Study of Tibetan Text Classification based on fastText

Wei Ma[1, a], Hongzhi Yu[1], Jing Ma[2, b]

[1]Key Laboratory of China's Ethnic Languages and Information Technology of Ministry of Education Northwest Minzu University Lanzhou, P. R. China

[2]Personnel Department Gansu Political Science and Law Institute Lanzhou, P. R. China

[a]mawei@xbmu.edu.cn, [b]mw1231@qq.com

Abstract. Tibetan text classification is an important research topic in Tibetan information processing. In this paper, we attempt to apply fastText text classification tool and fastText pre-training word vectors for Tibetan text classification. In the experiment, For the Tibetan language corpus segmented by Tibetan syllable points, we represent all the words in each document with the fastText pre-training word vectors, and then average all the word vectors in this data. The average vector (docvec) represent each piece of document, we put it into SVM classifier, and the results show that the model outperforms competitive the traditional Tibetan text classification method, and the F-measure has improved by 10%.

Keywords: text classification, Tibetan text, fastText.

1. Introduction

The rapid development of the information age, brought unprecedented a large number of texts, video, pictures, audio and other forms of data, the Internet application software can support more than 300 kinds of languages. The news data produces a lot of text class data, how to get valuable information in the messy news text data, becomes very important. Text classification, as a natural language processing technology which can classify messy text, can not only effectively manage text information, but also plays an important role in information retrieval, automatic document summary and so on.

Tibetan language research has made great progress, but its development is relatively lagging behind other resources-rich and widely used languages [1]. Due to the complexity of the Tibetan language, the study of Tibetan text classification is a challenging task. In languages such as English, spaces are used as natural delimiters between words, and as Chinese, there is no obvious distinction between Tibetan words. Text classification is a supervised learning process. First, an accurate model is used to train an optimal model, and then the model is used to classifier the data and output the classification result. The accuracy of the classification results reflects the accuracy of the model. Before the model is trained, the text needs to be represented and transformed into a form that the existing classification algorithm can handle. The current text representation methods are mainly Bag-of-Words (BOW) and distributed representation methods. The typical representative of distributed representation is Word Embedding. Word Embedding was first proposed by Hinton [2]in 1986. The word vector is a dense and low-dimensional real-value vector, and each dimension is represented by a real number. It also represents semantic and grammatical features, so that words with certain semantic relations are closer in the mathematical sense. This method better solves the dimensionality disaster problem existing in the traditional One-hot method, and also incorporates the semantic relationship between words into the representation of the text. The Word2Vec model is a word vector training model proposed by Mikolov et al. [3]in 2013. It is used to train learning words and phrase vectors, and the processing of text content is transformed into vector operations in vector space. The semantic similarity of the text is represented by the similarity in the vector space. Later, researchers extended the word vector representation to a phrase-level or sentence-level representation.

The Internet is continuously bringing us a huge amount of data. Faced with such large-scale data, fast and effective information processing technologies are emerging one after another. The traditional text classification method is slow, and it has been unable to meet the needs of high-speed information processing. For this reason, more researchers have focused their research on how to shorten the time

of text processing while satisfying performance. Therefore, it is urgent to design new methods that can shorten the accuracy of text classification while shortening Classification time. In 2016, Facebook AI Research proposed fastText [4]. The experimental data shows that by using a standard multi-core CPU, more than 1 billion words can be trained in 10 minutes, and also classify a half-million sentences among more than 300,000 categories in less than five minutes. In text categorization tasks, fastText can often achieve accuracy comparable to deep networks, but it is much faster than the deep learning method in training time. This paper attempts to apply the fastText text classification tool and the fastText Tibetan pre-training word vectors to the Tibetan text classification, and compare it with the traditional text classification method, and count the text classification results of different methods.

The rest of the paper is organized as follows; the second part is a brief review of the previous classification of Tibetan texts. The third part is a new way of expressing the Tibetan text. The fourth part gives the experimental results and analyzes them. Finally, in the fifth part, we draw conclusions and outline future work.

2. Related Work

In recent years, the classification of Tibetan texts has received more and more attention. Jiang tao[5]used the distributed representation of Tibetan words as a feature to significantly improve the performance of Tibetan text classification. Cao Hui [6] proposed an improved TF-IDF weighting algorithm. Jia Huiqiang[7] used the KNN algorithm to automatically classify Tibetan documents. Xu Guixian[8][9] introduced a Tibetan web page classification method, which uses a feature dictionary and cosine similarity algorithm to classify Tibetan web pages. Jiang Tao [10] uses a Tibetan text classifier based on SVM and NB algorithm to implement Tibetan word segmentation using CRF statistical methods [11]. The existing Tibetan classification system mainly adopts rules-based and statistical methods. The rule-based classification method requires manual construction of the classification dictionary, and the system scalability is not ideal. The traditional VSM method is used to represent documents lacking semantic information representation. Text preprocessing and text modeling methods have an important impact on Tibetan text categorization. This paper attempts to apply the fastText text classification tool and the fastText Tibetan pre-training word vectors to the Tibetan text classification. Our contribution is that this is the first attempt to introduce fastText into the Tibetan text classification. The experiment proves that the model is effective and has good classification results.

3. Method

The fastText algorithm is a supervised model, similar to the CBOW architecture of word2vec, and its structure is shown in Figure 1. CBOW predicts intermediate words through context, while fastText predicts tags through context (this tag is the type of text, which is determined by manual annotation). From the model architecture, like CBOW, the fastText model also has Three layers: input layer, hidden layer, output layer (Hierarchical Softmax, input is a number of words and their n-gram features, these features are used to represent a single document, the hidden layer is the superimposed average of multiple feature vectors, The hidden layer solves the maximum likelihood function, then constructs a Huffman tree according to the weights and model parameters of each category, and uses the Huffman tree as the output. If you use normal Softmax training, each label needs to be calculated, With the Huffman tree, the number of tags is large, the weight is high, and the Huffman coding is naturally shorter so that calculating the tags according to the Huffman coding path can greatly reduce the amount of calculation.

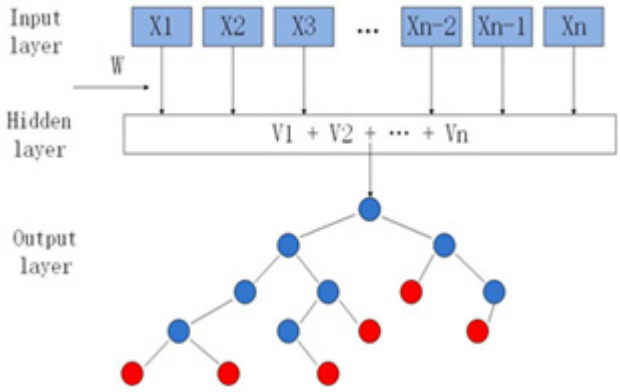

Fig 1. Structure of fastText

The figure1 shows the model of fastText. The weight value w can be regarded as a word lookup table for a certain piece of text. The representation of the word is then averaged into a representation of the text, which is then fed back to the linear classifier. Text representation is an implicit variable that can be reused. The difference between this architecture and the CBOW model is that the intermediate word in the CBOW model is replaced by a label. The SoftMax function is used in the fastText model to calculate the probability distribution of the category. For a set of datasets containing N documents, the goal of the model is achieved by minimizing the formula (1).

$$-\frac{1}{N}\sum_{n=1}^{N} y_n \log(f(BAX_n)) \tag{1}$$

Where N is the number of samples, y_n is the category corresponding to the nth sample, f is the loss function, X_n is the normalized feature of the NTH sample, A is the weight matrix (building words, embedding layer), B is the weight Matrix (hidden layer to output layer). The model trained by stochastic gradient descent and learning rate linear decay. fastText can achieve good results and fast speed, mainly because of two important factors, firstly using the sub-word n-gram information, and secondly using a Huffman coding tree-based hierarchical Softmax method.

3.1 N-gram Features

Most of the text vectorization research work is based on independent words in the text as the basic unit for training and learning. fastText introduces the concept of subword n-gram to solve the problem of morphology. It splits a word into character levels and uses character-level n-gram information to capture the order relationship between characters. During the training process, the corresponding training of each n-gram becomes a vector, and the word vector of the original complete word is obtained by summing all the vectors of its corresponding n-grams. All word vectors and character-level n-gram vectors are simultaneously summed and averaged as input to the training model.

3.2 Layered Softmax

Softmax is a generalization of logistic regression on multi-classification tasks and is the last layer in training neural networks. In general, Softmax takes the output of the hidden layer as input, and after linear and exponential transformation, performs global normalization to find the output with the highest probability. When the vocabulary quantity V is large (generally on the order of hundreds of thousands), the Softmax calculation is very costly and is of the O(V) magnitude. The idea of Hierarchical Softmax essentially transforms a global multi-classification problem into several binary classification problems, thus reducing the computational complexity from O(V) to O(logV). fastText uses a layered Softmax based on the Huffman tree. All internal nodes of the Huffman tree are similar to neurons in the hidden layer of the neural network, where the word vector of the root node corresponds to the projected word vector, and all leaf nodes are similar to the neurons of the neural network Softmax output layer, and the number of leaf nodes is the size of the vocabulary. The Huffman tree is constructed according to the number of occurrences of each category as the weight.

The more the number of occurrences of the sample, the shorter the path. If the depth of the node is l+1, the parent node is $n_1, \ldots, n_l$, then its probability is:

$$P(n_{l+1}) = \prod_{i=1}^{l} P(n_i) \tag{2}$$

This means that the probability of a node is always smaller than its parent.

4. Experiment

In this section, we present our experimental results and perform some analyses to better understand our models.

4.1 Dataset

Although researchers have made gratifying achievements in the classification of Tibetan texts, there is currently no public data set for evaluating Tibetan text classifications compared to a large number of English-Chinese text classification public corpora. Therefore, we choose the China Tibet News (Tibetan Edition), People's Daily (Tibetan Edition), Tibet Daily (Tibetan Edition) and other websites as our data source. These documents are divided into five categories: politics, economics, law, people, medicine and education. We used the pre-processing technique to eliminate text with a small amount of data and collected 7,000 articles. The corpus is divided into training sets and test sets. The training set accounts for 80% of the data set, with 5600 documents, the test set accounts for 20%, and there are 1400 documents. To evaluate the effect of different word segmentation methods on text classification, we construct two text classification datasets and divide the obtained documents into two ways. One is to divide the words according to the Tibetan word by the word segmentation tool, and obtain the Tibetan word segmentation data set Tibetan Classification Corpus 1 (TCC1), the other is to use the Tibetan Syllable point to segment, obtain the Tibetan Syllable point segmentation[12] data set Tibetan Classification Corpus 2 (TCC2), each of the tokens in two sets are separated by space and compared The effect of TCC1 and TCC2 in different classification methods. Every single text in all data is a line, and the "__label__ + tag" is added at the beginning of each line.

Pre-training data set: fastText publishes word vectors in 157 languages [13], which are trained on Common Crawl and Wikipedia using fastText. For Tibetan, they chose to use the Syllable point for word segmentation, and then train the word vectors, which contains 77425 vectors, vector dimension is 300, with character n-grams of length 5, a window of size 5 and 10 negatives.

4.2 Performance Measures

To evaluate the effectiveness of category assignments with classifiers to document, we adopt the precision, recall, and F1 measures that are widely used in text classification field. Precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances, while recall (also known as sensitivity) is the fraction of relevant instances that have been retrieved over the total amount of relevant instances. F1 measures that combine precision and recall is the harmonic mean of precision and recall, the traditional F-measure or balanced F-score:

$$F_1 = \frac{2RP}{R+P} \tag{3}$$

Precision and recall are then defined as:

$$p = \frac{tp}{tp+fp} \tag{4}$$

$$R = \frac{tp}{tp+fn} \tag{5}$$

tp, fn and fp are the number of true positives, false negatives, and false positives, respectively.

4.3 Results and Discussions

In this section, we present the results of the experiments conducted to demonstrate the effectiveness of the proposed method with two groups of the experiment.

We conducted experiments on different methods on two corpora. In order to verify the effect of fastText on the classification of Tibetan text, the results of Tibetan text classification are shown in Table 1.

Table 1. Performances on document classification.

Dataset	Model	P	R	F-measure
TCC1	SVM+TF-IDF	59.52	60.35	59.93
	SVM+Onehot	66.71	64.56	65.62
	fastText	65.31	62.48	63.86
TCC2	SVM+TF-IDF	70.13	68.23	69.17
	SVM+Onehot	73.28	72.31	72.79
	SVM+fastText	**81.19**	**75.14**	**78.05**
	fastText	70.23	71.63	70.92
	CNN+fastText	57.42	53.26	55.26
	LSTM+fastText	55.31	54.18	54.73

For two datasets obtained according to different word segmentation methods, TCC1 and TCC2, we evaluate the effectiveness of the fastText method based on comparison with baseline models such as naive Bayesian classifier (NB) and support vector machine (SVM). In addition, we also studied the performance of the Tibetan syllables as a neural networks model input. For the dataset TCC1 based on Tibetan words, first we use the most common SVM-based method, select TF-IDF as the feature, then use Onehot as the feature to train in the SVM, and finally use the FastText tool to build the text classifier, For the data set TCC2, we first use the same SVM+TF-IDF and SVM+Onehot methods for model training. Secondly, we use the fastText pre-training word vectors for each word in each data, and then All the word vectors in the data are averaged to obtain a document vector(docvec), its dimension is 300, which is used to represent each piece of data. All docvecs in the data set is trained in SVM to obtain the model. Third, we use the fastText tool to build a text classifier. Finally, we will convert all news corpora into a word index sequence, and combine the fastText pre-training word vector to generate a word vector matrix, and load the word vector matrix into the embedding layer of CNN [14][15] and LSTM [16][17][18]. and then carry out training.

The experimental results of the above methods are shown in Table 1. It can be seen that the difference between the scores obtained by the FastText-based method in the dataset TCC1 and the scores obtained by the traditional method is small, which is believed to be related to the size of the dataset and the accuracy of the word segmentation. For the data set TCC2, compared with the results of TCC1, the results of various methods have improved, especially the SVM+fastText method has been significantly improved, increased by more than 10%, fully demonstrating the fastText pre-training word vectors is very reliable. The classification results on the two datasets only using the fastText text categorization tool did not show a significant improvement. The classification results based on CNN and LSTM are not ideal, mainly because the size of the dataset is small and does not show the advantages of deep learning [19].

5. Conclusion

For the classification of Tibetan texts, this paper introduces fastText, the motivation is to explore its practicality in the field of Tibetan text classification, and try to use the fastText pre-training word vector. In text preprocessing, we use two segmentation methods, word segmentation, and syllable segmentation, respectively. Two datasets are obtained, we use SVM classification method and end-to-end neural network model training method. The results show that the combination of the fastText pre-training word vectors and the traditional SVM classifier can effectively improve the classifier performance. However, using the fastText text categorization tool for classification results is not ideal. We believe that if you have a large dataset and a more accurate word segmentation tool, it will inevitably make the improvement of the score more obvious.

Acknowledgments

This research has been supported by the Fundamental Research Funds for the Central Universities, Northwest Minzu University (31920160005) and (31920190094). National Science and Technology Major Project (2017YFB1002103). Key Laboratory of China's Ethnic Languages and Information Technology (Northwest Minzu University), Ministry of Education. The authors gratefully acknowledge the financial support from Northwest Minzu University.

References

[1]. N. Qun, L. Xing, X. Qiu, and X. Huang, “End-to-End Neural Text Classification for Tibetan,” 2017.

[2]. Hinton G E . Learning distributed representations of concepts.[C]// Eighth Conference of the Cognitive Science Society. 1989.

[3]. T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” pp. 1–12, 2013.

[4]. A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, “Bag of Tricks for Efficient Text Classification,” arXiv Prepr. arXiv1607.01759, 2016.

[5]. Jiang T, Yu H, Zhang B. Tibetan text classification using distributed representations of words[C]// 2015 International Conference on Asian Language Processing (IALP). IEEE, 2015.

[6]. Cao, H., Jia, H.: Tibetan text classification based on the feature of position weight. In: International Conference on Asian Language Processing (IALP), pp. 220–223. IEEE (2013).

[7]. Huiqiang Jia. Tibetan Text Classification Based on KNN. Journal of Northwest University for Nationalities, 24-29,2011.9.

[8]. Guixian Xu, Chuncheng Xiang, Xiaobing Zhao, and Guosheng Yang. Automatic Classification of Tibetan Web Pages. In Proceedings 2012 International Conference on Computer Science and Electronics Engineering,423-426,20 1 2.

[9]. Guixian Xu. Tibetan Web Pages Classification. Journal of Convergence Information Technology, Vol. 8, No. I,pp. 9-15,2013.

[10]. Tao Jiang, Yugang Dai, Ailin Li, and Hongzhi Yu. Tibetan Text Classification Using SVM and NB. In Proceeding of the 2nd National Conference on Information Technology and Computer Science,l160-11 65,2015.3.

[11]. Yachao Li, Yangkyi Jam, Chengqing Zong and Hongzhi Yu. Research and Implement of Tibetan Automatic Word Segmentation Based on Conditional Random Field. Journal of Chinese Information Processing, 27(4): 52-58,20.

[12]. Nguyen, T.-P. and Le, A.-C. (2016). A hybrid approach to Vietnamese word segmentation. In Computing & Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2016 IEEE RIVF International Conference on. IEEE.

[13]. E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, “Learning Word Vectors for 157 Languages,” in Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018), 2018.

[14]. Kim Y. Convolutional Neural Networks for Sentence Classification[J]. empirical methods in natural language processing, 2014: 1746-1751.

[15]. Johnson R, Zhang T. Effective Use of Word Order for Text Categorization with Convolutional Neural Networks[J]. Eprint Arxiv, 2014.

[16]. Zhou C, Sun C, Liu Z, et al. A C-LSTM Neural Network for Text Classification[J]. Computer Science, 2015.

[17]. Ji Young Lee, Franck Dernoncourt. Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks[J]. 2016.

[18]. Kalchbrenner N, Grefenstette E, Blunsom P. A Convolutional Neural Network for Modelling Sentences[J]. Eprint Arxiv, 2014, 1.

[19]. Yoshua Bengio, R'ejean Ducharme, Pascal Vincent, and Christian Janvin, A neural probabilistic language model. The Journal of Machine Learning Research, 3:1137-11 55, 2003.