

An End-to-End Method for Joint Extraction of Tibetan Entity Relations

Yuan Sun^{1,2*}, Sisi Liu^{1,2}, Tianci Xia^{1,2}, Xiaobing Zhao^{1,2}

¹School of Information Engineering, Minzu University of China, Beijing, China

²Minority Languages Branch, National Language Resource and Monitoring Research Center, Beijing, China

Email: *tracy.yuan.sun@gmail.com

How to cite this paper: Sun, Y., Liu, S.S., Xia, T.C. and Zhao, X.B. (2021) An End-to-End Method for Joint Extraction of Tibetan Entity Relations. *Journal of Computer and Communications*, 9, 132-142.
<https://doi.org/10.4236/jcc.2021.99010>

Received: July 1, 2021

Accepted: September 27, 2021

Published: September 30, 2021

Abstract

Entity relation extraction is to find entities and relations from unstructured texts, which is beneficial to the applications of knowledge graphs and question answering systems. The traditional methods handle this task in a pipelined manner which extracts the entities first and then recognizes their relations. This framework may lead to error delivery. In order to tackle this problem, this paper proposes an end-to-end method for joint extraction of Tibetan entity relations which can extract entities and relations at the same time. According to the Tibetan spelling characteristics, this paper processes the Tibetan corpus by word-level and character-level respectively. Combined with part of speech tagging, we use the end-to-end model to convert the entity relation extraction task to the tagging problem. Finally, the experimental results show that the proposed method is better than the baseline.

Keywords

End-to-End Model, Tibetan Entity Relation, Joint Method, Character-Level Processing

1. Introduction

The purpose of entity relation extraction is to extract the semantic relation between entity pairs in a sentence and make unstructured text into structured text. For example, `<e1>The Forbidden City</e1>` is located in the center of `<e2>Beijing </e2>`. Entity relation extraction can automatically identify entities The Forbidden City and Beijing as location relation. Hence, the extracted result is {The Forbidden City, located in, Beijing}, which called triplet here [1].

The traditional approach, which is called pipeline method, is divided into two steps: named entity recognize (NER) and relation classification (RC). It is to ex-

tract entities first, and then recognize their relations [2]. The typical NER model is based on statistical models. At present, many neural network models are also applied to NER tasks. There are mainly two methods for RC. The first is manual processing method based on feature extraction. The second is a processing method based on the neural network models. The advantage of pipeline method is that it can make the task easy to deal with, and each component can be more flexible, but it ignores the relevance between two subtasks, and each subtask can cause errors and propagate to next subtask. For example, errors generated by entity recognition will be passed to relation classification [3].

Different from the traditional methods, the framework of the joint model is to combine the entity extract task and the relation classification task with a simple model [1]. It integrates and displays the information of entities and relations to achieve better results. For example, Ren [6] proposes a framework based on Distant Supervision and Weakly Supervision to extract entities and relations jointly in texts, which includes generation of candidate sets, joint training of entities and vector spaces, and reasoning and prediction of entity types and relation types. Yang *et al.* [7] uses the joint reasoning model to extract viewpoint entities and viewpoint relations. In the viewpoint of entities identification task, the CRF model is used to transform the recognition into a tagging scheme task. In the viewpoint of relations extraction task, the viewpoint-parameter model is used to identify viewpoint relations. Singh [8] uses joint reasoning to perform three tasks: entities labeling, relations extraction, and common reference. The joint graph model is used to combine the three tasks together and to optimize the parameters of the joint inference model by learning and reasoning. Miwa and Bansal [9] propose a relation extraction model for joint entity detection parameter sharing. There are two bidirectional LSTM-RNNs in the model, one is based on word sequence (bidirectional sequential LSTM-RNNs), which is mainly used for entities detection. The other is based on Tree Structures (bidirectional tree-structures LSTM-RNNs) and is mainly used for relations extraction. The latter is stacked on the former, the former's output and the hidden layer are part of the latter's input. Zheng [10] uses the joint model to transform entity relation extraction tasks into tagging scheme tasks, and the end-to-end model is mainly used to extract entities and relation directly.

However, the existing joint model is feature-based structured system, which depends on complex features and NLP toolkits. In order to reduce manual processing errors, recently, an end-to-end neural network model is generally adopted for the entities and relations extraction. This model has been applied to various tagging scheme tasks like Named Entity Recognition (NER), CCG Supertagging [4] and Chunking [5]. The most commonly used neural network model is to use BiLSTM structure to obtain sentence information to complete these tasks.

The scope of entity relation extraction in Chinese and English is very wide, and the methods and models are also very advanced. In contrast, the technology of Tibetan information extraction is backward, and the pipeline method is

usually used for entities and relations extraction: Tibetan NER and Tibetan RC.

For the Tibetan NER, Jin Ming [11] proposes a research scheme based on rules and statistics model firstly. Luo Zhiyong [12] proposes to use the Tibetan person's name character-level features and naming rules, word frequency and word frequency comparison strategies in conjunction with the dictionary. Hua que cai rang [13] proposes a syllable-based Tibetan NER scheme, using a syllable training model to identify Tibetan person's name, places name, and organization name accurately. Liu Feifei *et al.* [14] propose a method for identifying Tibetan person's name based on hierarchical features. The internal and contextual features of a person's name are used as CRF features, then the characteristics of juxtaposition relation of the person's name are designed as rules to further improve the recognition effect.

For the classification of Tibetan relations, Wang Like [15] propose an improved distant supervised relation extraction model in Tibetan based on Piecewise Convolutional Neural Network. Ma Ning and Li Yachao [16] propose the template method that Tibetan texts are captured from the internet, the texts are word-segmented, part-of-speech is marked, NER, keywords and entities are filtered for extracting candidate templates, finally, the semantic similarity is calculated for the extracted candidate templates. If the threshold exceeds a certain value, it becomes a relation template.

In this paper, we focus on the joint model using the end-to-end model, extracting two (or more) entities and their relation from a sentence to form a triple. And we turn entities and relations extract task into a tagging scheme task. Firstly, we split the Tibetan sentence into a word or a character. And then, add the label group (BIESO) to each word or each character. In order to improve the accuracy of the results, we also assign part-of-speech (POS) tagging for each word or character. Finally, we use the end-to-end model and the Bi-LSTM framework for pre-training. In this way, we can build a simple joint model only through neural networks without the need for complex feature engineering.

2. Method

We propose an end-to-end model with part-of-speech to jointly extract entities and their relations. In this section, we firstly introduce the word-level and character-level processing. Then we introduce the work for word-pos framework. Finally, we detail the model we used to extract results. The overall framework is shown in **Figure 1**.

In **Figure 1**, the length of the input sentence is l , x_i represents embedding of each word or character, p_i represents the vector of part-of-speech, the final output y_i represents the values of tagging scheme.

2.1. Tagging Scheme

We process the tagging scheme by word-level and character-level respectively. Each word or character of the sentence is assigned a tag to extract the results.

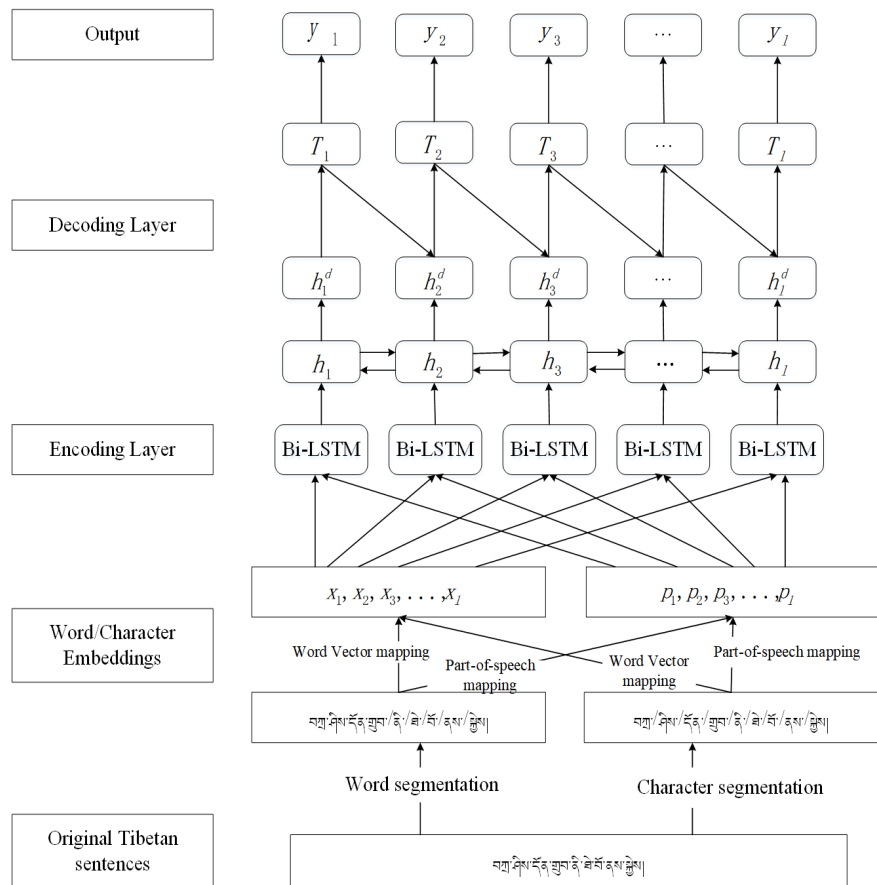


Figure 1. Framework.

The method of tag assignment is that the label “O” means that the word independent with the mentioned entity. In addition to “O”, other word tags are divided into three parts: entity position, relation type, and relation role. “BIES” (Begin, Inside, End, Single) is used to express the entity position. Relation type is looked from the known relation set. Relation roles are determined according to context information, “1” means the entity belongs to the first entity in the triple, while “2” belongs to the second entity.

For Tibetan word-level tagging scheme, the input sentence is “བཀྲ་ཤིས་དོན་གྲུབ་ནི་ཐེ་བོ་ནས་སྐྱེས།” (English interpretation: Tashidon was born in Diebu village), we use CRF tool to segment the Tibetan to “བཀྲ་ཤིས་དོན་གྲུབ་ནི་ཐེ་བོ་ནས་སྐྱེས།”, then tags are assigned to the word. The entity “བཀྲ་ཤིས་དོན་གྲུབ་” is related to the relation “སྐྱེས།” (English interpretation: Birthplace), so its tag is “S-1-BP”, The word “ཐེ” is the first word of entity “ཐེ་བོ་”, and is corresponding to “སྐྱེས།”, so its tag is “B-2-BP”.

For the Tibetan character-level tagging scheme, the input sentence is also “བཀྲ་ཤིས་དོན་གྲུབ་ནི་ཐེ་བོ་ནས་སྐྱེས།” (English interpretation: Tashidon was born in Diebu village), according to the Tibetan spelling characteristics, Tibetan syllable node is used to make a character-level process “བཀྲ་ཤིས་དོན་གྲུབ་ནི་ཐེ་བོ་ནས་སྐྱེས།”. Then tags are assigned to the syllables after character-segmentation. For example: The word “བཀྲ་” is labeled as “B-1-BP”, corresponding to the word-level by tagging scheme.

2.2. Pos Tagging

Since there is less information in Tibetan after the tagging scheme, the default label is “O” on words or syllables that are not related to the entity, and there is a large deviation in the extraction of the results, so we address this situation by part-of-speech of Tibetan words or characters after tagging scheme to reduce the error rate of extraction.

We choose the tool CRF for POS tagging [17]. We filter several labels like {N, U, K, V} and create the features templates. Then, using the below formula to compute the POS tagging probability:

$$P(y_i, X, i) = \exp\left(\sum_j \lambda_j t_j(y_{i-1}, y_i, i) + \sum_k \mu_k s_k(y_i, X, i)\right) \quad (1)$$

The expression $t()$ represents the transfer function, $s()$ represents the state function, j is the template number and k is the label number. We definition of POS about word-level (Tashi was born in Diebu Village) and character-level (Zwingran) shown in **Table 1**.

It is not difficult to find that many Tibetan-specific parts-of-speech, such as case-auxiliary words, verb, are helpful to judge the relation between two entities and improve the accuracy of extracting Tibetan entities.

2.3. End-to-End Model

In recent years, the end-to-end model based on neural network plays a good role in the sequence tagging task. In this paper, we adopted the end-to-end model to produce the tags sequence. The model mainly includes Word-POS vector, LSTM encoding layer, LSTM decoding layer and a softmax layer.

2.3.1. Word-POS Vector

Given a Tibetan sentence that the length is l , each word or character is represented as $W = \{x_1, x_2, x_3, \dots, x_l\}$ and word vector or character vector is represented as $T = \{t_1, t_2, t_3, \dots, t_l\}$, we use word2vec to train them firstly, then the part-of-speech of each word or character which is represented as $P = \{p_1, p_2, p_3, \dots, p_l\}$ is obtained by the CRF, finally, the two vectors are combined to form a new vector like $TP = \{(t_1, p_1), (t_2, p_2), (t_3, p_3), \dots, (t_l, p_l)\}$ by Word-POS [18].

For the Word-POS vector, we use the Skip-Gram method to train. The input vector represents the position corresponding to the Word-POS vector for being predicted, we set the dimensions of input vector as 100 and the context windows

Table 1. Definition of POS.

Word-level	བླ་མ་གྱིས་	ལྷོ་	མེ་	མེ་	ནས་	ལྷོ་
POS	N	U	N	N	K	V
Word-level	ཚོ་དབང་ལྷ་མོ་/N					
Character-level	ཚོ/N	དབང/N	ལྷ/N	མོ/N		

are 10. The Hidden Layer number is 300 and we use the softmax for Output Layer to classifier. The output represents probability that the word at a randomly chosen.

2.3.2. Encoding Layer

The vector representation TP generated by the preprocessing stage is transferred to the BiLSTM. BiLSTM can capture semantic information in sentences. It mainly includes the forward LSTM layer, the backward LSTM layer and a connection layer. The structure includes a series of cyclic link units, which is called memory blocks. Each current memory block can capture the current hidden vector h_t based on the hidden vector h_{t-1} of the previous layer, the unit vector c_{t-1} of the previous layer and the current input vector TP_{t-1} . The specific formula is defined as follows:

The input door:

$$i_t = \sigma(W_{wi}tp_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (2)$$

The forget door:

$$f_t = \sigma(W_{wf}tp_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (3)$$

The output door:

$$z_t = \tanh(W_{wc}tp_t + W_{hc}h_{t-1} + b_c) \quad (4)$$

$$c_t = f_t c_{t-1} + i_t z_t \quad (5)$$

$$o_t = \sigma(W_{wo}tp_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (6)$$

And the final feature output vector:

$$h_t = o_t \tanh(c_t) \quad (7)$$

In the Equations (1)-(6), b is a bias term, c is a memory unit, and $W_{()}$ is a randomly parameters. For each vector tp , the forward LSTM layer is encoded with context information and is set to \overline{h}_t . In the same way, the backward LSTM layer also encodes the vector tp and is set to \overleftarrow{h}_t . Finally, according the connection layer, the joint output expression vector \overline{h}_t and \overleftarrow{h}_t are denoted as $h_t = [\overline{h}_t, \overleftarrow{h}_t]$.

2.3.3. Decoding Layer

The LSTM structure is used to predict the sequence tagging. For a given vector tp , the input of the decoding layer is: the output h_t of the BiLSTM layer, the output p_{t-1} of the former prediction, the unit value c_{t-1} of the former, and the output value h_{t-1}^d of the former hidden layer. The specific formula is defined as follows:

The input door:

$$i_t^d = \sigma(W_{wi}^d h_t + W_{hi}^d h_{t-1} + W_{pi} p_{t-1} + b_i^d) \quad (8)$$

The forget door:

$$f_t^d = \sigma(W_{wf}^d h_t + W_{hf}^d h_{t-1}^d + W_{pf} p_{t-1} + b_f^d) \quad (9)$$

The output door:

$$z_t^d = \tanh(W_{wc}^d h_t + W_{hc}^d h_{t-1}^d + W_{tc}^d P_{t-1} + b_c^d) \quad (10)$$

$$c_t^d = f_t^d c_{t-1}^d + i_t^d z_t^d \quad (11)$$

$$o_t^d = \sigma(W_{wo}^d h_t + W_{ho}^d h_{t-1}^d + W_{co}^d c_t^d + b_o^d) \quad (12)$$

2.3.4. Softmax Layer

For the final softmax layer, based on the output vector P_t , the probability labels of the entities are predicted:

$$y_t = W_y P_t + b_y \quad (13)$$

$$p_t^i = \frac{\exp(y_t^i)}{\sum_{j=1}^{N_t} \exp(y_t^j)} \quad (14)$$

W_y is the input softmax matrix, N_t is the number of entire tags, b_y is the offset term.

3. Dataset and Evaluation Methods

3.1. Dataset and Parameters

It uses the Tibetan datasets processed by the Natural Language Processing Laboratory of Minzu University of China. The data format is the same as the NYT dataset [19]. The Tibetan dataset contains a total of 2400 triples and their original sentences. There are 11 relations in the relation set. In the experiment, we used 2000 train data and 400 test data.

We use the word2vec tool to generate word vectors and heuristic experiment whose dimensions are set to 50 dimensions. The number of LSTM encoding layer units is set to 300, the number of LSTM decoding layer units is set to 600, We regularize our network using dropout on embedding layer and the dropout ratio is 0.5, the length of sequence is set 200, the size of batch is set 64, and the learning rate is set to 0.002 from the grid search experiment.

3.2. Evaluation

The Precision, Recall, and F1-Score are mainly used as evaluation indicators. Unlike traditional machine learning methods, we don't use tag types to train the model. Therefore, entities type need not be considered during the evaluation. At the same time, we will randomly select 10% of the data in the train data as a validation data to optimize the parameters of our model

$$\text{Precision} = \frac{TP}{TP + FP} \quad (15)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (16)$$

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (17)$$

The FN represents the False Negative, FP represents the False Positive, TN represents the True Negative, TP represents the True Positive.

4. Experimental Results

4.1. Method Comparison

We compare the results of various algorithms in extracting Tibetan entity relations, including the traditional SVM [20] and LR [21] methods. We also compare the results of a single GRU method on tasks and our method yields the best results shown **Table 2**.

From **Table 2**, we can see that for the effect of Tibetan segmentation and part-of-speech tagging, the accuracy of our method is higher than the traditional machine learning method. At the same time, in the method of the neural network, a comprehensive comparison of different processing on Tibetan entities and relations extraction tasks is performed, especially, we use different granularity to process Tibetan and to divide Tibetan according to word-level and character-level, and add part-of-speech tagging and optimize it in the neural network learning. Our method has a higher improvement than single neural network model.

4.2. Comparison of POS

In addition, we compare the effect of part-of-speech of each word on the extraction of entity relation. we only make a comparison of part-of-speech tagging based on Tibetan character-level processing, we select NG (noun), P (lattice), part-of-speech V (verb), and A (adjective) as feature variables to input and choose the accuracy as the results evaluation, shown in **Table 3**.

Table 2. Experimental results of different models.

Method	Accuracy	recall	F1
SVM	0.34	0.24	0.28
LR	0.27	0.19	0.22
LSTM/GRU	0.60	0.32	0.41
LSTM + CRF	0.67	0.41	0.50
BiLSTM + word-level	0.47	0.27	0.34
BiLSTM + character-level	0.53	0.37	0.43
BiLSTM + word-level + POS	0.63	0.36	0.45
BiLSTM + character-level + POS	0.72	0.40	0.56

Table 3. Comparison of POS on character-level processing.

POS	Accuracy
+All	0.72
+NG	0.69
+P	0.59
+V	0.54
+A	0.56

As you can see in **Table 3**, it is not difficult to find that the influence of NG is relatively large. After analysis, we find that the percentage of NG in Tibetan is the largest, which is about 85%. At the same time, in the absence of NG, the accuracy of the final extraction has dropped by at least 10%. What can be seen is that the NG is very important for the extraction of Tibetan entities. While V accounts for the smallest proportion of all parts of speech, and it is about 2%. We also found that the bias of the influence of P and A on the results is very close, and the case-auxiliary and adjectives in Tibetan can help improve the accuracy of the extraction of Tibetan entities

Due to the scarcity of Tibetan, the processing needs to be corrected by a professional. It uses a machine to make a program processing in the all above-mentioned segmentation processes and manual correction. The processing is longer, and the results are also needed to be carried out by a professional. Neural network parameters are to be helped to optimize.

After that, we find that the following deficiencies existed in the experiment: 1) In the process of handling Tibetan words or characters, the length of the Tibetan was too long, and the corresponding entities and relations were often found after hundreds of lines; 2) The meaning of the expression in the Tibetan sentences is conflicting. An entity in the Tibetan language often expresses multiple meanings. That is, in the Tibetan sentence, in addition to the marked entity, the same meaning is expressed in other words or characters. It causes the misjudgment for the neural network model; 3) In the method of this paper, two entities in the same sentence often appear in other sentences, but the relational expression is inconsistent, resulting in a disorder in the relation and an increase in the error rate.

5. Conclusions

It proposes transformation of entities and relations extraction tasks into a tagging scheme task for the Tibetan. Our experiments achieved the highest accuracy compared with the traditional machine learning and single neural network models. However, our method also has some shortcomings in the Tibetan processing. At the same time, no comparison experiments have been conducted for the optimization of neural networks. We have not conducted in-depth research on the specific grammar rules and nature of Tibetan script.

In the future work, we will gradually optimize the processing of Tibetan, minimize human participation and optimize the model by adding Tibetan rules, as well as, provide a basis for the in-depth study of Tibetan natural language processing.

Acknowledgements

This work is supported by National Nature Science Foundation (No. 61972436).

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Vaswani, A., Bisk, Y., Sagae, K. and Musa, R. (2016) Supertagging with Lstms. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 232-237.
<https://doi.org/10.18653/v1/N16-1027>
- [2] Li, Q. and Ji, H. (2014) Incremental Joint Extraction of Entity Mentions and Relations. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, Volume 1: Long Papers, 402-412.
<https://doi.org/10.3115/v1/P14-1038>
- [3] Yu, X. and Lam, W. (2010) Jointly Identifying Entities and Extracting Relations in Encyclopedia Text via a Graphical Model Approach. *Coling 2010: Posters*, 1399-1407.
- [4] Zhai, F., Potdar, S., Xiang, B. and Zhou, B. (2017) Neural Models for Sequence Chunking. *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.
- [5] Fen, X. (2007) SVM and TSVM Based Chinese Entity Relation Extraction. D. Dissertation, School of National University of Defense Technology, Changsha.
- [6] Ren, X., Wu, Z., He, W., Qu, M., Voss, C.R., Ji, H. and Han, J. (2017) Cotype: Joint Extraction of Typed Entities and Relations with Knowledge Bases. *Proceedings of the 26th International Conference on World Wide Web*, 1015-1024.
<https://doi.org/10.1145/3038912.3052708>
- [7] Yang, B. and Cardie, C. (2013) Joint Inference for Fine-Grained Opinion Extraction. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Vol. 1: Long Papers, 1640-1649.
- [8] Singh, S., Riedel, S., Martin, B., Zheng, J. and McCallum, A. (2013) Joint Inference of Entities, Relations, and Coreference. *Proceedings of the 2013 workshop on Automated Knowledge Base Construction*, 1-6.
<https://doi.org/10.1145/2509558.2509559>
- [9] Miwa, M. and Bansal, M. (2016) End-to-End Relation Extraction Using Lstms on Sequences and Tree Structures. arXiv preprint arXiv:1601.00770
<https://doi.org/10.18653/v1/P16-1105>
- [10] Zheng, S., Wang, F., Bao, H., Hao, Y., Zhou, P. and Xu, B. (2017) Joint Extraction of Entities and Relations Based on a Novel Tagging Scheme. arXiv preprint arXiv:1706.05075 <https://doi.org/10.18653/v1/P17-1113>
- [11] Jin, M., Yang, H. and Shan, G.R. (2010) Research on Tibetan Named Entity Recognition. *Journal of Northwest University Nationalities: Natural Science*, 49-52.
- [12] Luo, Z.S., Song, R. and Zhu, X.J. (2009) Study on the Translation of Tibetan Names in Chinese. *Journal of the China Society for Scientific and Technical Information*, 475-480.
- [13] Hua, G.C., Jiang, W.B., Zhao, H.X. and Liu, Q. (2014) Title-Based Named Entity Recognition Based on Perceptron Model. *Computer Engineering and Applications*, **50**, 172-176.
- [14] Liu, F.F. and Wang, Z.J. (2009) Research on Tibetan Name Recognition Based on Hierarchical Features. *Journal of Computer Applications*, 1-7.
- [15] Wang, L.K., Sun, Y. and Xia, T.C. (2020) Distant Supervision for Tibetan Entity Relation Extraction. *Journal of Chinese Information Processing*, **34**, 74-76.
- [16] Ma, N., Li, Y.C., Yu, Y. and Gao, Y.J. Research on Template Acquisition Technology of Tibetan Entity Relations Facing the Internet. *Journal of the Central University for Nationalities (Natural Sciences)*, **24**, 35-39.

- [17] Lafferty, J., McCallum, A. and Pereira, F.C. (2001) Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data.
- [18] Zheng, S., Hao, Y., Lu, D., Bao, H., Xu, J., Hao, H. and Xu, B. (2017) Joint Entity and Relation Extraction Based on a Hybrid Neural Network. *Neurocomputing*, **257**, 59-66. <https://doi.org/10.1016/j.neucom.2016.12.075>
- [19] Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H. and Xu, B. (2016) Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Vol. 2: Short Papers, 207-212. <https://doi.org/10.18653/v1/P16-2034>
- [20] Liu, F.C., Zhong, Z.N., Lei, L. and Wu, H. (2013) Entity Relationship Extraction Method Based on Machine Learning. *Ordnance Industry Automation*, **32**, 57-62.
- [21] Lin, Y., Shen, S., Liu, Z., Luan, H. and Sun, M. (2016) Neural Relation Extraction with Selective Attention over Instances. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Vol. 1: Long Papers, 2124-2133. <https://doi.org/10.18653/v1/P16-1200>