Transition based neural network dependency parsing of Tibetan

Jiecairang Duo[1,2,3], *Quecairang* Hua[1,2,3,*], *Keyou* Huan[1,2,3], and *Rangdangzhi* Cai[1,2,3]

[1]College of Computer Science and Technology, Qinghai Normal University, Qinghai Xining 810016
[2]Tibetan Information Processing and Machine Translation Key Laboratory of Qinghai Province, Qinghai Xining 810008
[3]Key Laboratory of Tibetan Information Processing, Ministry of Education, Qinghai Xining 810008

Abstract. In order to improve the performance of Tibetan natural language processing applications such as machine translation, sentiment analysis and other tasks, this article proposes a neural network-based method for syntactic analysis of Tibetan language dependence. Part of the corpus of Qinghai Normal University's part-of-speech tag set is marked by the corresponding mapping relationship is transformed into the corpus annotated by the national standard part-of-speech tag set. At the same time, the CoNLL format Tibetan language dependency syntax tree library is constructed, and the method of shift-reduce plus neural network is adopted to systematically study and analyze the Tibetan language dependency syntax. Thereby improving the quality of Tibetan dependency syntactic analysis, and its accuracy rate reaches UAS:94.59%

1 Introduction

Tibetan dependent syntax analysis is one of the core technologies of Tibetan natural language processing. It mainly analyzes the components of Tibetan sentences effectively, lays an effective foundation for various high-level applications such as emotional classification, machine translation and extraction of entity relations, etc. At the same time, it also has certain requirements for basic task lexical analysis in Tibetan natural language processing. Reasoning serves as a link between the preceding and the following. Although the research on Tibetan natural language processing has also been rapidly improved with the development of network information technology, and now has also made more fruitful research results, such as Tibetan word segmentation, part-of-speech labeling [18], semantic dependency analysis [2][10].However, the research on Tibetan syntax analysis still faces many challenges. Dependent syntax analysis methods based on in-depth learning are basically blank. Therefore, it is necessary to draw on and consolidate some of the earlier statistical methods based on [1][16] and to use some appropriate neural network methods to extract and analyze the features of various Tibetan sentence patterns.

The more common methods in the field of dependency analysis work today are Transition-based [17] and Graph-based [9].The graph-based method establishes a directed complete graph in which the maximum spanning tree of dependency analysis is solved to

[*] Corresponding author: duojiecairang0626@qq.com

obtain the optimal solution. Transfer-based dependency parsing models the entire sentence decoding process as a finite automaton problem (analog to the parsing component in the compiler).Tibetan sentence patterns include statement sentences, exclamation sentences, rhetoric sentences, interrogative sentences and imperative sentences, which are basically represented by the wedge-shaped symbol '།'.We use the dependency analysis method based on transfer, which starts from the start state of the transfer and goes from one state to another continuously, and finally reaches the end state, with the corresponding state sequence tree of the end state as the analysis result. The state includes the stack of words being processed, the buffer holding the words to be processed, and the memory generating the dependent arc. The transfer sequence generally includes the move-in protocol and the generation of dependent arcs. The goal of transfer dependency parsing is to train a classifier to predict the next transfer sequence or action for a given transition state. Thus, the whole sentence structure can be expressed through the dependency between words.

2 Related research

Tibetan has a history of more than 1,500 years since its creation. After many years of long evolution, Tibetan has gradually moved to electronic information technology and achieved good results. Some experts and scholars at home and abroad have studied Tibetan natural language from various perspectives, including lexical analysis, syntax analysis, semantic analysis, machine translation and emotional classification. Dependent syntax analysis is also indispensable in Tibetan. At present, the study on Tibetan dependency analysis has only allowed others to propose a discriminant-based Tibetan dependency syntax analysis [16]. The perceptual machine method is used to train the parsing model, and the CYK bottom-up algorithm decodes to generate the maximum spanning tree. Xia Wuji and others have studied and proposed projection-based semantic dependency analysis of Tibetan [2], studied projection-based semantic dependency analysis of Tibetan and built a semantic dependency tree, and classified the dependency arcs by maximum entropy model. The dependent syntax analysis of Tibetan compound sentences [1] was also proposed by Genius et al. and its method is to use maximum entropy to extract the features and get the maximum entropy model to complete parameter estimation. The study of Tibetan dependency syntax analysis based on in-depth learning is basically blank. For this reason, this paper will make a transfer-based dependency analysis of Tibetan.

The Tibetan dependent syntax based on the neural network has a higher speed than the discriminant. In order to better assist Tibetan natural language processing technology, this paper conducts a transfer-based neural network Tibetan dependency syntax analysis study, to analyze the existence and type of dependency between predicted words, and to take the first step in deep language understanding. In this article, we use low-dimensional (50-dimensional), dense word embedding, but this method is not perfect. First, from a statistical point of view, too many feature weights with inaccurate predictions are used. In this way, Lexicalization and higher-order interaction features are important to improve the performance of these systems, but there is not enough data to properly weigh most of these features. For this reason, the introduction of more supportive features, such as word class features, will also provide significant support in improving the performance of dependency analysis.

3 Tibetan dependency syntax analysis and difficulties

Firstly, the Tibetan text corpus is preprocessed by word segmentation and part of speech tagging [18], which is divided into two categories: Qinghai Normal University annotated corpus specification and national standard tagging corpus specification. The dependency tagging of Tibetan language dependency syntax analysis uses the 36 types of dependency marked by Huaquecairang et al.

3.1 Tibetan dependency syntax analysis

The dependency syntacticstructure of Tibetan language relies on its part of speech, dependency tag and its phrase structure, as shown in Figure 1. An example: "དགེ་རྒན་གྱིས་ང་ལ་དཔྱད་རྩོམ་མཛུབ་སྟོན་གནང་།"（The teacher will guide my thesis.） In this example, the arrow (dependency arc) points to its child nodes, in which ROOT is the root node, SUB is the subject predicate relationship, GZCX is the case auxiliary in Tibetan grammar, OBJ is the object relationship, GENI is the case of the Ge auxiliary in Tibetan grammar, AUXIW is the auxiliary verb structure relationship, PUNCT is the punctuation structure relationship in Tibetan. All the above are based on the dependency labeling standard of Qinghai Normal University. The part of speech (POS tagging) of each word and its phrase structure are marked at the top of Tibetan characters in the figure.

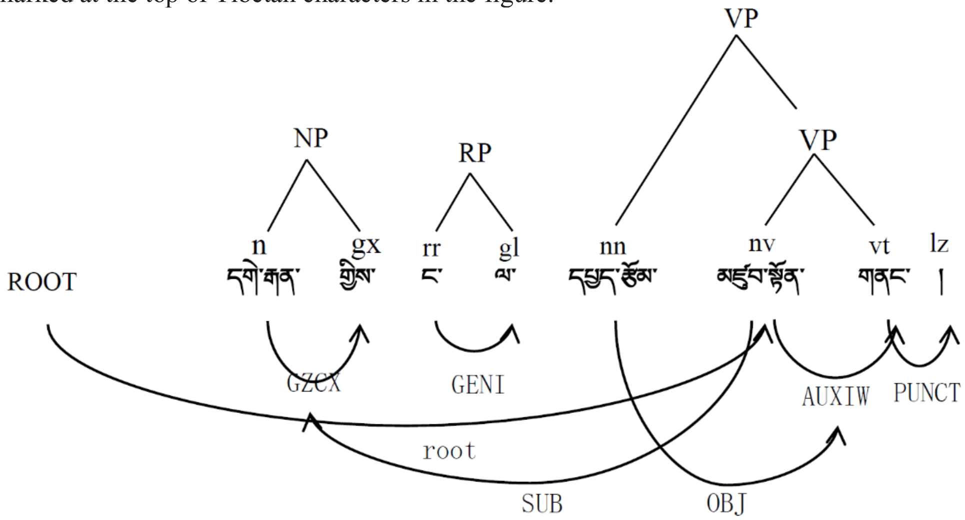


Fig. 1. Schematic diagram of Tibetan dependency syntax tree.

3.2 Difficulties existing

The main task of Tibetan lexical analysis is to separate Tibetan text into meaningful words, determine the category of each word and shallow ambiguity elimination, and identify some longer proper nouns. This paper uses the Tibetan part of speech tagging standards of 67 categories of Qinghai Normal University and the national standard 92 categories of Tibetan part of speech tagging norms, and makes corresponding mapping relations to transform each other.

Usually, the important information of a Tibetan sentence is located at the end or end of the sentence, the part taken from the front of the sentence and the part taken from the end of the sentence to calculate the core components of the sentence. The recognition of core verbs (ROOT nodes) by dependent parsing in Tibetan sentences is also of great importance.

Before the dependency relationship of case auxiliaries, it was only genius that allowed others to propose discriminant-based Tibetan dependency syntax analysis methods [16] and dependency syntax analysis of Tibetan compound sentences [1] where Tibetan case auxiliaries act as the father nodes of the belonging words.On the contrary, this paper presents the father nodes of the case auxiliaries, as shown in Figure 2,3 below “ང་ཚོ་ཡང་བཞི་པའི་གྲལ་དུ་ཞུགས་ཐུབ་སོང་།”(we joined the fourth group).Figure 2 is an example of a case assistant as a father node, while Figure 3 is a child node. The dependencies shown in Figure 3 are relatively comprehensive.

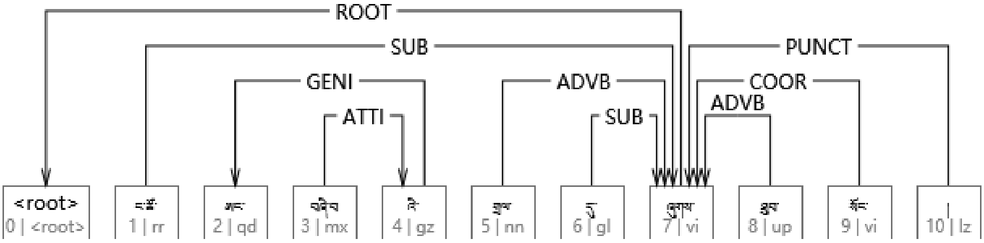


Fig. 2. Father node with participle as owner

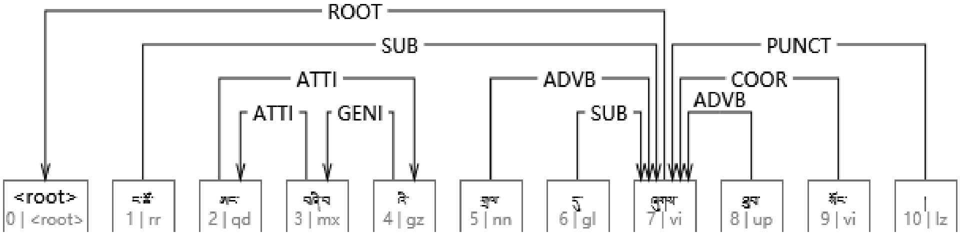


Fig. 3. Father node of ownership case auxiliary

4Transition +neural network dependency analysis method

Using the arc standard algorithm of one of the transfer systems as the basis of the transfer analyzer, a series of transformed sequences are made on the Tibetan text, resulting in dependent arcs and predicting the next transfer action. The transfer process is described in detail below.

4.1Tibetan dependent syntax analysis based on transition

First, each transfer process is based on the current conditions or state configuration to make a decision (transition), update the state after the transfer to the next decision process (greedy algorithm for decision making), each step selects the transfer result that is currently considered the best, which only loses a small accuracy and greatly improves the speed.

State configuration: $\mathrm{c} = (\mathrm{S}, \mathrm{B}, \mathrm{A})$， Sis the stack holding the transfer process (Stack), B is the input cache queue (Buffer), A is the set of dependency arcs for the currently analyzed dependency edges (dependencies, including dependency tags), and a sentence is assumed to be w where $w_1, w_2, \cdots, w_\mathrm{n}, w_\mathrm{n}$ is the word in the sentence, initial state $\mathrm{S} = [\mathrm{ROOT}]$, $\mathrm{B} = [w_1, w_2, \cdots, w_\mathrm{n}]$， $\mathrm{A} = \emptyset$， if the queue of one state is empty and $\mathrm{S} = [\mathrm{ROOT}]$, then this is the last state, the end state, to end the entire transfer process.

Transition: There are three states: LEFT-ARC, RIGHT-ARC and SHIFT.S_i represents the top element of the stack (stack first in, stack last out), b_i corresponds to the first element in the queue (queue FIFO), then there are three operations:

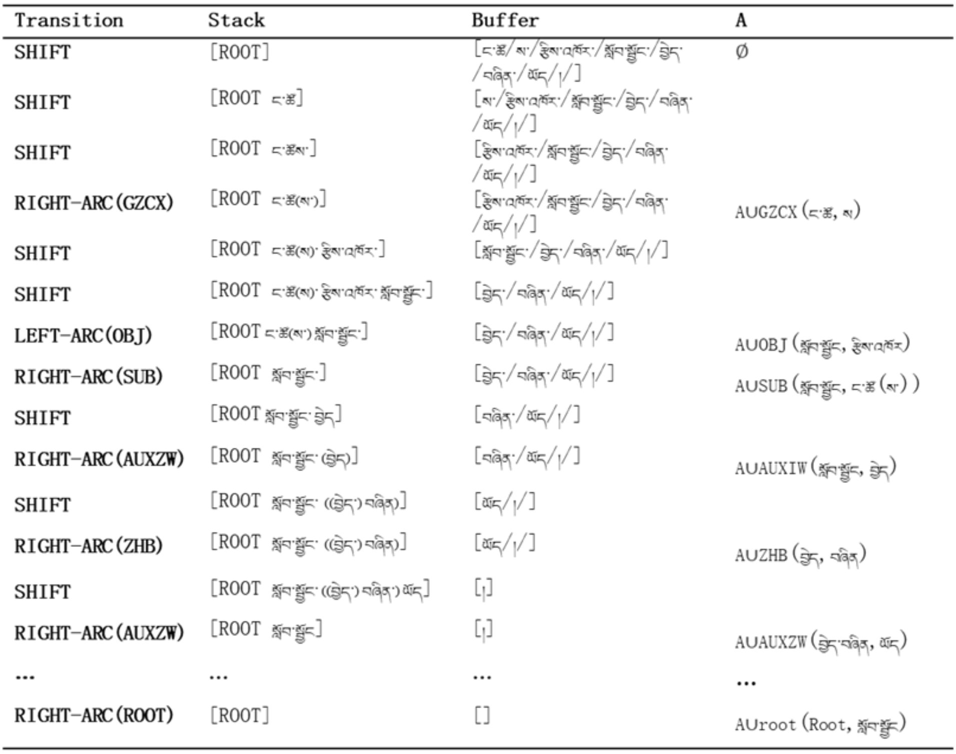
$\mathrm{LEFT}-\mathrm{ARC(l)}$:When the number of elements in the stack is greater than or equal to 2, add a dependency as $S_1 \rightarrow S_2$, and the label for the dependent arc is l, then S_2 Remove from the stack.

$\mathrm{RIGHT}-\mathrm{ARC(l)}$:When the number of elements in the stack is greater than or equal to 2, add a dependency as $S_2 \rightarrow S_1$, and the label for the dependent arc is l, then S_1 Remove from the stack.

SHIFT: When the number of elements in the buffer queue is greater than or equal to 1, b_1Remove from the queue and add to the top of the stack.

Table1 is an example of the whole process of transferring Tibetan sentence dependencies, with the example "ང་ཚོས་རྩིས་འཁོར་སློབ་སྦྱོང་བྱེད་བཞིན་ཡོད།" translated as: (We are learning computer.)

Table 1. Example of transfer sequence of Tibetan sentences



Transition	Stack	Buffer	A
SHIFT	[ROOT]	[ང་ཚོ/ས་/རྩིས་འཁོར་/སློབ་སྦྱོང་/བྱེད་/བཞིན་/ཡོད/།/]	∅
SHIFT	[ROOT ང་ཚོ]	[ས་/རྩིས་འཁོར་/སློབ་སྦྱོང་/བྱེད་/བཞིན་/ཡོད/།/]	
SHIFT	[ROOT ང་ཚོས་]	[རྩིས་འཁོར་/སློབ་སྦྱོང་/བྱེད་/བཞིན་/ཡོད/།/]	
RIGHT-ARC(GZCX)	[ROOT ང་ཚོ(ས་)]	[རྩིས་འཁོར་/སློབ་སྦྱོང་/བྱེད་/བཞིན་/ཡོད/།/]	AUGZCX (ང་ཚོ, ས)
SHIFT	[ROOT ང་ཚོ(ས)་ རྩིས་འཁོར་]	[སློབ་སྦྱོང་/བྱེད་/བཞིན་/ཡོད/།/]	
SHIFT	[ROOT ང་ཚོ(ས)་ རྩིས་འཁོར་ སློབ་སྦྱོང་]	[བྱེད་/བཞིན་/ཡོད/།/]	
LEFT-ARC(OBJ)	[ROOT ང་ཚོ(ས་) སློབ་སྦྱོང་]	[བྱེད་/བཞིན་/ཡོད/།/]	AUOBJ (སློབ་སྦྱོང, རྩིས་འཁོར)
RIGHT-ARC(SUB)	[ROOT སློབ་སྦྱོང་]	[བྱེད་/བཞིན་/ཡོད/།/]	AUSUB (སློབ་སྦྱོང, ང་ཚོ (ས་))
SHIFT	[ROOT སློབ་སྦྱོང་ བྱེད]	[བཞིན་/ཡོད/།/]	
RIGHT-ARC(AUXZW)	[ROOT སློབ་སྦྱོང་ (བྱེད)]	[བཞིན་/ཡོད/།/]	AUAUXIW (སློབ་སྦྱོང, བྱེད)
SHIFT	[ROOT སློབ་སྦྱོང་ ((བྱེད་) བཞིན)]	[ཡོད/།/]	
RIGHT-ARC(ZHB)	[ROOT སློབ་སྦྱོང་ ((བྱེད་) བཞིན)]	[ཡོད/།/]	AUZHB (བྱེད, བཞིན)
SHIFT	[ROOT སློབ་སྦྱོང་ ((བྱེད་) བཞིན་) ཡོད]	[།]	
RIGHT-ARC(AUXZW)	[ROOT སློབ་སྦྱོང]	[།]	AUAUXZW (བྱེད་བཞིན, ཡོད)
...	...	...	...
RIGHT-ARC(ROOT)	[ROOT]	[]	AUroot (Root, སློབ་སྦྱོང)

The dependency tags between the Tibetan text and the sentence components of the word segmentation are input into the neural network classifier through the transfer process.

4.2 Dependency analysis model based on neural network

A standard neural network with an implicit layer is established. As shown in Figure 4, where the corresponding embedded representations of the elements we selected from S^w, S^t, S^l are added to the input layer. The feature vectorization of word segmentation and part-of-speech tags and dependent tags is input.

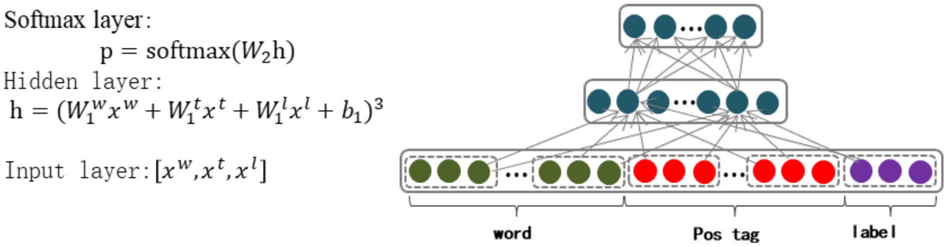

Softmax layer:
$p = \mathrm{softmax}(W_2 h)$
Hidden layer:
$h = (W_1^w x^w + W_1^t x^t + W_1^l x^l + b_1)^3$
Input layer: $[x^w, x^t, x^l]$
word
Pos tag
label

Fig. 4. Neural Network Structure

Assume the size of the input set is n_w, n_t, n_l, $S^w = [w_1, \cdots, w_{n_w}]$, for n_wVectorize all words to get $x^w = [e^w_{w_1}, e^w_{w_2}, \cdots, e^w_{w_{n_w}}]$, so $x^t$and x^lcan be obtained.Connect to a fully connected network and use the activation function (cube function) followed by a fully connected layer to get the classification probability using softmax.

4.3 Feature structure

A brief description of the feature design is as follows:

There are 18 features of the word class, of which $.w$ denotes the corresponding word, lc_1 means S_1at the top of the stack leftmost child node, rc_1 means S_1at the top of the stack rightmost child node.

Word Features:$S_1.w, S_2.w, S_3.w, b_1.w, b_2.w, b_3.w, lc_1(S_1).w, lc_1(S_2).w, rc_1(S_1).w,$
$rc_1(S_2).w, lc_2(S_1).w, lc_2(S_2).w, rc_2(S_1).w, rc_2(S_1).w, lc_1(lc_1(S_1)).w,$
$lc_1(lc_1(S_2)).w, rc_1(rc_1(S_1)).w, rc_1(rc_1(S_2)).w$

Eighteen parts of speech have part-of-speech characteristics, of which t denotes the part-of-speech of the corresponding words.

Features of pos tag:$S_1.t, S_2.t, S_3.t, b_1.t, b_2.t, b_3.t, lc_1(S_1).t, lc_1(S_2).t, rc_1(S_1).t, rc_1(S_2).t,$
$lc_2(S_1).t, lc_2(S_2).t, rc_2(S_1).t, rc_2(S_1).t, lc_1(lc_1(S_1)).t, lc_1(lc_1(S_2)).t,$
$rc_1(rc_1(S_1)).t, rc_1(rc_1(S_2)).t$

The dependency label has 12 features, of which l represents the corresponding dependency.

Arc labels:$lc_1(S_1).l, lc_1(S_2).l, rc_1(S_1).l, rc_1(S_2).l, lc_2(S_1).l, lc_2(S_2).l, rc_2(S_1).l,$
$rc_2(S_1).l, lc_1(lc_1(S_1)).l, lc_1(lc_1(S_2)).l, rc_1(rc_1(S_1)).l, rc_1(rc_1(S_2)).l$

All layers are fully connected, with the implied layer followed by softmax, and the output is one of the probabilities in $2N_l + 1$.The activation function of the implied layer uses a cubic function, which fits exactly the combination of 48 features in the participle+part of speech+dependent tag.

5 Experiment

To support Tibetan dependent parsing based on transfer neural network, more than 10,000 Tibetan dependent parsing corpuses are constructed into training, test and verification sets. The neural network described in the previous section has 48 features, since it is nonlinear, including cube functions, which can also help to extract the relationships between features, such as whether they occur together or not. However, traditional methods can only manually combine these 48 features to represent the situation that occurs together, which will result in hundreds of combinations, that is, hundreds of features, but each sentence may occur very rarely, so the feature vectors will be sparse, time-consuming and relatively poor generalization ability.

5.1 Experimental data

The corpus uses the new Tibetan-dependent corpus format, Conference on Computational Natural Language Learning, which ends in CoNLL, as shown in Table 2.

Table 2. Corpus labeled in CoNLL format.

ID	FORM	LEMMA	CPOSTAG	POSTAG	FEATS	HEAD	DEPREL		
1	བདག་	_	rr	rr	_	5	SUB	_	_
2	ལྷ་ས་	_	ns	ns	_	3	FOBJ	_	_
3	ནས་	_	gj	gj	_	4	ADVB	_	_
4	ཡོང་བ་	_	vi	vi	_	5	OBJ	_	_
5	ཡིན	_	up	up	_	0	ROOT	_	_
6	།	_	lz	lz	_	5	PUNCT	_	_

Only the first eight columns are used in this text, meaning:

1) (ID) denotes the ordinal number of the current word in a sentence, starting from 1;

2) (FORM) denotes the current word or punctuation;

3) (LEMMA) denotes the prototype or stem of the current word (or punctuation);

4) (CPOSTAG) denotes the part of speech (coarse grained) of the current word;

5) (POSTAG) denotes the part of speech (fine grained) of the current word;

6) (FEATS) denotes the characteristics of a sentence. In the experiment, this column was not used and was replaced with an underscore.

7) (HEAD) is the central word of the current word;

8) (DEPREL) denotes the dependence of the current word on the central word.

In CoNLL format, each word occupies one line, and the non-value column is underlined '_' instead, the sentence is separated by a blank line (see Table 2.), and a symbol '_' is added at the end of the dependency to increase the feature.

5.2 Experimental results and analysis

A total of 10023 sentences were used in the experiment, of which 10023 sentences were in the training set. 500 sentences in the validation set and 500 sentences in the test set were in the CoNLL format mentioned above.

The main measures of dependency analysis are as follows:

$$Acc = \frac{Correct\ Dependency}{Total\ Dependency} \times 100\% \tag{1}$$

$$UAS = \frac{Correct\ tag\ relationship}{All\ tag\ relationships} \times 100\% \tag{2}$$

$$LAS = \frac{correct\ (tag\ relation + relation\ label)}{all\ tag\ relations} \times 100\% \tag{3}$$

Table 3. Experimental result data

Label Specification	Training set(sentences)	Dev set(sentences)	Test set(sentences)	UAS	
				Test	**Dev**
Qinghai normal university	10023	500	500	94.59	86.44
National standard	10023	500	500	93.28	86.18

On a test corpus in Tibetan CoNLL format, the dependency correctness rate (UAS) for classifying non-dependent tags is classified by a neural network model. The results are shown in Table 3.

The experimental results show that the dependency parsing accuracy based on the neural network is 94.59%, which has room for improvement. This experiment is the first time to use the neural network method to analyze the dependency of Tibetan. The small size of the database has not completely covered all the Tibetan sentence patterns, and the lexical analysis has not been fully used, thus causing analysis errors in model training.

6 Conclusion

On the basis of Tibetan word segmentation and part-of-speech labeling, this paper builds a Tibetan dependency tree in CoNLL format. Combining the characteristics of Tibetan words, part-of-speech labeling and dependency, a neural network model and a transfer-based neural network method are selected as the dependent syntax analysis methods for part-of-speech labeling corpus and national standard part-of-speech labeling corpus of Qinghai Normal University, as well as the depth of Tibetan information processing. Layer theory provides theoretical basis and data support, and is of great significance to the construction of Tibetan dependent syntax tree, the progress of syntax analysis research, and the development of Tibetan information processing.

Qinghai Province Science and technology plan project (2017-GX-146, 2019-SF-129), Innovation and entrepreneurship project of Qinghai Normal University.

References

1. Hua Quecairang,Zhao Haixing.Dependency syntactic analysis of Tibetan Compound Sentences [J].Journal of Chinese Information Processing,2016,30(6):224-22.
2. Xia Wuji, Hua Quecairang. Research on Tibetan semantic dependency analysis based on projection [J]. Computer Engineering and Science, 2019, 41(10):1868-1873.
3. Wei Tiantian. A study on the completion of verb object structure in dependency parsing based on Linguistic Theory [J]. Journal of Hanjiang Normal University, 2019.
4. Liu Hang, Liu Mingtong, Zhang Yujie,etc. Chinese Integrative dependency analysis model based on Stack-Tree LSTM [J]. Journal of Chinese Information Processing, 2019, 33(1).
5. TURGHUN Osman, YANG Yating, WANG Lei,etc.Construction of Uyghur dependency tree database based on Chinese Uyghur mapping [J]. Journal of Chinese Information Processing, 2019, 033(001):103-110.
6. Liu Shiyi. Research model of dependency parsing based on deep learning [D]. 2019.
7. Wang Hengjun, Si Nianwen, Song Yulong,etc.Neural network dependency parsing model combined with global vector features [J]. Journal of Communications, 2018,

039(002):53-64.

8. Liu Mingtong, Zhang Yujie, Xu Jin'an, et al. Neural network dependency parsing model based on multi feature fusion coding [J]. Chinese Journal of information technology, 2018, 32 (12)
9. Chen Zhiqun, Wang Bing, Wang Rongbo, et al. Graph structure dependency parsing based on bidirectional LSTM [J]. Journal of Hangzhou University of Electronic Science and technology, 2018, 038 (001): 43-48
10. Xia Wuji, Hua Quecairang. Tibetan semantic analysis based on dependency tree [J]. Journal of Tsinghua University (NATURAL SCIENCE EDITION), 2019 (9)
11. Toudan cairang, Nimazaxi, wanmezhaxi. Research on the construction technology of Tibetan dependency tree database [J]. Plateau science research, 2018, 2 (03): 103-109.
12. Renzeng duojie, luosanggaden. The valence of Tibetan Verbs and sentence generation [J]. Computer knowledge and technology, 2017, 013 (026): 178-179.
13. Dorje droma. Tibetan "n_1+G+N_2" semantic relation analysis research [J]. Plateau science research, 2019, 3 (01): 108-114.
14. Zhang junchi. Research on dependency parsing model based on recurrent neural network [D]. 2016.
15. Bai Yawen. Analysis and Research on Kazakh sentence method based on transfer neural network [D]. 2019.
16. Hua Quecairang, Zhao Haixing. Tibetan dependency parsing based on discriminant analysis [J]. Computer engineering, 2013, 39(4):300-304.
17. Rasooli M S , Tetreault J . Yara Parser: A Fast and Accurate Dependency Parser[J]. Computer Science, 2015.
18. Hua Quecairang, Liu Qun, Zhao Haixing. Research on discriminative part of speech tagging in Tibetan texts [J]. Journal of Chinese Information Processing,2014,28(02):56-60.