Application of Convolutional Neural Network Based on Paragraph2vec in Text Sentiment Analysis

Jian Di [a], Jing Sang

School of Control and Computer Engineering, North China Electric Power University, Baoding 071003, China.

[a]1216446079@qq.com

Abstract

With the exploration of deep learning in the field of natural language processing,the word vector technology represented by word2vec and the neural network model represented by CNN and LSTM have been widely used in text sentiment analysis. In the sentence-level text sentiment analysis task,the traditional convolutional neural network lacks of effective attention to word order information,and ignores the influence of word order information on the accuracy of sentiment analysis.In order to solve this problem,this paper uses Paragraph2vec to obtain the paragraph vector of the sentence,which contains important information such as the subject and word order of the sentence.The paragraph vector was connected with the word vector trained by the skip-gram model to get the sentence vector as the input to the convolutional neural network.Experiments show that in the sentence-level text sentiment analysis, the model has a further improvement in accuracy compared to the benchmark models.

Keywords

Text sentiment analysis;Word embedding technique;Paragraph2vec;Convolutional neural network.

1. Introduction

Sentiment analysis, as one of the main tasks of natural language processing, also known as opinion mining,is the analysis of people's opinions, emotions, assessments, attitudes, and evaluations on products, problems, events, topics, and so on [1].With the deepening of sentiment analysis research, its research methods gradually form three mainstream methods: methods based on emotional dictionary and rule, machine learning and deep learning. The first two methods rely on manual design, and the results are deeply influenced by artificial prior knowledge, lacking wide adaptability, and not easy to promote. However, deep learning has no such drawbacks, and with deep learning shine in terms of image, audio, video and so on, more and more scholars have migrated it to the study of natural language processing, and achieved good results. Therefore, deep learning has become a popular research method for solving text sentiment analysis.

There are two important parts to using deep learning for sentiment analysis: word vector and neural network classification model.Word vectors can be divided into discrete representations and distributed representations according to the coding method.Discrete representations have the disadvantages of lack of word order, space consumption, and lack of inter-word relations. Therefore, the word vectors mentioned in the deep learning method mostly refer to distributed representation.Word2vec is one of the most widely used word vector learning tools proposed by Google's Mikolov in 2013 [2]. Word2vec uses a neural network to train the language model of thinking, which is essentially a simplified neural network to remove the hidden layer.However, Word2vec ignores the influence of word order on sentiment analysis, so Quoc V.Le and T.Mikolov

proposed a method for processing variable length text to improve this problem, which was called Paragraph2vec[3]. The only difference between Paragraph2vec and Word2vec is that the paragraph vector is added. The paragraph vector is shared in the whole paragraph. It records the missing information in the current context, which solves the problem of lack of word order information in the word2vec, while retaining the advantages of word2vec.In the choice of neural network model, RNN and CNN are two models commonly used in sentiment analysis. RNN is the earliest model to break through in sentiment analysis tasks. RNN is a continuous structure that can preserve the historical information of sentences. However, in sentiment analysis tasks, phrases with obvious emotional colors have a critical impact on classification results. The CNN model that extracts local features is better at solving such tasks.

This paper uses Paragraph2vec-based convolutional neural network model to solve the problem of sentence-level text sentiment analysis. Firstly, we divide the corpus into sentences, input the Paragraph2vec model and obtain the corresponding word vector and paragraph vector by training.Secondly, the word vector and the paragraph vector are connected to form a vector of sentences, and then input into the convolutional neural network model for training to obtain a deeper sentence feature vector, which is finally input into the classifier for classification.

2. Main Models

2.1 Paragraph2vec

Paragraph2vec is an unsupervised model that can learn variable-length continuous vector representations of sentences, paragraphs, and documents. This model can obtain fixed-length paragraph vectors and word vectors.The paragraph vector stores the subject of the current paragraph and the context information that is missing from the word vector.The core of Paragraph2vec is two models: PV-DM (Distributed Memory Model of Paragraph Vectors) and PV-DBOW (Distributed Bag of Words version of Paragraph Vector).The PV-DM model will be highlighted below.

The role of the PV-DM model is to predict the next word in the context by a given paragraph vector and the context word vector in this paragraph. The model is a simplified neural probabilistic language model with the hidden layer removed, leaving only the input layer, mapping layer, and output layer.the framework shown in Fig. 1.

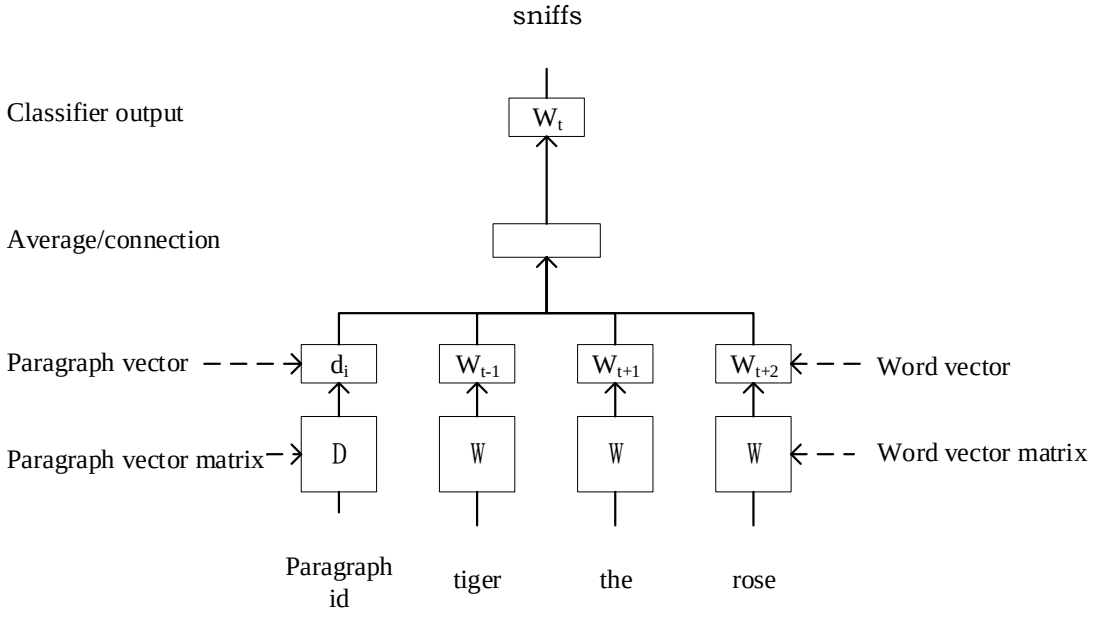

Fig.1 PV-DM model framework

In the model training phase, for a given training set P, there are sentences $X\{x_1, x_2, ..., x_N\}$ and words $Y\{y_1, y_2, ..., y_M\}$.For any sentence x_i, using a fixed-size window to slide over the words of the current sentence. Each time you slide to a position, using the paragraph vector d_i of the current sentence and the context word vector $(w_{t-k}, ..., w_{t+k})$ in the window to predict the word w_t that will appear at the maximum probability at that position. The PV-DM usually obtains a combined vector h by averaging or concatenating the paragraph vector d_i and the context word vector $(w_{t-k}, ..., w_{t+k})$, and then predicting it by the softmax function of the output layer.The Softmax function is often accelerated using the Hierarchical Softmax or Negative Sampling method [12] . Since the paragraph vector d_i is fixed during the training, the semantic information of the entire sentence is utilized every time the word is predicted. At this stage, the paragraph vector matrix D of the training set, the word vector matrix W, and the parameters U, b of softmax are obtained.

In the inference phase of the model, for the new sentence, using the trained PV-DM model, the word vector matrix W and the parameters U, b are fixed, and the new paragraph vector is obtained by the gradient descent method. Only the paragraph vector matrix D is updated at this stage.

The task of the PV-DM model is to maximize the average log-likelihood function:

$$\frac{1}{M}\sum_{t=k}^{M-k}\log p(w_t \mid d_i, w_{t-k}, ..., w_{t+k}) \tag{1}$$

Where M is the number of words, k is the size of the training window, and d_i is the paragraph vector of the sentence in which the context word in the current window is located. For the formula (1), the key is the conditional probability function,there has:

$$p(w_t \mid d_i, w_{t-k}, ..., w_{t+k}) = \frac{e^{z_{w_t}}}{\sum_j e^{z_j}} \tag{2}$$

Where z_j is the non-normalized logarithm probability of the output word j, which is calculated as:

$$z = b + Uh(d_i, w_{t-k}, ..., w_{t+k}; W, D) \tag{3}$$

Where b, U are parameters of the softmax function, and h is formed by averaging or concatenating vectors extracted from the word vector matrix W and the paragraph vector matrix D.

In the experiments in this paper, the paragraph vector was trained by the PV-DM model.

2.2 Convolutional Neural Network Based on Paragraph2vec

The convolutional neural network is a feedforward neural network, which differs from the traditional neural network in its unique feature extractor composed of convolutional layer and Pooling layer.In recent years, CNN has made great progress in speech recognition, image recognition, target tracking, natural language processing, etc. In natural language processing, CNN has short-distance dependence characteristics, and its convolutional layer extracts enough local features to obtain the overall semantic information of the sentence, while the Pooling layer can keep the features obtained by the convolutional layer unchanged, so it has certain advantages in dealing with sentence-level text sentiment analysis.

The convolutional neural network model used in this paper is shown in Fig.2. This model has four main parts: the representation of sentences; convolution; Max Pooling and Softmax classification.

2.2.1 The Representation of Sentences

This model is used to solve the sentence-level text sentiment analysis, so it is necessary to input by sentence.With a sentence S, S is a matrix consisting of word vectors of all the words of the sentence and paragraph vector of the sentence, in which the word vector is learned by the Skip-gram model.Let the dimension of the word vector be d, w_i be the d-dimensional word vector of the i-th word in

the sentence, p_{PV} be the paragraph vector of the sentence, then the sentence S of length l (containing l word) will expressed as:

$$S = w_1 \oplus w_2 \oplus ... \oplus w_l \oplus p_{PV} \tag{4}$$

Where $\oplus$ is a splicing operation, so the sentence matrix consists of vectors $S \in R^{d\times(l+1)}$.

This layer differs from the traditional input layer in that the word vector and the paragraph vector are spliced together to form a representation of the sentence, so that the word order information that was originally missing can be supplemented.

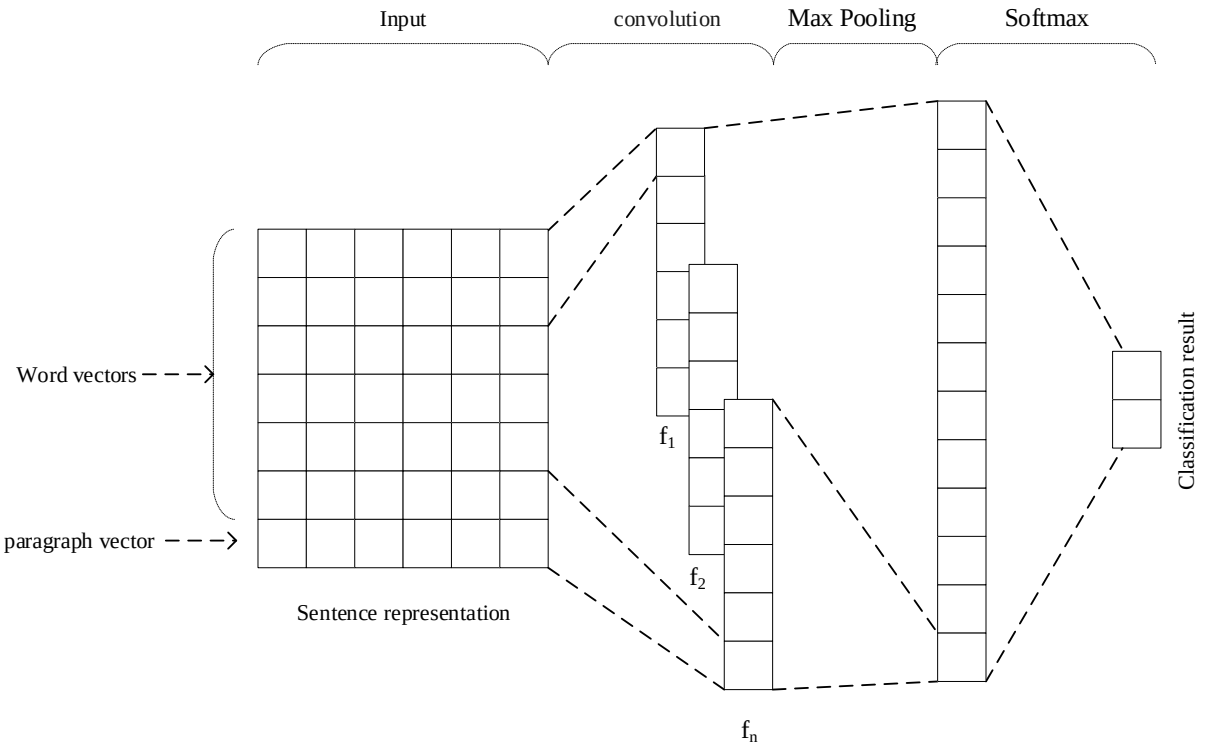

Fig. 2 Structure of convolutional neural network based on Paragraph2vec

2.2.2 Convolution

The convolution operation is a process of obtaining a feature map from a sentence matrix. It captures valuable semantic information throughout the sentence. The convolution operation mainly uses a filter matrix to calculate the local word vector matrix to obtain a feature value.Let $w_{i:j}$ be the matrix formed by the word vector w_i to w_j.For the input matrix S, a window of length t is swept on the matrix S with a step size of 1 to obtain the local word matrix $w_{i:i+t-1}$, using a filter of size $t\times d$ to operate on the matrix $w_{i:i+t-1}$, a new feature c_i can be obtained. The calculation method of c_i is:

$$c_i = f(m \cdot w_{i:i+t-1} + b) \tag{5}$$

Where c_i represents the i-th eigenvalue in the convolutional feature map, i has a value range of $(1, l-t+2)$, f is a nonlinear function, m is a filter matrix, and b is a bias value.After the filter matrix is convoluted with all the local word vector matrices obtained by window sliding, the feature matrix C can be obtained, $C = [c_1, c_2, ..., c_{l-t+1}], C \in R^{(l-t+2)\times 1}$.

In practical applications, one filter is not enough to get enough features, and usually multiple filters of different window sizes are used to get more different features. With n filters, the eigenvalues calculated by different filters are:

$$c_{ji} = f(m_j \cdot w_{i:i+t-1} + b) \tag{6}$$

Where j represents the j-th filter,the value of the filter window t is usually 2, 3, 4, 5, and the feature matrix obtained by the j-th filter is $C_j = [c_{j1}, c_{j2}, ..., c_{jl-t+2}]$.

2.2.3 Max Pooling

The Pooling layer performs dimensionality reduction on the features of the convolutional layer to form a feature vector that is ultimately passed to the classifier. This layer can ensure displacement invariance, reduce the input size and number of parameters of the next layer, and reduce the amount of calculation. On NLP applications, the Pooling layer usually has two types of operations: Mean-Pooling and Max-Pooling. The former is to average only the feature points in the neighborhood. Under this operation, all the feature values obtained by the convolutional layer are related to the final feature vector, which will weaken the strong eigenvalue, which is not applicable to the classification task. In contrast, Max-Pooling is more widely used [13].In the sentiment analysis of sentence-level texts, the input sentences are often of different lengths. Through the Max-Pooling operation, several values are fixedly extracted from each convolved feature vector. When the number of filters in the convolution layer is fixed, the Pooling layer also fixes the length of the final feature vector, so that the length of the feature vector output by the Pooling layer is the same regardless of the length of the input sentence, which facilitates the classification operation of the next layer.

This article uses the Max Pooling Over Time operation, which retains the feature value with the largest score for the feature values obtained by a filter, that is, retains the strongest feature.Let the convolution feature vector obtained by filter f_j be C_j, and the feature value obtained by the pooling layer after operation be p_j.The calculation method of p_j is:

$$p_j = \max(C_j) \tag{7}$$

With n filters in the convolutional layer, the pooled feature vector P obtained by all the convolutional feature vectors after the Pooling operation is $P = [p_1, p_2, ...p_n], P \in R^{n\times 1}$.

2.2.4 Softmax

In order to reduce the over-fitting and improve the performance of the model, the Dropout [4] strategy is added to the Softmax classification training. The idea is to perform a random Bernoulli transform on the input vector, that is, the elements of P are randomly set to 0. The output of the Softmax classifier is:

$$p(y \mid P) = soft\max(W_s \cdot (P \circ r) + b_s) \tag{8}$$

Where y is the categorical variable, W_s is the Softmax classifier parameter, r is the Bernoulli random variable vector, its elements are only 0 and 1, and b_s is the bias term.

2.2.5 Training Model

This model uses a back propagation algorithm to train the model. The goal of training is to minimize the cross entropy loss function:

$$loss = -\sum_i p(y_i \mid S_i) \log(\hat{p}(y_i \mid S_i)) + \lambda \|\theta\|_2^2 \tag{9}$$

Where i is the sentence number, $p(y_i \mid S_i)$ is the real category of the i-th sentence, $\hat{p}(y_i \mid S_i)$ is the prediction category obtained by the softmax classification of the i-th sentence, and θ is the parameter in the model. In order to alleviate the problem of over-fitting, this paper performs L2 regularization on the loss function, and λ is the parameter of the regular term.

3. Experiments and Analysis

In this paper, the convolutional neural network model based on Paragraph2vec is applied to two data sets for experiment.By comparison with the reference model to verify the proposed method to improve the accuracy of sentence-level sentiment analysis by supplementing the word order information.The word vector and paragraph vector in this paper are trained by the Doc2vec model

provided by gensim, where the word vector is trained by the skip-gram model and the paragraph vector is trained by PV-DM. The experimental environment and its configuration in this paper are shown in Table 1.

Table 1. Experimental environment configuration

Experimental environment	Configuration
operating system	Windows 10
CPU	I5-6300HQ, 2.30GHz
RAM	8GB
Programming language	Python 3.5.5
Deep learning framework	Tensorflow 1.8.0

3.1 Data Sets

There are two data sets used in this experiment. The first is the IMDB big movie review data set [5], which is a two-category data set containing 100,000 movie commentary data, which has been divided into 50,000 labeled and 50,000 unlabeled data. The tagged review data was divided into 25,000 positive comments and 25,000 negative data. The training set and test set were also divided by 1:1 during the training. The second is the Stanford sentiment treebank (SST) dataset [6], which contains 11,855 sentences from the Rotten Tomatoes film review, with 8544 as the training set, 2210 as the test set, and 1101 as the validation set.The SST dataset has two classification tasks, one is a fine-grained 5-category task, the label is: {Very Negative, Negative, Neutral, Positive, Very Positive}; the other is a coarse-grained 2-category task, labeled: {Negative , Positive}, this paper sets the 5 classification task data set as SST-1 data set, and the 2 classification task data set as SST-2. The document size of the IMDB data set is different from that of the SST data set. The former has multiple sentences in each comment, while the latter has a single sentence. Therefore, when training the IMDB data set, the whole comment is treated as a long sentence.

3.2 Experimental Parameter Settings

In this paper, the experimental parameters of the text are set according to the comments given in [7]. The experimental parameters include the word vector dimension, the paragraph vector dimension, the convolution filter window size, the number of convolution filters, and the Droupout algorithm ratio. The activation function of the convolutional layer of this experiment is the Relu function, and the specific parameters are shown in Table 2.

Table 2. Experimental parameters

Parameter name	Value
word vector dimension	100
paragraph vector dimension	100
filter window size	{3,4,5}
number of filters	{100,100,100}
Droupout algorithm ratio	0.5

3.3 Word Vector Models Comparison Experiment Results

Table 3. Comparison of word vector models on IMDB data set

Word vector models	Vector dimension	Accuracy (%)	F1-score
Word2vec	200	76.1	0.76
Paragraph2vec	200	86.4	0.86

he Word2vec and Paragraph2vec word vector models were compared on the IMDB film data, and the classifiers were Logistic Regression.In the word2vec-based experiment, the superposition mean of all word vectors in the sentence is used as the vector of the sentence. As shown in Table 3, the classification results from Paragraph2vec model are significantly better than Word2vec in the accuracy rate and F1-score.The reason is that the Word2vec model focuses on words during the

training process, ignoring the influence of sentence structure and word order information, so it is slightly inferior to Paragraph2vec in sentence-level sentiment analysis tasks.

3.4 Neural Network Models Comparison Experiment Results

Table 4. Comparison of the accuracy of neural network models on multiple data sets

Models	IMDB (%)	SST-1(%)	SST-2(%)
RNN	86.8	43.2	82.4
CNN	87.9	44.5	84.0
This paper	89.0	45.2	85.0

This paper compares the models on the IMDB, SST-1 (5-category) and SST-2 (2-category) data sets. The IMDB and SST-2 data sets are binary classification tasks,so the corresponding loss function is binary cross entropy. The loss function of the SST-1 multi-classification task is categorical cross entropy. The baseline models in this paper is the recurrent neural network RNN, the convolutional neural network CNN proposed by kim [8]. On the SST task, the results of RNN are from the paper [5]. As shown in Table 4, we can clearly see that in the sentence-level sentiment analysis task, the CNN model is significantly better than the RNN model. The proposed model obtains the current optimal results on both IMDB and SST binary classification and multi-classification tasks, and proves the superiority of Paragraph2vec paragraph vector combined with CNN.

4. Conclusion

Aiming at the problem of word order information in sentence-level sentiment analysis, this paper proposes a convolutional neural network model combined with Paragraph2vec. The Paragraph2vec model considers the context information in the process of learning the paragraph vector.Therefore, its vector retains enough important information of sentences not found in word vectors, and provides effective information such as word order for the final sentence feature vector, which improves the accuracy of the model processing sentence-level sentiment analysis.This paper describes the principle of the main model in detail, and tests it on multiple data sets. After repeated experiments, the results are better than the baseline model, which verifies the rationality and effectiveness of the proposed method. The method proposed in this paper does not pay attention to the emotional information. Next, the emotional information will be taken into account in the model, and the ability of the model to deal with multi-classification tasks will be improved.

References

[1] Liu B. Sentiment Analysis and Opinion Mining[J]. Synthesis Lectures on Human Language Technologies, 2012, 5(1): 1-167.
[2] Mikolov T,Chen K,Corrado G, et al. Efficient Estimation of Word Representations in Vector Space[J]. Corr, 2013, abs/1301.3781.
[3] Le QV,Mikolov T. Distributed Representations of Sentences and Documents[J]. Corr, 2014, abs/1405.4053.
[4] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a Simple Way to Prevent Neural Networks from Overfitting [J]. J. Mach. Learn. Res., 2014, 15(1): 1929-1958.
[5] Maas AL, Daly RE, Pham PT, et al. Learning Word Vectors for Sentiment Analysis[C]//Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, 2011: 142-150.
[6] Socher R, Perelygin A, Wu J, et al. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank[J]. In Proceedings of EMNLP, 2013: 1631-1642.
[7] Zhang Y, Wallace BC. A Sensitivity Analysis of (and Practitioners' Guide To) Convolutional Neural Networks for Sentence Classification [J]. Corr, 2015, abs/1510.03820.
[8] Kim Y. Convolutional Neural Networks for Sentence Classification [J]. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1746–1751.