Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 2410206, 9 pages
https://doi.org/10.1155/2018/2410206

Research Article

Towards Accurate Deceptive Opinions Detection Based on Word Order-Preserving CNN

Siyuan Zhao,[1] **Zhiwei Xu**,[1,2] **Limin Liu**,[1] **Mengjie Guo**,[1] **and Jing Yun**[1]

[1] *College of Information Engineering, Inner Mongolia University of Technology, Hohhot, Inner Mongolia, China*
[2] *Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China*

Correspondence should be addressed to Zhiwei Xu; xuzhiwei2001@ict.ac.cn

Received 2 February 2018; Revised 24 June 2018; Accepted 6 September 2018; Published 24 September 2018

Academic Editor: Emilio Insfran Pelozo

Convolutional neural network (CNN) has revolutionized the field of natural language processing, which is considerably efficient at semantics analysis that underlies difficult natural language processing problems in a variety of domains. The deceptive opinion detection is an important application of the existing CNN models. The detection mechanism based on CNN models has better self-adaptability and can effectively identify all kinds of deceptive opinions. Online opinions are quite short, varying in their types and content. In order to effectively identify deceptive opinions, we need to comprehensively study the characteristics of deceptive opinions and explore novel characteristics besides the textual semantics and emotional polarity that have been widely used in text analysis. In this paper, we optimize the convolutional neural network model by embedding the word order characteristics in its convolution layer and pooling layer, which makes convolutional neural network more suitable for short text classification and deceptive opinions detection. The TensorFlow-based experiments demonstrate that the proposed detection mechanism achieves more accurate deceptive opinion detection results.

1. Introduction

Artificial neural networks (ANNs) are a well known bio-inspired model that simulates human brain capabilities such as learning and generalization [1, 2]. ANNs consist of a number of interconnected processing units (*i.e.*, neurons), wherein each neuron performs a weighted sum followed by the evaluation of a given activation function [3]. ANNs have the ability of self-learning, associative storage capabilities, and high-speed search for the optimal solution [4]. In recent years, ANNs have been applied in natural language processing, pattern recognition, knowledge engineering, expert systems, etc. Deep learning model is developed from the study of artificial neural networks, which takes high dimensional data to achieve the feature selection and update itself adaptively according to the feedback. Deep learning model can achieve accurate perceptron with multiple hidden layers.

Deceptive opinions detection is an important application of deep learning model. The existence of deceptive opinions makes it difficult for customers who lack relevant experience to obtain accurate judgments of the reviewed products and buy the appropriate products. To detect deceptive opinions effectively, two types of detection mechanisms are proposed. One type of mechanisms is proposed based on the semantic or polarity analysis of online opinions [5], and the other type of mechanisms involves behaviors of users [6, 7] into opinion analysis. The technology, such as support vector machine [8, 9], suffers from its maladaptation to various short opinion texts. CNN is a feed-forward neural network whose artificial neurons can respond to the surrounding cells within a fixed coverage area. Currently, CNN has been used for deceptive opinions detection. However, online opinions are usually quite short and varied in types and content. The existing CNN based approaches cannot adapt to various short texts and detect deceptive opinions with high accuracy.

In order to achieve accurate deceptive opinion detection, we need to adapt all relevant features of deceptive opinions to obtain a comprehensive deep learning model of deceptive opinions detection. To achieve this, we introduce the word order among consecutive words in sentences into the process of the deceptive opinion analysis. In detail, we propose a novel word order-preserving pooling layer, which is embedded in the existing CNN model. In this way, the characteristics of our deceptive opinions model are enriched and we can detect deceptive opinions more effectively.

Contributions. The main contributions are listed as follows:

(1) A novel feature for representing opinion texts, the word order of sentences, is introduced into deceptive opinion detection, and a word order-preserving CNN (OPCNN) model is proposed to keep the word order characteristics during the process of opinion analysis

(2) We implement our deceptive opinion detection model on an open source deep learning platform, TensorFlow, and demonstrate that, compared with basic CNN model, OPCNN can achieve more accurate detection of deceptive opinions

Organization. In Section 2, we analyze the related work. Section 3 gives details of our proposed deceptive opinion detection model, and Section 4 provides the performance evaluation based on TensorFlow. We conclude in Section 5.

2. Preliminary and Related Work

2.1. CNN Model and Its Applications. The neural network model is connected with a large number of neurons to form a complex network system with adaptive and self-learning ability, suitable for dealing with the unclear inherent features of data. As a new type of neural network models, the deep learning model takes high dimensional inherent features among a large amount of data to update itself adaptively and achieve accurate perceptron with multiple hidden layers.

Convolutional neural network (CNN) model is a typical deep learning model with a good fault tolerance, parallel processing, and self-learning ability [10]. CNN has been widely used in image processing, speech recognition, and natural language processing. Compared with other popular deep learning models [11], CNN is more accurate in natural language processing and is more efficient to achieve training results.

2.2. Related Work. There are a lot of researches on deceptive opinion detection due to its important application role in Internet economies. Jindal and Liu [12] studied deceptive opinion problems and train models based on the features of opinion texts. Myle Ott et al. [13] created a benchmark data set for evaluating detection performance. Fei et al. [5] studied sudden bursts of the number of opinion replies and discovered that these bursts are caused by the sudden popularity of partiproducts or by a sudden invasion of a large number of fake opinions. Belief Propagation (BP) was used to infer whether a user is a fake user or not. Wang et al. [6] proposed an innovative heterogeneous opinion graph model to capture the relationship between the users and users' opinions in the store. They used the interaction and the role of the nodes in the figure to reveal the causes of deceptive opinion. Then they designed an iterative algorithm to identify deceptive opinions. Mukherjee [7] et al. found that more than 70% of deceptive opinion publishers issued opinions similarity greater than 0.3, and real opinion publishers published opinions similarity less than 0.18 in the Yelp data set. The content similarity calculation for the opinions made by the same commentator can reflect the characteristics of the opinion's behavior. Hernndez et al. [8] used PU-learning to detect deceptive opinions. PU-learning is a semisupervised technique for building a binary classifier on the basis of positive and unlabeled examples only. The paper gets an average improvement of 8.2% and 1.6% over the original approach in the detection of positive and negative deceptive opinions, respectively. Cagnina et al. [14] proposed the character n-grams in tokens to obtain a low dimensionality representation of opinions and used support vector machines classifier to evaluate their proposal on available corpora with reviews of hotels, doctors, and restaurants. The result shows that using character n-grams in tokens allows obtaining competitive results with a low dimensionality representation.

There have been some studies using deep learning models to identify deceptive opinions. Raymond's [9] team built a semantic language model to identify semantic repetitive opinions and made deceptive opinions detection. However, because the opinion itself has a certain degree of semantic similarity and content on the repeatability, there may be a miscarriage of justice. Li et al. [15] took the word vector as input, with CNN; the emotional polarity feature can also be applied to unsupervised methods for deceptive opinions text detection. However, only considering the emotional polarity of the deceptive opinion on the identification is not sufficient. At the same time, the local sampling of the CNN model cannot take into account the existence of the word order in the text. Jindal [16] thought that the same user that gives his all positive opinions or negative opinions to the same brand of products is a kind of abnormal behavior and the corresponding opinion may be deceptive opinion. The researchers proposed a "one-condition rules" and a "two-condition rules" to predict the falseness of the text by probabilistic prediction. Yapeng Jing [17] set the data set on the AMT of hotel opinions, used the information gained to select the feature of the word bag, and then detected deceptive opinions through the ordinary neural network, DBN-DNN network, and LBP network. However, the artificial data set cannot accurately reflect the true opinions. Ma et al. [18] presented a novel method that learns continuous representations of microblog events for identifying rumors such as deceptive information. The proposed model is based on recurrent neural networks (RNN) such as Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) for learning the hidden representations to capture the variation of contextual information of relevant posts over time. However, the detection performance of this model suffers from the limitation of its feature collection range.

Deceptive opinions detection is a type of complex text classification. The deceptive opinion is very short and varied in type and content. In order to effectively identify deceptive opinions, we need to find additional characteristics of deceptive opinions besides the textual semantics and emotional polarity that have been widely used in text analysis. To achieve this, we introduce the word order characteristics into the CNN model and design word order-preserving k-max pooling technology to preserve the word order characteristics of deceptive opinions in the deceptive opinions detection process.

3. Deceptive Opinion Detection Model

To achieve an accurate deceptive opinion detection mechanism, we study the word order characteristic of online opinion texts. In addition, we design a word order-preserving CNN network to model various short opinion texts. In this way, we involve a foundational textual characteristic into deceptive opinion detection process and obtain more accurate detection results.

3.1. Chinese Word Order. For most of languages, such as English and Chinese, the word order is an inherent type of features in texts. Almost all languages have their word ordering of the subject (S), object (O), and the verb (V), and among the languages all over the world, all six possible basic word orders exist [19], especially SVO (Subject-Verb-Object) and SOV (Subject-Object-Verb). The study has shown that the earliest human language had rigid word order. Nowadays, SOV basic word order is common among the languages all over the world, and many other word orders can be reconstructed back to an SOV stage. Firstly, it can be concluded that SOV must have been the word order of the "ancestral language" among the six possible word orders [20, 21]. In addition, there are researches to demonstrate that, besides SOV, SVO is such a prominent word order in the languages all over the world. For example, in a sentence like "fireman kicks boy," both nouns could in principle be the agents. SVO is used to avoid expressing two plausible agents ("fireman" and "boy") at the same side of the verb instead of SOV [22]. In this paper, we use the word order feature to optimize CNN (convolutional neural network) model for deceptive opinion detection.

3.2. Order-Preserving CNN Model. CNN model includes input layer, convolution layer, pooling layer, and output layer. We proposed an improved CNN model considering the word order characteristics of consecutive words in sentences. The input layer takes the opinions with a certain word order as input values. In the convolution layer and the pooling layer, we preserve the word order of the consecutive words in the inputted sentences and apply the word order persevering pooling method instead of the original pooling layer, as shown in Figure 1. Ultimately, we propose a novel type of CNN model, order-perserving CNN model (OPCNN). The detailed model is illustrated in Figure 2.

3.2.1. Input Layer. We use word vectors to represent the word frequency of each word [23] and take them as the training inputs of our model. We use the word2vec model to predict words that appear in the context. The input layer consists of an $n \times m$ two-dimensional matrix, where n is the length of the sentence and m is the dimension of word vectors. The text representation process can be formulated as (1), where a represents the matrix, w represents word vector, and v represents a specific word vector. Ultimately, each opinion is represented by a two-dimensional word vector matrix.

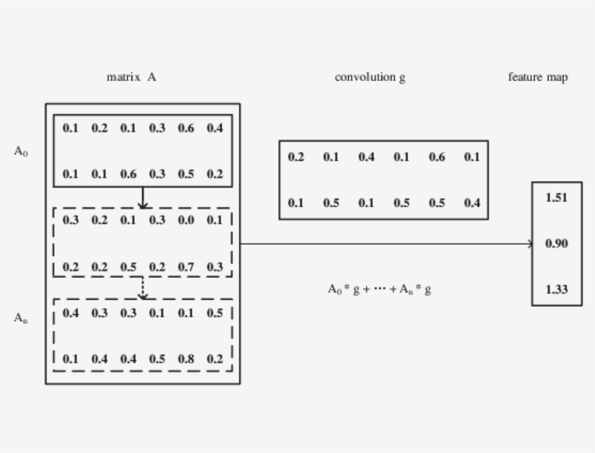


Figure 1: Convolution operation.

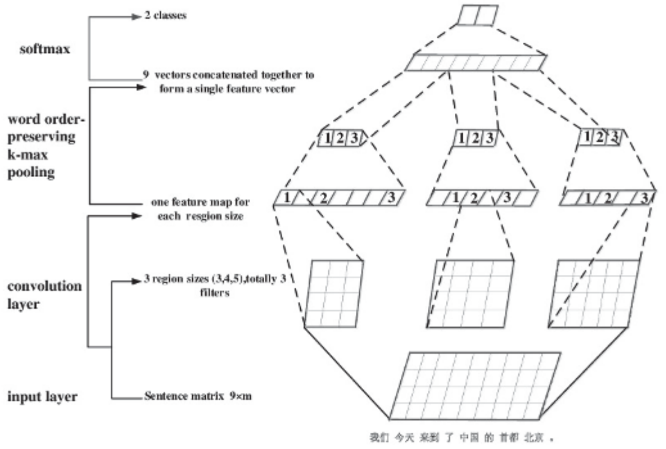
$$A = \left(\vec{w}_1, \vec{w}_2, \ldots, \vec{w}_n\right)^T$$
$$\vec{w}_i = \left(v_1, v_2, \ldots, v_m\right) \tag{1}$$

3.2.2. Convolution Layer. The input layer transfers the word vector matrix A to the convolution layer for convolution operations. The padding types of convolution operations include "same" and "valid." As is shown in (2), we perform the i-th convolution in the l-layer K_i^l on matrix A, taking the ReLU function as activation function. The bias b_i^l is the valid padding type of convolution operations, and the matrix a_i^l is a feature map operation. The size of the convolution window is $h \times m$, where h is the width of the convolution kernel and m is the dimension of vectors. The width of the convolution kernel, h, needs to be adjusted dynamically, as shown in Figure 1.

$$a_i^l = \sigma\left(\text{conv2}\left(A, K_i^l, valid\right) + b_i^l\right) \tag{2}$$

Each inputted word vector in the convolution window is converted to an eigenvalue by the nonlinear transformation of the CNN model. As the window moves down, the corresponding eigenvalue of the convolution kernel is generated and the corresponding eigenvector is obtained.

3.2.3. Word Order Persevering Pooling Layer. In pooling layer, the eigenvalues are selected to represent the features of the inputted word vectors. The position of each word in a word vector is a very important feature in the text analysis, so it is necessary to preserve the word order in the pooling layer. To preserve the word order of the inputted sentences, We replace the original max pooling method in the CNN model with the word order-preserving k-max pooling method. The order persevering pooling layer reduces the number of feature words, while preserving the order of these words. We use the



2 classes
softmax
9 vectors concatenated together to form a single feature vector
word order-preserving k-max pooling
1 2 3
1 2 3
one feature map for each resgion size
convolution layer
3 region sizes (3,4,5),totally 3 filters
input layer
Sentence matrix 9×m
我们 今天 来到 了 中国 的 首都 北京 。

Figure 2: The framework of OPCNN model.

order persevering pooling layer to preserve the word order of the inputted sentences.

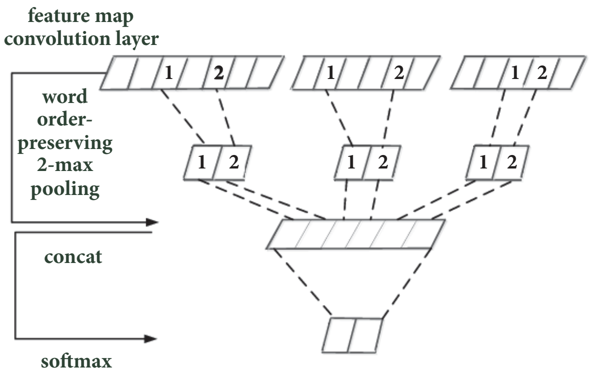
$$a_i^l = \sigma\left(\beta_i^l \times pooling\left(a_i^{l-1}, k\right) + b_i^l\right) \tag{3}$$

In (3), we use the order-persevering k-max pooling method to handle the result of the *l-1*-th convolution layer, a_i^{l-1}. In detail, the proposed pooling layer selects the k-max values from the one-dimensional features obtained from the previous convolution layer and discards the other eigenvalues.

As shown in Figure 3, the order-preserving k-max pooling method selects the k-max values in the sequence s. The order of the selected values remains the same in s. The word order-preserving k-max pooling method can discern more finely the number of times that the feature is highly activated in s [24]. In addition, the method can also distinguish the progression by which the features with high activations will cover the sequence s. Finally, we can obtain the k-max features in the sequence s.

3.2.4. Output Layer. We concat the obtained features from the pooling layer. It is a two-classification problem which distinguishes the deceptive opinions from real opinions. The result of the concat function is transmitted into the softmax function to evaluate the probability that this opinion is deceptive. Finally, the proposed model uses cross entropy as the loss function of the model to measure the difference between the deceptive opinions and the honest opinions.

3.3. Deceptive Opinions Detection Algorithm. To detect deceptive opinions with OPCNN model, we first construct the word vector model and preprocess the experimental data and then take OPCNN model to detect deceptive opinions online. The deceptive opinions detection process is depicted in Algorithms 1 and 2 . Specifically, we train the OPCNN model according to Algorithm 1 and detect deceptive opinions with Algorithm 2.



feature map convolution layer
1 2
1 2
1 2
word order-preserving 2-max pooling
1 2
1 2
1 2
concat
softmax

Figure 3: Word order persevering pooling layer.

Complexity analysis: assume that the number of iterations is k, the number of samples per input sentence is m, the number of words in a sentence is v, the word vector dimension is d, the convolution window size is w, and the number of output channels is n. The model tackles inputted sentences with a time complexity of O(v∗n∗(2d∗w^2+w-1)). Therefore, the time complexity of the OPCNN model can be expressed as O(w^2∗k∗m∗n∗d∗v).

4. Experiments

4.1. Experimental Data Set. To evaluate the performance of our deceptive opinion detection scheme, we construct a data set by collecting 24,166 online opinions about hotels and annotating these opinions of hotels according to the data annotation method presented by Li [25]. Among them, 4132 opinions are deceptive, as depicted in Table 1. 21750 opinions of this data set are used as the training set, and the others are used as the test set.

In detail, we first check whether the opinion is related to the hotel services or not, and if there is no relevance, the opinion is a deceptive opinion. In addition, we give

```
Input:train dataset $\{(x_n, t_n)\}_{n=1}^{N}$,the structure of CNN $\{h_l\}_{l=1}^{L}$
Output:The parameters of CNN
for $i < L$ do
  if $h_l$ is convolution layer Algorithm then
    for $\forall i$ do
      $a_i^l = \sigma(\mathrm{conv}\,2(A, K_i^1, valid) + b_i^l)$
    end for
  end if
  if $h_l$ is pooling layer then
    for $\forall i$ do
      $a_i^l = \sigma(\beta_i^l \times pooling(a_i^{l-1}, k) + b_i^l)$
    end for
  end if
  if $h_l$ is connection layer then
    $a_i^l = \sigma(z^l)$
  end if
end for
```

Algorithm 1: Forward Propagation Algorithm.

```
Input:The parameters of CNN,train dataset $\{(x_n, t_n)\}_{n=1}^{N}$,the structure of CNN $\{h_l\}_{l=1}^{L}$
Output:Response error
for $i < L$ do
  if $h_l$ is convolution layer then
    for $\forall i, j$ do
      $\frac{\partial J}{\partial K_{ij}^{l}} = \sum_{st} \left(\delta^{(l)}\right)_{st} \left(P_j^{(l-1)}\right)_{st}$
    end for
    for $\forall i$ do
      $\frac{\partial J}{\partial b_i^{(l)}} = \sum_{st} (\delta_i)_{st}$
    end for
  end if
  if $h_l$ is pooling layer then
    for $\forall i$ do
      $\frac{\partial J}{\partial b_i^{(l)}} = \sum_{st} (\delta_i)_{st}$
    end for
    for $\forall i$ do
      $\frac{\partial J}{\partial \beta_i^{(l)}} = \sum_{s,t} (\delta_i^{(l)} \circ d_i^{(l-1)})_{st}$
    end for
  end if
  if $h_l$ is connection layer then
    $\frac{\partial J}{\partial w^{(l)}} = \delta^{(l)} \left(a^{(l-1)}\right)^T$
    $\frac{\partial J}{\partial b^{(l)}} = \delta^{(l)}$
  end if
end for
```

Algorithm 2: Back Propagation Algorithm.

Table 1: The data used in the experiment.

Opinion type	#Opinions	Average length
Deceptive opinion	4132	78
Positive opinion	20034	112
All opinions	24166	106

Table 2: The type of the OTT data set.

Opinion type	P & H	P & D	N & H	N & D
Number	400	400	400	400

the opinions with strong emotional polarity an annotation of deceive opinions. Finally, opinions that contain a large number of negative words are marked as deceptive opinions [26, 27].

In addtion, we use a standard data set, the OTT data set [13], to evaluate the generalization capability of the proposed mechanism. The OTT data set includes 1600 opinions which are divided into four types equally, positive honest opinions (P & H), positive deceptive opinions (P & D), negative honest opinions (N & H), and negative deceptive opinions (N & D), as shown in Table 2.

4.2. Implementation. TensorFlow is an open source platform to implement deep learning model in practice. It is the second-generation artificial intelligence learning system based on DistBelief. In order to evaluate the performance of the proposed detection scheme, we implement the proposed detection scheme on TensorFlow.

In detail, the opinion data and the word vector processed by the word2vec tool are taken as input. We use a function, embedding_lookup(), to find the index of each word in a sentence and generate the corresponding word vector. In this way, the input layer can obtain a word vector matrix of all sentences in an opinion text. In the convolution layer, textual feature extraction is performed by means of convolution operation in the TensorFlow framework, conv2d(). In the pooling layer, the word order-preserving K-max pooling method is used instead of the max_pool() method of the TensorFlow framework to perform dimension reduction on the feature maps. We concat the features obtained from the pooling layer. Additionally, the dropout() method of TensorFlow is used to prevent the problem of overfitting. In the output layer, Softmax logistic regression function is used to identify deceptive opinions.

In order to evaluate the performance of our scheme extensively, three baseline schemes are implemented as follows:

(1) The first experimental baseline is based on the classical supports vector machine (SVM) [28], which uses a statistical method called TF-IDF for feature collection

(2) The second baseline is also based on support vector machine (SVM); however, Bi-Gram is used to collect text features [29]

(3) The third baseline uses the convolutional neural network (CNN) to detect deceptive opinions [30]. It is also implemented on TensorFlow. The hyper-parameters of this model are shown in Table 3.

4.3. Evaluation Metrics. We evaluate the proposed detection scheme by accuracy, precision, recall, f1-measure, and accuracy gain. True positives (TP), false positives (FP), false negatives (FN), and true negatives (TN) are also calculated for evaluating detection accuracy.

Accuracy (A) is calculated as the ratio of the number of opinions correctly classified by the classifier to the total number of opinions.

$$A = \frac{TP + TN}{TP + FP + TN + FN} \tag{4}$$

Precision (P) is calculated as the ratio of all "correctly classified opinions (TP)" to all "actually classified opinions (TP + FP)."

$$P = \frac{TP}{TP + FP} \tag{5}$$

Recall (R) is the ratio of all "opinions correctly classified (TP)" to all "opinions that should be classified as deceptive opinions (TP+FN)."

$$R = \frac{TP}{TP + FN} \tag{6}$$

F1-measure is the harmonic mean of precision and recall.

$$F_1 = \frac{2PR}{P + R} = \frac{2TP}{2TP + FP + FN} \tag{7}$$

Accuracy gain (α) is calculated as the ratio of A_e to A_c, where A_e represents the accuracy of OPCNN model and A_c represents the accuracy of tf-idf+svm, bigram+svm, and CNN, respectively. When the value of α is larger, the accuracy of the OPCNN model is better than that of other baselines.

$$\alpha = \frac{A_e}{A_c} \tag{8}$$

4.4. Analysis of Results.

4.5. Effect of Traning Set Size. In order to evaluate the detection performance with the training sets with different size, we change the size of the deceptive opinion training set. The accuracy gain(α) is used to evaluate the accuracy increment when the training data set is replaced by a larger one. As depicted in Figure 4, the accuracy gain relative to CNN increases as the other two when the training data is larger than 4000. A larger training data can improve the detection accuracy of the proposed detection model.

4.5.1. Word Vector Dimension Selection. In the paper, the word2vec model is used to generate word vector. The dimension of the word vector is related to the convolution kernel width of OPCNN and CNN. Therefore, the dimension of the word vector is an important factor to the performance of OPCNN and CNN. We evaluate the detection accuracy in terms of different dimensions of the inputted word vectors.

As depicted in Figure 5, the detection accuracy of the proposed detection scheme and the CNN based baseline shows the small fluctuations and achieves the best detection accuracy when the dimension of word vectors is equal to 100.

Table 3: Hyper-parameter setting.

Hyper-parameter	Description	Value
d	word vector dimension	100
d_{min}	convolution width	3,4,5
H	number of convolutions	64,64,64
L2	L2 regularization lambda	0.5
dropout	dropout keep probability	0.5
mini-batch	mini-batch gradient descent	50

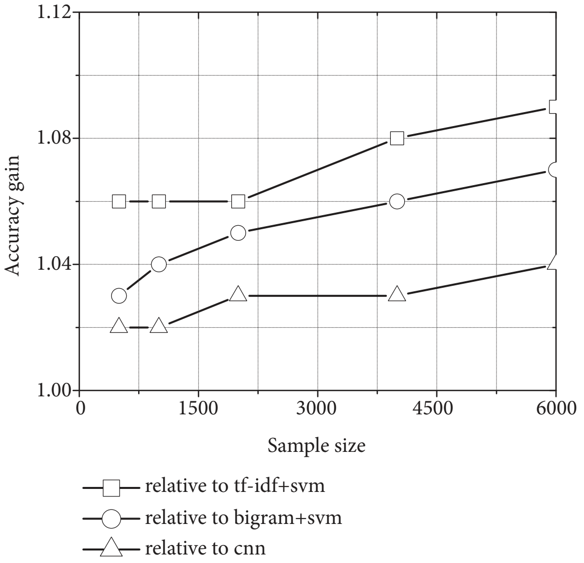


Figure 4: The result of accuracy gain(α) in different training set.

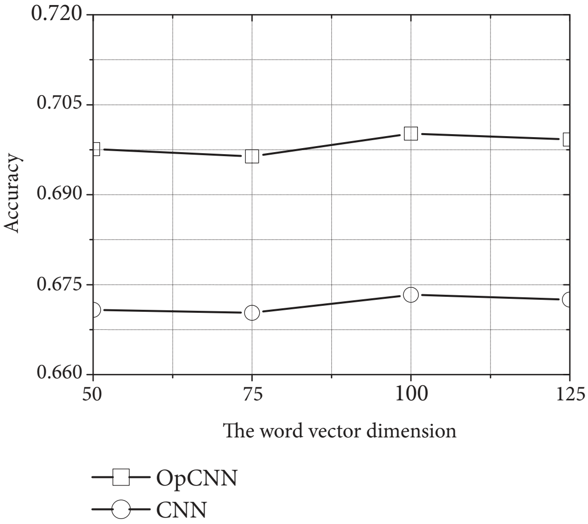


Figure 5: The effect of the word vector dimension on experimental results.

4.5.2. Effectiveness of K. In OPCNN, the order-preserved k-max pooling method is used in the pooling layer instead of the original max pooling method. The value of k is particularly important for improving the detection accuracy of OPCNN. As shown in Figure 6, the best detection accuracy is obtained when k is equal to 3. When the value of k is small, only a part of primary eigenvalues will be selected and the others will be lost. When the value of k is too large, the noise data will be involved in the detection model and affect the accuracy of the detection model. The detection accuracy in terms of different k for the OPCNN model is evaluated.

4.5.3. Accuracy Analysis. As shown in Table 4, the OPCNN method achieves best detection performance. The accuracy of OPCNN is equal to 70.02%, the recall of OPCNN is equal to 66.83%, and the F1 of OPCNN is equal to 69.76%. Compared to the other two baselines, CNN can collect more textual features cached to the surrounding input cells and thus improve detection accuracy. OPCNN enriches the deceptive opinion features by embedding the word order-preserving k-max pooling layer and thus improving the accuracy of deceptive opinion detection on the basis of the CNN-based baseline.

4.5.4. Generalization Capability Analysis. The generalization capability of the proposed detection method is evaluated on the Ott data set. As shown in Table 5, OPCNN obtains better detection performance in comparation with CNN based baseline. The accuracy, recall, and F1 have been improved by more than 2%.

It can be seen that the detection performance of the OPCNN is better than the CNN based baseline, no matter what data set is used. The performance comparison among the two SVM-based baselines and the CNN-based baseline has been discussed. The CNN-based baseline has better detection performance than the other two baselines [30]. Therefore, the proposed scheme has the best detection performance among all the four detection schemes, including the two SVM-based baselines and the CNN-based baseline. The generalization ability of the proposed detection scheme is quite good.

5. Conclusion

In this paper, an optimized CNN model is proposed to identify deceptive opinions. Considering the characteristics of short opinion texts, we introduce the text order into the deceptive opinion detection and extend the scope of the features of deceptive opinions. In order to preserve the text order while detecting online deceptive opinions, this paper

Table 4: Effect of OPCNN.

Experimental method	Accuracy	Recall	F1-measure
tf-idf+svm	64.53%	63.18%	64.42%
Bigram+svm	66.27%	64.13%	65.85%
CNN	67.33%	64.79%	67.20%
OPCNN	70.02%	66.83%	69.76%

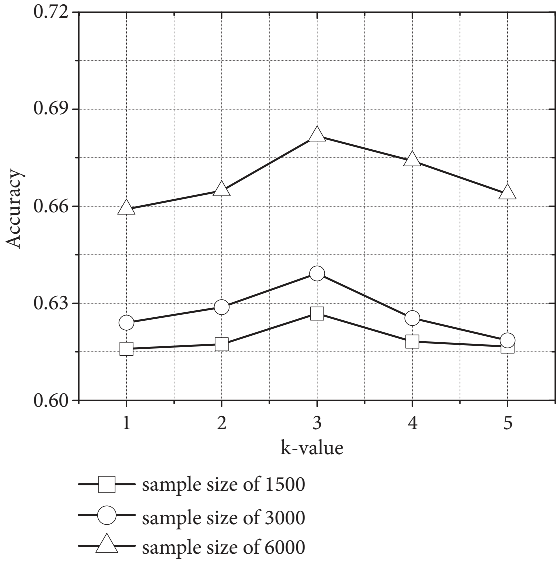


Figure 6: In the top-k method, the effect of k value on experimental results.

Table 5: Generalization ability analysis.

Experimental method	Accuracy	Recall	F1-measure
CNN	82.04%	78.20%	80.07%
OPCNN	84.50%	81.03%	82.84%

proposes an order-preserving k-max pooling operation for CNN model. The extensive experiments demonstrate that the proposed detection model can improve the detection accuracy for online deceptive opinions.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Natural Science Foundation of China (61540004, 61502255, and 61650205) and by Natural Science Foundation of Inner Mongolia Autonomous Region (2017MS(LH)0601 and 2018MS06003).

References

[1] C. L. Black, “Method And System For Training An Artificial Neural Network,” in *Proceedings of the IEEE International Conference on Systems, Man, & Cybernetics*, pp. 347–352, 2001.

[2] F. Pulvermüller and A. Knoblauch, “Discrete combinatorial circuits emerging in neural networks: A mechanism for rules of grammar in the human brain?” *Neural Networks*, vol. 22, no. 2, pp. 161–172, 2009.

[3] A. Knoblauch, “Neural associative memories based on optimal bayesian learning,” 2013.

[4] D. D’Amore and V. Piuri, “Behavioral simulation of artificial neural networks: A general approach for digital and analog implementation,” in *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol. 3, pp. 1327–1330, IEEE, 2002.

[5] G. Fei, A. Mukherjee, B. Liu, M. Hsu, M. Castellanos, and R. Ghosh, “Exploiting burstiness in reviews for review spammer detection,” 2013.

[6] P. Wang, J. Xu, B. Xu et al., “Semantic Clustering and Convolutional Neural Network for Short Text Categorization,” *Digital Content Technology and Service Research Center*, pp. 352–357, 2015.

[7] A. Mukherjee, V. Venkataraman, B. Liu, and N. Glance, “What yelp fake review filter might be doing?” 2013.

[8] P. Rosso, “Detecting positive and negative deceptive opinions using PU-learning,” *Information Processing Management*, vol. 51, no. 4, pp. 433–443, 2015.

[9] R. Y. Lau, S. Y. Liao, R. C. Kwok, K. Xu, Y. Xia, and Y. Li, “Text mining and probabilistic language modeling for online review spam detection,” *ACM Transactions on Management Information Systems (TMIS)*, vol. 2, no. 4, pp. 1–30, 2011.

[10] X. Wei, J. D. Zhang, and X. M. Qi, “Study for workpiece identity based on the BP neural network and the Zernike moment,” *Advanced Materials Research*, pp. 694–697, 2013.

[11] G. Arevian and C. Panchev, “Optimising the hystereses of a two context layer RNN for text classification,” in *Proceedings of the International Joint Conference on Neural Networks*, pp. 2936–2941, 2007.

[12] N. Jindal and B. Liu, “Analyzing and detecting review spam,” *IEEE Computer Society*, pp. 547–552, 2007.

[13] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock, “Finding deceptive opinion spam by any stretch of the imagination,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, pp. 309–319, Association for Computational Linguistics, 2011.

[14] L. C. Cagnina and P. Rosso, “Detecting Deceptive Opinions: Intra and Cross-Domain Classification Using an Efficient Representation,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 25, 2, pp. 151–174, 2017.

[15] L. Li, W. Ren, B. Qin, and T. Liu, “Learning Document Representation for Deceptive Opinion Spam Detection,” in *Chinese*

Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data, Springer International Publishing, 2015.

[16] N. Jindal, B. Liu, and E.-P. Lim, "Finding unusual review patterns using unexpected rules," in *Proceedings of the ACM International Conference on Information and Knowledge Management*, pp. 1549–1552, 2010.

[17] Y. Jing, *Research of Deceptive Opinion Spam Recognition Based on Deep Learning*, East China Normal University, Shagnhai, China, 2014.

[18] J. Ma, W. Gao, P. Mitra et al., "Detecting Rumors from Microblogs with Recurrent Neural Networks," in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 1–10, 2016.

[19] M. S. Dryer, "Order of subject and verb," 2005.

[20] F. J. Newmeyer, "On the reconstruction of proto-world word order," *Evolutionary Emergence of Language*, 2000.

[21] M. Gell-Mann and M. Ruhlen, "The origin and evolution of word order," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 108, no. 42, pp. 17290–17295, 2011.

[22] E. Gibson, S. T. Piantadosi, K. Brink, L. Bergen, E. Lim, and R. Saxe, "A Noisy-Channel Account of Crosslinguistic Word-Order Variation," *Psychological Science*, vol. 24, no. 7, pp. 1079–1088, 2013.

[23] R. Johnson and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks," 2015.

[24] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A Convolutional Neural Network for Modelling Sentences," 2014.

[25] F. Li, M. Huang, Y. Yang, and X. Zhu, "Learning to identify review spam," in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 2488–2493, 2011.

[26] M. Crawford, T. M. Khoshgoftaar, J. D. Prusa, A. N. Richter, and H. Al Najada, "Survey of review spam detection using machine learning techniques," *Journal of Big Data*, vol. 2, no. 1, article no. 23, 2015.

[27] Y. Ren, *Identifying deceptive reviews based on labeled and unlabeled data*, Wuhan University, 2015.

[28] S. M. Asadullah and S. Viraktamath, "Classification of Twitter Spam Based on Profile and Message Model Using Svm," 2017.

[29] S. Seneviratne, A. Seneviratne, M. A. Kaafar, A. Mahanti, and P. Mohapatra, "Spam mobile apps: Characteristics, detection, and in the wild analysis," *ACM Transactions on the Web (TWEB)*, vol. 11, no. 1, pp. 1–29, 2017.

[30] H. Huang, Y. Wen, and H. Chen, "Detection of False Online Advertisements with DCNN," in *Proceedings of the International Conference on World Wide Web Companion. International World Wide Web Conferences Steering Committee*, pp. 795-796, 2017.

Advances in Operations Research
Advances in Decision Sciences
Journal of Applied Mathematics
The Scientific World Journal
Journal of Probability and Statistics
International Journal of Mathematics and Mathematical Sciences
Journal of Optimization
International Journal of Engineering Mathematics
International Journal of Analysis
Journal of Complex Analysis
Advances in Numerical Analysis
Mathematical Problems in Engineering
International Journal of Differential Equations
Discrete Dynamics in Nature and Society
International Journal of Stochastic Analysis
Journal of Mathematics
Journal of Function Spaces
Abstract and Applied Analysis
Advances in Mathematical Physics