TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 10/14 pp 1085–1100
DOI: 10.26599/TST.2022.9010055
Volume 28, Number 6, December 2023

A Tibetan Sentence Boundary Disambiguation Model Considering the Components on Information on Both Sides of Shad

Fenfang Li, Hui Lv, Yiming Gao, Dolha, Yan Li, and Qingguo Zhou*

Abstract: Sentence Boundary Disambiguation (SBD) is a preprocessing step for natural language processing. Segmenting text into sentences is essential for Deep Learning (DL) and pretraining language models. Tibetan punctuation marks may involve ambiguity about the sentences' beginnings and endings. Hence, the ambiguous punctuation marks must be distinguished, and the sentence structure must be correctly encoded in language models. This study proposed a component-level Tibetan SBD approach based on the DL model. The models can reduce the error amplification caused by word segmentation and part-of-speech tagging. Although most SBD methods have only considered text on the left side of punctuation marks, this study considers the text on both sides. In this study, 465 669 Tibetan sentences are adopted, and a Bidirectional Long Short-Term Memory (Bi-LSTM) model is used to perform SBD. The experimental results show that the F1-score of the Bi-LSTM model reached 96%, the most efficient among the six models. Experiments are performed on low-resource languages such as Turkish and Romanian, and high-resource languages such as English and German, to verify the models' generalization.

Key words: Sentence Boundary Disambiguation (SBD); punctuation marks; ambiguity; Bidirectional Long Short-Term Memory (Bi-LSTM) model

1 Introduction

Natural Language Processing (NLP) is an essential subfield of artificial intelligence. Basic research in NLP includes Sentence Boundary Disambiguation (SBD), Word Segmentation (WS), Part-Of-Speech (POS) tagging, and syntactic analysis. These tasks form the basis of downstream tasks, such as machine translation, automatic question-answering systems, information extraction systems, automatic summarization, and search engines[1]. The development of NLP has progressed from rule-based to statistics-based, and then to Deep Learning (DL)[2]. The rule-based approach is based on linguistic theory, which emphasizes linguists' understanding of language phenomena and uses unambiguous rules to describe or explain ambiguous behaviors or features[3]. Therefore, researchers need to understand a language well to apply rule-based approaches. In the past several years, DL models have been successfully applied to many sequential labeling and classification tasks, such as SBD, speech recognition, WS, and POS tagging and chunking[4, 5]. DL models can learn a hierarchy of nonlinear feature detectors to capture complex statistical patterns[6]. With the application of DL technologies in NLP, various new tasks emerge. Given the high cost of labeling data, there is only a small amount of labeled training data, and the model cannot learn enrichment information while training[7]. Implementing most NLP tasks is now turned into the trend of pretraining + fine-tuning, training on mass data, and fine-tuning on specific task datasets[8]. Most of the

• Fenfang Li, Hui Lv, Yiming Gao, Yan Li, and Qingguo Zhou are with the School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China. E-mail: liff18@lzu.edu.cn; lvh19@lzu.edu.cn; gaoym2020@lzu.edu.cn; liyan_2007@lzu.edu.cn; zhouqg@lzu.edu.cn;
• Dolha is with the Key Laboratory of China's National Linguistic Information Technology, Northwest Minzu University, Lanzhou 730030, China. E-mail: duola67@126.com.
* To whom correspondence should be addressed.
Manuscript received: 2022-04-29; revised: 2022-08-14; accepted: 2022-11-16

Pretraining Language Models (PLMs) come at the cost of large datasets, and the input is usually in sentences, so the text must be segmented into sentences. Moreover, SBD can improve the amount of data and helps separate sentence components and analyze sentence structures[9].

Tibetan is an ancient and insufficiently researched language. Since Tibetan data are scarce and Tibetan PLMs are still in their infancy, structured Tibetan pretraining datasets must be built for Tibetan information processing. Standardized datasets for high-resource languages, such as English, Chinese, and French, provide a high-quality training corpus for PLMs and have considerable effects[10]. High-quality structured electronic corpora are scarce in low-resource languages, such as Thai, Turkish, and Romanian. The data crawled from the Internet for low-resource languages may vary from an article or a paragraph. However, too long sequences will increase the training cost, so the text must be segmented into sentences[11]. Segmenting text into sentences is a simple task for humans, but it is challenging for computers to learn sentence-ending features. Depending on the form of the data, SBD tasks can be divided into two categories. Text containing punctuation marks is processed by sentence boundary disambiguation, while sentence boundary detecction is performed for text without punctuation marks[12]. Corpus in this study contains punctuation masks, so SBD in this study is defined as sentence boundary disambiguation.

Tibetan is an ancient language with unique pronunciation, grammatical characteristics, and grammatical rules. These bases are still in use in modern written Tibetan. Only when the research technique of Tibetan SBD is mature can we accurately study the characteristics of Tibetan sentences, perform syntactic analysis, and effectively improve PLM performance. Modern Tibetan is composed of letters, including 30 consonant letters and four vowel letters, and consonant letters and vowel letters are arranged separately in the alphabet. The 30 consonant letters are divided into eight groups, which are (1) [“ཀ”, “ཁ”, “ག”, “ང”], (2) [“ཅ”, “ཆ”, “ཇ”, “ཉ”], (3) [“ཏ”, “ཐ”, “ད”, “ན”], (4) [“པ”, “ཕ”, “བ”, “མ”], (5) [“ཙ”, “ཚ”, “ཛ”, “ཝ”], (6) [“ཞ”, “ཟ”, “འ”, “ཡ”], (7) [“ར”, “ལ”, “ཤ”, “ས”], and (8) [“ཧ”, “ཨ”]. The four vowel letters are “ི”, “ུ”, “ེ”, and “ོ”. Vowel letters must be added to the upper or lower part of consonant letters and cannot be written independently[13]. Two kinds of characters are derived from the consonant letters: the superfix and the subfix. Neither the superfix nor the subfix can be written independently, and they should be added to the upper or lower part of the consonant letters[14].

Tibetan writing is from left to right, and syllables are separated by tsheg. Further, Tibetan syllables can be composed of at least one and at most seven components. The seven components are root, prefix, superfix, subfix, vowel, suffix, and postfix. Each letter in a syllable is called a component, and components are the constituent parts of a syllable. Figure 1 shows the structure of a Tibetan syllable (“བསྒྲིགས་”), which comprises seven parts. Each horizontal unit in a syllable is called a character. A syllable is composed of at least one and at most four characters: prefix, Vertical Combination Character (VCC, also called “character set”), suffix, and postfix. For example, in the syllable “བསྒྲིགས་”, “བ” is the prefix, “སྒྲི” is the VCC, “ས” is the suffix, and “ག” is the postfix. In the character set (“སྒྲི”), “ི” is a vowel, “ས” is a superfix, “ག” is a root, and “ྒ” is a subfix. A Tibetan syllable has only one VCC, and the prefix, the suffix, and the postfix are single consonant letters[15]. Therefore, Tibetan has not only horizontal spelling but also vertical spelling. This bidirectional spelling pattern is a unique feature of Tibetan.

We know that SBD mainly refers to processing compound sentences, and there is usually a pause between clauses in compound sentences[16]. As we all know, an integral Chinese or English sentence has noticeable ending punctuation marks. It usually ends with a period, question mark, exclamation point,

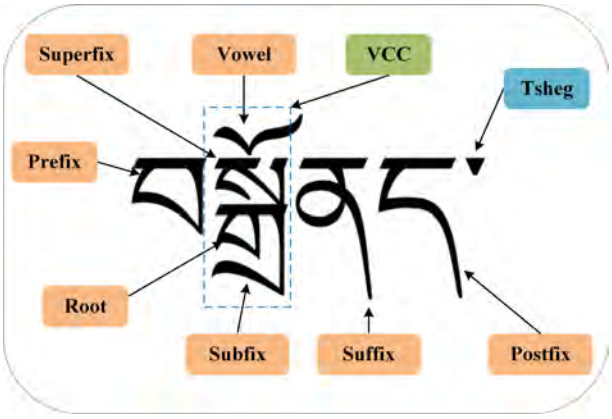


Fig. 1 Structure of the Tibetan syllable.

semicolon, or ellipsis. Table 1 shows the punctuation marks' functions in modern Tibetan. The most frequent punctuation mark used at the end of a sentence in Tibetan is shad or double shad. For ancient Tibetan, a double shad represents the end of a paragraph in a poem or prose, and it functions as a period, question mark, semicolon, exclamation point, or ellipsis. A shad behind words or phrases indicates a period or comma[17]. With the development of Tibetan culture, people gradually simplified a double shad into a shad. Therefore, the function of shad in Tibetan is vibrant. In modern Tibetan, the punctuation mark representing the end of a sentence is called a shad, but the unit ending with a shad is sometimes a complete sentence, a clause, or a sentence fragment. With predicates at the end of the sentence, the structure of Tibetan sentences adopts the "SOV" type (subject + object + predicate + shad). The predicate before the shad comprises auxiliary words, verbs, adjectives, pronouns, and adverbs[18]. English and Chinese can use punctuation marks, such as a period, question mark, or exclamation point, to express the types of sentences. The tone of Tibetan sentences is also wholly represented by the shad. The shad can be used as the end punctuation marks of a statement sentence, question sentence, exclamation sentence, imperative sentence, pause, or clause[19]. As we know, conjunctions in a text can also connect

Table 1 Modern Tibetan punctuation marks and their function.

Punctuation mark	Function
Tsheg ("་")	Tibetan tsheg is used to divide syllables.
Shad ("།")	Tibetan shad indicates a discourse's pause, turning point, or end.
Double shad ("༎")	The application of double shad is the same as the shad, but it emphasizes that a passage or sentence has ended.
Quadruple shad ("༎༎")	Quadruple shads are used in chapters, volumes, and book endings. They belong to Tibetan stylistic symbols.
Rinchenspung shad ("༑")	The left end of a sentence begins with Rinchenspung shad when it is less than three letters.
Snake-shaped shad ("༈")	In a long Tibetan text, when the last syllable of a sentence is at the first position of a line, the syllable is separated by a snake-shaped shad.
Double ornament ("༄༅།")	The Tibetan double ornament is mainly used at the beginning of articles and books to indicate the start of the text.

two clauses. In addition to common conjunctions, Tibetan conjunctions include many free words and other function words. Even the function words connect the clauses of sentences with 50–60 words. The existence of conjunctions leads to the absence of punctuation marks in a sentence. According to the above analysis of Tibetan sentences' characteristics, the use of shad in Tibetan leads to ambiguity for sentence endings, and a Tibetan sentence is difficult to extract in structured data building. However, in some cases, shads in text do not mark sentence endings, such as "གི།", "གྱི", "ཀྱི།", "ཡི།", "གིས།", "གྱིས།", "ཀྱིས།", "པས།", "བས།", "ཡིས།", "ར།", "ན།", "ལ།", "ལས།", "ནས།", "གོ།", "ངོ་།", "དོ།", "ནོ།", "བོ།", "མོ།", "འོ།", "རོ།", "ལོ།", "སོ།", "ཏོ།", "ཀྱང་།", "འང་།", "ཡང་།", "དེ།", "ཏེ།", "སྟེ།", "གམ།", "བམ།", "མམ།", "འམ།", "རམ།", "ལམ།", "སམ།", "ཏམ།", "ཅིང་།", "ཞིང་།", "ཤིང་།", "ཅེའོ།", "ཞེའོ།", "ཤེའོ།", "ཅེ་ན།", "ཞེ་ན།", "ཤེ་ན།", "ནི།", and "དང་།". Therefore, these problems in modern Tibetan are complex to address and need to be solved in Tibetan information processing[20].

Given the complexity of rule-based and statistical SBD methods, the Tibetan specificity characters, and the high cost of manual SBD methods, this study adopts the Bidirectional Long Short-Term Memory (Bi-LSTM)[21]model to disambiguate Tibetan Sentence Boundaries (SBs) automatically. This study selected punctuation marks and component sequences on both sides of punctuation marks for training and testing. Finally, the experimental result demonstrated that considering the sequence on both sides of punctuation marks is more advantageous and stable than only considering the left side sequence. Meanwhile, the Bi-LSTM model has the highest efficiency in Tibetan SBD among several basic DL models. The main contributions of this study are as follows:

(1) This study proposes an SBD model at the component level to determine whether the shad is the end of a sentence.

(2) The performance is based on text classification models, such as the Convolutional Neural Network (CNN)[22], LSTM[23], Gated Recurrent Unit (GRU)[24], Bidirectional GRU (Bi-GRU)[25], and Multi-Layer Perception (MLP)[26], is performed. Meanwhile, the performance based on sequence labeling models, such as the Hidden Markov Model (HMM), Conditional Random Fields (CRF), Bi-LSTM, and Bi-LSTM-CRF, is performed.

(3) To demonstrate the high performance and

generalization ability of the Bi-LSTM model, this study experimented on the public datasets of English, German, Romanian, and Turkish.

2 Related Work

As is well known, the development of many NLP technologies starts with English, followed by high-resource languages, such as German, French, and Chinese, and then low-resource languages. The development of SBD is probably the same[27]. For the SBD task, researchers mainly investigate new features and models that effectively discriminate between boundaries or non-boundaries[28, 29]. Researchers have adopted Decision Trees (DTs), MLP, HMM, Maximum Entropy (ME), and CRF to study SBD[30, 31].

The SBD research on English provides ideas for other languages. The critical problem in English SBD is distinguishing the abbreviation period as an effective SB mark. Read et al.[32] statistically analyzed 75 000 scientific abstracts, finding that 54.7%–92.8% of periods appear at the end of sentences, approximately 90% indicate a sentence ending, 10% suggest an abbreviation, and 0.5% include both. Riley[33] proposed the DT classifiers to determine whether the periods mark SBs. This approach used the probabilities of words being sentence-final or sentence-initial, word length, and word case as features to perform SBD. Palmer et al.[34] proposed the SATZ system to study the POS distribution of the context surrounding a potential SB to perform SBD. Reynar and Ratnaparkhi[35] employed supervised ME learning to study SBD. Their system variants treat segmentation as a disambiguation task and achieve good results. Gillick[36] adopted Support Vector Machines (SVMs) to discuss the English SBD. Mikheev[37] treated sentence segmentation with a small set of rules based on determining whether the words to the left or right of a potential SB are abbreviations or proper names. Mikheev[38] combines Ref. [37] with supervised POS tagging that includes tags for end-of-sentence markers to reduce the error rates. Kiss and Strunk[39] presented a fully unsupervised system named PUNKT. The system is rooted in identifying abbreviations by finding collocation bonds between candidates and periods.

SBD develops slowly in low-resource languages. The traditional Sanskrit text uses a script continuum, lacking some orthographic elements that make up the modern western text. Therefore, Sanskrit is not a consistent and straight forward system for marking SBs. From the content of the Sanskrit translation, Tibetan is the language that can restore Sanskrit regardless of a word's meaning. Tibetan translation has the habit of a literal translation. At the same time, the Chinese often adopt free translation to be elegant and easy to understand, leading to an understanding deviation between Chinese and Sanskrit. For Sanskrit SBD, Hellwig[40] applied the LSTM model to Sanskrit SBD, demonstrating exemplary performance. Hock[41] and Yang et al.[42] adopted CRF to identify Sanskrit SBs, and their input functions can come from keywords of any length and are trained to maximize classification accuracy. Zhao et al.[43] adopted RNN to capture the long-term interactions among morphology, vocabulary, and output symbols, which plays an essential role in Sanskrit SBD.

For the uncertainty function of Tibetan punctuation marks, researchers have conducted relevant research in the early stage and achieved good results. For Tibetan, rule-based research methods, machine learning approaches, and a combination of the two have been applied to study SBD. Zhao et al.[44] proposed a rule-based method based on the auxiliary suffix to detect Tibetan SBs. This study provided a preliminary analysis and discussion on the sentence pattern characteristics of Tibetan legal text. Cai and Ji[45] analyzed Tibetan syntax in terms of linguistics and studied the problem of SBD, which regards SBD as a study to eliminate the ambiguity of sentence-ending punctuation marks. Ren and An[46] proposed an SBD method for constructing three dictionaries (an ending word dictionary, a non-ending word dictionary, and a unique word dictionary). It transformed the SBD problem into querying the words on the left of the shad belonging to which dictionary. Cai[47] proposed a verb-centered dichotomy SBD method based on ME. First, the ME model detects Tibetan sentences by grammar rules and a thesaurus, and then ambiguous sentences are further identified. Li et al.[48] proposed a method based on the rules and ME. This method is the first in which combining rules and a machine learning method were used for the Tibetan SBD task. Ma et al.[49] proposed a POS tagging rule method for Tibetan SBD. First, the text was segmented into words and marked with POS, and then the text was scanned. When scanning shad or double shad, judge whether the word on the left of the shad or double shad was a conjunction, and whether the POS of the word is a noun, a number, or a status word. If so, the

model would continue scanning; otherwise, it performed sentence segmentation. Zhao et al.[50] studied the method of SBD for the ending of modern Tibetan auxiliary verbs. This method first identified the auxiliary verb on the left of the shad, then judged whether the auxiliary verb on the left was a verb through the auxiliary verb, and finally considered whether the number of syllables of the sentence was greater than seven and segmented from the shad. Zha and Luo[51] extracted Tibetan sentences using a reverse search of functional word positions and suffix POS. This method improved the efficiency of Tibetan sentence extraction and identified 11 POS sentence endings. Que et al.[52] studied the problem of the automatic recognition of Tibetan compact shads based on rules and an SVM. This method first uses terminal words and compact shads to establish a feature vocabulary and then uses the SVM to perform classification.

The above models solve the problem of SBD in modern Tibetan from different perspectives, using rule-based methods, machine learning based methods, or a combination of the two. Most methods require researchers to have a high language foundation, and the corpus of each study has different emphases. Before the experiment, the data must be extracted for a unique practice environment, and the participants in the study must be selected. Rule-based studies need to perform WS and POS tagging, and because of the error amplification principle, the performance of WS and POS tagging has a considerable influence on SBD. Because of the scarcity of Tibetan electronic data resources, these data and rules for Tibetan SBD tasks are unpublished.

3 Proposed Deep Bi-LSTM Approach

Tibetan data resources are scarce, and there is no publicly available Tibetan SBD data. This study proposes a component-level Tibetan SBD method on DL models to improve the data preprocessing efficiency of Tibetan. Tibetan SBD based on the DL model includes three stages: (1) feature extraction, (2) model training and preservation, and (3) sentence boundary disambiguation. First, we input the shad and sequence on the left and right of the shad into the model. Then, we train the DL model to find the discrimination pattern from the basic features through nonlinear changes. Finally, the global decision is realized through the saved model. Figure 2 depicts the framework of SBD in this study.

3.1 Definition

As mentioned above, the SBD can be considered as a text classification problem. {eos, neos} is the class of examples, eos means the end of a sentence, neos means not the end of a sentence. According to Bayesian formula, given the input example x, the predicted label $\hat{y}$ of the example is shown in the following:

$$\begin{aligned}\hat{y} &= \arg\max_{y} p(y|x) = \\ &\arg\max_{y} p(x,y) = \\ &\arg\max_{y} p(y)p(x|y)\end{aligned} \tag{1}$$

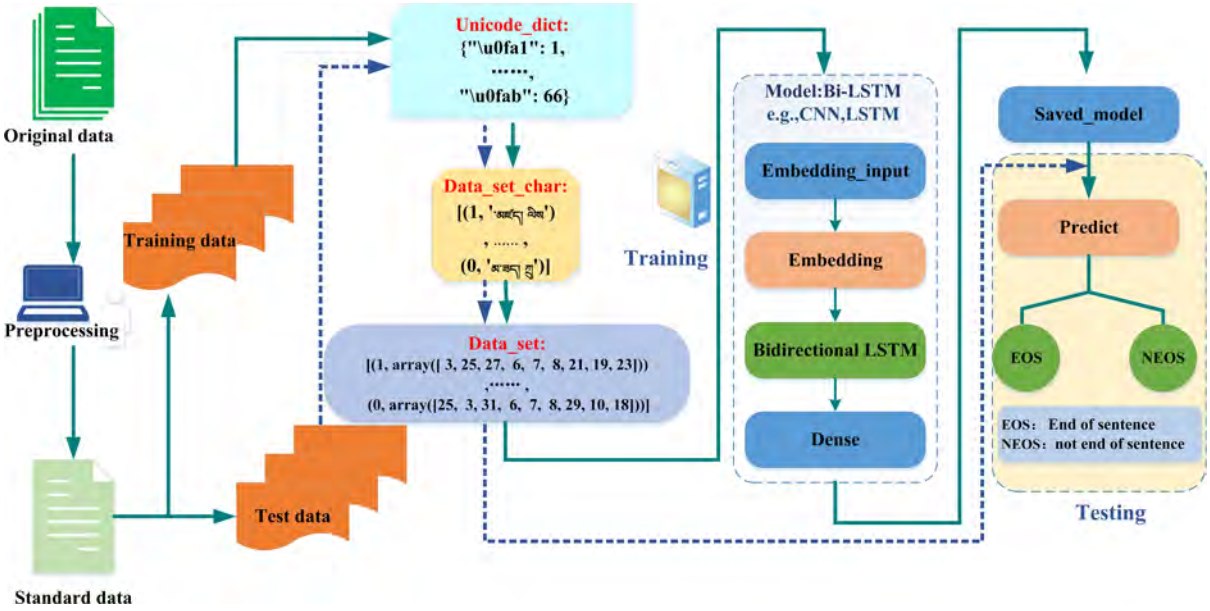


Fig. 2 Framework for applying preprocessing in the different stages of SBD.

In this study given the example t, x_t represents the input feature of t, and is conditionally independent. T is the length of the input sequence, y_t is the predicted class of example t. According to Eq. (1), the probability of $p(x|y)$ is calculated as follows:

$$p(x|y)=\prod_{t=1}^{T}p(x_t|y_t)=\prod_{t=1}^{T}\frac{p(y_t|x_t)p(x_t)}{p(y_t)} \quad (2)$$

where $p(y_t|x_t)$ is the posterior probability and is calculated to decide which class ($y_t \in \{\text{eos}, \text{neos}\}$) t belongs to. From Eq. (2), we find that $p(x|y)$ is calculated as the product of $p(x_t|y_t)$ when t takes different values.

Since $p(x_t)$ is fixed and thus can be ignored in the maximization operation, the probability $p(y)$ is the final probability approximated as follows:

$$p(y) = p(y_1)\prod_{t=2}^{T}p(y_t|y_{t-1}) \quad (3)$$

From Eqs. (1)–(3), the most likely predicted label can be obtained as follows:

$$\hat{y} = \arg\max_{y} p(y)\prod_{t=1}^{T}\frac{p(y_t|x_t)}{p(y_t)} \quad (4)$$

The posterior probability $p(y_t|x_t)$ is the neural network's output.

3.2 Model training processing

This section presents the Bi-LSTM model in detail. The Bi-LSTM model was introduced to control the degree of historical information retained by each LSTM unit, memorize the current input information, keep the essential features, and discard the unimportant parts. The previous cell state is simultaneously introduced to calculate the input gate, forgetting gate, and new information. For sequence modeling, the future and history information at each moment is equally important, but the standard LSTM model cannot capture the future information according to its order. Therefore, this study adopts the Bi-LSTM model, adding a reverse LSTM layer to the forward LSTM network layer. Figure 3 shows the structure of the Bi-LSTM model. This study introduces window size to reduce the requirements of the hardware platform. Figure 3 is an example that sets the window size as 4, and the fifth part is a shad.

One LSTM unit includes an input gate i, a forget gate f, an output gate o, and a cell state c. These four parts are described as follows:

(1) The activation value of the input gate i_t contains the current input x_t, the last hidden state h_{t-1}, and the last cell state c_{t-1},

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (5)$$

where $\sigma(\cdot)$ is the activation function, W_{xi} is the weight matrix for the input gate to determine how much new information to add, W_{hi} is the matrix between the hidden state and the input gate, W_{ci} is the matrix between the cell state and the input gate, and b_i is the bias vector for the input gate.

(2) The activation value of the forget gate f_t is shown as follows:

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (6)$$

where W_{xf} is the weight matrix for forget gate to determine how much information to forget, W_{hf} is the matrix between the hidden state and the forget gate, W_{cf} is the matrix between the cell state and the forget gate, and b_f is the bias vector for forget gate.

(3) The activation value of the cell state c_t includes

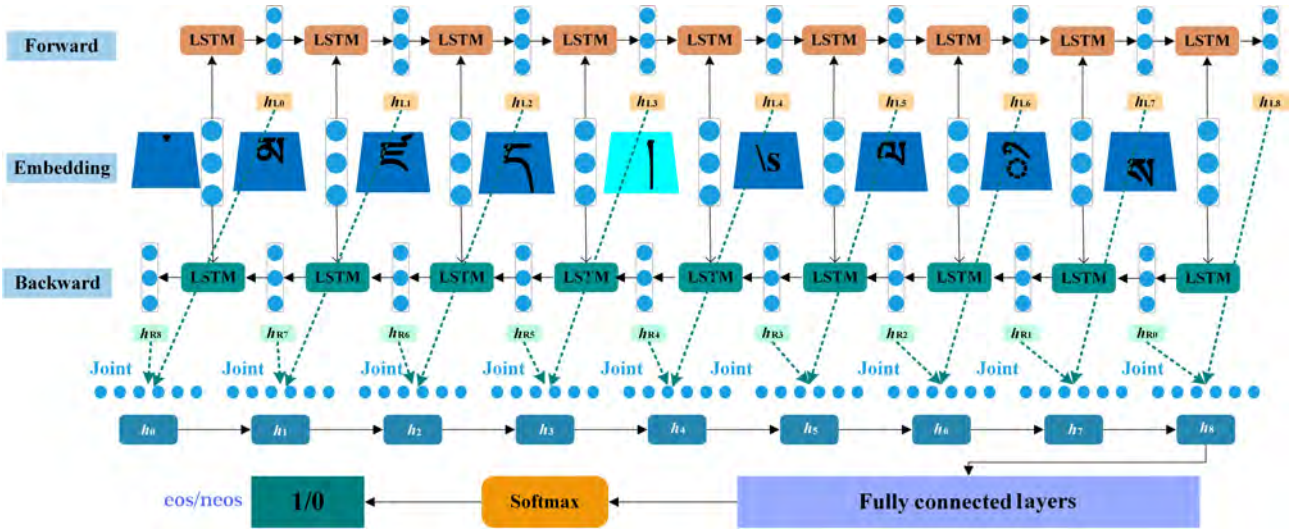


Fig. 3 Bi-LSTM model for SBD.

c_{t-1} and h_{t-1}, shown as follows:

$$c_t = i_t g_t + f_t c_{t-1} \tag{7}$$

$$g_t = \sigma(W_{xc} x_t + W_{hc} h_{t-1} + W_{cc} c_{t-1} + b_c) \tag{8}$$

where g_t is an intermediate variable, W_{xc} is the weight matrix of the cell state, W_{hc} is the weight matrix between the hidden state and the cell state, W_{cc} is the matrix between c_t and c_{t-1}, and b_c is the bias vector for cell state.

(4) The activation value of the output gate o_t is shown as follows:

$$o_t = \sigma(W_{xo} x_t + W_{ho} h_{t-1} + W_{co} c_t + b_o) \tag{9}$$

where W_{xo} is the weight matrix of the output gate, W_{ho} is the weight matrix between the hidden state and the output gate, W_{co} is the weight matrix between the cell state and the output gate, and b_o is the bias vector for output gate.

Finally, the currently hidden state h_t of the output gate is obtained by multiplying c_t with the weight matrix of the outputs, shown as follows:

$$h_t = o_t \tanh(c_t) \tag{10}$$

$$h_t = [h_t^+ \oplus h_t^-] \tag{11}$$

where $\tanh(\cdot)$ is the activation function, h_t is the conjunction result of Bi-LSTM that uses the element-wise sum to combine the hidden state of forward LSTM h_t^+ and backward LSTM h_t^-.

3.3 Classification

For a text classification problem, the label should be decided globally. This section uses the softmax classifier to predict the label $\hat{y}$ from a discrete set of classes $\{\text{eos}, \text{neos}\}$ for a segment. The classifier takes h_i as input,

$$p(y|S) = \text{soft}\max(W^{(S)} h_i + b^{(S)}) \tag{12}$$

$$\widehat{y} = \arg\max_y p(y|S) \tag{13}$$

where S is the input sequence, $W^{(S)}$ is the weight matrix of S, and $b^{(S)}$ is the bias vector of S. $y \in \mathbf{R}_m$ is the estimated probability for each class by softmax.

The cost function is the negative log-likelihood of $\hat{y}$, shown as follows:

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} k_i \log(y_i) + \lambda \|\theta\|_F^2 \tag{14}$$

where $k \in \mathbf{R}_m$ is the represented ground truth, m is the number of target classes, λ and θ are hyper regularization parameters. This study combines the dropout with L_2 regularization to alleviate overfitting.

4 Simulation Experiments and Analysis

4.1 Experiment environment

4.1.1 Network architecture and settings

An NVIDIA V100 GPU was adopted to train the model. The mini-batch size was set to 32, word-vector dimension to 100, dropout to 0.5, and epochs to 10 during the training. The checkpoint was saved for each epoch. Finally, we selected the checkpoint model with the highest performance for testing.

4.1.2 Metrics

This study considers the following metrics to evaluate the proposed model. (1) Accuracy is chosen as a standard metric of model performance. However, when considering the differences in samples between two categories, the accuracy is likely to deviate. (2) Precision, recall, and F1-score are introduced as the second metric. (3) This study also uses the Receiver Operating Characteristic (ROC) curve and the Area Under Curve (AUC) of the ROC to measure the performance. AUC is the probability that the predicted positive samples are in front of the negative ones.

4.1.3 Data

Currently, the most extensive Tibetan SBD corpus in the existing literature is 100 000 lines but unpublished. This study crawled a Tibetan SBD news dataset from the YongZin search engine (https://www.yongzin.com). The dataset is organized by the Key Laboratory of China's National Linguistic Information Technology of the Northwest Minzu University, and 465 669 Tibetan sentences are constructed. To avoid the influence of WS errors on the accuracy of Tibetan SBD, we choose the Tibetan component as the training unit in this study. Meanwhile, English and other experimental languages are at the character level. The data used in this study are a complete sentence on one line, with one or more shads in the sentence interior and a shad at the end. Sentences are divided into segments by shads; the segment at the end of the sentence is labeled "1", and the rest are labeled "0". After separating 465 669 complete sentences by shad, 12.66 million marked data are obtained. We select English, German, Romanian, and Turkish datasets from the "Europarl" corpus[53] to validate the model's generalization. The experiments are conducted to study English, German, Turkish, and Romanian SBD based on the period, semicolon, exclamation, or question mark. This study divides the corpus in the proportion of 8:2 for training and testing. The number of training data is

372 536 sentences, and the number of test data is 93 133 sentences for Tibetan. After analyzing the cost of the experiment hardware platform, this study selects the sequence on both sides of the shad and sets the window size as 4, 6, 8, and 10. Table 2 lists the statistics of the four languages' datasets.

4.2 Experimental result of sequence labeling models

Two SBD methods based on DL are text classification and sequence labeling. This study examines four sequence labeling SBD methods to compare the performance of Tibetan SBD methods based on text classification. These four models are HMM, CRF, Bi-LSTM, and Bi-LSTM-CRF.

4.2.1 Labeling strategies for sequence labeling SBD

The methods based on sequence labeling mainly study WS, POS tagging, and named entity recognition. These methods require labeled data, and the labeling strategy is also diverse. Table 3 shows the sequence labeling strategies.

Standard datasets of the Tibetan word segmentation models are scarce. The words level-labeled approach easily brings WS error amplification, and the component-based method can easily destroy the structure of Tibetan words, so this study selects syllables as the labeling unit for sequence labeling SBD. This study selects the labeling method of BME based on syllable level to mark the text. The first syllable in the sentence is labeled as B, the last syllable is labeled as E, and the remaining syllables are labeled as M.

Table 2 Dataset sizes of four languages.

Language	Train	Test	Total
German	1 476 646	18 4581	1 661 227
English	1 474 820	184 352	1 659 172
Romanian	148 925	18 617	167 542
Turkish	144 586	18 075	162 661

Table 3 Sequence labeling strategies.

Label	Label meaning
IO	(inner, outer)
BIO	(begin, inner, outer)
BIE	(begin, inner, end)
BIES	(begin, inner, end, single)
BIEO	(begin, inner, end, outer)
BIESO	(begin, inner, end, single, outer)
BME	(begin, middle, end)
BMES	(begin, middle, end, single)
BMESO	(begin, middle, end, single, outer)
BIOX	(begin, inner, outer, affix)

4.2.2 Experimental result of sequence labeling SBD

In this study, sequence labeling experiments are performed on HMM, CRF, Bi-LSTM, and Bi-LSTM-CRF models. The experimental results are shown in Table 4.

Table 4 shows the SBD results based on the sequence labeling methods. We know that when the sequence labeling methods for SBD are trained, most syllables are labeled with label M, so a problem of label imbalance emerges. We see from Table 4 that the F1-score of label M is the highest among the four models, followed by label E, and label B is the worst. Table 4 shows that the average F1-score of the three labels under the four models ranges from 98.82% to 99.63%, but the values of the three labels varies greatly. Since label M is the most important proportion of labels in the training data, its F1-score is the highest, ranging from 99.34% to 99.81%, while label B ranges from 82.38% to 94.67%, and label E ranges from 85.7% to 94.68%. This study focuses on SBD, and we should pay more attention to the evaluation index of label E. For the metric of label E, CRF and Bi-LSTM-CRF have the best effect among the four models, and their performance is approximately 94.68%, followed by Bi-LSTM, which is 86.08%, and the F1-score of HMM is the lowest, which is 85.7%. Among the sequence labeling methods based on DL, the CRF method outperforms Bi-LSTM, and Bi-LSTM-CRF considerably outperforms Bi-LSTM. We can conclude that CRF can improve the performance of sequence labeling SBD. It can be seen that for the sequence labeling method to realize SBD, because of

Table 4 Experimental result of sequence labeling SBD. (%)

Models' label	HMM			CRF			Bi-LSTM			Bi-LSTM-CRF		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
B	81.22	91.59	86.10	**95.27**	**94.07**	**94.67**	71.60	97.00	82.38	90.53	96.91	93.61
E	81.14	90.79	85.70	**95.27**	**94.09**	**94.68**	77.78	96.37	86.08	90.52	96.85	93.58
M	99.68	99.24	99.46	99.79	99.83	99.81	99.88	98.81	99.34	**99.89**	**99.63**	**99.76**
Average/total	99.04	98.96	98.99	**99.63**	**99.63**	**99.63**	99.00	98.74	98.82	99.56	99.54	99.55

the imbalance of labels, the evaluation results show substantial differences among labels.

4.3 Experimental results of Bi-LSTM

4.3.1 Experimental results of Tibetan SBD considering the information on the left, right, and both sides of the shad

(1) A comparison of considering information on the left, right, and both sides of the shad

To verify that the information on both sides of punctuation marks is efficient, this section compares the experiments considering the information on the left, right, and both sides of punctuation marks on six models. The experimental results are shown in Table 5. L, B, and R represent the left, both (both left and right), and right.

We see from Table 5 that with increasing window size, the metrics of SBD generally show a trend of increasing first and then decreasing, which proves that the performance of Tibetan SBD has an apparent relationship with the window size. At the same time, we compare the performance in the three cases of extracting the left side sequence (SBD (L)), the right side sequence (SBD (R)), and both sides sequence (SBD (B)) of the shad. Table 5 shows that the performance of SBD (B) is the best, followed by SBD (L), and SBD (R) is the worst. This result indicates that the SBD is insufficient when only considering the information on one side of the shad. Considering the information on both sides of the shad can thoroughly learn the text's past and future information. The information on the right of the shad cannot make use of its value without considering the information on the left. Under the four window sizes, LSTM, Bi-LSTM, GRU, and Bi-GRU can reach the maximum F1-score in SBD (B) models. In this study, under the four window sizes, F1-score improved range for Bi-LSTM on SBD (B) of shad is from 2.16% to 12.59%, and the maximum average improvement under the four window sizes is 5.985%. The model considering the information on both sides of the punctuation marks can comprehensively learn the context information of punctuation marks and effectively improve the efficiency of Tibetan SBD.

For Tibetan, the Bi-LSTM model has the highest accuracy, followed by the LSTM model, and the maximum difference between LSTM and Bi-LSTM models is 0.21%. The maximum F1-score difference between Bi-LSTM and the other four models is 17.35%. With the window sizes increasing, the F1-score of the Bi-LSTM model increases from 95.19% to 96%, a difference of 0.81 percentage.

(2) Comparison of F1-score and average F1-score on SBD (L) and SBD (B) models

Figures 4 and 5 show that F1-score is higher for SBD (B) than for SBD (L). When setting different window sizes, the values of F1-score of LSTM and Bi-LSTM are the highest, followed by CNN, GRU, Bi-GRU, and MLP. For the SBD (B), the values of F1-score of GRU, Bi-GRU, and MLP are promoted higher than those

Table 5 Experimental results of Tibetan on SBD (L), SBD (B), and SBD (R) models. (%)

Model	Window size=4				Window size=6				Window size=8				Window size=10			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
CNN (L)	95.45	92.38	94.41	93.39	95.80	93.82	93.83	93.82	95.56	92.26	94.91	93.57	95.43	94.27	92.18	93.21
LSTM (L)	96.29	93.21	96.10	94.63	96.89	94.32	96.67	95.48	97.07	94.56	96.94	95.74	97.09	94.61	96.98	95.78
Bi-LSTM (L)	**96.37**	**93.82**	**95.61**	**94.71**	**96.63**	**94.78**	**96.28**	**95.53**	**97.12**	**94.39**	**97.32**	**95.83**	**97.11**	**94.61**	**97.04**	**95.81**
GRU (L)	93.88	88.67	94.01	91.26	95.18	90.73	95.60	93.10	95.22	92.47	93.56	93.01	95.01	91.53	94.03	92.76
Bi-GRU (L)	95.71	92.65	94.92	93.77	94.76	90.47	94.55	92.47	87.36	81.04	82.00	81.52	93.21	86.74	94.48	90.45
MLP (L)	88.29	92.37	71.45	80.57	88.38	86.28	78.25	82.07	87.14	84.95	75.57	79.98	84.43	90.34	60.70	72.61
CNN (B)	95.34	93.05	93.08	93.16	96.11	93.11	95.62	94.35	95.97	92.20	96.29	94.20	95.79	91.80	96.23	93.96
LSTM (B)	96.64	93.59	96.75	95.15	96.97	94.35	96.90	95.61	97.16	94.29	97.55	95.89	97.19	94.52	97.38	95.93
Bi-LSTM (B)	**96.70**	**94.30**	**96.09**	**95.19**	**97.18**	**94.63**	**97.21**	**95.91**	**97.23**	**94.88**	**97.09**	**95.97**	**97.25**	**95.01**	**97.01**	**96.00**
GRU (B)	96.10	92.24	96.68	94.41	96.3	93.25	96.06	94.63	96.12	94.76	93.76	94.26	96.02	94.01	94.31	94.16
Bi-GRU (B)	96.13	92.47	96.49	94.44	96.30	93.25	96.06	94.63	95.90	92.12	96.17	94.11	95.84	92.79	95.16	93.96
MLP (B)	87.68	88.98	72.77	80.06	86.72	79.69	81.78	80.72	88.31	89.67	74.17	81.19	85.66	79.61	77.72	78.65
CNN (R)	70.25	70.30	21.65	33.10	71.05	68.99	26.98	38.79	71.34	67.73	29.98	41.56	71.20	70.19	26.58	38.56
LSTM (R)	70.39	64.15	29.28	40.21	71.47	67.15	31.52	42.91	71.71	65.45	35.60	46.12	72.00	68.46	32.72	44.28
Bi-LSTM (R)	70.47	66.03	27.05	38.38	**71.40**	**65.96**	**32.83**	**43.84**	**71.69**	**64.98**	**36.27**	**46.55**	**71.91**	**67.53**	**33.51**	**44.79**
GRU (R)	70.45	65.20	28.08	39.25	70.66	71.20	23.01	34.78	68.82	63.11	19.99	30.37	68.74	76.68	11.58	20.12
Bi-GRU (R)	**70.17**	**61.41**	**33.01**	**42.94**	70.31	76.19	18.43	29.68	69.84	72.60	18.14	29.03	69.72	76.07	15.97	26.40
MLP (R)	68.91	81.34	11.11	19.55	67.90	61.14	15.32	24.51	68.30	66.04	13.95	23.03	68.21	65.49	13.73	22.69

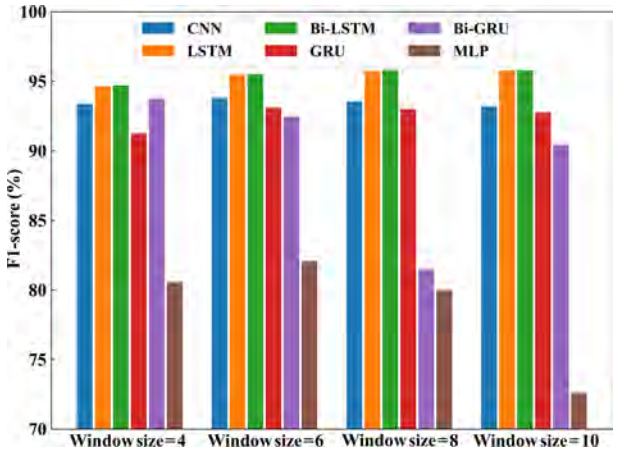


Fig. 4 Comparisons of F1-score on SBD (L).

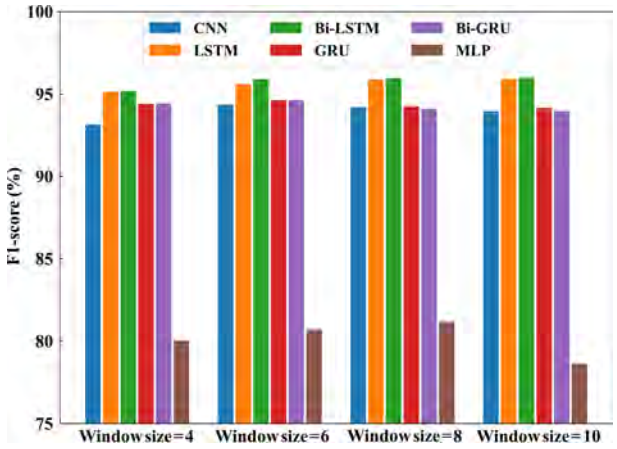


Fig. 5 Comparisons of F1-score on SBD (B).

of the SBD (L) models. Therefore, the introduction of the SBD (B) in this study effectively improves the SBD's efficiency and shows less sensitivity to the window size than that of SBD (L).

Figure 6 compares the average F1-score of six models in SBD (L) and SBD (B). This comparison shows that

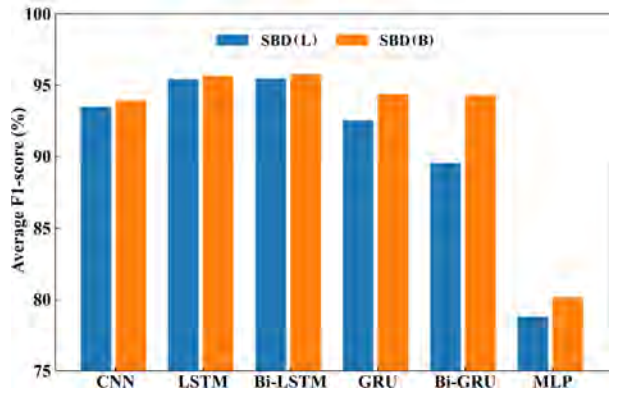


Fig. 6 Comparisons of average F1-score on SBD (L) and SBD (B).

the average F1-score of SBD (B) is higher than that of SBD (L) under the six models, proving that SBD (B) is more stable and efficient in SBD tasks. Under the six models, the average F1-score ranges from 78.81% to 95.47% for the SBD (L) models, and from 80.16% to 95.77% for the SBD (B) models. The gap between SBD (B) and SBD (L) is smaller for CNN, LSTM, and Bi-LSTM than for GRU, Bi-GRU, and MLP, which also indicates that CNN, LSTM, and Bi-LSTM models are stable and efficient in Tibetan SBD tasks.

4.3.2 Experimental results of SBD (B) models for four languages with four window sizes

To verify the generalization of the SBD (B), we compare SBD in English, German, Romanian, and Turkish. Table 6 shows the experimental results on each language, adopting window sizes of 4, 6, 8, and 10. E, G, R, and T represent English, German, Romanian, and Turkish, respectively.

The F1-score of the four languages remain unchanged under the LSTM and Bi-LSTM models, and a small downward trend occurs under the CNN, GRU, Bi-GRU, and MLP models. Therefore, the language characteristics influence the SBD performance, and the window sizes' changes show different characteristics for different languages. For English, German, Romanian, and Turkish, the maximum accuracy changes caused by window sizes are 0%, 0.02%, 0.07%, and 0.08%, respectively. Therefore, for language characteristics, the gap of F1-score caused by window size is less obvious in four languages. These results provide a reference for different languages when selecting window sizes for SBD tasks. The SBD performance of the four languages is affected more by the models' differences than by the window sizes' difference.

4.3.3 ROC curve

This study adds the ROC curve and AUC value to comprehensively evaluate model performance. The ROC curve involves two indices: the true positive rate and the false positive rate. The ROC curve is used to qualitatively analyze model performance.

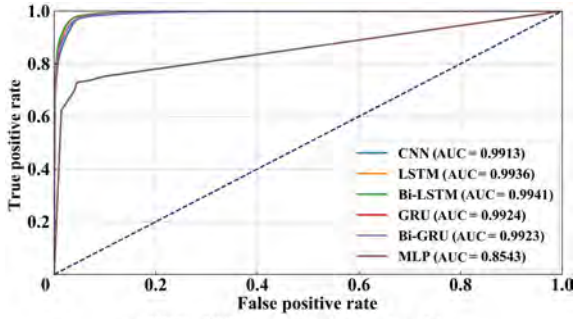
(1) ROC for Tibetan with window sizes of 4 and 10

We choose window sizes of 4 and 10 for the ROC experiments for the six models, 4 being the smallest window size and 10 being the window size with the highest performance (see Section 4.3.2). Figure 7 shows the ROC curve and AUC values of the six models under window sizes 4 and 10. The AUC value of the Bi-LSTM model is the highest, followed by LSTM, and the MLP

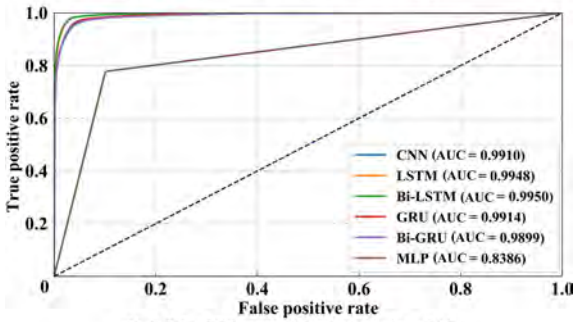
Table 6 Experimental results for four languages with four window sizes.

(%)

Model and language	Window size=4				Window size=6				Window size=8				Window size=10			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
CNN(E)	97.4	97.31	99.90	98.59	97.41	97.31	99.90	98.59	97.39	97.32	99.88	98.58	97.31	97.20	99.91	98.54
LSTM(E)	97.48	97.37	99.92	98.63	97.48	97.39	99.90	98.63	**97.49**	**97.40**	**99.90**	**98.64**	**97.50**	**97.39**	**99.91**	**98.64**
Bi-LSTM(E)	**97.48**	**97.38**	**99.91**	**98.63**	**97.49**	**97.39**	**99.91**	**98.63**	97.49	97.39	99.91	98.63	97.48	97.39	99.90	98.63
GRU(E)	97.38	97.29	99.89	98.58	97.13	97.01	99.91	98.44	97.10	97.00	99.89	98.42	94.25	94.05	99.98	96.93
Bi-GRU(E)	97.41	97.32	99.89	98.59	97.01	96.92	99.87	98.38	96.56	96.41	99.93	98.14	95.00	94.81	99.96	97.32
MLP(E)	95.10	96.90	97.73	97.31	95.85	96.50	99.02	97.74	94.49	96.98	96.94	96.96	94.21	96.98	96.62	96.80
CNN(G)	95.78	95.61	99.55	97.54	95.89	95.71	99.57	97.60	95.85	95.65	99.58	97.58	95.74	95.65	99.45	97.51
LSTM(G)	95.95	95.75	99.60	97.63	95.96	95.74	99.62	97.64	95.97	95.75	99.62	97.65	**95.97**	**95.73**	**99.65**	**97.65**
Bi-LSTM(G)	**95.94**	**95.73**	**99.61**	**97.63**	**95.96**	**95.74**	**99.62**	**97.64**	**95.97**	**95.75**	**99.63**	**97.65**	95.97	95.73	99.63	97.64
GRU(G)	95.80	95.59	99.59	97.55	94.94	94.69	99.56	97.06	93.21	92.66	99.82	96.11	90.25	89.65	99.91	94.50
Bi-GRU(G)	95.82	95.65	99.55	97.56	95.52	95.19	99.70	97.39	94.91	94.52	99.71	97.05	90.55	90.00	99.83	94.66
MLP(G)	93.13	94.57	97.40	95.97	94.06	94.80	98.32	96.53	94.06	94.64	98.50	96.53	92.41	94.74	96.31	95.52
CNN(R)	97.66	98.16	99.37	98.76	98.21	98.22	99.91	99.06	98.13	98.36	99.68	99.02	98.10	98.17	99.85	99.00
LSTM(R)	**98.19**	**98.18**	**99.93**	**99.05**	98.23	98.26	99.88	99.06	**98.26**	**98.31**	**99.87**	**99.08**	**98.25**	**98.36**	**99.81**	**99.08**
Bi-LSTM(R)	98.10	98.31	99.70	99.00	**98.23**	**98.27**	**99.88**	**99.07**	98.20	98.22	99.90	99.05	98.24	98.26	99.9	99.07
GRU(R)	98.17	98.26	99.83	99.04	98.07	98.33	99.65	98.98	97.15	97.11	99.95	98.51	95.34	95.29	100.00	97.59
Bi-GRU(R)	97.87	98.51	99.24	98.87	98.15	98.23	99.83	99.02	97.93	98.24	99.59	98.91	97.38	97.36	99.93	98.63
MLP(R)	97.36	97.76	99.47	98.61	97.34	97.74	99.47	98.60	96.41	97.81	98.39	98.10	96.31	97.63	98.47	98.05
CNN(T)	97.44	97.73	99.28	98.50	97.25	97.06	99.77	98.40	97.35	97.29	99.64	98.45	97.37	97.57	99.37	98.46
LSTM(T)	**97.55**	**97.84**	**99.29**	**98.56**	97.55	97.84	99.30	98.56	97.63	98.12	99.10	98.61	97.63	97.91	99.32	98.61
Bi-LSTM(T)	97.50	97.64	99.45	98.54	**97.62**	**98.00**	**99.21**	**98.60**	**97.65**	**98.04**	**99.21**	**98.62**	**97.63**	**97.76**	**99.48**	**98.61**
GRU(T)	97.34	97.83	99.05	98.44	97.46	97.57	99.48	98.52	97.00	97.02	99.51	98.25	97.28	97.55	99.29	98.41
Bi-GRU(T)	97.42	97.81	99.17	98.49	97.41	97.50	99.5	98.49	97.29	97.32	99.54	98.42	97.03	97.22	99.33	98.26
MLP(T)	94.66	97.03	96.65	96.84	95.13	96.99	97.26	97.13	95.06	96.65	97.55	97.10	95.78	95.92	99.23	97.55



(a) ROC of Tibetan with window size of 4



(b) ROC of Tibetan with window size of 10

Fig. 7 ROC for Tibetan with window sizes of 4 and 10.

is the lowest. From Figs. 7a and 7b, the AUC values of Bi-LSTM under the two window sizes are 0.9950 and 0.9941, respectively, with a difference of 0.0009. However, for the CNN, LSTM, GRU, Bi-GRU, and MLP models, the differences in AUC values between the window sizes of 4 and 10 are 0.0003, –0.0012, –0.0010, –0.0024, and –0.0157, respectively. Therefore, except for the Bi-LSTM and CNN models, the AUC values are more important when the window size is 4 rather than 10, in contrast to F1-score in Section 4.3.2. Figure 7 shows that the most suitable window sizes are inconsistent, and adding the ROC curve and AUC value can thoroughly evaluate model performance. In conclusion, from this experiment, the window size has a specific influence on SBD when considering the different models.

(2) Tibetan ROC for the Bi-LSTM model with four window sizes

Figure 8 shows Tibetan's ROC curves and AUC values under the Bi-LSTM model with different window sizes. Figure 8 shows that the ROC values of the four window sizes of the Bi-LSTM model are in slight

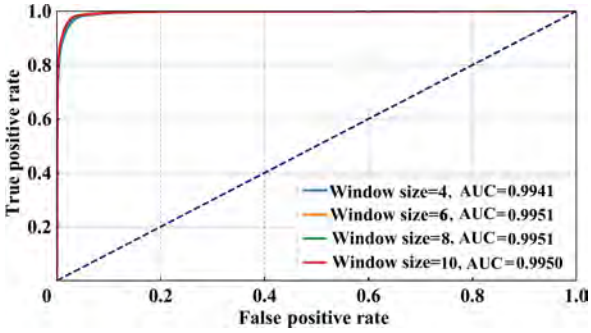


Fig. 8 ROC for Tibetan for the Bi-LSTM model with four window sizes.

gap, and the maximum interval of AUC is only 0.001. Figure 7 shows that when the window sizes are 4 and 10, the maximum difference in AUC between the models are 0.1398 and 0.1564, respectively. Therefore, the model considerably influences SBD performance compared to the window size. At the same time, the experimental data are consistent with the results in Table 5. Therefore, to improve SBD's efficiency, we can choose the appropriate window size according to the practical hardware platform of the models.

4.4 Tibetan SBD examples

To evaluate the availability of SBD models in data building for downstream tasks, this study selects a paragraph to verify the effectiveness of the models. The original text, the English translation, and golden contrast text with </eos> are shown in Table 7. The SBD result of six models are shown in Table 8, </eos> indicates correct labeling, and </neos> indicates incorrect labeling. From Table 8, we find that the results differ slightly among these models, the SBD results of the LSTM and Bi-LSTM models are closer to the golden contrast text, while other models have SBD errors of various degrees.

5 Conclusion

The need for data scales is increasing because of the popularity of NLP's pretraining + fine-tuning trend. The pretraining models need to segment text into sentences, so this study proposes a Tibetan SBD method based on the Bi-LSTM model. This study investigates the Tibetan SBD based on the idea of text classification, and performs experiments on six models. The model selects components as an experiment unit and extracts components with window sizes from both sides of punctuation marks to train and evaluate. At the same time, this study compares four sequence labeling methods to realize Tibetan SBD. This study set 465 669 Tibetan sentences to evaluate the model's performance and selects four languages from the "Europarl" corpus to verify the model's generalization. The experimental results show that considering the information on both sides of punctuation marks in SBD is more stable than on one side. The Bi-LSTM model is more efficient than other models in the SBD study. The results prove that the window size remarkably impacts the performance of Tibetan SBD, demonstrating the scientific nature of relevant literature to distinguish the SB according to the word's POS to the left of the shads. In the future, we will verify the SBD model on downstream tasks, such as machine translation and extract summarization.

6 Future Work

In the present work, we aim to transform compSutational models that consider the text information on both sides of a punctuation mark to view the text information. We

Table 7 Original Tibetan text and golden contrast text with </eos>.

Text type	Content
Original text	ཀྲུང་གོ་གུང་ཁྲན་ཏང་བཙུགས་ནས་ལོ་འཁོར་བརྒྱ་ལོན་པར་བསུ་མ་བྱེད་པ་དང་། བརྗོད་བྱ་གཙོ་བོ་དར་སྤེལ་དང་སྟ་མང་ཅན་འདོན་སྤྲོད། གངས་ལྗོངས་མེ་ལོང་དྲ་བའི་པར་རིས་ཆེད་སྒྲིག་ལེ་ཚན་གྱི་ནང་དོན་ཕུན་སུམ་ཚོགས་སུ་གཏོང་བ། རྒྱ་ཆེའི་པར་ལེན་མཁན་ཚོའི་གསར་རྩོམ་གྱི་གསོན་ཤུགས་ཇེ་འཕེལ་དང་། སྔར་ལས་ལྷག་པའི་སྤྲོ་ནས་མང་ཚོགས་དང་ཁ་ཐག་ཉེ་ཞིང་དྲང་ནུས་ཀྱི་ཕྱུལ་ཕྱུང་པར་རིས་བརྩམས་ཆོས་སྔོག་འདོན་དང་འཚོལ་ཞིབ་བྱ་ཆེད། མི་ཆོས་ཡུལ་ལྗོངས་དང་། འགོ་འགྱུར་ཉམས་སྣང་། དམངས་སྲོལ་གོམས་གཤིས། སྤྱི་ཚོགས་འཚོ་བ་སོགས་རྟེན་གཞིར་བཟུང་ནས་པར་ལེན་སྒྱུ་རྩལ་སྤྱད་དེ་བོད་འབྲེལ་ས་ཁུལ་གྱི་ལས་རིགས་ཁག་གི་དཔལ་འབྱོར་དང་སྤྱི་ཚོགས་འཕེལ་རྒྱས་ཀྱི་མངོན་གསལ་དོད་པའི་གྲུབ་འབྲས་གང་ལེགས་མཚོན་པར་བྱ་ཆེད། མཚོ་སྔོན་བོད་སྐད་དུ་ལམ་རྒྱང་སྒྲོག་བརྙན་འཕྲིན་ཁང་གི་གངས་ལྗོངས་མེ་ལོང་དྲ་བས་ཐུར་ཡོད་ཀྱི་པར་རིས་ལེ་ཚན་གྱི་རྨང་གཞིའི་སྟེང་“ལོ་བརྒྱའི་གྲུབ་འབྲས། རབ་མཛེས་མཚོ་སྔོན་”ཞེས་པ་བརྗོད་དོན་ཡིན་པའི་ཀྲུང་གོ་གུང་ཁྲན་ཏང་བཙུགས་ནས་ལོ་འཁོར་བརྒྱ་ལོན་པར་རྟེན་འབྲེལ་ཞུ་བའི་པར་ལེན་བརྩམས་ཆོས་བསྡུ་རུབ་ཀྱི་བྱེད་སྒོ་སྤེལ་འཆར་ཡིན་ལ། བདམས་ཐོན་བྱུང་བའི་བརྩམས་ཆོས་ཚང་མ་གངས་ལྗོངས་མེ་ལོང་དྲ་བ་དང་།གངས་ལྗོངས་མེ་ལོངAPP ཉིན་རེའི་གངས་ལྗོངས་སྐད་འཕྲིན་སྤྱི་སྟེགས་སོགས་ནས་འགྲེམ་སྟོན་བྱ་རྒྱུ། བདམས་ཐོན་བྱུང་བའི་བརྩམས་ཆོས་ཚང་མ་དྲ་དམངས་ཀྱིས་འདེམས་ཤོག་འཕེན་པ་དང་ཆེད་མཁས་པའི་དཔྱད་འདེམས་བྱ་ཐབས་སྤྱད་དེ་བྱ་དགའ་སྟེར་རྒྱུ་ཡིན།
English translation	Hardships and great changes have taken place in the past century. </eos> To celebrate the 100th anniversary of the founding of the Communist Party of China, carry forward the main theme, eulogize the great achievements of the Party, enrich the content of Xiangyun.com's photography column, stimulate the innovative and creative vitality of photographers, further explore excellent photography works with enthusiasm and positive energy, and use video art to fully reflect the remarkable achievements of the economic and social development of all walks of life in Tibet-related areas with cultural landscape, style changes, folk customs, and social life as carriers. </eos> Xiangyun.com of the Qinghai radio and television station held a collection activity for photographic works celebrating the 100th anniversary of the founding of the Party with the theme of "Red Centennial Focus on Qinghai" based on the original photography column. </eos> All selected works are broadcast through Xiangyun.com, the gold Kelsang APP, the Snow-Capped Mountain official WeChat account, and other platforms. </eos> The selected works are awarded by online netizens voting and offline expert evaluation.
Golden contract text with "</eos>"	ཀྲུང་གོ་གུང་ཁྲན་ཏང་བཙུགས་ནས་ལོ་འཁོར་བརྒྱ་ལོན་པར་བསུ་མ་བྱེད་པ་དང་། </eos> བརྗོད་བྱ་གཙོ་བོ་དར་སྤེལ་དང་སྟ་མང་ཅན་འདོན་སྤྲོད། གངས་ལྗོངས་མེ་ལོང་དྲ་བའི་པར་རིས་ཆེད་སྒྲིག་ལེ་ཚན་གྱི་ནང་དོན་ཕུན་སུམ་ཚོགས་སུ་གཏོང་བ། རྒྱ་ཆེའི་པར་ལེན་མཁན་ཚོའི་གསར་རྩོམ་གྱི་གསོན་ཤུགས་ཇེ་འཕེལ་དང་། སྔར་ལས་ལྷག་པའི་སྤྲོ་ནས་མང་ཚོགས་དང་ཁ་ཐག་ཉེ་ཞིང་དྲང་ནུས་ཀྱི་ཕྱུལ་ཕྱུང་པར་རིས་བརྩམས་ཆོས་སྔོག་འདོན་དང་འཚོལ་ཞིབ་བྱ་ཆེད། མི་ཆོས་ཡུལ་ལྗོངས་དང་། འགོ་འགྱུར་ཉམས་སྣང་། དམངས་སྲོལ་གོམས་གཤིས། </eos> སྤྱི་ཚོགས་འཚོ་བ་སོགས་རྟེན་གཞིར་བཟུང་ནས་པར་ལེན་སྒྱུ་རྩལ་སྤྱད་དེ་བོད་འབྲེལ་ས་ཁུལ་གྱི་ལས་རིགས་ཁག་གི་དཔལ་འབྱོར་དང་སྤྱི་ཚོགས་འཕེལ་རྒྱས་ཀྱི་མངོན་གསལ་དོད་པའི་གྲུབ་འབྲས་གང་ལེགས་མཚོན་པར་བྱ་ཆེད། མཚོ་སྔོན་བོད་སྐད་དུ་ལམ་རྒྱང་སྒྲོག་བརྙན་འཕྲིན་ཁང་གི་གངས་ལྗོངས་མེ་ལོང་དྲ་བས་ཐུར་ཡོད་ཀྱི་པར་རིས་ལེ་ཚན་གྱི་རྨང་གཞིའི་སྟེང་“ལོ་བརྒྱའི་གྲུབ་འབྲས། རབ་མཛེས་མཚོ་སྔོན་”ཞེས་པ་བརྗོད་དོན་ཡིན་པའི་ཀྲུང་གོ་གུང་ཁྲན་ཏང་བཙུགས་ནས་ལོ་འཁོར་བརྒྱ་ལོན་པར་རྟེན་འབྲེལ་ཞུ་བའི་པར་ལེན་བརྩམས་ཆོས་བསྡུ་རུབ་ཀྱི་བྱེད་སྒོ་སྤེལ་འཆར་ཡིན་ལ། </eos> བདམས་ཐོན་བྱུང་བའི་བརྩམས་ཆོས་ཚང་མ་གངས་ལྗོངས་མེ་ལོང་དྲ་བ་དང་།གངས་ལྗོངས་མེ་ལོངAPP ཉིན་རེའི་གངས་ལྗོངས་སྐད་འཕྲིན་སྤྱི་སྟེགས་སོགས་ནས་འགྲེམ་སྟོན་བྱ་རྒྱུ། </eos> བདམས་ཐོན་བྱུང་བའི་བརྩམས་ཆོས་ཚང་མ་དྲ་དམངས་ཀྱིས་འདེམས་ཤོག་འཕེན་པ་དང་ཆེད་མཁས་པའི་དཔྱད་འདེམས་བྱ་ཐབས་སྤྱད་དེ་བྱ་དགའ་སྟེར་རྒྱུ་ཡིན།

Table 8 Tibetan SBD results predicted by six models.

Model	Result
CNN	ཀྲུང་གོ་གུང་ཁྲན་ཏང་བཙུགས་ནས་ལོ་འཁོར་བརྒྱ་ལོན་པར་བསུ་མ་བྱེད་པ་དང་། </eos> བརྗོད་བྱ་གཙོ་བོ་དར་སྤེལ་དང་སྣ་མང་ཅན་འདོན་སྤྲོད། གངས་ལྗོངས་མི་ལོང་དྲ་བའི་པར་རིས་ཆེད་སྒྲིག་ལེ་ཚན་གྱི་ནང་དོན་ཕུན་སུམ་ཇེ་ཚོགས་སུ་གཏོང་བ། རྒྱ་ཆེའི་པར་ལེན་མཁན་ཚོའི་གསར་རྩོམ་གྱི་གསོན་ཤུགས་ཇེ་འཕེལ་དང་། </eos> སྔར་ལས་ལྷག་པའི་སྒོ་ནས་མང་ཚོགས་དང་ཁ་ཐག་ཉེ་ཞིང་དྲང་ནུས་ཀྱི་ཕུལ་བྱུང་པར་རིས་བརྩམས་ཆོས་སྤྲོག་འདོན་དང་འཚོལ་ཞིབ་བྱ་ཆེད། མི་ཆོས་ཡུལ་ལྗོངས་དང་། འཕོ་འགྱུར་ཉམས་སྣང་། དམངས་སྲོལ་གོམས་གཤིས། </neos> སྤྱི་ཚོགས་འཚོ་བ་སོགས་རྟེན་གཞིར་བཟུང་ནས་པར་ལེན་སྒྱུ་རྩལ་སྤྱད་དེ་བོད་འབྲེལ་ས་ཁུལ་གྱི་ལས་རིགས་ཁག་གི་དཔལ་འབྱོར་དང་སྤྱི་ཚོགས་འཕེལ་རྒྱས་ཀྱི་མངོན་གསལ་དོད་པའི་གྲུབ་འབྲས་གང་ལེགས་མཚོན་པར་བྱ་ཆེད། མཚོ་སྔོན་བོད་སྐད་དུ་ལམ་རྒྱུང་སྒྲོག་བརྙན་འཕྲིན་ཁང་གི་གངས་ལྗོངས་མི་ལོང་དྲ་བས་སྔར་ཡོད་ཀྱི་པར་རིས་ལེ་ཚན་གྱི་རྨང་གཞིའི་སྟེང་”ལོ་བརྒྱའི་གྲུབ་འབྲས། རབ་མཛེས་མཚོ་སྔོན་”ཞེས་པ་བརྗོད་དོན་ཡིན་པའི་ཀྲུང་གོ་གུང་ཁྲན་ཏང་བཙུགས་ནས་ལོ་འཁོར་བརྒྱ་ལོན་པར་རྟེན་འབྲེལ་ཞུ་བའི་པར་ལེན་བརྩམས་ཆོས་བསྡུ་རུབ་ཀྱི་བྱེད་སྒོ་སྤེལ་འཆར་ཡིན་ལ། </eos> བདམས་ཐོན་བྱུང་བའི་བརྩམས་ཆོས་ཚང་མ་གངས་ལྗོངས་མི་ལོང་དྲ་བ་དང་།གངས་ལྗོངས་མི་ལོངAPP ཉིན་རེའི་གངས་ལྗོངས་སྐད་འཕྲིན་སྤྱི་སྟེགས་སོགས་ནས་འགྲེམ་སྟོན་བྱ་རྒྱུ། </eos> བདམས་ཐོན་བྱུང་བའི་བརྩམས་ཆོས་ཚང་མ་དུ་དམངས་ཀྱིས་འདེམས་ཤོག་འཕེན་པ་དང་ཆེད་མཁས་པའི་དཔྱད་འདེམས་བྱ་ཐབས་སྤྱད་དེ་བྱ་དགའ་སྟེར་རྒྱུ་ཡིན།
LSTM	ཀྲུང་གོ་གུང་ཁྲན་ཏང་བཙུགས་ནས་ལོ་འཁོར་བརྒྱ་ལོན་པར་བསུ་མ་བྱེད་པ་དང་། </eos> བརྗོད་བྱ་གཙོ་བོ་དར་སྤེལ་དང་སྣ་མང་ཅན་འདོན་སྤྲོད། གངས་ལྗོངས་མི་ལོང་དྲ་བའི་པར་རིས་ཆེད་སྒྲིག་ལེ་ཚན་གྱི་ནང་དོན་ཕུན་སུམ་ཇེ་ཚོགས་སུ་གཏོང་བ། རྒྱ་ཆེའི་པར་ལེན་མཁན་ཚོའི་གསར་རྩོམ་གྱི་གསོན་ཤུགས་ཇེ་འཕེལ་དང་། སྔར་ལས་ལྷག་པའི་སྒོ་ནས་མང་ཚོགས་དང་ཁ་ཐག་ཉེ་ཞིང་དྲང་ནུས་ཀྱི་ཕུལ་བྱུང་པར་རིས་བརྩམས་ཆོས་སྤྲོག་འདོན་དང་འཚོལ་ཞིབ་བྱ་ཆེད། མི་ཆོས་ཡུལ་ལྗོངས་དང་། འཕོ་འགྱུར་ཉམས་སྣང་། དམངས་སྲོལ་གོམས་གཤིས། </eos> སྤྱི་ཚོགས་འཚོ་བ་སོགས་རྟེན་གཞིར་བཟུང་ནས་པར་ལེན་སྒྱུ་རྩལ་སྤྱད་དེ་བོད་འབྲེལ་ས་ཁུལ་གྱི་ལས་རིགས་ཁག་གི་དཔལ་འབྱོར་དང་སྤྱི་ཚོགས་འཕེལ་རྒྱས་ཀྱི་མངོན་གསལ་དོད་པའི་གྲུབ་འབྲས་གང་ལེགས་མཚོན་པར་བྱ་ཆེད། མཚོ་སྔོན་བོད་སྐད་དུ་ལམ་རྒྱུང་སྒྲོག་བརྙན་འཕྲིན་ཁང་གི་གངས་ལྗོངས་མི་ལོང་དྲ་བས་སྔར་ཡོད་ཀྱི་པར་རིས་ལེ་ཚན་གྱི་རྨང་གཞིའི་སྟེང་”ལོ་བརྒྱའི་གྲུབ་འབྲས། རབ་མཛེས་མཚོ་སྔོན་”ཞེས་པ་བརྗོད་དོན་ཡིན་པའི་ཀྲུང་གོ་གུང་ཁྲན་ཏང་བཙུགས་ནས་ལོ་འཁོར་བརྒྱ་ལོན་པར་རྟེན་འབྲེལ་ཞུ་བའི་པར་ལེན་བརྩམས་ཆོས་བསྡུ་རུབ་ཀྱི་བྱེད་སྒོ་སྤེལ་འཆར་ཡིན་ལ། </eos> བདམས་ཐོན་བྱུང་བའི་བརྩམས་ཆོས་ཚང་མ་གངས་ལྗོངས་མི་ལོང་དྲ་བ་དང་།གངས་ལྗོངས་མི་ལོངAPP ཉིན་རེའི་གངས་ལྗོངས་སྐད་འཕྲིན་སྤྱི་སྟེགས་སོགས་ནས་འགྲེམ་སྟོན་བྱ་རྒྱུ། </eos> བདམས་ཐོན་བྱུང་བའི་བརྩམས་ཆོས་ཚང་མ་དུ་དམངས་ཀྱིས་འདེམས་ཤོག་འཕེན་པ་དང་ཆེད་མཁས་པའི་དཔྱད་འདེམས་བྱ་ཐབས་སྤྱད་དེ་བྱ་དགའ་སྟེར་རྒྱུ་ཡིན།
Bi-LSTM	ཀྲུང་གོ་གུང་ཁྲན་ཏང་བཙུགས་ནས་ལོ་འཁོར་བརྒྱ་ལོན་པར་བསུ་མ་བྱེད་པ་དང་། </eos> བརྗོད་བྱ་གཙོ་བོ་དར་སྤེལ་དང་སྣ་མང་ཅན་འདོན་སྤྲོད། གངས་ལྗོངས་མི་ལོང་དྲ་བའི་པར་རིས་ཆེད་སྒྲིག་ལེ་ཚན་གྱི་ནང་དོན་ཕུན་སུམ་ཇེ་ཚོགས་སུ་གཏོང་བ། རྒྱ་ཆེའི་པར་ལེན་མཁན་ཚོའི་གསར་རྩོམ་གྱི་གསོན་ཤུགས་ཇེ་འཕེལ་དང་། སྔར་ལས་ལྷག་པའི་སྒོ་ནས་མང་ཚོགས་དང་ཁ་ཐག་ཉེ་ཞིང་དྲང་ནུས་ཀྱི་ཕུལ་བྱུང་པར་རིས་བརྩམས་ཆོས་སྤྲོག་འདོན་དང་འཚོལ་ཞིབ་བྱ་ཆེད། མི་ཆོས་ཡུལ་ལྗོངས་དང་། འཕོ་འགྱུར་ཉམས་སྣང་། དམངས་སྲོལ་གོམས་གཤིས། </eos> སྤྱི་ཚོགས་འཚོ་བ་སོགས་རྟེན་གཞིར་བཟུང་ནས་པར་ལེན་སྒྱུ་རྩལ་སྤྱད་དེ་བོད་འབྲེལ་ས་ཁུལ་གྱི་ལས་རིགས་ཁག་གི་དཔལ་འབྱོར་དང་སྤྱི་ཚོགས་འཕེལ་རྒྱས་ཀྱི་མངོན་གསལ་དོད་པའི་གྲུབ་འབྲས་གང་ལེགས་མཚོན་པར་བྱ་ཆེད། མཚོ་སྔོན་བོད་སྐད་དུ་ལམ་རྒྱུང་སྒྲོག་བརྙན་འཕྲིན་ཁང་གི་གངས་ལྗོངས་མི་ལོང་དྲ་བས་སྔར་ཡོད་ཀྱི་པར་རིས་ལེ་ཚན་གྱི་རྨང་གཞིའི་སྟེང་”ལོ་བརྒྱའི་གྲུབ་འབྲས། རབ་མཛེས་མཚོ་སྔོན་”ཞེས་པ་བརྗོད་དོན་ཡིན་པའི་ཀྲུང་གོ་གུང་ཁྲན་ཏང་བཙུགས་ནས་ལོ་འཁོར་བརྒྱ་ལོན་པར་རྟེན་འབྲེལ་ཞུ་བའི་པར་ལེན་བརྩམས་ཆོས་བསྡུ་རུབ་ཀྱི་བྱེད་སྒོ་སྤེལ་འཆར་ཡིན་ལ། </eos> བདམས་ཐོན་བྱུང་བའི་བརྩམས་ཆོས་ཚང་མ་གངས་ལྗོངས་མི་ལོང་དྲ་བ་དང་།གངས་ལྗོངས་མི་ལོངAPP ཉིན་རེའི་གངས་ལྗོངས་སྐད་འཕྲིན་སྤྱི་སྟེགས་སོགས་ནས་འགྲེམ་སྟོན་བྱ་རྒྱུ། </eos> བདམས་ཐོན་བྱུང་བའི་བརྩམས་ཆོས་ཚང་མ་དུ་དམངས་ཀྱིས་འདེམས་ཤོག་འཕེན་པ་དང་ཆེད་མཁས་པའི་དཔྱད་འདེམས་བྱ་ཐབས་སྤྱད་དེ་བྱ་དགའ་སྟེར་རྒྱུ་ཡིན།
GRU	ཀྲུང་གོ་གུང་ཁྲན་ཏང་བཙུགས་ནས་ལོ་འཁོར་བརྒྱ་ལོན་པར་བསུ་མ་བྱེད་པ་དང་། </eos> བརྗོད་བྱ་གཙོ་བོ་དར་སྤེལ་དང་སྣ་མང་ཅན་འདོན་སྤྲོད། གངས་ལྗོངས་མི་ལོང་དྲ་བའི་པར་རིས་ཆེད་སྒྲིག་ལེ་ཚན་གྱི་ནང་དོན་ཕུན་སུམ་ཇེ་ཚོགས་སུ་གཏོང་བ། རྒྱ་ཆེའི་པར་ལེན་མཁན་ཚོའི་གསར་རྩོམ་གྱི་གསོན་ཤུགས་ཇེ་འཕེལ་དང་། </eos> སྔར་ལས་ལྷག་པའི་སྒོ་ནས་མང་ཚོགས་དང་ཁ་ཐག་ཉེ་ཞིང་དྲང་ནུས་ཀྱི་ཕུལ་བྱུང་པར་རིས་བརྩམས་ཆོས་སྤྲོག་འདོན་དང་འཚོལ་ཞིབ་བྱ་ཆེད། མི་ཆོས་ཡུལ་ལྗོངས་དང་། </neos> འཕོ་འགྱུར་ཉམས་སྣང་། དམངས་སྲོལ་གོམས་གཤིས། སྤྱི་ཚོགས་འཚོ་བ་སོགས་རྟེན་གཞིར་བཟུང་ནས་པར་ལེན་སྒྱུ་རྩལ་སྤྱད་དེ་བོད་འབྲེལ་ས་ཁུལ་གྱི་ལས་རིགས་ཁག་གི་དཔལ་འབྱོར་དང་སྤྱི་ཚོགས་འཕེལ་རྒྱས་ཀྱི་མངོན་གསལ་དོད་པའི་གྲུབ་འབྲས་གང་ལེགས་མཚོན་པར་བྱ་ཆེད། མཚོ་སྔོན་བོད་སྐད་དུ་ལམ་རྒྱུང་སྒྲོག་བརྙན་འཕྲིན་ཁང་གི་གངས་ལྗོངས་མི་ལོང་དྲ་བས་སྔར་ཡོད་ཀྱི་པར་རིས་ལེ་ཚན་གྱི་རྨང་གཞིའི་སྟེང་”ལོ་བརྒྱའི་གྲུབ་འབྲས། རབ་མཛེས་མཚོ་སྔོན་”ཞེས་པ་བརྗོད་དོན་ཡིན་པའི་ཀྲུང་གོ་གུང་ཁྲན་ཏང་བཙུགས་ནས་ལོ་འཁོར་བརྒྱ་ལོན་པར་རྟེན་འབྲེལ་ཞུ་བའི་པར་ལེན་བརྩམས་ཆོས་བསྡུ་རུབ་ཀྱི་བྱེད་སྒོ་སྤེལ་འཆར་ཡིན་ལ། </eos> བདམས་ཐོན་བྱུང་བའི་བརྩམས་ཆོས་ཚང་མ་གངས་ལྗོངས་མི་ལོང་དྲ་བ་དང་།གངས་ལྗོངས་མི་ལོངAPP ཉིན་རེའི་གངས་ལྗོངས་སྐད་འཕྲིན་སྤྱི་སྟེགས་སོགས་ནས་འགྲེམ་སྟོན་བྱ་རྒྱུ། </eos> བདམས་ཐོན་བྱུང་བའི་བརྩམས་ཆོས་ཚང་མ་དུ་དམངས་ཀྱིས་འདེམས་ཤོག་འཕེན་པ་དང་ཆེད་མཁས་པའི་དཔྱད་འདེམས་བྱ་ཐབས་སྤྱད་དེ་བྱ་དགའ་སྟེར་རྒྱུ་ཡིན།
Bi-GRU	ཀྲུང་གོ་གུང་ཁྲན་ཏང་བཙུགས་ནས་ལོ་འཁོར་བརྒྱ་ལོན་པར་བསུ་མ་བྱེད་པ་དང་། </eos> བརྗོད་བྱ་གཙོ་བོ་དར་སྤེལ་དང་སྣ་མང་ཅན་འདོན་སྤྲོད། གངས་ལྗོངས་མི་ལོང་དྲ་བའི་པར་རིས་ཆེད་སྒྲིག་ལེ་ཚན་གྱི་ནང་དོན་ཕུན་སུམ་ཇེ་ཚོགས་སུ་གཏོང་བ། </neos> རྒྱ་ཆེའི་པར་ལེན་མཁན་ཚོའི་གསར་རྩོམ་གྱི་གསོན་ཤུགས་ཇེ་འཕེལ་དང་། སྔར་ལས་ལྷག་པའི་སྒོ་ནས་མང་ཚོགས་དང་ཁ་ཐག་ཉེ་ཞིང་དྲང་ནུས་ཀྱི་ཕུལ་བྱུང་པར་རིས་བརྩམས་ཆོས་སྤྲོག་འདོན་དང་འཚོལ་ཞིབ་བྱ་ཆེད། མི་ཆོས་ཡུལ་ལྗོངས་དང་། འཕོ་འགྱུར་ཉམས་སྣང་། དམངས་སྲོལ་གོམས་གཤིས། </eos> སྤྱི་ཚོགས་འཚོ་བ་སོགས་རྟེན་གཞིར་བཟུང་ནས་པར་ལེན་སྒྱུ་རྩལ་སྤྱད་དེ་བོད་འབྲེལ་ས་ཁུལ་གྱི་ལས་རིགས་ཁག་གི་དཔལ་འབྱོར་དང་སྤྱི་ཚོགས་འཕེལ་རྒྱས་ཀྱི་མངོན་གསལ་དོད་པའི་གྲུབ་འབྲས་གང་ལེགས་མཚོན་པར་བྱ་ཆེད། མཚོ་སྔོན་བོད་སྐད་དུ་ལམ་རྒྱུང་སྒྲོག་བརྙན་འཕྲིན་ཁང་གི་གངས་ལྗོངས་མི་ལོང་དྲ་བས་སྔར་ཡོད་ཀྱི་པར་རིས་ལེ་ཚན་གྱི་རྨང་གཞིའི་སྟེང་”ལོ་བརྒྱའི་གྲུབ་འབྲས། རབ་མཛེས་མཚོ་སྔོན་”ཞེས་པ་བརྗོད་དོན་ཡིན་པའི་ཀྲུང་གོ་གུང་ཁྲན་ཏང་བཙུགས་ནས་ལོ་འཁོར་བརྒྱ་ལོན་པར་རྟེན་འབྲེལ་ཞུ་བའི་པར་ལེན་བརྩམས་ཆོས་བསྡུ་རུབ་ཀྱི་བྱེད་སྒོ་སྤེལ་འཆར་ཡིན་ལ། </eos> བདམས་ཐོན་བྱུང་བའི་བརྩམས་ཆོས་ཚང་མ་གངས་ལྗོངས་མི་ལོང་དྲ་བ་དང་།གངས་ལྗོངས་མི་ལོངAPP ཉིན་རེའི་གངས་ལྗོངས་སྐད་འཕྲིན་སྤྱི་སྟེགས་སོགས་ནས་འགྲེམ་སྟོན་བྱ་རྒྱུ། བདམས་ཐོན་བྱུང་བའི་བརྩམས་ཆོས་ཚང་མ་དུ་དམངས་ཀྱིས་འདེམས་ཤོག་འཕེན་པ་དང་ཆེད་མཁས་པའི་དཔྱད་འདེམས་བྱ་ཐབས་སྤྱད་དེ་བྱ་དགའ་སྟེར་རྒྱུ་ཡིན།
MLP	ཀྲུང་གོ་གུང་ཁྲན་ཏང་བཙུགས་ནས་ལོ་འཁོར་བརྒྱ་ལོན་པར་བསུ་མ་བྱེད་པ་དང་། </eos> བརྗོད་བྱ་གཙོ་བོ་དར་སྤེལ་དང་སྣ་མང་ཅན་འདོན་སྤྲོད། གངས་ལྗོངས་མི་ལོང་དྲ་བའི་པར་རིས་ཆེད་སྒྲིག་ལེ་ཚན་གྱི་ནང་དོན་ཕུན་སུམ་ཇེ་ཚོགས་སུ་གཏོང་བ། རྒྱ་ཆེའི་པར་ལེན་མཁན་ཚོའི་གསར་རྩོམ་གྱི་གསོན་ཤུགས་ཇེ་འཕེལ་དང་། སྔར་ལས་ལྷག་པའི་སྒོ་ནས་མང་ཚོགས་དང་ཁ་ཐག་ཉེ་ཞིང་དྲང་ནུས་ཀྱི་ཕུལ་བྱུང་པར་རིས་བརྩམས་ཆོས་སྤྲོག་འདོན་དང་འཚོལ་ཞིབ་བྱ་ཆེད། མི་ཆོས་ཡུལ་ལྗོངས་དང་། འཕོ་འགྱུར་ཉམས་སྣང་། དམངས་སྲོལ་གོམས་གཤིས། </eos> སྤྱི་ཚོགས་འཚོ་བ་སོགས་རྟེན་གཞིར་བཟུང་ནས་པར་ལེན་སྒྱུ་རྩལ་སྤྱད་དེ་བོད་འབྲེལ་ས་ཁུལ་གྱི་ལས་རིགས་ཁག་གི་དཔལ་འབྱོར་དང་སྤྱི་ཚོགས་འཕེལ་རྒྱས་ཀྱི་མངོན་གསལ་དོད་པའི་གྲུབ་འབྲས་གང་ལེགས་མཚོན་པར་བྱ་ཆེད། མཚོ་སྔོན་བོད་སྐད་དུ་ལམ་རྒྱུང་སྒྲོག་བརྙན་འཕྲིན་ཁང་གི་གངས་ལྗོངས་མི་ལོང་དྲ་བས་སྔར་ཡོད་ཀྱི་པར་རིས་ལེ་ཚན་གྱི་རྨང་གཞིའི་སྟེང་”ལོ་བརྒྱའི་གྲུབ་འབྲས། </neos> རབ་མཛེས་མཚོ་སྔོན་”ཞེས་པ་བརྗོད་དོན་ཡིན་པའི་ཀྲུང་གོ་གུང་ཁྲན་ཏང་བཙུགས་ནས་ལོ་འཁོར་བརྒྱ་ལོན་པར་རྟེན་འབྲེལ་ཞུ་བའི་པར་ལེན་བརྩམས་ཆོས་བསྡུ་རུབ་ཀྱི་བྱེད་སྒོ་སྤེལ་འཆར་ཡིན་ལ། </eos> བདམས་ཐོན་བྱུང་བའི་བརྩམས་ཆོས་ཚང་མ་གངས་ལྗོངས་མི་ལོང་དྲ་བ་དང་།གངས་ལྗོངས་མི་ལོངAPP ཉིན་རེའི་གངས་ལྗོངས་སྐད་འཕྲིན་སྤྱི་སྟེགས་སོགས་ནས་འགྲེམ་སྟོན་བྱ་རྒྱུ། བདམས་ཐོན་བྱུང་བའི་བརྩམས་ཆོས་ཚང་མ་དུ་དམངས་ཀྱིས་འདེམས་ཤོག་འཕེན་པ་དང་ཆེད་མཁས་པའི་དཔྱད་འདེམས་བྱ་ཐབས་

hope to combine rule-based and DL models to promote performance in the future. At the same time, our research will increase the amount of Tibetan data and attempt to optimize word vectors and pretraining models for downstream tasks. We hope to stimulate new research in Tibetan SBD to improve nonstandard data performance and increase robustness and generalization to the domain and genre variation.

Acknowledgment

This work was supported by the National Key R&D Program of China (No. 2020YFC0832500), the Ministry of Education-China Mobile Research Foundation (No. MCM20170206), the Fundamental Research Funds for the Central Universities (Nos. lzujbky-2022-kb12, lzujbky-2021-sp43, lzujbky-2020-sp02, lzujbky-2019-kb51, and lzujbky-2018-k12), the National Natural Science Foundation of China (No. 61402210), the Science and Technology Plan of Qinghai Province (No. 2020-GX-164), the Google Research Awards and Google Faculty Award, the Provincial Science and Technology Plan (Major Science and Technology Projects-Open Solicitation) (No. 22ZD6GA048), the Gansu Provincial Science and Technology Major Special

Innovation Consortium Project (No. 21ZD3GA002), and the Gansu Province Green and Smart Highway Key Technology Research and Demonstration. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Jetson TX1 used for this research.

References

[1] K. Sirts and K. Peekman, Evaluating sentence segmentation and word tokenization systems on Estonian web texts, in *Human Language Technologies-the Baltic Perspective* (*HIT 2020*). Amsterdam, the Netherlands: IOS Press, 2020, pp. 174–181.

[2] J. Asghar, S. Akbar, M. Z. Asghar, B. Ahmad, M. S. Al-Rakhami, and A. Gumaei, Detection and classification of psychopathic personality trait from social media text using deep learning model, *Comput. Math. Methods Med*., vol. 2021, p. 5512241, 2021.

[3] H. B. Wang, J. X. Wang, Q. Shen, Y. T. Xian, and Y. F. Zhang, Maximum entropy Thai sentence segmentation combined with Thai grammar rules correction, *Univ. Politehn. Bucharest Sci. Bull. Seri. C-Electr. Eng. Comput. Sci.*, vol. 82, no. 1, pp. 19–34, 2020.

[4] T. N. Ho, T. Y. Chong, V. H. Do, V. T. Pham, and E. S. Chng, Improving efficiency of sentence boundary detection by feature selection, In *Intelligent Information and Database Systems*, N. T. Nguyen, B. Trawiński, H. Fujita, and T. P. Hong, eds. Berlin, Germany: Springer, 2016, pp. 169–174.

[5] L. Zhao, A. Zhang, Y. Liu, and H. Fei, Encoding multi-granularity structural information for joint Chinese word segmentation and POS tagging, *Pattern Recogn. Lett*., vol. 138, pp. 163–169, 2020.

[6] A. Elnagar, R. Al-Debsi, and O. Einea, Arabic text classification using deep learning models, *Informat. Process. Manag.*, vol. 57, no. 1, p. 102121, 2020.

[7] P. Ke, H. Ji, S. Liu, X. Zhu, and M. Huang, SentiLR: Linguistic knowledge enhanced language representation for sentiment analysis, in *Proc. Empirical Methods in Natural Language Processing*, Punta Cana, Dominican Republic, 2020, pp. 6975–6988.

[8] J. Ainslie, S. Ontañón, C. Alberti, V. Cvicek, Z. Fisher, P. Pham, A. Ravula, S. Sanghai, Q. Wang, and L. Yang, ETC: Encoding long and structured inputs in transformers, in *Proc. 2020 Conf. Empirical Methods in Natural Language Processing EMNLP*, Punta Cana, Dominican Republic, 2020, pp. 268–284.

[9] B. Bi, C. Li, C. Wu, M. Yan, W. Wang, S. Huang, F. Huang, and L. Si, PALM: Pre-training an autoencoding & autoregressive language model for context-conditioned generation, in *Proc. 2020 Conf. Empirical Methods in Natural Language Processing EMNLP*, Punta Cana, Dominican Republic, 2020, pp. 8681–8691.

[10] A. Singh, B. P. Singh, A. K. Poddar, and A. Singh, Sentence boundary detection for Hindi-English social media text, in *Recent Findings in Intelligent Computing Techniques*, P. K. Sa, S. Bakshi, I. K. Hatzilygeroudis, and M. N. Sahoo, eds. Singapore: Springer, 2018, pp. 207–215.

[11] C. Özbey and Ö. Dinçsoy, Sentence boundary detection in Turkish news with regular expressions, in *Proc. of 2019 27^{th} Signal Processing and Communications Applications Conf*., Sivas, Turkey, 2019, pp. 1–4.

[12] A. Mekki, I. Zribi, M. Ellouze, and L. H. Belguith, Sentence boundary detection of various forms of Tunisian Arabic, *Lang. Res. Eval*., vol. 56, no. 1, pp. 357–385, 2022.

[13] N. Sun and C. Du, News text classification method and simulation based on the hybrid deep learning model, *Complexity*, vol. 2021, p. 8064579, 2021.

[14] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, Deep learning-based text classification: A comprehensive review, *ACM Comput. Surv*., vol. 54, no. 3, p. 62, 2021.

[15] F. Wan and X. He, Tibetan syntactic parsing based on syllables, in *Proc. 3^{rd} Int. Conf. Mechatronics and Industrial Informatics*, Zhuhai, China, 2015, pp. 753–756.

[16] M. Maimaiti, Y. Liu, H. Luan, and M. Sun, Enriching the transfer learning with pre-trained lexicon embedding for low-resource neural machine translation, *Tsinghua Science and Technology*, vol. 27, no. 1, pp. 150–163, 2022.

[17] G. Lobsang, W. Lu, K. Honda, J. Wei, W. Guan, Q. Fang, and J. Dang, Tibetan vowel analysis with a multi-modal Mandarin-Tibetan speech corpus, in *Proc. 2016 Asia-Pacific Signal and Information Processing Association Ann. Summit and Conf.* (*APSIPA*), Jeju, Republic of Korea, 2016, pp. 1–6.

[18] F. C. Wan, H. Z. Yu, X. H. Wu, and X. Z. He, Tibetan syntactic parsing for Tibetan-Chinese machine translation, in *Proc. Int. Conf. Advanced Computer Science and Engineering* (*ACSE 2014*), Guangzhou, China, 2014, pp. 371–376.

[19] L. Liang, F. Tian, and B. Sun, Current status of Tibetan sentiment analysis and cross-language analysis, in *Proc. 2018 6^{th} Int. Conf. Machinery, Materials and Computing Technology* (*ICMMCT 2018*), Jinan, China, 2018, pp. 324–329.

[20] Y. Bie and Y. Yang, A multitask multiview neural network for end-to-end aspect-based sentiment analysis, *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 195–207, 2021.

[21] C. Xu, L. Xie, and X. Xiao, A bidirectional LSTM approach with word embeddings for sentence boundary detection, *J. Signal Process. Syst.*, vol. 90, no. 7, pp. 1063–1075, 2018.

[22] S. Yu, D. Liu, W. Zhu, Y. Zhang, and S. Zhao, Attention-based LSTM, GRU and CNN for short text classification, *J. Intell. Fuzzy Syst. Appl. Eng. Technol,*, vol. 39, no. 1, pp. 333–340, 2020.

[23] S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural Computat*., vol. 9, no. 8, pp. 1735–1780, 1997.

[24] T. A. Le, Sequence labeling approach to the task of sentence boundary detection, in *Proc. 4^{th} Int. Conf. Machine Learning and Soft Computing* (*ICMLSC 2020*), Haiphong City, Vietnam, 2020, pp. 144–148.

[25] H. Wang, J. He, X. Zhang, and S. Liu, A short text

classification method based on N-gram and CNN, *Chin. J. Electron*., vol. 29, no. 2, pp. 248–254, 2020.

[26] Y. Gao, M. Wang, Y. Yu, and C. Zhang, Human motion sequence recognition based on correlation feature selection and multilayer perceptron, in *Proc. SPIE 11584, 2020 Int. Conf. Image, Video Processing and Artificial Intelligence*, Shanghai, China, 2020, p. 115841D.

[27] Y. Liu, E. Shriberg, A. Stolcke, D. Hillard, M. Ostendorf, and M. Harper, Enriching speech recognition with automatic detection of sentence boundaries and disfluencies, *IEEE Trans. Audio Speech Lang. Process*., vol. 14, no. 5, pp. 1526–1540, 2006.

[28] S. Li and B. Gong, Word embedding and text classification based on deep learning methods, in *Proc. 2020 2^{nd} Int. Conf. Computer Science Communication and Network Security* (*CSCNS2020*), Sanya, China, 2020, p. 06022.

[29] A. Al-Doulat, I. Obaidat, and M. Lee, Unstructured medical text classification using linguistic analysis: A supervised deep learning approach, in *Proc. 2019 IEEE/ACS 16^{th} Int. Conf. Computer Systems and Applications* (*AICCSA 2019*), Abu Dhabi, the United Arab Emirates, 2019, pp. 1–7.

[30] A. Zhang, B. Li, W. Wang, S. Wan, and W. Chen, MII: A novel text classification model combining deep active learning with Bert, *Comput. Mater. Con*., vol. 63, no. 3, pp. 1499–1514, 2020.

[31] M. V. Abrahams and M. G. Kattenfeld, The role of turbidity as a constraint on predator-prey interactions in aquatic environments, *Behav. Ecol. Sociobiol*., vol. 40, no. 3, pp. 169–174, 1997.

[32] J. Read, R. Ridan, S. Oepen, and L. J. Solberg, Sentence boundary detection: A long solved problem? in *Proc. COLING 2012: Posters*, Mumbai, India, 2012, pp. 985–994.

[33] M. D. Riley, Some applications of tree-based modelling to speech and language, in *Proc. Workshop on Speech and Natural Language*, Cape Cod, MA, USA, 1989, pp. 339–352.

[34] D. D. Palmer and M. A. Hearst, Adaptive multilingual sentence boundary disambiguation, *Computat. Linguist*., vol. 23, no. 2, pp. 241–267, 1997.

[35] J. C. Reynar and A. Ratnaparkhi, A maximum entropy approach to identifying sentence boundaries, in *Proc. 5^{th} Conf. Applied Natural Language Processing*, Washington, DC, USA, 1997, pp. 16–19.

[36] D. Gillick, Sentence boundary detection and the problem with the U.S., in *Proc. Human Language Technologies: 2009 Ann. Conf. North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, Boulder, CO, USA, 2009, pp. 241–244.

[37] A. Mikheev, Tagging sentence boundaries, in *Proc. 1^{st} North American Chapter of the Association for Computational Linguistics Conf.*, Seattle, WA, USA, 2000, pp. 264–271.

[38] A. Mikheev, Periods, capitalized words, etc., *Comput. Linguist*., vol. 28, no. 3, pp. 289–318, 2002.

[39] T. Kiss and J. Strunk, Unsupervised multilingual sentence boundary detection, *Comput. Linguist*., vol. 32, no. 4, pp. 485–525, 2006.

[40] O. Hellwig, Detecting sentence boundaries in Sanskrit texts, in *Proc. COLING 2016, 26^{th} Int. Conf. Computational Linguistics: Technical Papers*, Osaka, Japan, 2016, pp. 288–297.

[41] H. H. Hock, Some issues in Sanskrit syntax, in *Proc. Seminar on Sanskrit Syntax and Discourse Structures*, Pairs, France, 2013, pp. 13–15.

[42] L. Yang, A. Stolcke, E. Shriberg, and M. P. Harper, Using conditional random fields for sentence boundary detection in speech, in *Proc. 43^{rd} Ann. Meeting of the Association for Computational Linguistics*, Ann Arbor, MI, USA, 2005, pp. 451–458.

[43] Y. Zhao, C. Wang, and G. Fu, A CRF sequence labeling approach to Chinese punctuation prediction, in *Proc. 26^{th} Pacific Asia Conf. Language, Information, and Computation*, Bali, Indonesia, 2012, pp. 508–514.

[44] W. N. Zhao, H. D. Liu, X. Yu, J. Wu, and P. Zhang, The Tibetan sentence boundary identification based on legal texts, (in Chinese), in *Proc. National Symp. on Computational Linguistics for Young People* (*YWCL2010*), Wuhan, China, 2010, pp. 490–496.

[45] R. Cai and T. Ji, Researches of speech classification methods based on Tibetan repertoire, (in Chinese), *J. Northwest Univ. Nat*. (*Nat. Sci.*), vol. 26, no. 2, pp. 39–42, 2005.

[46] Q. J. Ren and J. C. R. An, Research on automatic recognition method of Tibetan sentence boundary, (in Chinese), *China Comput. Commun*., vol. 8, no. 316, pp. 62–63, 2014.

[47] Z. T. Cai, Research on the automatic identification of Tibetan sentence boundaries with maximum entropy classifier, (in Chinese), *Comput. Eng. Sci*., vol. 34, no. 6, pp. 187–190, 2012.

[48] X. Li, Z. Cai, W. Jiang, Y. Lv, and Q. Liu, A maximum entropy and rules approach to identifying Tibetan sentence boundaries, (in Chinese), *J. Chin. Informat. Proc*., vol. 25, no. 4, pp. 39–44, 2011.

[49] W. Z. Ma, Z. Wanme, and Z. Nima, Method of identification of Tibetan sentence boundary, (in Chinese), *J. Tibet Univ*., vol. 27, no. 2, pp. 70–76, 2012.

[50] W. Zhao, X. Yu, H. Liu, L. Li, L. Wang, and J. Wu, Modern Tibetan auxiliary ending sentence boundary detection, (in Chinese), *J. Chin. Informat. Proc*., vol. 27, no. 1, pp. 115–119, 2013.

[51] X. Zha and B. Luo, Based on function words and sentence patterns Tibetan sentence extraction method, (in Chinese), *J. Northwest Minzu Univ*. (*Nat. Sci.)*), vol. 39, no. 4, pp. 39–43&62, 2018.

[52] C. Z. M. Que, Q. C. R. Hua, R. D. Z. Cai, and W. J. Xia, Tibetan sentence boundary recognition based on mixed strategy, (in Chinese), *J. Inner Mongolia Normal Univ.* (*Nat. Sci. Ed.*), vol. 48, no. 5, pp. 400–405, 2019.

[53] P. Koehn, EUROPARL: A parallel corpus for statistical machine translation, in *Proc. Machine Translation Summit X: Papers*, Phuket, Thailand, 2005, pp. 79–86.

Fenfang Li received the MEng degree from Northwest Normal University, China in 2016. She is a lecturer at the School of Computer Science and Technology, Northwest Normal University, Lanzhou, China. She is currently a PhD candidate at the School of Information Science and Engineering, Lanzhou University, Lanzhou, China. Her research interests include deep learning, natural language processing, computational linguistics, and Tibetan information processing.

Hui Lv received the MEng degree from Lanzhou Jiaotong University, Lanzhou, China in 2014. She is a lecturer at the College of Electrical Engineering, Northwest Minzu University, China. She is currently a PhD candidate at the School of Information Science and Engineering, Lanzhou University, Lanzhou, China. Her research interests include deep learning, natural language processing, computational linguistics, and Tibetan information processing.

Yiming Gao received the BEng degree from Henan University of Economics and Law, Zhengzhou, China in 2020. She is currently a master student at the School of Information Science and Engineering, Lanzhou University, Lanzhou, China. Her research interests cover deep learning, natural language processing, and Tibetan information processing.

Qingguo Zhou received the PhD degree from Lanzhou University, Lanzhou, China in 2005. He is currently a professor at the School of Information Science and Engineering, Lanzhou University, Lanzhou, China. His research interests include deep learning, natural language processing, embedded real-time systems, safety-critical systems, and smart transportation. He is a fellow of the International Institute of Electrical Engineers and a recipient of the New Century Talent Fund of the Ministry of Education.

Dolha received the PhD degree from Northwest Minzu University, Lanzhou, China in 2012. He is currently a professor at the Key Laboratory of China's National Linguistic Information Technology, Northwest Minzu University, Lanzhou, China. His research interests include computational linguistics, natural language processing, and Tibetan information processing.

Yan Li received the PhD degree from Lanzhou University, Lanzhou, China in 2012. She is currently an assistant professor at the School of Information Science and Engineering, Lanzhou University, Lanzhou, China. Her research interests include machine learning, natural language processing, and Tibetan information processing.