AUTOMATIC TEXT CLASSIFICATION-EFFICACIOUSLY EMPLOYING THE CONVOLUTED NEURAL NETWORKS TO HARNESS THE NATURAL LANGUAGE PROCESSING TOOLS & TECHNIQUES

Pushkar Garg

Delhi Public School, R.K. Puram, New Delhi

ABSTRACT

Automatic text classification is a fundamental task in the field of natural language processing and it can help users select vital information from massive text resources. To better represent the semantic meaning of a text, and to solve the problem that traditional methods need to extract features manually, we use TF-IDF algorithm to calculate the weight of each word in a text, then weight the word vectors by TF-IDF value. This method will generate text vectors, which have clearer semantic meanings. Then we input the text vector matrix into Convolution Neural Network (CNN), so that the CNN will automatically extract text features. Through extensive experiments conducted on two data sets, experiments demonstrate that our approach can effectively improve the accuracy of classification, and the classification accuracy of the two data sets are 96.28% and 96.97% respectively.

Keywords-*text classification; word vector; deep learning; machine learning; convolution neural network*

INTRODUCTION

As computer technology develops, text data produced by increasing internet users grows explosively with each passing day, ranging from e-mail, BBS, news text, and so on. How to manage and use this text information effectively has become an urgent need, which has promoted the rapid development and extensive application of automatic text classification technology. The most important issue in text classification is facing how to express the text on the computer, this representation needs to find the information that can represent text, Efficient text representation and better text feature extraction are difficult problems for text classification. Therefore, to find an efficient method of text representation and feature extraction, and to improve the accuracy of classification, becomes an important problem in automatic text categorization.

To achieve an automatic and rapid classification of various texts, researches have been conducted relating to Support Vector Machine (SVM) [1], K-Nearest Neighbor (KNN) [2], Naive Bayesian (NB) [3] and other classification methods. The existing text representation methods are Bag-of-words (BOW) [4], Term Frequency-Inverse Document Frequency (TF-IDF) [5], Latent Semantic Indexing (LSI) [6] and so on.

The above-mentioned methods are shallow machine learning, in which the semantic meaning of texts is not clear, and feature selection and feature extraction need to be done manually. Therefore, a method of text representation and efficient feature extraction is urgently needed. In this paper, we propose the W2V_TFIDF_CNN method to better represent text and extract features. We use TF-IDF algorithm to calculate the weight of words in each text, then use the TF-IDF value to weight the word vectors to get text vectors. The text vectors are input in the CNN in a form of a matrix, using CNN to automatically extract text features, and then classifying texts into different types.

WORD VECTOR LEARNING METHODS

Word2vec [7] was put forward by Mekolov after he learned from Bengio's Neural Network Language Model (NNLM) [8] and Hinton's Log_Linear model [9]. Word2vec can train word vectors quickly and efficiently and it includes two sub-models: CBOW model and Skip-gram model. The CBOW model predicts the word "w(t)" by using c words ahead and behind it. While, on the contrary, the Skip-gram model uses the word "w(t)" to predict c words ahead and behind it. In this paper, we use CBOW model to train word vectors.

When the training is completed, we can get all the word vectors in the text. The trained word vector is like the equation: "king" - "man" + "woman" = "queen" [10]. We can learn that training word vector through Word2vec is quite conducive to express semantic features.

CONVOLUTION NEURAL NETWORK

A. Basic Structure of CNN

CNN is a multi-layer neural network and is composed of multiple convolution layers and pooling layers in turn. Each layer consists of many two-dimensional planes and each plane consists of independent neurons. For the basic working principle of CNN and its principle derivation, as in [11]. The basic structure of CNN is shown in Fig. 1.

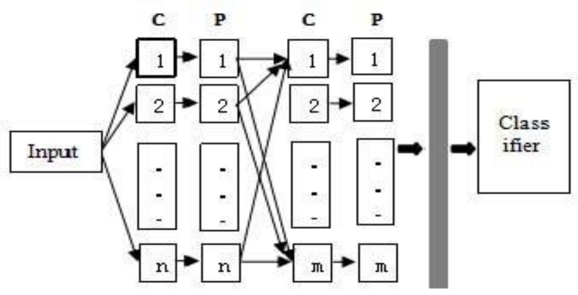

Figure 1. Basic Structure of CNN.

"C" represents a convolution layer which is also known as a feature extraction layer, where the input of each neuron here is connected to the receptive field [12] of the previous layer and local features are extracted. The weight of the same feature graph is shared, that is, similar graphs share the same convolution kernel. The layer C preserves different local features so that the extracted features will be invariant to translation and rotation. "P" refers to a pooling layer, also called a feature mapping layer, which pools the features obtained by layer C so that the extracted features will be invariant to scaling. Besides, the number of Layer C and Layer P is set according to actual needs. The last layer of CNN is generally a fully connected layer, and the number of output nodes is the number of classification targets. For more information about the structure of CNN, as in [13].

B. Text Classification Model Based on CNN

1) Input Layer

The first layer of CNN is the input layer. The shape of the input text matrix is (n, s, k), where n refers to the number of texts, s refers to the fixed text length (All the texts are padded to the fixed length s during preprocessing if the text is shorter than s), and k is the dimension of word vector. x_i□R^k represents the k-dimensional word vector corresponding to the i-th word in the text. The input text can be expressed as

$$x_{1:s} = x_1 \heartsuit x_2 \heartsuit \bullet\bullet\bullet \heartsuit x_s$$

i

Here ♥ is the concatenation operator. In general, let$x_{i:i+j}$ refertotheconcatenationofwordsx_i,

x_{i+1}, ... , x_{i+j} ·

2)Convolution Layer

The second layer of CNN is a convolution layer. The convolution operation involves a filter w$\chi R^{h\times k}$, which is applied to a window of h words to produce a new feature, for example, a feature ci is generated from a window of words $x_{i:i+h-1}$ by

$$c_i = f(w \bullet x_{i:i+h-1} + b)$$

Here b is a bias term and f is the non-linear activation function. When the convolution filter is moved by one step at a time, all the input matrixes are convoluted by each window in turn, which will produce a feature map

$$c = [c_1, c_2, \bullet\bullet\bullet, c_{s-h+1}]$$

3) PoolingLayer

The third layer of CNN refers to a pooling layer. In this paper, the CNN model uses MAX Pooling. After the convolution operation, the feature maps of the convolution layer are pooled, and all the feature maps are aggregated and calculated. We take its maximum value as the feature of the pooling layer because it would extract the most prominent features.

4) Fully Connected Layer

The fourth layer of CNN is the fully connected layer, which connects all the features and output values to classifiers. While training the model, if the number of training samples is too small, or the model is excessively trained, it will cause over-fitting. In this paper, we applied the Dropout method [14] proposed by Hinton to improve the generalization of the model, thus to prevent over-fitting. At the same time, the interaction between the hidden layer neurons is reduced and the structure of the model is optimized. For more information about CNN, as in [15].

C. The Method Proposed in this Paper

1) Weighting Word2vec by TF-IDF

For the set D containing M texts, where Di (i = 1, 2, ..., M), has been segmented by "Jieba", and then it is trained by word2vec model to get the 100-dimensional word vector of each word. For each word in a text, its weight value tfidf(t, D) is calculated through the TF-IDF algorithm, which refers to the weight value of the word t in the text Di (i=1,2,…, M). TF-IDF considers the word frequency tf in a single text and the word frequency idf in the text set. The formula of idf for the word t.

$$tf\text{-}idf(t, D) = \frac{\Box tf(t, D_i) \times idf(t)}{\sqrt{\sum_{t \in D_i} [tf(t, D_i) \times idf(t)]^2}}$$

Here tf(t, Di) is the word frequency of word t in the i-th text, and its denominator is a normalization factor.

The word vector of each word is weighted to represent a text, as in (6):

$vec(D_i)$refers to the vector of each text D_i, $w_t$represents the N-dimensional word vector of the word t, and tfidf(t, Di) represents the TF-IDF weight value of the word t in the text Di.

2) W2V_TFIDF_CNN

When we get each text vector, we will modify all the resulting text vectors to the shape of a matrix that is required for CNN processing. The input texts can be expressed as

$$T_{1:n} \square\ vec(D_1)\ \square vec(D_2) \square\square\square\square vec(D_n)$$

Here $T_{1:n}$ represents all input texts of CNN, □ is the concatenation operator.

A. DataPreprocessing

The text data cannot be directly inputted into CNN before being preprocessed. Data preprocessing is divided into the following three steps:

1) Word Segmentation and Stop-WordsDeletion

In this paper, the word segmentation tool is based on an open source library of "Jieba". We make a text of stop-words to reduce redundancy and make the text classification more accurate.

2) Padding

Padding all the texts with the word "filling" to the same length as the longest text among all the texts, which will produce the needed matrix for the subsequent input into CNN.

3) Word Vectorization

In order to test the influence of BOW, TF-IDF, LSI and Word2vec model dimension on the classification results, eachmodelselectsdifferentdimensiontests.Weselect[100, 200, 300, 400, 500] as the dimensions of the BOW, TF-IDF, and LSI models. Also, we select 50 topics for the LSImodel and use SVM and KNN as classifiers. We apply the Word2vec tool to train text and select [100, 200, 300, 400, 500] as the dimensions of the Word2vec model. The classification effect of each model is shown in Fig.2

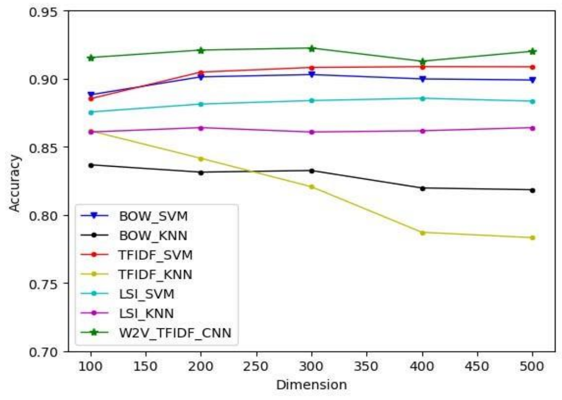

Figure 2. The Accuracy of Classification in different Model Dimensions

From Fig. 2 (in Fig. 2, BOW_SVM means that text is trained using the BOW model to obtain text features, and then SVM classifier is used to classify texts. Other symbols are similar), the increase of the training word vector dimensions hardly influences the classification accuracy. Taking the size of a text, training speed and the efficiency of classification into consideration, we finally select 200- dimensional training text for BOW, TF-IDF and LSI models, and 100-dimensional training text for Word2vec model. IV.

EXPERIMENTAL RESULTS AND ANALYSIS

To verify the validity of the text classification method based on a convolution neural network proposed in this paper, two data sets are selected in the experiments.

A. Data Sources and Experimental Environment

The experimental corpus is selected from the NetEase news corpus and text classification corpus of Fudan University as the trained corpus. Then, we choose 8160 texts about 6 classifications: automobile, culture, economics, medicine, military, and sports from the NetEase news corpus, as well as 7789 texts about 10 classifications: art, education, philosophy, history, computer, environment, agriculture, economics, politics, sports from text classification corpus of Fudan University. This experiment is supported by Python under Windows 7.

B. The result of the Proposed Method

1) The Effect of Text Length on Text Classification

In order to test the effect of text length on the training effect of CNN, this paper intercepts texts from NetEase news and text classification corpus of Fudan University. These texts are with lengths (word number in one text) of 100, 200, 300, 400, 500, and 1000, and iterations of 1, 10, 20, 30, 40, 50, 60,

70, 80, 90, and 100. For the criteria of text classification, we constantly take the classification accuracy as indicators for evaluation. The formula is as in (7):

$$Accuracy = \frac{correct\ number\ of\ text\ classification}{total\ number\ of\ text}$$

The experimental results are shown in Table ĉ.

Number of Iterations	Length of NetEase News Text						Length of Text Classification Corpus of Fudan University					
	100	*200*	*300*	*400*	*500*	*1000*	*100*	*200*	*300*	*400*	*500*	*1000*
1	85.98	87.86	88.88	89.11	89.97	90.60	86.52	84.89	78.21	76.33	76.76	76.12
10	91.74	93.79	94.73	94.40	95.66	95.01	95.67	95.97	94.90	95.12	94.77	92.55
20	93.66	94.44	95.38	95.66	96.11	95.54	96.02	96.70	95.84	95.84	95.37	94.13
30	94.28	94.97	95.58	95.58	96.16	96.11	96.66	96.57	96.53	95.76	95.76	95.25
40	94.24	95.50	95.83	95.66	95.99	95.91	96.87	97.0	96.02	96.49	95.42	95.33
50	94.64	95.62	95.62	96.20	95.84	96.32	96.83	97.0	96.74	96.61	95.89	93.53
60	94.52	95.58	95.75	95.46	96.03	96.48	96.96	96.87	96.53	96.83	96.70	94.95
70	94.93	95.42	95.95	95.87	96.36	96.65	96.83	97.13	96.02	96.40	96.61	95.50
80	94.56	95.75	95.99	96.16	96.52	96.69	96.91	97.04	96.70	96.83	96.10	96.02
90	94.32	95.99	95.62	96.16	96.11	96.48	97.09	97.04	96.79	96.49	95.67	95.89
100	94.64	95.46	95.91	96.24	96.65	96.60	96.83	96.79	96.57	96.49	96.06	95.72

From the NetEase news text data set in Table ĉ, the text classification accuracy grows as the increase of text length from 100 to 500 at an interval of 100 after convergence. While when the text length increases from 500 to 1000, the classification accuracy is basically unchanged. And as it can be seen from the text classification corpus of Fudan University in Table ĉ, the text classification accuracy has increased slightly with the text length rising from 100 to 200 after convergence. Then with text length increasing, the text classification accuracy shows a downward trend. The reason for this result is the difference between the contents of the two data sets, which differentiates the selected features at the beginning ofthe selection, and thus it causes different CNN training effects. In this way, the accuracy of the texts about 10 classifications is better than these about 6 classifications.

Taking text classification accuracy and training speed into consideration, this paper chooses the text length of 500 and 200 from the NetEase news text and text classification corpus of Fudan University respectively.

2) Effect of Number of Iterations

CNN gets weights by iterative calculations and obtains the ideal parameters after several iterations. In this experiment, the results of our proposed method W2V_TFIDF_CNN of different iterations on two data sets are shown in Table II and Table III.

TABLE II. EXPERIMENT RESULTS OF DIFFERENT ITERATIONS (NETEASE NEWS TEXT)

Performance	Number of Iterations						
	1	*10*	*20*	*30*	*40*	*50*	*100*
Training time (s)	57.6	674.2	1364.8	2057.5	2820.2	3444.0	6945.9
Test time (s)	9.4	9.6	9.3	10.3	10.2	10.2	9.1
Accuracy (%)	90.25	95.66	96.11	96.16	95.99	95.84	96.65
Loss (%)	35.63	24.63	17.89	16.53	17.02	17.35	17.10

TABLE III. EXPERIMENT RESULTS OF DIFFERENT ITERATIONS (TEXT CLASSIFICATION CORPUS OF FUDAN UNIVERSITY)

Performance	Number of Iterations						
	1	*10*	*20*	*30*	*40*	*50*	*100*
Training time (s)	40.6	430.3	870.1	1222.9	1583.9	1972.6	3661.7
Test time (s)	3.7	3.9	4.2	4.0	3.6	3.8	3.5
Accuracy (%)	84.89	95.97	96.70	96.57	97.0	97.0	96.79
Loss (%)	67.49	15.08	11.55	11.46	9.88	12.15	13.57

In CNN training, when the number of iterations is not adequate, the network learning is not sufficient and the trained model is not ideal, just as it is shown in Table Ċ and Table ċ, so is the accuracy rate. With the increasein the number of iterations, the accuracy of classification increases, the rising trend of classification accuracy is gradually reduced. When the number of training reaches a certain level, the network parameters nearly remain the same. Then, the convolution network converges and the classification performance is optimal.

The training time of CNN and the times of iterations are positively correlated, but the test time is not affected by the number of iterations. In this experiment, the length of each text of the intercepted NetEase news text is 500, and the length of each text of the interception of Fudan text isonly 200, so the time spent in the training and testing of the two data sets are quite different.

In this experiment, we choose the CNN training method of mini-batch training, that is, one group includes several samples which are called batch-size. Mini-batch training will lead to network jitter when training loss happens. The previous batch-size samples will update the network weights to ensure that the new weights will reduce losses in the training process. But for the latter batch-size, losses may increase. However, the overall number of losses will gradually reduce with more training being carried out. A total of 100 iterations were performed in this experiment, and the effect of the first 20 iterations is shown in Fig. 3.

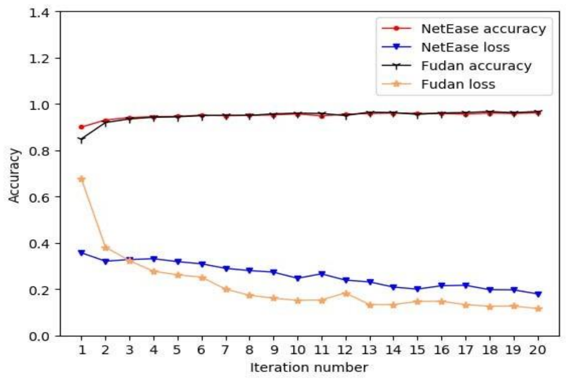

Figure 3. Accuracy and Loss of Two Data Sets Based on Convolution Neural Network

3) Comparison of the Proposed Method with other Methods

To verify the effectiveness of this method, we compared our proposed method W2V_TFIDF_CNN with other classification methods mentioned in Fig. 2. The results of comparing them with the other text classification methods on the two data sets are shown in Table IV.

TABLE IV. CLASSIFICATION ACCURACY OF DIFFERENTTEXT CLASSIFICATION METHODS(%)

Classification Methods	Corpora Name	
	NetEase News Text	*Text Classification Corpus of Fudan University*
BOW_SVM	92.97	90.13
BOW_KNN	86.27	83.13
TFIDF_SVM	94.10	90.48
TFIDF_KNN	89.09	84.15
LSI_SVM	91.89	88.13
LSI_KNN	89.66	86.40
W2V_TFIDF_CNN	96.28	96.97

In Table IV, the accuracy of the method based on CNN is superior to other methods, thus the effectiveness of this classification method is verified. The reason for this effect is as follows: 1) Using Word2vec to produce word vectors can get higher-quality features; 2) The impact of a single word on the entire document is better taken into consideration through TF-IDF weighting; 3) The post-processed text features through CNN can better represent the high-level features of a text.

CONCLUSION

This paper is mainly about the text categorization method based on CNN, which does not need to extract text features in advance. The process is as follows: First, pre-process texts through Word2vec to generate word vectorsbased on Chinese characteristics. Then, weight the word vectors to get the text vector by the generated TF-IDF value. Finally, use CNN to extract higher-level features and improve network recognition ability. At the same time, prevent data from over-fitting in the method of Dropout to improve the generalization capacity of the network.

The text classification performance of CNN is tested in various aspects including different text lengths, iteration times and so on. Also, it is compared with other classification methods. It is shown in the experimental results that our method is superior to other classification methods in this paper, which can produce better classification results. The classification accuracy of W2V_TFIDF_CNN on the two data sets are 96.28% and 96.97% respectively.

REFERENCES

[1] Cui J M, Liu J M, Liao Z Y. Research of Text Categorization Based on Support Vector Machine[J]. Computer Simulation, 2013, 30 (2): 299-302.

[2] Gui Y N. Chinese Text Classification Based on KNN Algorithm [D]. China University of Petroleum (Beijing), 2012.

[3] Li D. Chinese Text Classification Based on Naive Bayesian Method [D]. Hebei University, 2011.

[4] Wallach H M. Topic modelling: beyond bag-of-words[C]// International Conference on Machine Learning. ACM, 2006:977-984.

[5] Hu J, Yao Y. Research on the Application of an Improved TFIDF Algorithm in Text Classification[J]. Journal of Convergence Information Technology, 2013, 8(7):639-646.

[6] Vastenhouw B. Latent Semantic Indexing[J]. 1990.

[7] Bengio Y, Schwenk H, Senécal J S, et al. Neural Probabilistic Language Models [M]//Innovations in Machine Learning. Springer Berlin Heidelberg, 2006.

[8] Mnih A, Hinton G. Three New Graphical Models for Statistical Language Modelling [C]//Proceedings of the 24th International Conference on Machine Learning. ACM, 2007: 641-648.

[9] Le Q V, Mikolov T. Distributed Representations of Sentences and Documents[J]. EprintArxiv, 2014, 4:1188-1196.

[10] Mikolov T, Yih W, Zweig G. Linguistic Regularities in Continuous Space Word Representations [C]//HLT-NAACL. 2013:746-751.

[11] Behnke S. Hierarchical Neural Networks for Image Interpretation [M]. Berlin Heidelberg: Springer, 2003.

[12] Hubel D H, Weisel T N. Wiesel, T.N. Receptive Fields, Binocular Interaction and Functional Architecture in Cat's Visual Cortex. J. Physiol. (London) 160, 106-154 [J]. 1962, 160.

[13] Jia S J, Yang D P, Liu J H. Product Image Fine-grained Classification Based on Convolutional Neural Network [J]. Journal of Shandong University of Science and Technology (Natural Science Edition), 2014, 33 (6): 91-96.

[14] Hinton G E, Srivastava N, Krizhevsky A, et al. Improving Neural Networks by Preventing Co-adaptation of Feature Detectors [J]. Computer Science, 2012, 3 (4): pages. 212-223.

[15] Huang W, Wang J. Character-level Convolutional Network for Text Classification Applied to Chinese Corpus[J]. 2016.