Computers, Materials & Continua CMC, vol.60, no.2, pp.707-719, 2019

Tibetan Sentiment Classification Method Based on Semi-Supervised Recursive Autoencoders

Xiaodong Yan[1, 2], Wei Song[1, 2, *], Xiaobing Zhao[1, 2] and Anti Wang[3]

Abstract: We apply the semi-supervised recursive autoencoders (RAE) model for the sentiment classification task of Tibetan short text, and we obtain a better classification effect. The input of the semi-supervised RAE model is the word vector. We crawled a large amount of Tibetan text from the Internet, got Tibetan word vectors by using Word2vec, and verified its validity through simple experiments. The values of parameter α and word vector dimension are important to the model effect. The experiment results indicate that when α is 0.3 and the word vector dimension is 60, the model works best. Our experiment also shows the effectiveness of the semi-supervised RAE model for Tibetan sentiment classification task and suggests the validity of the Tibetan word vectors we trained.

Keywords: Recursive autoencoders (RAE), sentiment classification, word vector.

1 Introduction

With the rapid development of Web 2.0, users participate in the manufacture of website content. Consequently, there are a large number of user-involved valuable comments on people, events, products and so on which are generated on the Internet. By analyzing this information, potential users can mine people's views and opinions to make business decisions, political decisions, and so on. It is a hard work to deal with such massive amounts of data manually. How to use help users quickly analyze and process these web texts automatically and extract useful emotional information by computer has become the focus of many researchers. Text sentiment analysis is the process of analyzing, processing, summarizing and disposing words, sentences and texts with emotional color[0]. At present, the research on sentiment classification of Chinese and English texts is relatively mature. However, for Tibetan information that started late, the study of Tibetan sentiment tendencies is relatively lagging behind. With the increasing content of network information such as Tibetan web pages and Tibetan digital libraries, more and more Tibetan compatriots express their opinions and opinions in Tibetan on the Internet. The emotional analysis of Tibetan texts has become an urgent research issue. On the basis of the analysis of sentence sentiment tendency, it is convenient to analyze the sentiment orientation of the text, and even get the overall tendencies of massive information. Therefore, sentence-level sentiment classification

[1] Minzu University of China, Beijing, 100081, China.

[2] National Language Resource Monitoring & Research Center Minority Languages Branch, Beijing, China.

[3] New Jersey Institute of Technology, Newark, NJ, 07102, USA.

[*] Corresponding Author: Wei Song. Email: songwei@muc.edu.cn.

has important research value and is also the research focus of this paper.

2 Related work

Sentiment classification is one of the hot issues in natural language processing. There have been many researches on text sentiment classification at home and abroad. In general, it can be divided into a machine learning based method and sentiment dictionary-based method. The basic idea of the machine learning method is to get an estimate of the dependence between the input and output of the system based on known training samples, so that it can make the most accurate prediction of the unknown output. In 2002, Pang et al. [Pang, Lee and Vaithyanathan (2002)] used common machine learning techniques to make propensity judgments, and compared the propensity judgment effects of support vector machine (SVM), naive Bayes (NB), and maximum entropy. It shows that the SVM has the best classification effect. The literature[0] studied the classification of news texts, and used the naive Bayesian method and the maximum entropy method to divide the news text into positive emotions and negative emotions, and used word frequency and binary value as feature weights, and finally achieved better results. Classification effect, the highest classification accuracy rate of more than 90%. Based on sentiment dictionary or knowledge system, the literature uses the existing semantic dictionary to judge the semantic tendency of the sentiment words in the sentence. Then according to the syntactic structure and other information, indirectly get the semantic tendency of the sentence. Riloff et al. [Riloff and Shepherd (1997)] proposed a corpus-based approach to construct sentiment dictionary to achieve emotional classification. Later et al. [Riloff, Wiebe and Phillips (2005)] used the Bootstrapping algorithm, which used the elements of pronouns, verbs, adjectives and adverbs in the text as features. And they also treated differently according to the position of the sentence in the paragraph to realize the objective and subjective classification of corpus data. Zhu et al. [Zhu, Min, Zhou et al. (2006)] artificially constructed the word set of positive and negative seed sentiment words in the literature, and then used HowNet to calculate the semantic similarity between candidate words and seed sentiment words to determine their emotional polarity.

In terms of sentiment classification of Tibetan texts, research at home and abroad is not yet mature, and relevant literature is very limited. The literature used Tibetan three-level segmentation system to segment the Tibetan texts and part-of-speech tagging, and used the hand-built Tibetan sentiment analysis vocabulary to extract the emotional features in combination with the existing feature selection methods, and used the similarity classification algorithm to classify the sentiment of Tibetan texts. In literature, the sentiment analysis of Tibetan Weibo was carried out based on the combination of statistics and dictionary-based methods. The accuracy of this method was significantly higher than that of TF-IDF-based Tibetan microblog sentiment analysis.

Based on the above related work, this paper applies the semi-supervised recursive autoencoders RAE model to sentiment classification task of Tibetan short text. Through the extensive training of the word vector and the determination of the dimension, we get good classification result.

3 Sentiment classification method based on semi-supervised RAE

3.1 Tibetan word vector training

The input of the semi-supervised RAE Tibetan sentiment classification model method is a sequence of word vectors of Tibetan text. There are two methods for initializing the word vector. In the first method, we simply initialize the vector of each word $x \in R^n$ to a value (sample) that follows the Gaussian distribution: $x \sim N(0, \delta^2)$, and then put the word vectors into a matrix $L \in R^{n \times |V|}$, where |V| is the length of the vocabulary. This initialization method works well in an unsupervised neural network that can optimize these word vectors by capturing valid information in the training data. The second method to get a word vector is through an unsupervised neural language model [Bengio, Ducharme,Vinent et al. (2003); Collobert and Weston (2008)]. In the processing of training the word vector by neural language model, they get the grammatical and semantic information in the training corpus by calculating the words' co-occurrence statistical information. And then they transform the information into a vector space, so after getting the word vector the verbal similarity of the two words in the training corpus can be predicted.

This paper uses the Word2vec tool to train and to obtain the neural language model of Tibetan. The Word2vec tool is a tool that Google developed and open sourced in 2013 to represent words as real numbers. Its main idea is to calculate the context relevant statistics of each word in all text and other co-occurring words by training a large amount of text. And then use these statistics to represent the words appearing in the text as a K-dimensional vector, and the value of K is nor very large. After obtaining the vectorized representation of the words, we can get the semantic similarity of the text by the word vector operation. Word vectors trained through Word2vec can be used to do a lot of research in the field of natural language, such as text clustering, text categorization, sentiment analysis, and so on. If we regard words as features, then Word2vec can express the features of the text in K-dimensional vector space, and this representation with semantic information is a deeper feature representation. The corpus used to train Tibetan word2vec includes wiki encyclopedia Tibetan version, primary and secondary school textbooks, news and Sina Weibo, totaling 253 M Tibetan texts.

The evaluation method of word vector can be mainly divided into two types. The first one is to apply the word vector to an existing system, and compare the running results of the system before and after the addition; the second one is to evaluate the word vector directly from the perspective of linguistics, such as text similarity one-level semantic offset and so on. To test the effect and quality of trained Tibetan word vectors, we first test some semantic similarities of words, such as input words: མཚོ་སྔོན་ཁེང་ཚེན་ལི་དམངས་འཛུས་ཚོགས་རྒྱན་ལས་ཀྱི་ཡོན་ལྷན་ཁང་རྒྱན་ལས་ཀྱི་ཡོན་ལྷན་ཁང་གི་ (Qinghai Provincial People's Congress Standing Committee) to find out which words are similar in the training expectation (see in Fig. 1). The number on the right side of the figure measures the degree of similarity between the word and the input target word, and its value ranges from [0, 1]. The larger the value, the higher the similarity.

མཚོ་སྔོན་ཞིང་ཆེན་དཔལ་འཛུམ་ཚོགས་ཆེན་རྒྱུན་ལས་ཀུ་ཡོན་སྐྱབ་ཁང་།	Qinghai Provincial People's Congress Standing Committee	
ཞིང་ཆེན་དཔལ་འཛུམ་ཚོགས་ཆེན་རྒྱུན་ལས་ཀུ་ཡོན་སྐྱབ་ཁང་།	Standing Committee of the Provincial People's Congress	0.810125565
ཞིང་ཆེན་ཁང་ཨའི་སིང་ཁྲིམས་ཀུ་ཡོན་སྐྱབ་ཁང་།	Provincial Committee of Politics and Law	0.807753325
ཞིང་ཆེན་དཔལ་ཆེན་རྒྱུན་ལས་ཀུ་ཡོན་སྐྱབ་ཁང་།	Provincial People's Congress Standing Committee	0.772617459
མཚོ་སྔོན་ཞིང་ཆེན་སྒྲིག་ཁྲིམས་ཞིབ་བཤེར་ཀུ་ཡོན་སྐྱབ་ཁང་།	Qinghai Provincial Commission for Discipline Inspection	0.7700755
མཚོ་སྔོན་ཞིང་ཆེན་ཁང་ཨའི་དཔུང་ཁྲི།	Qinghai Provincial Party Committee Secretary	0.745360911
མཚོ་སྔོན་ཞིང་ཆེན་ཁང་ཨའི་སིང་ཁྲིམས་ཀུ་ཡོན་སྐྱབ་ཁང་།	Qinghai Provincial Party Committee Political and Legal Committee	0.719981909
ཞིང་ཆེན་སྒྲིག་ཁྲིམས་ཞིབ་བཤེར་ཀུ་ཡོན་སྐྱབ་ཁང་།	Provincial Commission for Discipline Inspection	0.701914907
ཞིང་ཆེན་ཐད་གཀོག་ལས་ལྷང་བ་བའི་ཀུ་ཡོན་སྐྱབ་ཁང་།	Provincial Direct Work Committee	0.697856188
སིད་སྐྱི་ཁྲལ་ཁང་།	The State Council	0.689607143
མཚོ་སྔོན་ཞིང་ཆེན་སྒྲིག་ཁྲིམས་ཞིབ་བཤེར་ཀུ་ཡོན་སྐྱབ་ཁང་།	Qinghai Provincial Direct Working Committee	0.6777733

Figure 1: Tibetan word vector test (1)

In the training corpus find the similar words with བདེ་སྐྱིད་ (happiness) (see Fig. 2).

From the above test results, it can be found that the similar candidate words calculated by the Tibetan word vector trained in this paper have a large or a certain degree of semantic similarity with the calculated original words. Therefore, we believe that the Tibetan word vector trained in this paper has good effect and quality. In this paper, the Tibetan word vector trained by the Word2vec tool is used as the input of the semi-supervised RAE model. For a small number of words not found in the word vector trained by Word2vec, we use the first method above to initialize.

བདེ་སྐྱིད་	happiness	
འཕྱོར་སྤྱད་	rich	0.6101334095
རྨོད་སྐྱིད་	happy	0.599177002907
མཛེས་སྡུག་ལྡན་	beautiful	0.550595283508

དྲོད་ཚྱིང་	warm	0.493547737598
མཛེས་རྡུག་	beauty	0.492604732513
ཁྱིམ་གཞིས་	family property	0.491671025753
བདེ་སྡུད་ལས་བརྟེན་	living and working in peace	0.485439032316
འཚོ་བ་	life	0.477133393288
དགའ་ཚྱིད་	elegance	0.476655

Figure 2: Tibetan word vector test (2)

3.2 Semi-supervised RAE model for Tibetan sentiment classification

The Tibetan sentiment classification based on semi-supervised RAE model is shown in Fig. 3.

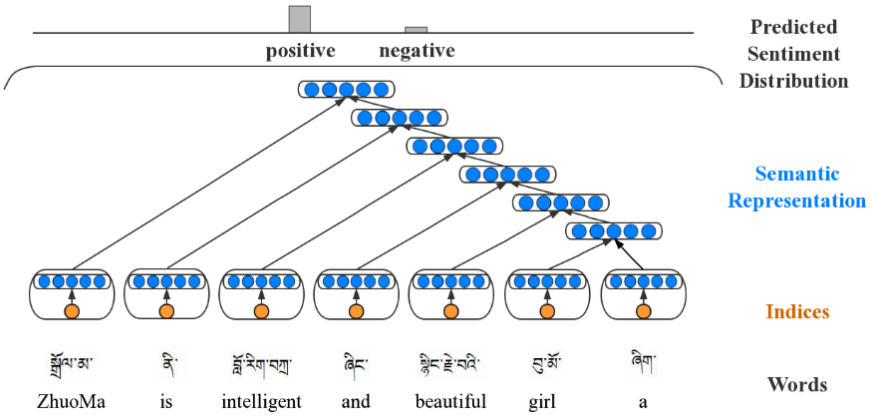

Figure 3: Semi-supervised RAE sentiment classification model

With the Tibetan word vector, the unsupervised RAE method has been able to obtain the distribution feature vector representation of the text sentence without giving the text structure tree. In order to apply it to Tibetan sentiment classification, it needs to be expanded to a semi-supervised RAE. The basic idea is to add a classifier to the top of its RAE and supervise it with labeled sample data. To do this, we add a simple softmax layer to the root node of the tree structure representing the sentence for classification, as defined by formula (1) where $d \in R^K$ is a K-dimensional polynomial distribution, $\sum_{k=1} d_k = 1$ and K is the number of emotional labels. This paper focuses on both negative and positive categories. That is k=2.

$$d(p;\theta) = soft\ \max\left(W^{label} p\right) \tag{1}$$

The output of the softmax layer represents the conditional probability A, which is the probability that the current text belongs to each category, so that the category of the text

can be predicted. The calculation method of the cross-entropy error is shown in formula (2) where t is the distribution of the labels.

$$E_{cE}(p,t;\theta) = -\sum_{k=1}^{K} t_k \, log \, d_k \, (p;\theta) \tag{2}$$

After adding the softmax layer, the training process of the semi-supervised RAE model needs to consider not only the reconstruction error of the parent node in the text sentence structure tree, but also the cross-entropy error of the softmax layer to learn the semantic and sentiment classification information in the text. Fig. 4 shows the root node RAE unit of the sentence structure tree.

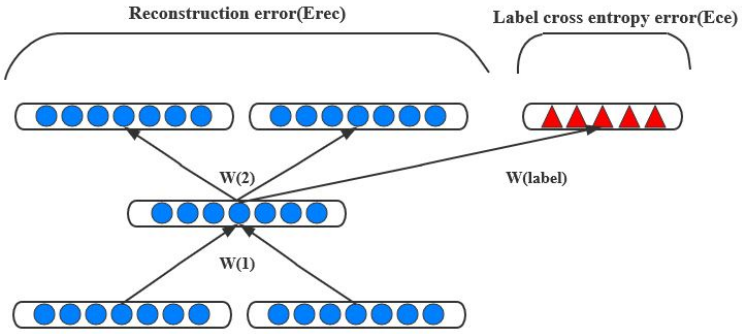

Figure 4: Root node RAE unit

Therefore, the optimization objective function based on the semi-supervised RAE method on the tagged training data set can be expressed as Eq. (3) where (x, t) represents a sample in the training corpus, that is (text sentence, label), indicating the error of a sample.

$$J = \frac{1}{N}\sum_{(x,t)} E(x,t;\theta) + \frac{\lambda}{2}\|\theta\|^2 \tag{3}$$

The error of a text sentence is the sum of the reconstruction error and the cross-entropy error of all non-terminal nodes in the sentence tree structure, so it can be expressed as shown in formula (4) Where s represents the non-terminal node in the sentence tree structure.

$$E(x,t;\theta) = \sum_{s\in T(RAE_\theta(X))} E([c_1;c_2]_s \, , p_s \, , t, \theta) \tag{4}$$

For the root node, the reconstruction error and the cross-entropy error need to be considered at the same time. The error calculation formula in the formula (4) can be written as shown in the formula (5) Where α is a parameter that adjusts the reconstruction error and the cross-entropy error weight.

$$E([c_1;c_2]_s \, , p_s \, , t, \theta) = \alpha E_{rec}([c_1;c_2]_s;\theta) + (1-\alpha)E_{cE}(p_s \, , t;\theta) \tag{5}$$

When the value of α is adjusted, the change will propagate back and affect the parameters of the RAE model as well as the vector representation of the text. In this paper, we will adjust the parameters to study its impact on the classification results in subsequent experiments.

4 Experiment and result analysis

4.1 Experimental data set

We crawl Tibetan Weibo and Tibetan comments on Sina Weibo by writing a web crawler, and saves the crawled text corpus in txt format. After pre-processing, we also need to

manually tag and verify these Tibetan texts to get a library of Tibetan emotional corpus. The labeling rules are as follows: positive text entries are marked with the label '+1', negative text entries are marked with the label '-1' neutral text entries are marked with the label '0', and the useless text entries are not removed in the preprocessing are marked with the label '2'. The final marked result is shown in Tab. 1.

Table 1: Marked corpus statistical result

Category	Negative	Positive	Neutral	Useless
Number	3717	8836	16842	1083

For a better comparison of the experimental results, the data set of this experiment are all 3717 negative samples and randomly selecting 4000 samples from the positive samples. At the same time, 400 positive and negative samples were randomly selected from the experimental data set as test sets, and the remaining samples were used as training sets.

4.2 Parameter setting

The semi-supervised RAE algorithm used in this paper is an open source project on Github by Sanjeev Satheesh from Stanford University. The source code is re-implemented in Java based on Richard Socher's documentation and MATLAB source code. For the hardware platform, the operating system of the server is Ubuntu Linux 14.04, 128 G memory, 16-core processor.

We use the default values of the parameters in the softmax layer classifier. And we compare two sets of experiments to find the optimal values of super parameter α for measuring cross entropy error and reconstruction error and the dimension of Word2Vec word vector. The model needs multiple iterations to get the final optimized result. And the number of iterations required for model convergence is different under the different parameter settings state. For making the experiment result is always best at every parameter setting, after observing the number of iterations of multiple experiments, it is found that the number of iterations will not exceed 1000. So we set 1000 as the number of iteration of the experiment.

4.3 The value of α selection experiment and result analysis

The value of the parameter α determines the degree of attention to the cross-entropy error and reconstruction error of the semi-supervised RAE model during training. Therefore, when the value of α is larger, the model pays more attention to the reconstruction error, and the sentence distribution vector obtained by training will contain more syntactic information and less emotional information; when the value of α is smaller, the opposite is true. Therefore, this paper studies the influence of the parameter α on the training process and finds the optimal parameter value by setting a series of different values of α in each training process. In addition, we fix the dimension of the word vector as a moderate value 50, to avoiding its impact on the results. The details of the experiment results are shown in Tab. 2.

Table 2: Experimental results with different A values

Value of α	Category	Accuracy/%	Recall rate/%	Value of F/%	Overall correct rate/%
0.1	Negative	82.59	78.25	80.36	80.88
	Positive	79.33	83.50	81.36	
0.2	Negative	83.55	78.75	81.08	81.63
	Positive	79.91	84.45	82.14	
0.3	Negative	83.91	79.50	81.64	**82.13**
	Positive	80.52	84.75	82.58	
0.4	Negative	86.11	62.00	72.09	76.00
	Positive	70.31	90.00	78.95	
0.5	Negative	79.17	14.25	24.15	55.25
	Positive	52.88	96.25	68.26	

In order to more intuitively observe and analyze the influence of the value of α on the experimental results, the following Fig. 5 shows the trend of the fold line change of the negative polarity, positive polarity and overall classification accuracy under different values of α.

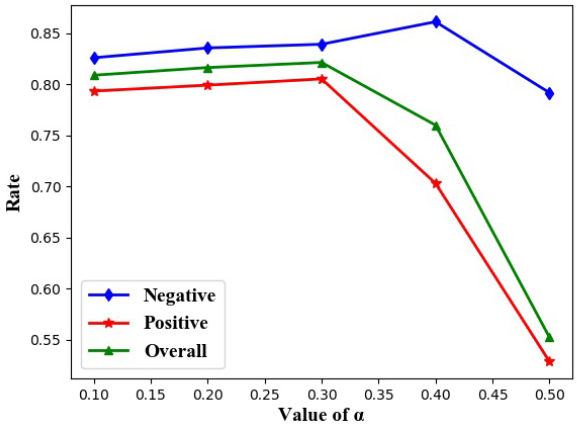

Figure 5: Trend of classification effect under different value of α

It can be found from Tab. 2 and Fig. 5 that when the value of α changes from 0.1 to 0.3, the classification effect of the model becomes better as the value of α increases, and when α is 0.3, the whole model reaches the best; when the value of α changes from 0.3 to 0.4, the effect of the negative polarity classification increases with the increase of α, but the classification effect of the positive polarity decreases sharply; when the value of α is greater than 0.4, the classification effect of the positive polarity and the positive polarity decrease greatly with the increase of α, indicating that the model over-represented the syntactic information of the text and could not obtain the emotional information of the corpus.

In the experiment of this paper, the whole model achieves the best effect when α is 0.3, and the optimal value of α is 0.2 in the comparative experiment of Pu et al. [Pu, Hou, Liu et al. (2017)]. Therefore, we can find that the optimal value of α in the semi-supervised RAE method is not the same for different corpus of texts in the same language; however, the best value is always small. Therefore, the cross-entropy error of the softmax layer should be paid more attention to when training of the model.

4.4 Word vector dimension selection experiment and result analysis

When using the Word2Vec tool to train Tibetan word vectors, you can set the dimension (length) of the word vector. The size of the vector dimension has a significant impact on the accuracy of the semi-supervised RAE model and the training efficiency. If the word vector is too short, it cannot effectively contain the semantic information of the word; if the word vector is too long, not only will the training data be sparse, but also the subsequent model training will be inefficient, wasting time and computing resources. To find the best word vector, we set a group of experiments that the values of the word vector are different in each training process while the parameter α is fixed to the optimal value 0.3. The results are shown in the Tab. 3.

Table 3: Word vector dimension comparison test result

Number	Category	Accuracy/%	Recall rate/%	Value of F/%	Overall correct rate/%
10	Negative	67.21	61.50	64.23	65.75
	Positive	64.52	70.00	67.15	
20	Negative	81.30	75.00	78.02	78.90
	Positive	76.85	82.79	79.71	
30	Negative	83.98	76.00	79.79	80.75
	Positive	78.08	85.50	82.31	
40	Negative	84.28	77.75	80.88	81.63
	Positive	79.35	85.50	82.31	

50	Negative	83.91	79.50	81.64	82.13
	Positive	80.52	84.75	82.58	
60	Negative	84.68	78.75	81.61	**82.25**
	Positive	80.14	85.75	82.85	
70	Negative	83.96	78.50	81.13	81.75
	Positive	79.81	85.00	82.32	
80	Negative	84.30	76.50	80.21	81.13
	Positive	78.49	85.75	81.96	
90	Negative	84.53	76.50	80.31	81.25
	Positive	78.54	86.00	82.10	
100	Negative	84.34	76.75	80.37	81.25
	Positive	78.67	85.75	82.06	

To more intuitively observe and analyze the influence of word vector dimension on the classification effect of the model under different values, Fig. 6 shows the variation trend of the negative line, positive polarity and overall classification accuracy under different vector dimensions.

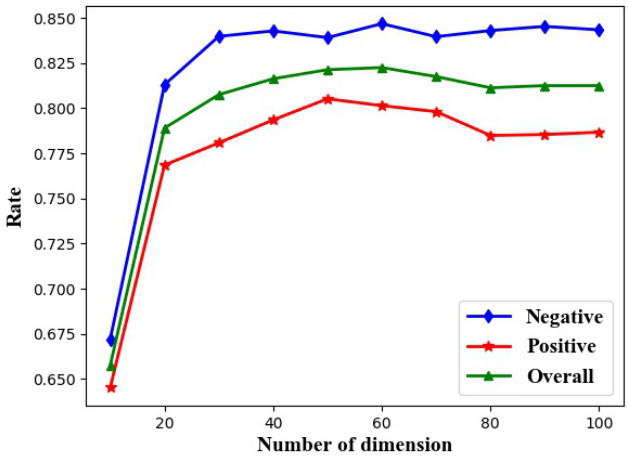

Figure 6: Trend of classification effect under different feature dimensions

From Tab. 3 and Fig. 6, we can see that when the word vector dimension is 10, the classification effect of the model is particularly poor. The reason may be that the vector dimension is too short to better represent the text information; when the word vector is increased from 10 to 20, the classification effect of the model has been greatly improved;

from 20 to 60, the classification effect still improves with the increase of the vector dimension, but its growth is slow and its amplitude is getting smaller, and the overall effect of the model is best when the vector dimension is 60-dimensional; When the vector dimension is larger than 60, the overall classification effect decreases and fluctuates up and down with a small amplitude, which indicate that the dimension of the word vector increasing can not only fail to better express the text information, but also bring the noise data to the model and then affect the classification effect.

In the literature of Pu et al. [Pu, Hou, Liu et al. (2017)], the optimal value of the word vector is 110, and when the corpus volume is 10,000, the overall classification accuracy of the model reaches 86.2%, which is higher than the best classification result of this paper. In theory, because the word vector used in this paper is obtained through training, the final classification effect should be higher than the method of initializing the word vector randomly. It is very likely that the Tibetan sentiment corpus collected in this paper covers a wider range of fields, and the number of samples in some areas is insufficient, resulting in the model not being able to learn the emotional characteristics of the field well. Therefore, this comparison is not strong scientific and rigorous. I hope that with the continuous development of informatization in Tibetan and other minority languages, relevant research institutions can launch relevant evaluation platforms, thereby achieving evaluation and comparison in the same corpus environment. It can better promote the progress and development of minority languages in the field of sentiment analysis.

4.5 Comparison and analysis of experimental results

In this paper, SVM experiments based on artificial extraction features, SVM Tibetan sentiment classification experiments based on algorithm extraction features, SVM Tibetan sentiment classification experiments based on multi-feature fusion, and Tibetan sentiment classification based on semi-supervised RAE model are respectively presented in the same dataset. The comparison of the results obtained by the four sets of experiments with their optimal parameters is shown in Fig. 7:

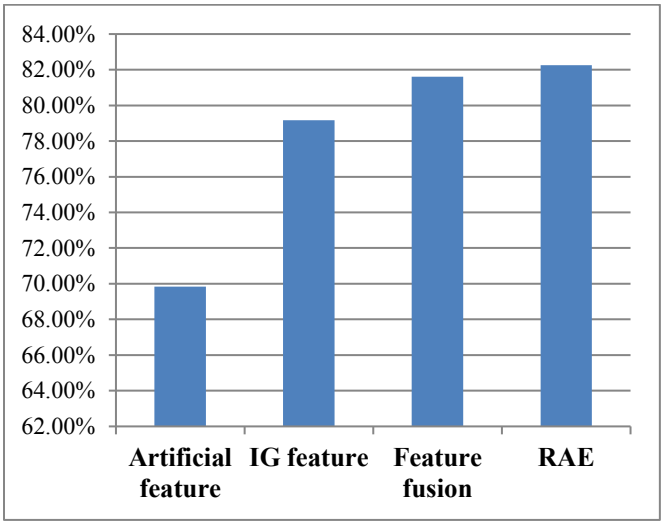

Figure 7: Comparison of experimental results

It can be seen from Fig. 7 that the classification effect of the Tibetan sentiment classification model based on semi-supervised RAE is better than that of other models in this paper, and the overall accuracy of classification is 82.25%. We could find the reason. SVM is a statistical machine learning method which can only learn the probability statistics of words, while semi-supervised RAE model used in this paper can perform distributed vector representation of the text sentences. The vector not only contains the statistical distribution information of the feature words in the text, but also learns the sentence context structure information of the text, which can better understand the text, so that achieving a better emotional classification result.

5 Conclusion

This chapter applies the semi-supervised RAE model to the emotional classification task of Tibetan short texts, and achieves a better classification effect. The method is compared with SVM experiments based on artificial extraction features, SVM Tibetan emotion classification experiments based on algorithm extraction features, and SVM Tibetan emotion classification experiments based on multi-feature fusion. The results show that the proposed method is superior to the other three classification methods.

Acknowledgment: The work in this paper is supported by the National Natural Science Foundation of China project "Research on special video recognition based on deep learning and Markov logic network" (61503424).

References:

Bengio, Y.; Ducharme, R.; Vincent, P.; Janvin, C. (2003): A neural probabilistic language model. *Journal of Machine Learning Research*, no. 3, pp. 1137-1155.

Collobert, R.; Weston, J. (2008): A unified architecture for natural language processing: Deep neural networks with multi-task learning. *Proceedings of the 25th International Conference on Machine Learning*, pp. 160-167.

Li, G.; Cheng, Y. Y.; Kou, G. Z. (2010): Sentence analysis of sentences and its key issues. *Library and Information Work*, vol. 54, no. 11, pp. 114-117.

Li, H. G.; Yu, H. Z. (2011): Tibetan text sentiment classification system design. *Gansu Science and Technology*, vol. 40, no. 1, pp. 106-107.

Pang, B.; Lee, L.; Vaithyanathan, S. (2002): Thumbsup sentiment classification using machine learning techniques. *Processing of the Conference on Empirical Methods in Natural Language Processing*, vol. 10, pp. 79-86.

PattunnaRajam, P.; Korah, R.; Kalavathy, G. M. (2018): Test vector optimization using pocofan-poframe partitioning. *Computers, Materials & Continua*, vol. 54, no. 3, pp. 251-268.

Pu, C. R.; Hou, J. L.; Liu, Y.; Zhai, D. H. (2017): Deep learning algorithms in Tibetan emotional analysis. *Journal of Frontiers of Computer Science and Technology*, no. 7, pp. 1-8.

Riloff, E.; Shepherd, J. (1997): A corpus-based approach for building semantic lexicons. *Processing of the Conference on Empirical Methods in Natural Language Processing*, pp.

117-124.

Riloff, E.; Shepherd, J. (1999): A corpus-based bootstrapping algorithm for semi-automated semantic lexicon construction. *Journal of Natural Language Engineering*, vol. 5, no. 2, pp. 147-156.

Riloff, E.; Wiebe, J.; Phillips, W. (2005): Exploiting subjectivity classification to improve information extraction. *Proceedings of the 20th National Conference on Artificial Intelligence*, vol. 20, no. 3, pp. 1106.

Riloff, E.; Patwardhan, S.; Wiebe, J. (2006): Feature subsumption for opinion analysis. *Processing of the Conference on Empirical Methods in Natural Language Processing*, pp. 440-448.

Riloff, E.; Wiebe, J. (2003): Learning extraction patterns for subjective expressions. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 105-112.

Riloff, E.; Wiebe, J.; Wilson, T. (2003): Learning subjective nouns using extraction pattern bootstrapping. *Proceedings of the Seventh Conference on Natural Language Learning*, pp. 25-32.

Wiebe, J.; Riloff, E. (2011): Finding mutual benefit between subjectivity analysis and information extraction. *IEEE Transactions on Affective Computing*, vol. 2, no. 4, pp. 175-191.

Xu, J.; Ding, Y. X.; Wang, X. L. (2007): Emotional automatic classification of news using machine learning methods. *Journal of Chinese Information Processing*, vol. 21, no. 6, pp. 95-100.

Zhang, J.; Li, Y. X. (2014): Sentiment analysis of Tibetan Weibo based on emotional dictionary. *Technology Innovation Forum*, pp. 220-222.

Zhao, Y. Y.; Qin, B.; Liu, T. (2010): Summary of text sentiment analysis. *Journal of Software*, vol. 21, no. 8, pp. 1834-1848.

Zhu, Y. L.; Min, J.; Zhou, Y. Q.; Huang, X. Q.; Wu, L. D. (2006): Vocabulary semantic tendency calculation based on HowNet. *Journal of Chinese Information Processing*, vol. 20, no. 1, pp. 16-22.