feature selection.

Online social network is a new mean of obtaining information from people. Due to the large volume of user-generated content, researchers use various techniques, such as content credibility evaluation or data mining to evaluate this information automatically[1–4]. Microblog is one of the important platforms in onilne social networks. The development of microblogging has greatly accelerated the depth and speed of information exchange between users[5]. However, while microblogging improves convenience for users, it also reduces the cost of disseminating false content. The dissemination of false content hurts social stability, disrupts people's normal lives and endangers network information security[6,7]. It is therefore important to evaluate the credibility of microblog content, a practice that has numerous benefits[8,9]. Related deep learning methods have strong feature learning capabilities and can learn deep features from microblogs to achieve better credibility evaluation results[10].

Microblog content is created and disseminated at specific times and via specific channels, which complicates research into deep learning-based content credibility evaluation[11]. The concept of content semantics began to be put forward, and the content semantics after mining are called features[12–15]. False content is highly confusing, meaning that analyzing and mining the characteristics of false content can produce a better evaluation effect. Therefore, feature selection is of great importance to content credibility evaluation[16]. To solve the feature selection problem in the content credibility evaluation context, researchers have proposed that deep learning be used to mine microblog text features[17]. Geng et al.[18] use the attention mechanism to obtain the features that are most useful for the task in question. Kumar et al.[19] developed a multi-head attention mechanism to obtain sentence-level key features. The multi-head attention model has achieved outstanding performance in mining multiple key point features[20]. Sangeetha et al. employ a multi-head attention mechanism to process sentence input sequences in parallel[21]. Khan et al. introduced a multi-head attention mechanism in a convolutional neural network to ensure that the model automatically selects key features[22].

The acquisition of microblog features begins at the word level and moves to the sentence level. The introduction of the lexicon can add more task-related word information, leading to stronger semantics[23,24]. However, no lexicon designed for content credibility evaluation has yet been developed.

Based on the above analysis, this paper proposes a microblog content credibility evaluation model based on collaborative key points. The main innovations of our work can be summarized as follows: (1) using fake microblogs to extract the basic rumor word set, then employing an iterative algorithm based on the Word2Vec word vector cosine similarity calculation method to expand the rumor word database; (2) using the improved TF-IDF algorithm to calculate the comprehensive rumor value of the words in the rumor lexicon, after which the comprehensive rumor value of the words and the words themselves are vectorized, such that the semantics are strengthened to a degree that aids in the acquisition of microblog key points; (3) employing the multi-head

College of Information Engineering, Henan University of Science and Technology, Luoyang 471023, Henan, China. ✉email: xingling_my@haust.edu.cn

attention mechanism twice—at the word level and sentence level—to obtain the microblog key points, which improves the text self-attention performance and enhances the model's ability to evaluate the credibility of the content.

The rest of this paper is structured as follows. "Related works" section introduces the related works of content credibility evaluation based on deep learning. Section "Proposed credibility evaluation model" section describes the model proposed in this paper. "Experimental results and evaluation" section discusses our experiments. And "Conclusion" section concludes.

Related works

As microblogging has continued to develop, information security issues caused by false information have attracted the attention of researchers. To date, researchers have proposed many methods to solve the feature selection problem in content credibility evaluation.

Common methods of this kind are mainly based on deep learning[25]. Unlike classifier-based methods[26,27], deep learning methods can mine deep features of the content. Ma et al.[28] used Recurrent Neural Networks (RNNs) to learn the features of the content. Duong et al.[29] used RNN to combine the text and text source characteristics, while Chen et al.[30] used RNN to learn the features in text and comments. It can be observed from these works that the introduction of other effective features in addition to the text boosts the performance of these methods. Torshizi et al.[31] clustered the data, then used a long short-term memory network (LSTM) to analyze each cluster and determine whether the content is truthful. To solve the context information acquisition problem, an improved bidirectional long short-term memory network (BiLSTM) method is proposed[32,33]. Guo et al.[34] used BiLSTM to process data in two directions and obtained text context information. Recently, substantial work has shown that pre-trained models on the large corpus can learn universal language representations, which are beneficial for content credibility evaluation tasks and can avoid training a new model from scratch[35]. And various pre-training tasks are proposed for different purposes, such as GloVe[36] and BERT[37].

When performing collaborative deep learning tasks, establishing or expanding a lexicon based on the characteristics of the task can help neural networks to more efficiently learn relevant information features. Wang et al.[38] proposed an emotional vocabulary expansion method based on word spacing and mutual point information. Jia et al.[39] added new contents and emotional symbols to the HowNet Emotion Dictionary to analyze the evolution of public opinion. Wang et al.[40] used an improved dictionary classification method to calculate and label the emotional score of the content in the dataset and achieve emotional classification for microblogs. Zhang et al.[41] used the TF-IDF algorithm to extract keywords from comments and construct an emotion dictionary based on word similarity. However, due to the wide variety of false information contained in microblogs, it is not possible to expand the thesaurus by mining the emotional information of certain words in a similar way to the sentiment classification task[42], resulting in the current absence of a content credibility-related lexicon.

Subsequently, the researchers found that introducing the attention mechanism into the model can effectively improve the performance of the model. The attention mechanism processes large amounts of information and selects the information that is more critical to the goal. Xu et al.[43] introduced the content focus mechanism to aggregate keywords in original tweets. Ghanem et al.[44] focused on the emotional features generated by data to identify false information. Wu et al.[45] constructed a propagation graph based on the propagation characteristics of false information and dynamically adjusted the weight of nodes in the graph using the attention mechanism. Fang et al.[46] combined the multi-head attention model with Convolutional Neural Networks (CNNs) to select words that are more conducive to classification between levels, thereby achieving fake news detection.

Proposed credibility evaluation model

Hierarchical attention networks (HANs) encode from the word level to the sentence level, which is an effective means of obtaining key points[47]. This paper uses a multi-head attention mechanism based on HAN; at the same time, it introduces a rumor lexicon to facilitate word coding, and accordingly builds a microblog content credibility evaluation model based on collaborative key points (CECKP). The overall structure of the model is illustrated in Fig. 1. This model is divided into four parts: data processing, the key points of words, the key points of sentences, and content credibility evaluation.

Data processing. Due to the dearth of large, open and complete datasets appropriate for the present task, most relevant studies use the application program interface (API) provided by the platform to obtain data for their experiments[48]. In this paper, 15,000 points of false information and 15,000 of true information were extracted, from which an experimental dataset for content credibility assessment based on Sina Weibo (named the CECKP-Dataset) was constructed. The false information is derived from the results announcement section of the false information report in the Sina Microblog Community Management Center, while the true information is made up of verified content posted by well-known official accounts. In the subsequent word vector representation, it is necessary to convert the words and the comprehensive rumor value of these words into a vector form. It is accordingly necessary to construct a rumor vocabulary based on the characteristics of the words in microblogs containing false information. The remainder of this section will explain the construction of the rumor lexicon, its acquisition, its expansion, and the calculation of the comprehensive rumor value.

Construction of the rumor lexicon This paper sorts the false content published by the Sina Weibo community management center and uses a combination of manual screening and automatic expansion to construct a microblog rumor lexicon. The preprocessed lexicon is manually screened to obtain the basic rumor word set $\{B_1,B_2,B_3,\ldots,B_i\}$. The remaining part is used as the candidate lexicon word set $\{A_1,A_2,A_3,\ldots,A_i\}$, after which the Word2vec word vector cosine similarity calculation method is employed to calculate the similarity between the candidate words and the basic words[49]. Words that meet the initial requirements will enter the extended lexicon,

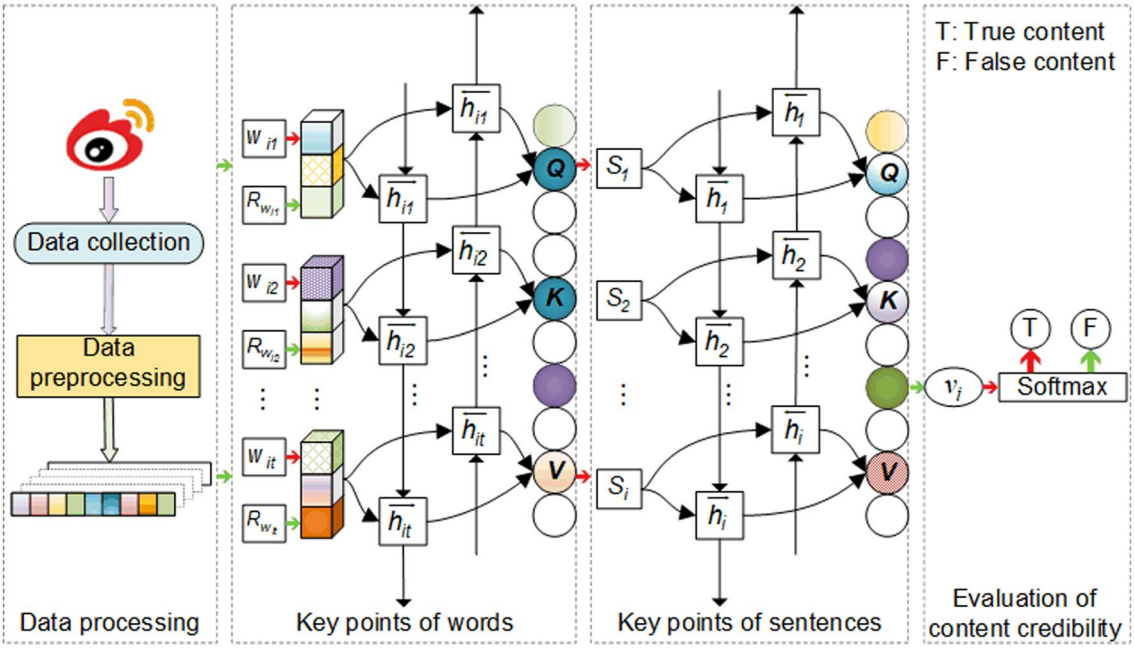

Figure 1. Microblog content credibility evaluation model based on collaborative key points.

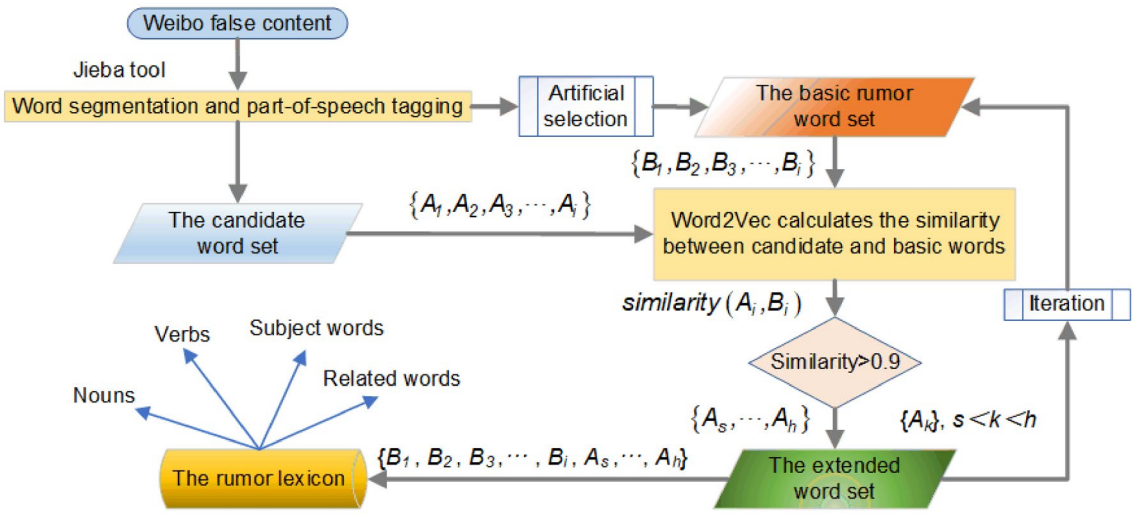

Figure 2. Flowchart of microblog rumor word database construction.

after which some words in the extended lexicon will be put into the basic rumor lexicon again for iterative calculations until no new words can be obtained. The construction process is illustrated in Fig. 2.

Acquisition of basic rumor word set: First, duplicate content removal is carried out on the microblog false information corpus; here, we delete "@", "&", and other special symbols that cause interference. The added dictionary function of the Jieba word segmentation tool is then employed to integrate the Chinese word segmentation dictionary as an added dictionary used to mark the word segmentation and parts of speech on the microblog corpus. The dictionary integrates the Baidu and Sogou word banks, as well as some names of individuals and popular new terms. True and false content differs minimally in terms of structure and thus needs to be considered in combination with the actual situation. Nouns are generally the subject or indicator of the main meaning, and are thus better able to represent false information; moreover, "#" or "[]" are often used to mark key nouns and thus highlight the theme of the post. The application scenarios of verbs are limited by content and often appear in some false information. Related words refer to words that appear together in the text: when these words appear together, the text is more likely to be false. In this paper, nouns, verbs, subject words and related words are selected to construct a rumor lexicon.

Category	Number	Example
Noun	1687	Health, disaster, elderly
Verb	756	Crash, lead to, provocation
Subject	354	Blast, vacation, plastic
Connective	49	Child…abducted…, forward…free…

Table 1. Information about the rumor lexicon.

Expansion of the rumor lexicon: To efficiently filter the candidate words in the corpus, the Word2Vec word vector cosine similarity calculation function is employed to calculate the similarity between each word in the candidate vocabulary set and the basic rumor word set. The similarity calculation formula is as shown in Eq. (1):

$$similarity(A,B) = \frac{\boldsymbol{A}\cdot\boldsymbol{B}}{\|\boldsymbol{A}\|\|\boldsymbol{B}\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}} \tag{1}$$

Here, $\boldsymbol{A}$ and $\boldsymbol{B}$ represent the word vector in the candidate word set and the basic rumor word set, respectively. If the similarity exceeds 0.9, the word is added to the extended lexicon $\{A_s,\ldots,A_h\}$; the manually filtered extension $\{A_k\}$ is then added to the basic rumor word set, where $s<k<h$, after which incremental iterations are performed to mine more related words. When the algorithm cannot find new words, the iterations stop, and we can obtain the final rumor lexicon $\{B_1,B_2,B_3,\ldots,B_i,A_s,\ldots,A_h\}$. The relevant information regarding the constructed rumor lexicon is presented in Table 1.

Calculation of comprehensive rumor value: In this paper, the improved TF-IDF algorithm is used to calculate the importance of the words in the rumor word database; these calculated results are then used as the comprehensive rumor value of the words in question. Because only part of the noun is modified in the false information, the requirements for the text frequency are relatively high, while the requirements for the inverse text frequency are relatively low. Accordingly, adjusting the weight in the formula makes TF more powerful than IDF. At the same time, to eliminate the influence of different microblog text lengths on the weight of words, the calculation formula is subjected to cosine normalization, with the word frequency taken as a logarithm to eliminate the influence of different word frequencies on the overall calculation.

The calculation formula of the comprehensive rumor value R is presented in Eq. (2):

$$R(w) = \frac{\log_2(\varphi tf+1)\times\log_2\left(\frac{N}{(1-\varphi)idf}\right)}{\sqrt{\sum_i\left(\log_2(\varphi tf_i+1)\times\log_2\left(\frac{N}{(1-\varphi)idf_i}\right)\right)^2}} \tag{2}$$

Here, N represents the total number of fake microblogs and φ is the weight of TF. The calculated comprehensive rumor value of the words in the fake microblog will be transformed into vectors and words, which are entered simultaneously into the model for training.

Key points of words. The acquisition of word key points comprises three steps: word vector representation, word sequence coding, and word attention.

Word vector representation: The quality of word vector expression has an important influence on both the semantic expression of microblog texts and the effectiveness of credibility evaluation tasks. This paper introduces the constructed rumor lexicon into the word vector representation layer. The input word vector comprises two key parts: the word vector and the comprehensive rumor value of the word. The calculation formula is as shown in Eq. (3):

$$\boldsymbol{x}_{it} = \boldsymbol{W}_e w_{it} \oplus R_{w_{it}}, t \in [1,T] \tag{3}$$

Here, $\boldsymbol{x}_{it}$ represents the word vector of the tth word in the ith sentence, $\boldsymbol{W}_e w_{it}$ represents the word vector of word w_{it}, R_{wit} represents the word's comprehensive rumor value, T represents the length of the sentence, and $\boldsymbol{W}_e$ is a 200-dimensional word vector obtained via pre-training with the Word2Vec tool.

Word sequence encoding: The forward LSTM layer processes the sequence from left to right by connecting two adjacent units, such as the current unit input x_1 and the hidden state h_{t-1} of the previous unit input. For a given input sequence $x_1, x_2, \ldots, x_{it}$, the forward LSTM layer generates an output sequence "$\overrightarrow{h}$". The formula used to calculate the forward LSTM layer is presented in Eq. (4):

$$\overrightarrow{h_{it}} = \overrightarrow{LSTM(x_{it})} \tag{4}$$

The reverse LSTM layer processes the sequence from right to left by connecting two adjacent units; for example, the hidden state of the input x_1 of the current unit and the input of the next unit h_{t+1}. For a given input

sequence $x_{it}, \ldots, x_2, x_1$, the reverse LSTM layer generates the output sequence "$\overleftarrow{h}$". The formula used to calculate the reverse LSTM layer is shown in Eq. (5):

$$\overleftarrow{h_{it}} = \overleftarrow{LSTM(x_{it})} \tag{5}$$

The forward and reverse output are combined in Eq. (6):

$$\overleftrightarrow{h_{it}} = \overrightarrow{h_{it}} \oplus \overleftarrow{h_{it}} \tag{6}$$

Here, $\overrightarrow{h_{it}}$ represents the forward LSTM layer output value of the tth word in the ith sentence, $\overleftarrow{h_{it}}$ represents the output value of the reverse LSTM layer of the tth word in the ith sentence, and $\overleftrightarrow{h_{it}}$ represents the BiLSTM encoding output value of the tth word in the ith sentence.

Word attention: The multi-head attention model involves stacking several basic units of scaled dot-product attention. Here the input matrix is Query($\boldsymbol{Q}$), Key($\boldsymbol{K}$), Value($\boldsymbol{V}$), and $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{n \times d}$, the scaled Dot-Product Attention consists of h layers, and the attention calculation of each layer is as shown in Eq. (7):

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{\mathrm{d_k}}}\right)\boldsymbol{V} \tag{7}$$

Here, d is the number of hidden units in the neural network. Because the attention mechanism used by the multi-head attention model is self-attention, the input vector $\boldsymbol{Q}=\boldsymbol{K}=\boldsymbol{V}$. Linear transformation is required for calculation, and the parameters of $\boldsymbol{Q}$, $\boldsymbol{K}$ and $\boldsymbol{V}$ differ each time. When calculating the weights of $\boldsymbol{Q}$ and all $\boldsymbol{K}$, the point product similarity function is used, and is scaled through dividing by K dimensions to avoid the problem of an overlarge internal product value. The softmax function is then used to normalize the weights, which are then weighted and summed with the corresponding key values to obtain the attention. The results obtained after h iterations of attention reduction are spliced, after which the values obtained via linear transformation are used as the results of the multi-head attention model. The calculation equation is as shown in (8) and (9):

$$\text{head}_i = \text{Attention}\left(\boldsymbol{Q}\boldsymbol{W}_i^{Q}, \boldsymbol{K}\boldsymbol{W}_i^{K}, \boldsymbol{V}\boldsymbol{W}_i^{V}\right) \tag{8}$$

$$\text{MultiHead}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_h)\boldsymbol{W}^{o} \tag{9}$$

Here, $\boldsymbol{W}^{o}$ represents the weight of linear transformation, while s represents the calculated MultiHead ($\boldsymbol{Q}$, $\boldsymbol{K}$, $\boldsymbol{V}$) value, which is used to represent more feature information learned from different positions or spaces, as shown in Eq. (10):

$$s = \text{MultiHead}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) \tag{10}$$

The Max pooling layer is then used for compression change to obtain the most influential sequence $S_i, i \in [1, L]$, where L represents the number of statements.

Key points of sentences. Sentence key points are obtained from the output of word key points. There are two steps involved, including sentence sequence coding and sentence attention.

Sentence sequence coding: Because the semantic features of each sentence affect the credibility of the entire microblog, BiLSTM is used to mine the semantic features between sentences in text. Its calculation equation is shown below in (11):

$$\begin{cases} \overrightarrow{h_i} = \overrightarrow{LSTM(S_i)} \\ \overleftarrow{h_i} = \overleftarrow{LSTM(S_i)} \\ \overleftrightarrow{h_i} = \overrightarrow{h_i} \oplus \overleftarrow{h_i} \end{cases} \tag{11}$$

Here, $\overrightarrow{h_i}$ represents the output of the ith statement through forward LSTM coding, $\overleftarrow{h_i}$ indicates the output of the ith statement through reverse LSTM coding, and $\overleftrightarrow{h_i}$ represents the output of the ith statement encoded by BiLSTM. The encoded output will then be sent to the sentence attention layer to identify the most important part.

Sentence attention: To identify high-impact sentences in the text, each word must be combined with all other words in the sentence weight calculation of the sentence coding sequence using multivalent attention. The characteristic representation v is obtained as shown in Eq. (12):

$$v = \text{MultiHead}\left(\boldsymbol{Q}', \boldsymbol{K}', \boldsymbol{V}'\right) \tag{12}$$

Following calculation, the Max pooling layer is used to compress and change, after which the sentence sequence v_i with the most influence is obtained.

Content credibility evaluation. Once the previous steps are complete, the key point v_i of a microblog represents its semantic features. The microblog content credibility evaluation layer determines the deep-level feature information following multi-layer learning by constructing a true and false binary classification decider for the semantic features of the microblog text, thereby obtaining the final content credibility evaluation result.

Adjustable parameters	Value
Vector embedding dimension	200
Learning_rate	0.001
Optimizer	Adam
Batch_size	64
Dropout	0.3
Number of layers for multi-head attention	8

Table 2. Adjustable parameter settings.

Model	Classification	Accuracy	Precision	Recall	F1 Score
SVM	False	0.7082	0.7052	0.7153	0.7102
	True		0.7119	0.7010	0.7061
CNN	False	0.8280	0.8222	0.8370	0.8295
	True		0.8340	0.8190	0.8264
R-CNN	False	0.8452	0.8348	0.8607	0.8475
	True		0.8562	0.8297	0.8427
H-BLSTM	False	0.8475	0.8428	0.8543	0.8485
	True		0.8523	0.8407	0.8465
Att- BiLSTM	False	0.8607	0.8585	0.8637	0.8611
	True		0.8628	0.8577	0.8602
CECKP	False	0.8988	0.8966	0.9017	0.8991
	True		0.9011	0.8960	0.8985

Table 3. Experimental results for CECKP model and comparison models.

In this paper, the softmax function is used to construct the credibility classifier, and the calculation equation is as shown in (13):

$$p = \text{softmax}(W_c v_i + b_c) \tag{13}$$

Here, p represents the probability of the microblog content being true or false. In this paper, the objective function uses the negative log-likelihood function as the training loss function, the calculation equation for which is as shown in (14):

$$\mathcal{L} = -\sum_{d} \log(p_{dz}) \tag{14}$$

Here, z represents the true or false label of the text d.

Experimental results and evaluation

Experimental environment and related settings. In this paper, the experimental hardware platform is Intel Xeon(2.20 GHz), 12G memory, NVIDIA Tesla P100 16 GB. The experimental software platform is Ubuntu 18.04 operating system and development environment is Python3.6 programming language.

Ten-fold cross validation was used in the experiments[50], with the average score of ten-fold cross validation being used to indicate the final model performance. The evaluation indexes were accuracy, precision, recall and F1 score. The adjustable parameter settings of CECKP are listed in Table 2.

There are many deep learning-based models at this stage. These novel models have different emphases for different processing objects. When comparing experiments with the model proposed in this paper, a lot of problems may arise. Therefore, it is a good choice to choose baseline models when comparing. To test the performance of the CECKP model, a comparative experiment was conducted with the relevant baseline models, including a classifier method (SVM[51]) and several deep learning methods (CNN[52], R-CNN[53], H-BLSTM[34], Att-BILSTM[54]). The parameter settings of the comparison experimental models are set according to the relevant settings in the reference papers, while ten-fold cross validation is also adopted for the data division method.

Experimental results and analysis. In this paper, several models are used to verify the validity of the CECKP model. The total accuracy, precision, recall and F1 score for each model in the CECKP-Dataset were obtained and presented in Table 3. To visually illustrate the differences in the obtained values, a bar chart is used to compare the results. A comparison of "true" evaluation results is plotted in Fig. 3, while a comparison of "false" evaluation results is shown in Fig. 4.

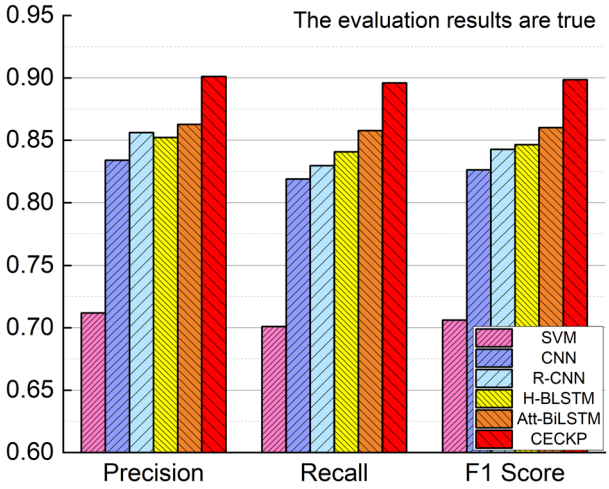

Figure 3. Performance comparison of models on the CECKP-dataset when evaluation results are true.

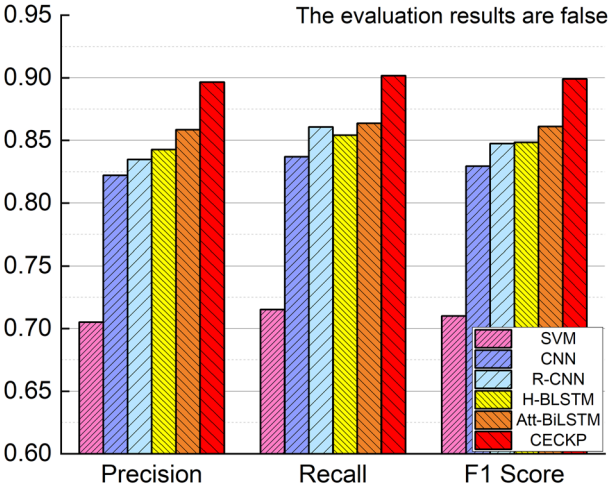

Figure 4. Performance comparison of models on the CECKP-dataset when evaluation results are false.

The CECKP model obtains the key points of words and sentences through multi-part processing of microblogs, composes the important words contained in a sentence into a representation of the sentence, then composes the important sentences in the text into a representation of the text, and subsequently obtains the final key points of the microblog; the obtained key points reflect the semantic features of the microblog text to the greatest extent, and were thus found to yield good evaluation results. Compared with other models, our approach enables more accurate microblog semantics to be obtained, so that good evaluation results can be achieved. Att-BiLSTM can learn top-down and down-top data features, while also adding an attention mechanism to focus the model on more important data, meaning that the overall effect is better. The difference between Att-BiLSTM and the model proposed in this paper lies in the attention mechanism in Att-BiLSTM being calculated only once, while in CECKP, multiple attentions are calculated several times to get the key points of the microblog; this indicates that the attention mechanism plays an important role in content credibility assessment-related research.

Model simplification test. To further verify the effectiveness of the CECKP model, the following models were constructed: (1) the CECKP-NT model, in which only the multi-head attention model is used, while the collaboration with the rumor word database is not introduced when acquiring the word encoding; (2) the CECKP-NK model, which represents the use of two-way LSTM to encode text content. When this encoding is performed, it is coordinated with the rumor vocabulary, and the multi-head attention model is not used. The experimental results of the model simplification test are presented in Table 4.

As can be seen from the above results, the collaborative key points-based method for the credibility evaluation of microblog content is significantly more effective. When the multi-attention model is not used, the key points in the microblog cannot be obtained and the text is not fully mined, resulting in the F1 score for fake microblog detection decreasing by 0.0288. When acquiring the word encoding, there is no coordination of the lexicon, and the semantic enhancement of part of the known rumor words is lost, resulting in a decrease of 0.0252 in the false

Model	Classification	Accuracy	Precision	Recall	F1 Score
CECKP-NT	False	0.8752	0.8827	0.8653	0.8739
	True		0.8679	0.8850	0.8764
CECKP-NK	False	0.8683	0.8576	0.8833	0.8703
	True		0.8797	0.8533	0.8633
CECKP	False	0.8988	0.8966	0.9017	0.8991
	True		0.9011	0.8960	0.8985

Table 4. Experimental results of model simplification test.

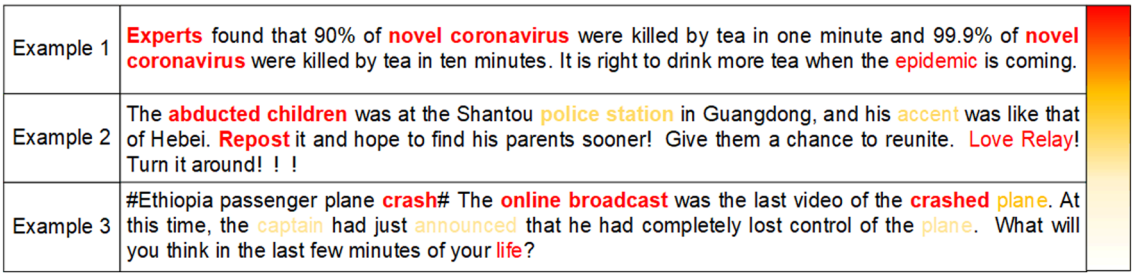

Figure 5. Visual analysis of the weights of key points for rumor words.

microblog detection F1 score. According to the experimental results, the acquisition of key points has a slightly greater influence on the credibility evaluation task than the rumor lexicon. However, if the rumor lexicon was to be further expanded in subsequent research, its effect may be improved.

Visualized analysis. To further verify the validity of the CECKP model proposed in this paper in terms of word attention, three fake microblog messages are selected for more in-depth visual analysis. In Fig. 5, the color depth is used to represent the weight of words following key point calculation: the larger the weight, the darker the color. Combine these words as the key points of the text.

It can be seen that the model proposed in this paper is able to select words with a certain comprehensive rumor value, and can also select words that appear repeatedly in the text and have significant meaning; examples include nouns such as experts, novel coronavirus, epidemic, etc. The key points can better represent the text semantics and improve the evaluation effect. However, because these words appear more frequently in both false and true microblogs, it is necessary to get the key points of the full text in conjunction with other words and sentences composed of keywords in the full text, then jointly evaluate the content credibility; otherwise, assessment errors may arise.

Pre-trained language models analysis. Pre-trained language models were very important in the task of credibility evaluation.We analyzed several commonly used pre-trained language models early in our research, including Word2Vec, BERT, and random embeddings.

Among them, Word2Vec was suitable for non-context-related word vectors. Under this model, similar words will have similar vectors. Bert applied to context-based word vectors, and obtained word vectors for each token based on the input sentence and context. Random embeddings were more advantageous in saving training time and computing resources, but its capabilities were limited.

Bert was better at analyzing sentences with complex structures or ambiguous words in sentences, but it required a long training time, and BERT has a great influence on the distribution of the corpus. When the length of the text we want to represent changes, it will have an impact on model training.

Although Word2Vec cannot obtain the context vector, it can obtain a stable vector training result, and the word vector of the same word can be directly used. The difference is that the word vectors obtained with Bert will change with the context, resulting in a large increase in computational time.

Considering these factors, we choose the Word2Vec model as the pre-trained language model, and the rumor lexicon we build can complement the Word2Vec model to a certain extent.

Conclusion

To solve the problem of feature selection in content credibility evaluation, this paper proposes a microblog content credibility evaluation model based on collaborative key points (CECKP). In this model, the key points of words and sentences in the text are acquired by means of a multi-attention mechanism, while a rumor lexis is jointly constructed during the acquisition of the key points of words to strengthen the word semantics; subsequently, the credibility of the microblog content is evaluated through the obtained microblog key points. Ten-fold cross validation experiments prove that the proposed model has high accuracy, precision, recall, and F1 score when evaluating the credibility of microblog content. In addition to the text semantic features, microblogs also

have many other related features. Currently, multi-modal feature fusion is being increasingly applied in various classification tasks[55,56]. In future work, we will focus on the application of attention mechanisms in multi-modal feature fusion to automatically distinguish various modal features according to weight. A more convenient and efficient assessment of content credibility can be achieved through the use of a relevant attention mechanism.

Data availability

The data used to support the findings of this study are available from the corresponding author upon reasonable request.

Received: 9 January 2022; Accepted: 29 August 2022
Published online: 08 September 2022

References

1. Huo, Y., Fan, J., Wen, Y. & Li, R. A cross-layer cooperative jamming scheme for social internet of things. *Tsinghua Sci. Technol.* **26**(4), 523–535 (2021).
2. Hou, Q., Han, M. & Cai, Z. Survey on data analysis in social media: A practical application aspect. *Big Data Min. Anal.* **3**(4), 259–279 (2020).
3. Liao, X., Zheng, D. & Cao, X. Coronavirus pandemic analysis through tripartite graph clustering in online social networks. *Big Data Min. Anal.* **4**(4), 242–251 (2021).
4. Evans, J. Social computing unhinged. *J. Soc. Comput.* **1**(1), 1–13 (2020).
5. Cencetti, G., Battiston, F., Lepri, B. & Karsai, M. Temporal properties of higher-order interactions in social networks. *Sci. Rep.* **11**(1), 1–10 (2021).
6. Gallotti, R., Valle, F., Castaldo, N., Sacco, P. & De Domenico, M. Assessing the risks of 'infodemics' in response to COVID-19 epidemics. *Nat. Hum. Behav.* **4**(12), 1285–1293 (2020).
7. Aldo Tenis, A. & Santhosh, R. Challenges and security issues of online social networks (OSN). *Mob. Comput. Sustain. Inform.* **68**, 703–709 (2022).
8. Voloch, N., Levy, P., Elmakies, M. & Gudes, E. An access control model for data security in online social networks based on role and user credibility. in *International Symposium on Cyber Security Cryptography and Machine Learning* 156–168 (Springer, Cham, 2019).
9. Xing, L., Ma, Q. & Jiang, L. Microblog user recommendation based on particle swarm optimization. *China Commun.* **14**(5), 134–144 (2017).
10. Islam, M. R., Liu, S., Wang, X. & Xu, G. Deep learning for misinformation detection on online social networks: A survey and new perspectives. *Soc. Netw. Anal. Min.* **10**(1), 1–20 (2020).
11. Li, Z., Zhang, Q., Du, X., Ma, Y. & Wang, S. Social media rumor refutation effectiveness: Evaluation, modelling and enhancement. *Inf. Process. Manag.* **58**(1), 102420 (2021).
12. Qi, L. *et al.* Compatibility-aware web API recommendation for mashup creation via textual description mining. *ACM Trans. Multimed. Comput. Commun. Appl.* **17**(1s), 1–19 (2021).
13. Gong, W. *et al.* Keywords-driven web APIs group recommendation for automatic app service creation process. *Softw. Pract. Exp.* **51**(11), 2337–2354 (2021).
14. Liu, H., Kou, H., Yan, C. & Qi, L. Keywords-driven and popularity-aware paper recommendation based on undirected paper citation graph. *Complexity* https://doi.org/10.1155/2020/2085638 (2020).
15. Qi, L. *et al.* Finding all you need: Web APIs recommendation in web of things through keywords search. *IEEE Trans. Comput. Soc. Syst.* **6**(5), 1063–1072 (2019).
16. Geng, Y., Sui, J. & Zhu, Q. Rumor detection of Sina Weibo based on SDSMOTE and feature selection. in *2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA). IEEE* (2019).
17. Sharma, U. & Kumar, S. Feature-based comparative study of machine learning algorithms for credibility analysis of online social media content. in *Data Engineering for Smart Systems* 13–25 (Springer, Singapore, 2022).
18. Geng, Y., Lin, Z., Fu, P. & Wang, W. Rumor detection on social media: A multi-view model using self-attention mechanism. in *International Conference on Computational Science* 339–352 (Springer, Cham, 2019).
19. Kumar, A., Narapareddy, V. T., Srikanth, V. A., Malapati, A. & Neti, L. B. M. Sarcasm detection using multi-head attention based bidirectional LSTM. *IEEE Access* **8**, 6388–6397 (2020).
20. Li, J., Wang, X., Tu, Z. & Lyu, M. R. On the diversity of multi-head attention. *Neurocomputing* **454**, 14–24 (2021).
21. Sangeetha, K. & Prabha, D. Sentiment analysis of student feedback using multi-head attention fusion model of word and context embedding for LSTM. *J. Ambient Intell. Humaniz. Comput.* **12**(3), 4117–4126 (2021).
22. Khan, Z. N. & Ahmad, J. Attention induced multi-head convolutional neural network for human activity recognition. *Appl. Soft Comput.* **110**, 107671 (2021).
23. Priya, K., Dinakaran, K. & Valarmathie, P. Multilevel sentiment analysis using domain thesaurus. *J. Ambient Intell. Humaniz. Comput.* **12**(5), 5017–5028 (2021).
24. Zhang, Z. *et al.* Twin-incoherent self-expressive locality-adaptive latent dictionary pair learning for classification. *IEEE Trans. Neural Netw. Learn. Syst.* **32**(3), 947–961 (2020).
25. Asghar, M. Z. *et al.* Exploring deep neural networks for rumor detection. *J. Ambient Intell. Humaniz. Comput.* **12**(4), 4315–4333 (2021).
26. Castillo, C., Mendoza, M. & Poblete, B. Information credibility on twitter. in *Proceedings of the 20th International Conference on World Wide Web*, 675–684 (2011).
27. Kumar, A. & Sangwan, S. R. Rumor detection using machine learning techniques on social media. in *International Conference on Innovative Computing and Communications* 213–221 (Springer, Singapore, 2019).
28. Ma, J. *et al.* Detecting rumors from microblogs with recurrent neural networks. in *Proceedings of the 25th International Joint Conference on Artificial Intelligence* 3818–3824 (New York, USA, 2016).
29. Duong, C. T., Nguyen, Q. V. H., Wang, S. & Stantic, B. Provenance-based rumor detection. in *Australasian Database Conference* 125–137 (Springer, Cham, 2017).
30. Chen, W., Zhang, Y., Yeo, C. K., Lau, C. T. & Lee, B. S. Unsupervised rumor detection based on users' behaviors using neural networks. *Pattern Recogn. Lett.* **105**, 226–233 (2018).
31. Torshizi, A. S. & Ghazikhani, A. Automatic Twitter rumor detection based on LSTM classifier. in *International Congress on High-Performance Computing and Big Data Analysis* 291–300 (Springer, Cham, 2019).
32. Smagulova, K. & James, A. P. A survey on LSTM memristive neural network architectures and applications. *Eur. Phys. J. Spec. Top.* **228**(10), 2313–2324 (2019).
33. Tripathi, S., Singh, S. K. & Lee, H. K. An end-to-end breast tumour classification model using context-based patch modelling–a BiLSTM approach for image classification. *Comput. Med. Imaging Graph.* **87**, 101838 (2021).

34. Guo, H., Cao, J., Zhang, Y., Guo, J. & Li, J. Rumor detection with hierarchical social attention network. in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* 943–951 (2018).
35. Qiu, X. *et al.* Pre-trained models for natural language processing: A survey. *Sci. China Technol. Sci.* **63**(10), 1872–1897 (2020).
36. Nandanwar, A. K. & Choudhary, J. Semantic features with contextual knowledge-based web page categorization using the GloVe model and stacked BiLSTM. *Symmetry* **13**(10), 1772 (2021).
37. Singh, M., Jakhar, A. K. & Pandey, S. Sentiment analysis on the impact of coronavirus in social life using the BERT model. *Soc. Netw. Anal. Min.* **11**(1), 1–11 (2021).
38. Wang, Q. *et al.* Extending emotional lexicon for improving the classification accuracy of Chinese film reviews. *Connect. Sci.* **33**(2), 153–172 (2021).
39. Jia, F. & Chen, C. C. Emotional characteristics and time series analysis of Internet public opinion participants based on emotional feature words. *Int. J. Adv. Rob. Syst.* **17**(1), 1729881420904213 (2020).
40. Wang, H. & Zhao, D. Emotion analysis of microblog based on emotion dictionary and Bi-GRU. in *2020 Asia-Pacific Conference on Image Processing, Electronics and Computers. IEEE* 197–200 (2020).
41. Zhang, Y., Sun, J., Meng, L. & Liu, Y. Sentiment analysis of E-commerce text reviews based on sentiment dictionary. in *2020 IEEE International Conference on Artificial Intelligence and Computer Applications. IEEE* 1346–1350 (2020).
42. Zeng, X., Chen, Q., Chen, S. & Zuo, J. Emotion label enhancement via emotion wheel and lexicon. *Chin. J. Comput.* **44**(6), 1080–1094 (2021).
43. Xu, N., Chen, G. & Mao, W. MNRD: A merged neural model for rumor detection in social media. in *2018 International Joint Conference on Neural Networks. IEEE* 1–7 (2018).
44. Ghanem, B., Rosso, P. & Rangel, F. An emotional analysis of false information in social media and news articles. *ACM Trans. Internet Technol. (TOIT)* **20**(2), 1–18 (2020).
45. Wu, Z., Pi, D., Chen, J., Xie, M. & Cao, J. Rumor detection based on propagation graph neural network with attention mechanism. *Expert Syst. Appl.* **158**, 113595 (2020).
46. Fang, Y., Gao, J., Huang, C., Peng, H. & Wu, R. Self multi-head attention-based convolutional neural networks for fake news detection. *PLoS ONE* **14**(9), e0222713 (2019).
47. Ma, T., Lv, S., Huang, L. & Hu, S. HiAM: A hierarchical attention based model for knowledge graph multi-hop reasoning. *Neural Netw.* **143**, 261–270 (2021).
48. Bai, H., Yu, H., Yu, G., Rocha, A. & Huang, X. Analysis on an auto increment detection system of Chinese disaster Weibo text. *JUCS J. Univers. Comput. Sci.* **27**, 230 (2021).
49. Yilmaz, S. & Toklu, S. A deep learning analysis on question classification task using Word2vec representations. *Neural Comput. Appl.* https://doi.org/10.1007/s00521-020-04725-w (2020).
50. Alrubaian, M., Al-Qurishi, M., Hassan, M. M. & Alamri, A. A credibility analysis system for assessing information on twitter. *IEEE Trans. Dependable Secure Comput.* **15**(4), 661–674 (2016).
51. Yang, F., Yu, X., Liu, Y. & Yang, M. Automatic detection of rumor on Sina Weibo. in *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*. 1–7 (2012).
52. Liu, Z., Wei, Z. & Zhang, R. Rumor detection based on convolutional neural network. *J. Comput. Appl.* **37**(11), 3053–3056 (2017).
53. Lai, S., Xu, L., Liu, K. & Zhao, J. Recurrent convolutional neural networks for text classification. in *Twenty-ninth AAAI Conference on Artificial Intelligence* 2267–2279 (2015).
54. Zhou, P. *et al*. Attention-based bidirectional long short-term memory networks for relation classification. in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Vol. 2, 207–212 (2016).
55. Xue, J. *et al.* Detecting fake news by exploring the consistency of multimodal data. *Inf. Process. Manag.* **58**(5), 102610 (2021).
56. Song, C., Ning, N., Zhang, Y. & Wu, B. A multimodal fake news detection model based on crossmodal attention residual and multichannel convolutional neural networks. *Inf. Process. Manag.* **58**(1), 102437 (2021).

Acknowledgements

This work is fully supported by the National Natural Science Foundation of China (62171180, 62071170, 62072158), Henan Province Science Fund for Distinguished Young Scholars (222300420006), the Program for Innovative Research Team in University of Henan Province (21IRTSTHN015), in part by the Key Science and the Research Program in University of Henan Province (21A510001), and the Science and Technology Research Project of Henan Province under Grant (222102210001).

Author contributions

L.X. and J.Y. conceived the experiments. All authors conducted the experiments and analysed the results. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to L.X.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.