The Text modeling method of Tibetan text combining Word2vec and improved TF-IDF

Tao Jiang¹,², Li Jia², Ma Cao Wan³,⁴, Jia Hao Meng⁴,⁵
²Key Laboratory of China's Ethnic Languages and Information Technology of Ministry of Education, Northwest Minzu University, Lanzhou, Gansu, China
³Department of Chinese Language and Literature, Northwest Minzu University, Lanzhou, Gansu, China
⁴Key Laboratory of China's Ethnic Languages and Information Technology of Ministry of Education, Northwest Minzu University, Lanzhou, Gansu, China

Abstract: In view of the problem of ignoring the importance of words in Tibetan text representation, this paper proposes a Tibetan text representation method combining Word2vec and improved TF-IDF. First of all, the method uses the Word2vec model to train all the word vectors of the text, which can capture the semantic information of the text. Secondly, the improved TF-IDF algorithm is used to calculate the weight of each word and word vector in the text. Fusion of Word2vec and improved TF-IDF algorithm to construct a Tibetan text vector representation model based on word vectors and weights. Finally, it uses the BiLSTM neural network model classifier to effectively classify the Tibetan text. The experimental results show that this method is better than the traditional method in the classification of Tibetan text, which verifies the effectiveness of the method.

1. INTRODUCTION
In recent years, with the rapid growth of Tibetan information data on the Internet, Tibetan information processing technology has been developing and deepening. However, due to the lack of unified standards, the lack of mature word processing technology and the serious shortage of Tibetan corpus, its research has been progressing slowly. As an important basic task in Tibetan information processing, word vectorization in Tibetan language directly restricts the development of Tibetan information processing technology and has a great impact on the research fields of automatic retrieval, text classification and machine translation[1].

The first problem to be solved in Tibetan text analysis is to transform unstructured Tibetan text into computer-understandable features[2]. This step is called text modeling. However, text modeling to some extent determines the effectiveness of the analysis. At present, the study of Tibetan word representation
is still in the initial stage. [3] studied the vector representation method of Tibetan characters, which is the first literature to put forward the Tibetan language representation model. [4] proposed a Tibetan text modeling method based on word vectors according to the grammatical characteristics of Tibetan. This method adopts the maximum entropy model for the part-of-speech tagging of Tibetan texts, using nouns and verbs as the features of the text, and then uses Word2vec training to get the word categories and calculate the probability distribution in each text. The word category probability matrix is used to represent the text, thereby implementing the text modeling has achieved good results. Combining the vector space model, [5] proposes to use some term items with high TF-IDF values in the text as contrast terms, perform sentence segmentation processing on the Tibetan text, and then use each sentence as the context topic. Quantify the degree of relevance between the terms in the text and the contrast terms. The experimental results show that the method can represent Tibetan text more accurately. [6] used the Skip-gram and CBOW models of Word2Vec to train Tibetan word vectors, and detected the effect of word vectors through semantics and grammar, and used the word vector training results to perform name recognition on deep learning models. Experimental results show that the performance of the CBOW model is better than the Skip-gram model.

Word Embedding is a distributed low-dimensional real number vector generated by a neural network language model training large-scale corpus. Its text representation is not restricted by the sparse features of the text and the update speed of the knowledge base. The role is to make related or similar words closer in distance. Word Embedding can not only calculate the semantic similarity between words, but also improve the accuracy of semantic similarity calculation between texts. Considering the advantages of word embedding, based on the existing research results, this paper proposes a Tibetan text modeling method that combines Word2vec and improved TF-IDF. This method uses Word2vec for word representation, the resulting word vector is a low-dimensional dense real number, and the semantic information is better retained. However, Word2vec cannot express the importance of words, so the TF-IDF algorithm is introduced to calculate the weight of each word vector in the text. In view of the fact that the TF-IDF algorithm only considers the frequency of words in the total corpus and ignores the distribution in different categories, an improved TF-IDF algorithm based on class frequency variance is proposed for this problem, which will take into account the distribution information of words in different categories. This method uses an improved TF-IDF algorithm to form a vector text representation of word vectors and weights, which better preserves the contextual relationship of words. Finally, BiLSTM model is used to take into account the global characteristics of the text, and the text context semantic information is fully considered to classify the text.

2. RELATED OVERVIEW

2.1. Word Embedding and Word2vec model
Word Embedding as a distributed representation of a word in the deep learning model was first proposed by Hinton. The basic idea is to perform unsupervised language model training on a large amount of unlabeled text data, and express words as a set of low-dimensional real number vectors to characterize the words. The resulting vector contains the semantic and grammatical information of the words. Word2vec is a word vector training tool released by Google in 2013, which can efficiently generate vector forms of words from large-scale unlabeled corpora, and provides two models of CBOW (continuous bag of words) and Skip-gram. The CBOW model obtains the probability of the current word by calculating the context of the current word, and the Skip-gram model predicts the context by the current word. The model includes three layers: input, mapping and output. The goal is to obtain a better word vector representation with a smaller amount of calculation. Figure 1 shows the architecture of the two models.
The distance between two word vectors is used to indicate the degree of similarity between two words. The distance between words indicates that cosine similarity is the more commonly used and better method. As shown in formula (1), where \( n \) represents the vector dimension.

\[
S(x, y) = \cos(x, y) = \frac{\sum_{i=0}^{n} (x_i \times y_i)}{\sqrt{\sum_{i=0}^{n} (x_i)^2 \times \sum_{i=0}^{n} (y_i)^2}}
\]  

(1)

This article uses Google's open source Word2vec as a training tool to divide the collected text into training data and test data. Use the Skip-gram model to train the training data to obtain the word vector of each word in the training data, and obtain vectors with dimensions of 100, 200, and 300, respectively. Through the experiment, the final dimension is 200. Table 1 gives examples of word vector training results arranged in reverse order of distance.

### Table 1. Word vector result training example

| Words           | Similar words | Similarity value | Words           | Similar words | Similarity value | Words           | Similar words | Similarity value |
|-----------------|---------------|------------------|-----------------|---------------|------------------|-----------------|---------------|------------------|
| ཕྲོང་གསེབ་ཀྱི། | ཆེན་པོ་ཆུ། དངུལ་བསྟན། བལྟ་སྐུལ། | 0.802021 0.802003 0.801960 0.751817 0.701767 | སྙན་ སྙན་ | ཐོ་མོ་ ཐོ་མོ་ བཏ་མོ་ སྤྱད་པ་ སྤྱད་པ་ | 0.792507 0.791998 0.752018 0.701961 0.691926 | ཁྱིམ་ ཁྱིམ་ | ཁྱིམ་ ཁྱིམ་ ཁྱིམ་ | 0.651824 0.551811 0.531672 0.501598 0.501604 |

2.2. BiLSTM Network model

LSTM is a one-way neural network structure, which is a special kind of RNN. Its appearance can be used to solve the problem that RNN cannot deal with long-term dependence in training\(^{[13]}\). From the perspective of the network structure, the problem is that the pre-order sequence information can be used well when calculating the current state of the neural unit, and the post-order sequence information cannot be effectively used. When performing more fine-grained classification tasks, we need to pay attention to the interaction between the texts. However, the unidirectional LSTM cannot fully utilize the global information of the text in the learning process of the representation of words, and cannot effectively capture the weaker semantic information\(^{[14]}\). In addition, the last sequence of long-sequence hidden layer output is used as a vector representation of the sentence sequence, which will be inconsistent due to the sequence head and tail. In order to solve the above problem, the information is input to the model in reverse, and the unidirectional LSTM network structure model is designed as a
bidirectional long-short-term memory network (BiLSTM) structure model. The model network structure is shown in Figure 2.

![BiLSTM model network](image)

Figure 2. BiLSTM model network

3. A TEXT MODELING METHOD FOR TIBETAN CLASSIFICATION

3.1 Improved TF-IDF algorithm

TF-IDF is a classic statistical method for calculating word weights. It consists of two parts of data: term frequency (TF) and inverse document frequency (IDF). The calculation of word frequency is shown in formula (2):

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$  \hspace{1cm} (2)

$tf_{i,j}$ represents the frequency of the word $w_i$ appearing in the document $d_j$, $n_{i,j}$ is the number of times the word appears in the document, the denominator is the total number of occurrences of all words in the document $d_j$, k is the number of different words in the document. The IDF calculation is shown in formula (3):

$$idf_i = \log \frac{n_d}{df(d, w_i) + 1}$$  \hspace{1cm} (3)

Where $idf_i$ represents the IDF of the word $w_i$ in the text library $d$, $n_d$ is the total number of documents in the text library $d$, $df(d, w_i)$ is the number of documents containing the word $w_i$ in the text library $d$, and 1 is added to prevent the situation from being zero. Finally, the calculation formula (4) of TF-IDF normalization processing is shown as follows:

$$tf-idf_{i,j} = \frac{tf_{i,j} \times idf_i}{\sqrt{\sum_{w_i \in d_j} [tf_{i,j} \times idf_i]^2}}$$  \hspace{1cm} (4)

It can be seen from the formula that the importance of word $w_i$ to document $d_j$ is proportional to the frequency with which it appears in document $d_j$. At the same time, it is inversely proportional to the number of documents that contain words in the entire text library. In the text classification research, the text in the experimental data is marked as different categories, and the TF-IDF algorithm only considers the total frequency of feature words in the entire data, ignoring the distribution between different categories. For example, the word ‘ཐུའིག་ཤེས’ appears more frequently in texts in the economic and political categories, but less frequently in texts in other categories. However, the traditional TF-IDF algorithm does not consider the distribution of different categories, resulting in the loss of certain words that have an effect on the category. Therefore, this paper proposes the TF-IDF algorithm of quasi-frequency variance. The quasi-frequency variance measures the distribution of words in different categories. The formula (5) is as follows:

$$q_i = \frac{\sqrt{\sum_{j=1}^{N} \left( \frac{df(d, w_i)}{N} - df(c_j, w_i) \right)^2}}{N}$$  \hspace{1cm} (5)
q$_i$ is the quasi-frequency variance of words, N is the text category score, df(d,w$_i$) is the number of documents containing the word w$_i$ in the text library d, df(c$_j$,w$_i$) is the number of documents containing the word w$_i$ in category c$_j$. The larger the q$_i$, the greater the fluctuation of the words in the category, the more uneven the distribution, and the greater the effect on the text classification. Therefore, the calculation of the TF-IDF algorithm based on the frequency-like variance is shown in formula (6):

$$tf-idf_q = df(d,w_i) \times q_i$$  \hspace{1cm} (6)

### 3.2 Tibetan text representation of Word2vec and improved TF-IDF

Perform word segmentation processing $Fi$=$[f_1...f_n]$ on the text $di$, and n is the number of words. According to the Word2vec word vector, replace the text after word segmentation with a low-dimensional numerical vector $VFi$=$[Vf_1...Vf_n]$, where $VFi$ is the word vector of $Fi$. k is the dimension of the word vector. This method is simpler than the traditional machine learning text classification algorithm, and can retain the information of the original text. However, Word2vec word vectors cannot describe the importance of words, so the formula (5) improved TF-IDF algorithm is used to calculate the weight of vector words. The final text representation is shown in formula (6). The overall work flow chart 3 is shown in the figure.

#### 4. EXPERIMENT AND ANALYSIS

### 4.1. Experimental data and parameters

In order to verify the effectiveness of the proposed text modeling method, this article uses web crawler technology to crawl the website data of Tibet Daily (Tibetan Version), Tibet Science and Technology News, China Tibet News Network (Tibetan Version) and other websites. Obtain a comprehensive corpus of Tibetan texts. The text categories mainly include three categories, namely politics, economy, culture and education. Distribution of various texts: 830 articles on politics, 307 articles on economy, and 287 articles on culture and education. The experiment uses 80% as the training set and 20% as the test set. Word2vec, a Google open source word vector tool, is used for word vector training. The
parameter settings are shown in Table 2. The experimental environment is Core i7 3.4GHz, 12G memory, Window10 64-bit operating system, and the deep learning framework is Keras.

| Hyperparameter | Parameter Description               | Value |
|----------------|-------------------------------------|-------|
| size           | Dimension of Word Embedding         | 200   |
| window         | The size of the context window      | 5     |
| min-count      | Minimum threshold for words         | 1     |
| cbow           | Whether to use the cbow model       | NO    |

4.2. Data preprocessing

Tibetan text is different from English and other written forms, and there is no clear lexical boundary symbol. The model of Tibetan word representation requires training a large amount of corpus to obtain vocabulary information, so the effect of segmenting Tibetan text is directly related to the final effect of word representation. In the Tibetan text, there are many words that are widely used but do not contribute much to the theme. Such words only serve to expand sentences and punctuation marks, and are mainly distributed in Tibetan function words. Tibetan function words are very frequent in the text and have a great influence on the word representation. The preprocessing of Tibetan text corpus includes data cleaning, removal of stop words and automatic word segmentation in Tibetan. First, delete invalid data and duplicate data. Then, remove stop words and use Zhu Jie[15] and others to obtain stop word lists for processing. This article puts the stop word removal before the word segmentation, and prevents the Tibetan word segmentation before deleting the format. The word segmentation adopts the Tibetan word segmentation system developed by Northwest University for Nationalities. The accuracy of the word segmentation system can reach 90%. In order to get an accurate word segmentation result, each file of the word segmentation result is manually proofread to correct its word segmentation error.

4.3. Experimental evaluation index

In classification, the accuracy rate (Precision, P) refers to the ratio of the number of correctly classified texts to the number of all classified texts, and the recall rate (Recall, R) refers to the ratio of the number of correctly classified texts to the actual number of texts in the class, as shown in Equation 7 And formula 8:

\[
\text{Precision} = \frac{TP}{TP + FP} \quad (7)
\]
\[
\text{Recall} = \frac{TP}{TP + FN} \quad (8)
\]

Among them, TP, FP, FN, TN represent the number of documents that meet certain conditions, as shown in Table 3:

| Reality      | Forecast results | Wrong example |
|--------------|------------------|---------------|
| Correct example | Really correct example | Fake wrong example |
| Wrong example              | False correct example | Really wrong example |

The F1 indicator is an indicator that considers both accuracy and recall. The macro average is the arithmetic average of performance indicators for each category. The F1 indicator is shown in Equation 9:
\[
F_1 = \frac{2TP}{2TP + FP + FN}
\]  

(9)

4.4. Analysis of results

In this paper, the bidirectional long-term and short-term memory network is used as the classifier, and the training set is used to train the model to obtain the Tibetan text classification results; then the test set is used to test the effect of the classification result model. From the accuracy rate, recall rate, and F1 value, the test results are shown in Table 4:

| Category          | Precision | Recall | F1-measure |
|-------------------|-----------|--------|------------|
| political         | 0.896     | 0.899  | 0.893      |
| economic          | 0.825     | 0.827  | 0.829      |
| Cultural education| 0.807     | 0.810  | 0.802      |
| **Average**       | 0.842     | 0.845  | 0.841      |

The experimental results show that the average accuracy rate, recall rate, and F1 value reach 84.2%, 84.5%, and 84.1%, respectively. Observing the experimental results, the accuracy of political text classification is high, and the effect of cultural education text classification is not ideal. Due to the small amount of Tibetan text in the experimental data, the cultural education training text accounted for only 20.5%. In order to verify the effectiveness of the improved TF-IDF word vector weighting algorithm, classification experiments were carried out on non-word vector weighting (NONE), traditional TF-IDF calculating word vector weighting and the weighting calculation method (TF-IDF-q) proposed in this paper Compared. The experimental results are shown in Figure 4:

![Figure 4. Performance comparison results of different word vector weights](image)

Through experimental comparison, it is found that the improved fusion TF-IDF algorithm proposed in this paper is better than the traditional TF-IDF algorithm, which proves the effectiveness of the introduced word vector weight calculation. The weight is to enhance the feature of the word vector, so that the performance of the algorithm is improved, and a good classification effect can be obtained. And observe the experimental results. Under the premise that the amount of Tibetan text data is small, without introducing any word vector weights, the text classification algorithm based on the two-way long short-term memory network still has good classification results. Preserving the original text information to the maximum extent, on the other hand, benefits from the powerful learning ability of the neural network.

5. Conclusion

In view of the problems of the current Tibetan text vector representation method, This paper proposes a new Tibetan text modeling method, which comprehensively considers the semantic information of words and the importance of words, and quantifies the degree of word contribution and integrates it with the word vector obtained through Word2vec. Experiments show that the method proposed in this
paper is better than the text vector representation method without word vector weights and traditional TF-IDF for calculating word vector weights on classification tasks. However, this method also has shortcomings. Due to the lack of a Tibetan text corpus, the overall classification effect of the neural network model is not ideal. In the future, the corpus of Tibetan texts will be expanded to improve the learning ability of the classifier to obtain more excellent classification effect.

ACKNOWLEDGMENTS
The work was supported by the Ministry of Education Humanities and Social Sciences Research Project (No. 18YJCZH065).

REFERENCES
[1] Cai Z J. The Key Technologies of Representation of Tibetan Word vector[D].Qinghai Normal University, 2018.
[2] Zha-xi L M. Research on Tibetan Word Representation Techniques [D].Qinghai Normal University,2019.
[3] Zhi-Jie C, Rang-Zhuoma C. Vector Space Models and Component Features Analysis of Tibetan Characters[J]. Journal of chinese information processing, 2016,30(2):202-206.
[4] Tao J, Hong-zhi Y. A text modeling method for Tibetan clustering[J].Journal of Northwest University for Nationalities (Natural Science Edition),2016,37(3):24-28.
[5] Tao X U, Hong-Zhi Y U, Yang-Ji J. Tibetan Document Representation Method Based on Improved Chi-squared Statistic[J]. computer engineering, 2014, 40(6):185-189.
[6] Jie Z, Tian-rui L. Tibetan name recognition method based on deep learning model[J].Plateau Science Research,2017,1(1):112-124.
[7] Ming Liu, Bo Lang, Zepeng Gu, Ahmed Zeeshan. Measuring Similarity of Academic Articles with Semantic Profile and Joint Word Embedding[J].Tsinghua Science and Technology,2017,22(6):619-632.
[8] Ma S D, Liu D F.Text Classification Method Based on Weighted Word2vec[J].Information Science,2019,37(11):38-42.
[9] Wang G S,Huang X J. Convoluition Neural Network Text Classification Model Based on Word2vec and Improved TF-IDF[J].Journal of Chinese Mini-Micro Computer Systems,2019,40(05):1120-1126.
[10] Hinton G E. Learning Distributed Representations of Concepts[C]//Proc of the 8th Annual Conference of the Cognitive Science Society. Amherst, USA,1986:1-12.
[11] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[C]//Advances in neural information processing systems,2013.
[12] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space[J]. Computer science,2013: arxiv:1301.3781.
[13] Huang L, Du C S. Application of recurrent neural networks in text classification[J].Journal of Beijing University of Chemical Technology (Natural Science Edition),2017,44(1):98-104.
[14] Gao T C, Zhang Y H. Sentiment Classification of Chinese Product Reviews Based on cw2vec and BiLSTM[J]. Software Guide.,2020,19(04):79-83.
[15] Zhu J,Li T R. Research on Tibetan Stop Words Selection and Automatic Processing Method[J].Journal of Chinese Information,2015,29(02):125-132.