Research on Tibetan Text Classification Method Based on Neural Network

Zhensong Li [1,2], Jie Zhu[1,2], Zhixiang Luo[1,2], Saihu Liu [1,2]
[1] School of Electronics Engineering and Computer Science Tibet University
[2] Tibetan Information Technology National Joint Engineering Research Center
Lhasa, China
{lizhensong_3.14159, rocky_tibet, 2328866327, 593731006}@qq.com

***Abstract*—Text categorization is an important task in natural language processing, and it has a wide range of applications in real life. In this paper, two N-Gram feature models (MLP, FastText) and two sequential models (sepCNN, Bi-LSTM) are used to study the automatic classification for Tibetan text based on syllables and vocabulary. The experiment on Tibetan language data collected by China Tibet News Network shows that the classification accuracy is about 85%.**

Keywords- Tibetan text classification; neural network; N-Gram model; sequence model

I. Introduction

Text classification is to classify and mark text information according to certain rules and standards. Tibetan text categorization is an important research task of Tibetan information processing technology, which plays a fundamental role in search, public opinion analysis, document retrieval, and other application fields.

With the development of Tibetan information processing technology and the arrival of the era of big data, the Tibetan text data in websites, emails, forums, comments, and real-time communications are growing rapidly. These data contain abundant knowledge and bad information. To classify these data, we can analyze and process the information. And use brings great convenience.

At present, the text categorization process can be divided into feature engineering and classification model construction. Feature engineering can be divided into text preprocessing, feature extraction and text representation, to convert the original text into structured information that can be recognized and processed by computer. According to the difference of text preprocessing, the classification model can be divided into the model using a word set (bag of words) and the model using a word order (sequence model). Among them, the bag of words model includes logistic regression, support vector machine (SVM), Bayesian and Multi-Layer Perception, and the sequence model includes convolutional neural network (CNN) and cyclic neural network (RNN).

In this paper, we will use Multi-Layer Perception (MLP), fast text classifier (Fast Text), depthwise separable convolutional network (SepCNN) and Bi-directional Long Short-Term Memory network (Bi-LSTM) to study the Tibetan text classification method based on syllables and vocabulary.

II. RELATED WORK

In the early stage of text categorization, people used the rules constructed by professionals to mark the text and developed a classification method based on knowledge engineering. This method relies heavily on rules and has poor generalization [1]. Later, the emergence and development of machine learning replaced the rule classification method and became the mainstream method of text classification. The main machine learning methods in text classification are Bayesian classification [2], KNN classification [3], Logistic classification [4], SVM classification [5], neural network classification [6].

With the development of word embedding [7,8], the neural network has achieved better results than other machine learning methods in many aspects of natural language processing [9]. It has also become the preferred classification method and an important research direction for text categorization tasks.

In 2016, Facebook opened a text categorization tool, FastText [10], which uses only shallow neural networks. Compared with other categorization tools, it keeps the categorization effect while greatly shortening the training time. Convolutional neural networks have achieved great success in image processing. In 2014, Yoon Kim applied a convolutional neural network to text categorization task [11]. It used the convolution window to obtain local information of sequential text and achieved good classification results in many text categorization tasks. Cyclic neural networks have great advantages when dealing with variable length sequences. In many text classification tasks, variants of the cyclic neural network Bi-LSTM [12] can be found, which can acquire the relationship between context and context well, and the use of gate mechanism can control the length of memory and avoid the problem of long-term dependence.

At present, there are less research results in the Tibetan text classification. In 2011, Chuncheng Xiang [13] and others studied the automatic text classification of Tibetan web pages based on rules and implemented the classification tag of Tibetan web pages by establishing the Tibetan category feature vocabulary and comparing the entries of Tibetan web pages. Huiqiang Jia [14] used the KNN algorithm to classify Tibetan texts. In 2014, Renqing Nobu [15] used the maximum entropy model design to implement the Tibetan bad text recognition system. In 2018, Hongyun Jia [16,17] studied the Tibetan text classification method using Logistic and SVM models. It can be found that in the current Tibetan classification research, there are only a small number of classification studies based on rules and using traditional machine learning methods, and no literature on applying neural network models to Tibetan text classification research has been found. Because Tibetan lacks open classification test corpus, and the corpus used by each researcher is different, the experimental results vary greatly, and the accuracy is difficult to evaluate and analyze.

This paper experiments on the Tibetan corpus collected by China Tibet News Network, and compares the four neural network models of MLP, FastText, sepCNN and Bi-LSTM, based on the syllable and vocabulary classification of Tibetan text.

III. Neural Network Model

A. N-Gram feature model

MLP and FastText belong to the N-Gram feature model. They use N-Gram of lexical granularity and N-Gram of syllable granularity in preprocessing. The original sample can be regarded as a sequence of Tibetan syllables or Tibetan vocabulary $T_i = [w_1, w_2 \cdots w_n]$, where T_i represents the nth original sample and w_n is the nth syllable or vocabulary in text T_i . The N-Gram operation is performed on the text to generate the corresponding N-Gram feature set $G_i = [x_1, x_2 \cdots x_m]$, where G_i is the N-Gram feature set generated by text T_i, and x_m is an N-Gram feature in the G_i feature set. In this paper, three N-Gram methods, unigram, bigram, and trigram, are used to generate the N-Gram feature set of Tibetan syllables and the N-Gram feature set of Tibetan vocabulary. Specific Tibetan N-Gram feature generation is shown in TABLE I.

TABLE I. RESULTS OF N-GRAM FEATURE GENERATION IN TIBETAN

sentence	སློབ་མ་རྣམས་རྩེད་མོ་རྩེ་བ(The students are playing.)
syllable	སློབ མ རྣམས རྩེད མོ རྩེ བ
N-Gram Characteristics of Syllable Granularity	‘སློབ’, ‘སློབ མ’, ‘སློབ མ རྣམས’, ‘མ’, ‘མ རྣམས’, ‘མ རྣམས རྩེད’, ‘རྣམས’, ‘རྣམས རྩེད’, ‘རྣམས རྩེད མོ’, ‘རྩེད’, ‘རྩེད མོ’, ‘རྩེད མོ རྩེ’, ‘མོ’, ‘མོ རྩེ’, ‘མོ རྩེ བ’, ‘རྩེ’, ‘རྩེ བ’, ‘བ’
vocabulary	སློབ་མ རྣམས རྩེད་མོ་རྩེ་བ
N-Gram Characteristics of Lexical Granularity	‘སློབ་མ’, ‘སློབ་མ རྣམས’, ‘སློབ་མ རྣམས རྩེད་མོ་རྩེ་བ’, ‘རྣམས’, ‘རྣམས རྩེད་མོ་རྩེ་བ’, ‘རྩེད་མོ་རྩེ་བ’

In-text representation, all N-Gram feature sets G are merged to remove repetitive generated list $L = [g_1, g_2 \cdots g_l]$, and all N-Gram features in list L are numbered, where g_l is the lth N-Gram feature in list L.

In MLP model, TF-IDF algorithm is used to process text into TF-IDF vector $V(MLP)_i = [t_1, t_2 \cdots t_l]$, where $V(MLP)_i$ represents the TF-IDF vector of N-gram feature set G_i and t_l represents the TF-IDF value of the lth feature in list L in the T_i text. The formula for calculating t_l value is (1):

$$t_l = \frac{fnum(G_i, g_l)}{fnum(G_i)} \cdot log \frac{dnum(all)}{dnum(all, g_l)+1} \quad (1)$$

Among them, $fnum(G_i, g_l)$ represents the number of occurrences of g_l features in N-Gram feature set G_i in list L, $fnum(G_i)$ is the total number of occurrences of all features in N-Gram feature set G_i, and the value of the front part of multiplication sign is Term Frequency (TF). The rear part of the multiplication sign is Inverse Document Frequency (IDF), $dnum(all)$ represents the total number of documents, and $dnum(all, g_l)$ represents the number of documents containing g_l features, and denominator plus 1 prevents the document set from containing g_l features.

MLP classifier uses a fully connected layer and softmax classification layer. The calculation formula between the input layer and the fully connected layer is as follows (2):

$$a_i^{(1)} = f(W^{(1)} V(MLP)_i + b^{(1)}) \quad (2)$$

$V(MLP)_i$ is the input of text T_i in the neural network. $a_i^{(1)}$ is the value of all neurons in the first layer of the neural network after input text T_i; f is the activation function of the first layer; $W^{(1)}$ is the weight matrix of all neurons in the first layer and the input layer; and $b^{(1)}$ is the bias unit of all neurons in the first layer.

The word vector is used in the FastText model. First, each feature in list L is coded with one-hot. Then, according to the one-hot vector encoded by list L, the sample is transformed into one-hot vector expression $O(FT)_i = [o_1, o_2 \cdots o_m]$, where $O(FT)_i$ represents one-hot vector set of sample T_i and o_m represents one-hot vector corresponding to feature x_m in N-Gram feature set G_i. Finally, the one-hot vector set is transformed into a dense vector through the word vector layer, and the corresponding set of word vectors $V(FT)_i = [v_1, v_2 \cdots v_m]$ is obtained. $V(FT)_i$ represents the set of word vectors of sample T_i. v_m is the word vector corresponding to o_m in one-hot vector set $O(FT)_i$.

FastText uses the average pooling layer and the softmax classification layer, and the formula between the input layer and the average pooling layer is (3):

$$aver_i = \frac{1}{m} W . \sum_{j=1}^{m} v_j^i \quad (3)$$

Where $aver_i$ represents the weighted average of all word vectors of the text T_i, the weight is W, m represents the number of text T_i word vectors, and v_j^i is the jth word vector in the word vector set $O(FT)_i$.

B. Sequence model

The sepCNN and Bi-LSTM models used in this paper belong to the sequence model. The original sample can be regarded as the sequence $T_i = [w_1, w_2 \cdots w_n]$ of Tibetan syllables or Tibetan vocabulary. w_n is the nth syllable or vocabulary in this text, and it is also the nth feature. T_i can also be regarded as a feature sequence. The feature sequence is transformed into the word vector sequence $V(sequence)_i = [v_1, v_2 \cdots v_n]$ by word vector operation. $V(sequence)_i$ is the corresponding word vector sequence of T_i. v_n represents the word vector of w_n feature in T_i after training.

SepCNN is a variant of CNN that mentioned in the conference paper of CVPR (Computer Vision and Pattern Recognition Conference) in 2017 [18]. It decomposes ordinary convolution operation into a depthwise process and a pointwise process. Compared with conventional CNN, the number of parameters of sepCNN is reduced and the training efficiency is improved.

In this paper, sepCNN classifier uses deep separable convolution layer, maximum pooling layer, average pooling layer, and softmax classification layer, in which the input word vector sequence is convoluted, such as formula (4):

$$A_i^{[1]} = g^{[1]}(W^{[1]} \star V(sequence)_i + b^{[1]}) \quad (4)$$

Among them, $V(sequence)_i$ is the input of text T_i in the neural network; $A_i^{[1]}$ is the value of all neurons in the first convolution layer after input text T_i ; $g^{[1]}$ is the activation function of the first convolution layer; $W^{[1]}$ is the convolution core used by the first convolution layer; $\star$ is the convolution operation ; and $b^{[1]}$ is the bias term.

Bi-LSTM classifier uses Bi-direction Long Short-Term Memory layer, fully connected layer and soft Max classification layer. The Bi-direction Long Short-Term Memory layer has one update gate, one forget gate, and one output gate, wherein the formula of update gate is as shown in (5):

$$\Gamma_u^{<n>} = \sigma(W_u[a^{<n-1>}, v^{<n>}] + b_u) \quad (5)$$

Among them, $a^{<n-1>}$ is the activation value of $n-1$ time; $v^{<n>}$ is the input of n time; W_u is the weight matrix of the update gate; b_u is the bias unit of the update gate. The formula of forget gate is as shown in (6):

$$\Gamma_f^{<n>} = \sigma(W_f[a^{<n-1>}, v^{<n>}] + b_f) \quad (6)$$

W_f is the weight matrix of the forget gate and b_f is the bias unit of the forget gate. The formula of output gate is as shown in (7):

$$\Gamma_o^{<n>} = \sigma(W_o[a^{<n-1>}, v^{<n>}] + b_o) \quad (7)$$

W_o is the weight matrix of the output gate and b_o is the bias unit of the output gate. The formula of the memory cell is as shown in (8), (9):

$$\tilde{c}^{<n>} = tanh(W_c[a^{<n-1>}, v^{<n>}] + b_c) \quad (8)$$

$$c^{<n>} = \Gamma_f^{<n>} * c^{<n-1>} + \Gamma_u^{<n>} * \tilde{c}^{<n>} \quad (9)$$

W_c is the weight matrix of memory cells and b_c is the biased unit of memory cells. The formula of activation value is as shown in (10):

$$a^{<n>} = \Gamma_o^{<n>} * tanh(c^{<n>}) \quad (10)$$

The activation value of the current neuron is determined by the memory cell value and the output gate. In this paper, Bi-direction time series is used, and its formula is (11):

$$y^{<n>} = g(w_y[\vec{a}^{<n>}, \overleftarrow{a}^{<n>}] + b_y) \quad (11)$$

$\vec{a}^{<n>}$ is the positive activation value of the neural network at time n, and $\overleftarrow{a}^{<n>}$ is the reverse activation value of the neural network at time n.

IV. EXPERIMENTS

A. Experimental Data

Because there is no open Tibetan text categorization corpus, Python is used to implement the crawler system, and the news corpus is crawled from "China Tibet News Network", with a total of 66 622 articles in 9 categories. The data set was cleaned to remove the text whose length was too short to express the clear meaning, and the final number of texts was 66310. Because the number of texts is different, each type is divided into 80%, 10% and 10% data sets, which are used as the training set, the validation set, and the test set respectively. The specific number is shown in TABLE II.

TABLE II. DATA SET DISTRIBUTION STATISTICS

categories	training	validation	test
government affairs	15000	1875	1875
farming	1648	206	206
legal	15992	1999	1999
scientific education	6880	860	860
Literature and Art	6816	852	852
religion	1784	223	223
medicine	1224	153	153
Historical geography	568	71	71
Eco-tourism	3136	392	392
total	53048	6631	6631

The non-Tibetan characters are cleaned when the text is processed. In the syllable level experiment, the Tibetan syllable-dividing mark is used as the segmentation point, and then the syllable-dividing mark is removed, and the syllables are separated by spaces. In the vocabulary level experiment, the data were segmented using the Tibetan automatic word segmentation system of Tibetan University. After processing, the words were separated by spaces, and the syllables contained in each word are connected by Tibetan syllable-dividing mark.

B. Parameter Settings and Evaluation Metrics

In this experiment, the neural network is constructed using the TensorFlow framework. The batch size is set to 1024; the learning rate is set to 0.001 using Adam optimizer; the loss function uses Multi-class log loss.

In MLP classifier, the number of neurons in the fully connected layer is set to 64, and the activation function uses ReLU.

The average pooling layer in FastText classifier uses GlobalAveragePooling1D.

Four deep separable convolution layers are set in sepCNN, the length of the convolution window is set to 3, the number of output channels of the first two convolution layers is set to 64, and the number of output channels of the last two layers is set to 128.

In Bi-LSTM, the output dimension of Bi-direction Long Short-Term Memory layer is set to 64, and the number of neurons in the fully connected layer is set to 64.

In this experiment, accuracy, precision, recall, and F1-Measure were used as evaluation criteria.

C. Experimental Results and Discussions

TABLE III shows the accuracy of MLP, FastText, sepCNN and Bi-LSTM based on Tibetan syllables or vocabularies in the training set, the validation set, and the test set. It can be found from the comparison table that the accuracy of each model based on vocabulary classification is higher than that based on syllable classification in the test set. This is because Tibetan words are the smallest unit of semantic load rather than syllables.

TABLE III. ACCURACY RESULTS BASED ON SYLLABLE AND VOCABULARY MODELS

models		training (%)	validation (%)	test (%)
MLP	***syllable***	93.77	84.63	84.15
	vocabulary	94.64	85.07	84.68
FastText	***syllable***	91.91	86.65	86.25
	vocabulary	91.92	86.58	86.38
sepCNN	***syllable***	87.09	83.67	83.09
	vocabulary	91.60	83.37	83.56
Bi-LSTM	***syllable***	88.83	84.77	84.20
	vocabulary	91.98	85.97	85.55

In TABLE III, by comparing the accuracy of the training set, the validation set and the test set, it can be found that the accuracy of the training set is significantly higher than that of the other two sets, which is due to the existence of over-fitting phenomenon in training. Some problems can be found in the analysis of samples. Samples vary in length. The shortest sample has only one sentence with four syllables, while the longer sample has thousands of sentences. The distribution of samples is uneven. TABLE II shows that the largest number of legal samples is 19 990, while the smallest number of historical and geographical samples is 710, accounting for only 1/28 of legal samples. The number of samples is relatively small. In this experiment, there are only 60,000 Tibetan News Texts in 9 categories, with an average of 7,000 in one category. It is difficult to compare the number of data sets with those of Chinese and English news classifications. The sample quality is not good; some samples have little information content; the classification of the original corpus is mixed. So, it is easy to have over-fitting problems in training.

TABLE IV and TABLE V shows the detailed precision, recall, and F1-Measure of each model in the test set.

TABLE IV. PERFORMANCE COMPARISON OF SYLLABLE-BASED MODELS ON TEST SETS

		government affairs	farming	legal	scientific education	Literature and Art	religion	medicine	Historical geography	Eco-tourism
MLP	***precision (%)***	90.52	68.66	87.95	71.00	81.03	73.86	89.38	54.84	78.12
	recall (%)	97.28	44.66	90.95	74.30	83.22	58.30	66.01	23.94	63.78
	F1-Measure(%)	93.78	54.12	89.42	72.61	82.11	65.16	75.94	33.33	70.22
FastText	***precision (%)***	93.91	64.58	90.73	78.01	78.41	74.19	87.80	64.52	79.55
	recall (%)	96.21	60.19	91.55	78.37	86.97	61.88	70.59	28.17	71.43
	F1-Measure(%)	95.05	62.31	91.14	78.19	82.47	67.48	78.26	39.22	75.27
sepCNN	***precision (%)***	93.79	68.66	86.00	70.90	79.31	73.94	88.18	46.67	66.36
	recall (%)	91.04	44.66	91.00	77.33	83.22	54.71	63.40	9.86	74.49
	F1-Measure(%)	92.40	54.12	88.43	73.97	81.21	62.89	73.76	16.28	70.19
Bi-LSTM	***precision (%)***	92.59	58.13	87.70	75.03	86.19	63.24	78.57	36.59	74.68
	recall (%)	93.92	57.28	91.00	73.72	79.11	71.75	71.90	21.13	74.49
	F1-Measure(%)	93.25	57.70	89.32	74.37	82.50	67.23	75.09	26.79	74.58

TABLE V. PERFORMANCE COMPARISON OF VOCABULARY-BASED MODELS ON TEST SETS

		government affairs	farming	legal	scientific education	Literature and Art	religion	medicine	Historical geography	Eco-tourism
MLP	***precision (%)***	93.50	80.28	88.62	70.02	79.43	69.68	85.83	59.57	76.40
	recall (%)	96.59	55.34	91.95	75.23	81.10	58.74	71.24	39.44	62.76
	F1-Measure(%)	95.02	65.52	90.25	72.53	80.26	63.75	77.86	47.46	68.91
FastText	***precision (%)***	94.62	75.00	89.69	78.02	80.42	71.28	88.35	69.70	74.30
	recall (%)	95.68	61.17	93.15	78.02	86.27	60.09	59.48	32.39	74.49
	F1-Measure(%)	95.15	67.38	91.39	78.02	83.24	65.21	71.09	44.23	74.39
sepCNN	***precision (%)***	96.52	77.14	90.50	72.03	73.68	62.26	74.65	56.67	65.87
	recall (%)	88.64	52.43	90.50	76.98	87.09	59.19	69.28	23.94	77.30
	F1-Measure(%)	92.41	62.43	90.50	74.42	79.83	60.69	71.86	33.66	71.13
Bi-LSTM	***precision (%)***	91.07	66.50	91.91	80.08	79.61	67.13	80.00	60.98	74.69
	recall (%)	96.27	63.59	90.95	71.05	85.68	65.02	67.97	35.21	77.55
	F1-Measure(%)	93.60	65.01	91.43	75.29	82.53	66.06	73.50	44.64	76.10

From TABLE II, TABLE IV and TABLE V, all kinds of F1-Measures are closely related to the number of samples it has. In all classification models, the F1-Measures of the category of "historical geography" are the lowest, while the F1 values of the two categories of "government affairs" and "legal" which have the largest number of samples are very high. Comparing their F1-Measures based on the syllable and vocabulary in each model, we can find that in most categories, and the F1-Measures are improved after having finished word segmentation. This is because of the structural characteristics of Tibetan; the use of word segmentation is better than syllable-based classification. Because the accuracy of the Tibetan word segmentation system is not too high, the F1-Measure has not been significantly improved after the existence of some categories of word segmentation.

Compared with the MLP model and FastText model in TABLE IV and TABLE V, FastText is superior to the MLP model in accuracy, precision, recall, and F1-Measure. These two models belong to shallow neural networks in structure. The difference is that MLP uses TF-IDF coding while FastText uses word embedding. Word embedding has more advantages than TF-IDF in terms of meaning expression and meaning association. It can be found that the FastText model has a good classification effect, and because of its simple structure, it is also fast in training.

Observation TABLE IV and TABLE V show that compared with other models, the accuracy, recall and F1 values of sepCNN are not very high. Compared with the advantage of convolutional neural networks in image processing, it has no great advantage in text processing. The reason for this may be related to the data samples used in this experiment, or it may be because only the basic convolutional network is used without improvement.

In TABLE IV and TABLE V, the cyclic neural network model Bi-LSTM performs well, because the Bi-LSTM model can get the context information of words in text very well and is good at long-distance information transmission of text. However, due to its complex structure and the problems of the small amount of data and uneven distribution in the Tibetan text set, it is very easy to produce over-fitting when training the Bi-LSTM model. Moreover, the Bi-LSTM model is difficult to parallelize in training due to its cyclic structure. Therefore, compared with other neural network models, the training time is too long, and the training cost is very high.

V. CONCLUSION AND FUTURE WORK

This research selects the typical models of full-connected neural network, convolution neural network, and cyclic neural network and apply them to Tibetan text categorization. The classification results based on Tibetan syllables and vocabulary were summarized and analyzed. The classification effect based on Tibetan vocabulary is better than that based on Tibetan syllables, and the use of neural networks can better solve the problem of Tibetan text classification. Because there is no open standard classification sample set in Tibetan corpus, the crawler technology is used to collect data in this study. There are some problems in the sample set, such as the small amount of data and the unbalanced distribution of data. After that, we will expand and improve the sample set, and further study the application of neural network in Tibetan information processing, to improve the effect of Tibetan text classification.

ACKNOWLEDGMENT

This paper is supported by The National Natural Science Foundation of China (61751216), the National Team of Computer and Tibetan Information Technology and the Construction of Key Laboratories (Tibetan Finance and Education Instruction [2018] No. 81).

REFERENCES

[1] M. Sasaki, and K. Kita, "Rule-based text categorization using hierarchical categories," SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.98CH36218), San Diego, CA, USA, 1998, pp. 2827-2830 vol.3.

[2] M. E. Maron, and J. L. Kuhns, "On Relevance, Probabilistic Indexing and Information Retrieval," J. ACM, vol. 7, no. 3, pp.216-244, Jul. 1960.

[3] B. Trstenjak, S. Mikac, and D. Donko, "KNN with TF-IDF based Framework for Text Categorization," Procedia Engineering, vol. 69, pp.1356-1364, 2014.

[4] D. Liu, T. Li, and D. Liang, "Incorporating logistic regression to decision-theoretic rough sets for classifications," International Journal of Approximate Reasoning, vol. 55, pp.197-210, Jan. 2014.

[5] L. Demidova, I. Klyueva, Y. Sokolova, N. Stepanov, N. Tyart, "Intellectual Approaches to Improvement of the Classification Decisions Quality on the Base of the SVM Classifier," Procedia Computer Science, vol. 103, pp. 222-230, 2017.

[6] G. Vinodhini, and R. M. Chandrasekaran, "A comparative performance evaluation of neural network based approach for sentiment classification of online reviews," Journal of King Saud University - Computer and Information Sciences, vol. 28, pp. 2-12, Jan. 2016.

[7] G. E. Hinton, "Learning distributed representations of concepts," Eighth Conference of the Cognitive Science Society, 1989, pp. 1-12.

[8] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," arXiv preprint, arXiv:1301.3781, 2013.

[9] J. Hartmann, J. Huppertz, C. Schamp, and M. Heitmann, "Comparing automated text classification methods," International Journal of Research in Marketing, vol. 36, pp. 20-38, Mar. 2019.

[10] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," arXiv preprint, arXiv:1607.04606, 2016.

[11] Y. Kim, "Convolutional Neural Networks for Sentence Classification," arXiv preprint, arXiv:1408.5882, 2014.

[12] M. Schuster, and K. K. Paliwal, "Bidirectional recurrent neural networks," in IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673-2681, Nov. 1997.

[13] C. Xiang, Y. Weng, X. Zhao, G. Yang, "Automatic Text Classification of Tibetan Web Pages Based on Column," Journal of Chinese Information Processing, vol. 25, no. 04, pp. 20-23, 2011.

[14] H. Jia, "Research on Key Technologies of Tibetan Text Classification Based on KNN Algorithms," Journal of Northwest University for Nationalities (Natural Science), vol. 32, no. 03, pp. 24-29, 2011.

[15] R. Nobu, Y. Su, Y. Sun, "Design and Implementation of Tibetan Bad Text Recognition System Based on Maximum Entropy Model," Tibet's Science and Technology, no. 03, pp. 77-78, 2014

[16] N. Qun, H. Jia, "Research and Implementation of Based on the Logistic Regression Model for Tibetan Text Classification," China Computer & Communication, no. 05, pp. 70-73, 2018.

[17] H. Jia, N. Qun, H. Su, L. Ciren, Z. Basang, "Research and Implementation of Tibetan Text Classification Based on SVM," Electronic Technology & Software Engineering, no. 09, pp.144-146, 2018.

[18] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 1800-1807.