Received March 22, 2019, accepted April 9, 2019, date of publication April 17, 2019, date of current version May 1, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2911320

A Hybrid Network Model for Tibetan Question Answering

YUAN SUN AND TIANCI XIA
School of Information Engineering, Minzu University of China, Beijing 100081, China
Minority Language Branch, National Language Resource and Monitoring Research Center, Minzu University of China, Beijing 100081, China

Corresponding author: Yuan Sun (tracy.yuan.sun@gmail.com)

This work was supported in part by the National Nature Science Foundation of China under Grant 61501529 and Grant 61331013, and in part by the National Language Committee Project of China under Grant ZDI125-36.

ABSTRACT Currently, research on question answering (QA) with deep learning methods is a hotspot in natural language processing. In addition, most of the research mainly focused on English or Chinese since there are large-scale open corpora, such as WikiQA or DoubanQA. However, how to use deep learning methods to QA of the low resource languages, like Tibetan becomes a challenge. In this paper, we propose a hybrid network model for the Tibetan QA, which combines the convolutional neural network and long short memory network (LSTM) to extract effective features from small-scale corpora. Meanwhile, since the strong grammar rules of Tibetan, we use the language model to decode the output of the LSTM layer which makes the answer more accurate and smoother. In addition, we add the batch normalization to accelerate deep network training and prevent overfitting. Finally, the experiments show that the ACC@1 value of the proposed model in Tibetan QA is 126.2% higher than the baseline model.

INDEX TERMS Tibetan question answering, hybrid network, convolutional neural network, long short memory network, language model.

I. INTRODUCTION

Question answering (QA) is concerned with building systems that automatically answer questions posed by humans in a natural language [1]. From keyword-based retrievable QA to community-oriented QA (e.g. Google QA system, Yahoo answer, Baidu knows, etc.), a variety of frameworks for QA are proposed. Currently, the question answering over knowledge base (KB) is proposed [2], [3].

Most of these systems are currently based on the end-to-end network model [4]–[10], which includes the encoding layer and decoding layer. The two layers usually use deep learning methods. In the end-to-end network model, there are two key points.

(1) Sequential sentences are processed using the recurrent neural network (RNN) model. For example, if you want to predict the word ''play'' in the sentence ''I want to play tennis'', you usually need to use the previous words ''I'', ''want'', ''to'', because these words in this sentence are relevant. However, the traditional neural network cannot solve this problem because the nodes are unconnected. RNN can remember the previous information and apply it to the calculation of the current output since the nodes between hidden layers are connected.

(2) Model structures can be combined freely. The model combination is a method to achieve balance between the error and overfitting. When the training data is limited, it is easy to cause overfitting. If the predicted results of different models are averaged, the risk of overfitting can be reduced.

The end-to-end network model has been successful used in English and Chinese, since there are large-scale open corpora, such as Natural Questions (English) [11], SimpleQuestions (English) [12], WikiQA (English) [13], SQuAD (English) [14], TREC QA (English) [15], TriviaQA (English) [16], WebQA (Chinese) [17] InsuranceQA (Chinese) [18], DoubanQA (Chinese) [19]. However, Tibetan QA systems and corpora are few. So how to use end-to-end network model in Tibetan QA and extract the effective features from small-scale corpora is a key problem.

Tibetan is an alphabetical language, established in the 7th century. Comparing with English and Chinese, Tibetan has strong grammar rules. For example, Chinese has three

The associate editor coordinating the review of this manuscript and approving it for publication was Khalid Aamir.

TABLE 1. Comparing examples of grammar rules in chinese and tibetan.

Chinese	Tibetan
帐篷上的绳子 (Ropes are on the tent.)	གུར་གྱི་རྩྭ་ཐག (Ropes belong to the tent.)
帐篷上有绳子 (There are ropes on the tent.)	གུར་གྱི་རྩྭ་ཐག (Ropes belong to the tent.)
绳子在帐篷上 (Ropes are on the tent.)	གུར་གྱི་རྩྭ་ཐག (Ropes belong to the tent.)

different expressions for the same meaning sentence, while Tibetan has only one way of expression, shown in TABLE 1. How to use the characteristics of Tibetan to the QA model is another key problem.

Based on the above description, the contributions of this paper are as follows:

(1) Since the scarcity of Tibetan corpora, we use a hybrid network which includes CNN and LSTM to extract the effective features from small-scale corpora.
(2) Since the grammar rules of Tibetan are strong, we use the language model (LM) to decode the output of LSTM layer, which makes it more accurate and smoother.
(3) Due to the complex structure and low running speed of the neural network, we use batch normalization (BN) acceleration to improve the model speed on the premise of better results.

II. RELATED WORK

In recent years, many end-to-end network models have been proposed, such as BiDAF [4], R-net [5], DCN [6], ReasonNet [7], Document Reader [8], Interactive AoA Reader [9] and Reinforced Mnemonic Reader [10]. These models are all based on deep learning methods, such as RNN, CNN, LSTM, etc.

RNN has successful used in QA because it can easily deal with the text sequence problem. However, it is difficult to deal with the long-term dependency problem. To solve this problem, LSTM [20] is proposed. Meanwhile, attention mechanism breaks the restriction that the traditional encoder-decoder structure depends on fixed length vectors, and has a great promotion effect on sequential learning tasks [21]–[23]. Li *et al.* [24] proposed a question categorization method based on LSTM. Firstly, words of questions are transformed to vectors. Then, a novel LSTM with attention mechanism is used to capture the most important features in a question. Finally, features are fed into the classifier to predict the category of the question. Chen *et al.* [8] proposed a method based on LSTM which uses an external memory to store the knowledge. The memory is read and written on the fly with respect to the attention, and these attentive memories are combined for inference. Wang and Manning [25] tried to compare the question and answer sentence by the syntactical matching in parse trees. Rocktaschel *et al.* [26] utilized a two-way attention method based on LSTM which can read the question and related answer tokens for improving encoding.

Another problem is that RNN prevents parallel computing because tokens must be put into RNN sequentially. Recently, the efficient progress has been made in the application of CNN in natural language processing (NLP) [27], [28]. Yu *et al.* [19] proposed a CNN based network to answer selection of a given question, which uses distributed representations and learns to match questions with answers by considering the semantic encoding. Yih *et al.* [12] constructed models for single-relation QA with triples in knowledge base. Bordes *et al.* [13] used a type of siamese network for learning to map question and answer pairs into a joint space. Iyyer *et al.* [29] worked on the QA task that requires identifying an entity described by a series of sentences.

There are also some related works in the combination of LSTM and CNN. Tan *et al.* [30] built embedding representations of questions and answers based on the LSTM model. They extended the model in two directions. One is defining a more composite representation for questions and answers by combining CNN with the basic framework. The other is utilizing a simple but efficient attention mechanism to generate the answer representation according to the question context. Li [31] used the general deep learning model to solve the multi-choice QA task. And they used a two layers LSTM with attention which gets a significant result. Feng *et al.* [32] did not rely on any linguistic tools, and the model can be applied to different languages or domains which require to specify an answer candidate pool for each question in the development. Yih *et al.* [33] focused on improving the performance using models of lexical semantic resources and evaluated on the TREC-QA [15]. Santos *et al.* [34] proposed a new neural network architecture called BOW-CNN which combines a bag-of-words representation with a distributed vector representation created by the CNN model. Zhou *et al.* [35] proposed a way to build a concept thesaurus based on the semantic relations extracted from Wikipedia. These works usually used CNN to calculate the similarity between questions and answers, and used LSTM to feature extraction, so the dependence of the CNN-LSTM networks is not high. Different from previous works, our work mainly uses CNN-LSTM networks to extract features.

LM is an important component in NLP applications such as machine translation and speech recognition [36]. LM provides the probability of a word sequence in languages. In recent years, using LM in LSTM has made great progresses [37]–[39].

The research of the Tibetan QA is relatively few. Sun *et al.* [40] proposed a method for Tibetan QA based on KB, which includes the understanding of Tibetan questions, judging the types of questions, and retrieving the appropriate answers by similarity calculation. Chen [41] designed a Tibetan encyclopedia knowledge QA system, which contains three main modules: knowledge base management, question analysis module and answer extraction module. These works only used keywords extraction to analyze questions, did not understand the true meaning of the questions.

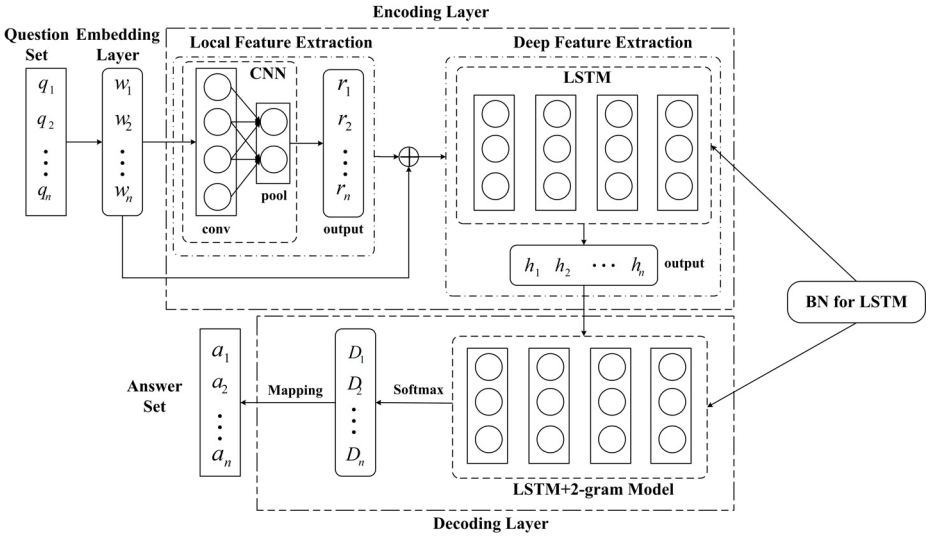

FIGURE 1. Architecture of Tibetan QA.

III. MODEL ARCHITECTURE

A. PROBLEM FORMULATION

The task considered in this paper is defined as follows. Given the question set with n questions $q = \{q_1, q_2, \cdots, q_n\}$ and the answer set with n answers $a = \{a_1, a_2, \cdots a_n\}$. We assign a label to each answer 0 or 1, respectively stands for wrong or right. And the ratio of right and wrong is 1:10. Wrong answers are selected randomly from the corpus. So, the corpus format can be defined as $(q, a, label)$, and this is the training data of our model. Using the training data, we can get the hybrid network model. For the input question, the model can output the answer which score is the highest. The architecture is shown in Fig 1.

B. FRAMEWORK

The model mainly includes three layers:

(1) Embedding layer
For the input question set $q = \{q_1, q_2, \cdots, q_n\}$, this layer vectorizes the words in the questions and gets the vector representation $w = \{w_1, w_2, \cdots, w_n\}$.

(2) Encoding layer
This layer includes two parts: local feature extraction using CNN and deep feature extraction using LSTM. The vector $r = \{r_1, r_2, \cdots, r_n\}$ is the CNN output through the convolution layer and the pooling layer. $w = \{w_1, w_2, \cdots, w_n\}$ and $r = \{r_1, r_2, \cdots, r_n\}$ are added as input to LSTM, we get the output vector $h = \{h_1, h_2, \cdots, h_n\}$. From these two parts, we get the effective features to represent questions.

(3) Decoding layer
This layer uses LSTM and 2-gram model to decode the vector $h = \{h_1, h_2, \cdots, h_n\}$, and gets the answer's score $D = \{D_1, D_2, \cdots, D_n\}$ through a softmax function. Choosing the highest score of answers, we get the answer $a = \{a_1, a_2, \cdots a_n\}$.

Also, in order to improve the speed of the model, we use the BN function for LSTM layer.

IV. MODEL DETAILS

A. EMBEDDING LAYER

We use the Word2Vec [42] tool to obtain the embedding representation of each word. The dimension of word embedding is set to 50. All the out-of-vocabulary words are mapped to an <UNK> token with random initialization.

B. ENCODING LAYER

In the encoding layer, we use the combination of CNN and LSTM to extract local features and deep features respectively. The numbers of the CNN layer and the LSTM layer are determined by the experiment, as shown in TABLE 3 of section V. We select two CNN layers and three LSTM layers as the structure of our model.

1) LOCAL FEATURE EXTRACTION

In the convolution layer, the input of current layer depends on the output of the previous layer. Each convolution layer has multiple input feature maps shown in Equation (1).

$$x_j^l = f\Big(\sum_{i \in M_j} x_i^{l-1} * k_{ij}^l + b_j^l\Big) \tag{1}$$

where x_j^l is the j^{th} feature map of layer l, and x_i^{l-1} is the i^{th} feature map of $l-1$ layer. k_{ij}^l is the i^{th} feature map associated with j^{th} feature map of layer l. M_j is the total input maps. The f is the ReLU activation function [43]. Each output of feature maps is given an additive bias b_j^l, that is the j^{th} bias of layer l. However, for a specific output feature map, the input feature map will be convolved with different kernels. That is to say, the input feature map i is associated with the output feature map j and k, then kernels applied to the input

map i are different from the output feature map j and k. So, in our model, selected feature maps and kernels are as relevant as possible to the current feature map.

In the pooling layer, the down sampling task of the input map is generated, shown in Equation (2). If there are N inputs maps, it will be exactly N output maps.

$$x_j^l = f(\beta_j^l task(x_j^{l-1}) + b_j^l) \tag{2}$$

x_j^{l-1} is the output of the convolution layer in Equation (1), and $task(x_j^{l-1})$ is the selected task function of the pooling layer. x_j^l is the output of the current pooling layer. Each output feature map is given the multiplicative bias β_j^l that represents the j^{th} multiplicative bias of layer l. b_j^l is the j^{th} additive bias of layer l. f is same as Equation (1). So, for the QA task, the final question expression can be transformed into the following Equations (3)-(4).

$$r_i = f(e \cdot w_{i:i+slid-1} + b_i) \tag{3}$$
$$r = (r_1, r_2, \cdots, r_n) \tag{4}$$

r_i is one of features from the output layer, and e is the filter of the CNN layer. b_i is the bias of feature i. $w_{i:i+slid-1}$ is the word embedding vector including the words from i to $i + slid - 1$, and $slid$ is the size of the selected window.

2) DEEP FEATURE EXTRACTION

After the CNN layer, we use LSTM to deep feature extraction. $w = \{w_1, w_2, \cdots, w_n\}$ and $r = \{r_1, r_2, \cdots, r_n\}$ are added as input to LSTM. Then, we get the output vector $h = \{h_1, h_2, \cdots, h_n\}$ by calculating the network unit activations using the Equations (5)-(10), iteratively from time $t = 1$ to n.

$$i_t = \sigma(W_{ix}[r_t, w_t] + W_{ic}c_{t-1} + b_i) \tag{5}$$
$$f_t = \sigma(W_{fx}[r_t, w_t] + W_{fc}c_{t-1} + b_f) \tag{6}$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_{ct}[r_t, w_t] + b_c) \tag{7}$$
$$o_t = \sigma(W_{ox}[r_t, w_t] + W_{oc}c_{t-1} + b_o) \tag{8}$$
$$h_t = o_t \odot \tanh(c_t) \tag{9}$$
$$h_t = \phi(W_o h_t + b_o) \tag{10}$$

where i_t, f_t , c_t and o_t are respectively the input gate, forget gate, cell state and output gate in time t. h_t is the output of LSTM. σ is the logistic sigmoid function. $W_{(\cdot)x}$ terms denote weight matrices (e.g. W_{ix} is the matrix of weights from the input gate.). $W_{(\cdot)c}$ terms denote diagonal weight matrices for full connections. $[r_t, w_t]$ is the adding value of the CNN output and the word vector in time t. $b_{(\cdot)}$ terms denote bias vectors (e.g. b_i is the bias vector of the input gate). $\odot$ is the element-wise product of the vectors and ϕ is the network output activation function.

C. DECODING LAYER

In the decoding layer, we use the LSTM and LM to get the answer. We use the n-gram model as the training function of LM because it can handle large scale unlabeled corpora. Through the experiment, we use the 2-gram model trained by the KenLM toolkit [18] on cleaned texts from the Tibetan corpus, shown in TABLE 4, described in section V.

$D(a)$ is the score value of answer a, shown in Equation (11). It is a linear combination of log probabilities from the LSTM and LM, along with a word insertion term.

$$D(a) = \log(p_{LSTM}(a|x)) + \alpha \log(p_{lm}(a)) + \beta wordcount(a) \tag{11}$$

where $p_{LSTM}(a|x)$ is the probability of the answer a from LSTM in the decoding layer when the sequence x has been predicted. $p_{lm}(a)$ is the decoding probability of the answer a using the language model. $wordcount(a)$ is the number of words contained in the answer a. α and β are weights. Finally, we use the beam search method to find the optimal answer.

D. BATCH NORMALIZATION FOR SPEEDING UP

We increase the depth of the network by adding more hidden layers, rather than making each layer larger. Previous work [44] has been proved this method is practical by increasing the number of successive two-way repeated layers. However, there are still problems in overfitting and running speed.

Therefore, we use BN to increase the speed of model and prevent overfitting in the process of training. BN can improve the convergence rate of recursive networks without reducing generalization performance [45]. And the $B(x)$ is shown in Equation (12).

$$B(x) = \gamma \frac{x - E[x]}{(Var[x] + \varepsilon)^{1/2}} + \beta \tag{12}$$

$E[x]$ and $Var[x]$ are the empirical mean and variance over a mini-batch. The bias b of the layer is dropped since its effect is cancelled by the mean removal. The learnable parameters γ and β allow the layer scale and shift each hidden unit as desired. The constant ε for numerical stability is small and positive.

In our model, we add the BN transformation to LSTM, and replace $\phi(Wh + b)$ with $\phi(B(Wh))$ in the Equation (10). So the output h_t is converted to the following, shown in Equation (13).

$$h_t = \phi(B(W_o h_t)) \tag{13}$$

V. EXPERIMENTS AND ANALYSIS

A. DATASET

We collect the Tibetan QA corpus from the website https://zhidao.yongzin.com, and get about 4,000 QA pairs through preprocessing. The data format is shown in TABLE 2. We split the corpus to training set (60%), validation set (20%) and test set (20%) because the Tibetan QA corpus is less. The average number of Tibetan words in question/answer is 15/40 respectively. The dataset contains 7,781 different words. Among them, the frequencies of 440 words are more than 100 times.

B. EVALUATION

We use the Mean Average Precision (MAP), which is a single-value metric that reflects the model's performance on all

TABLE 2. Examples in Tibetan QA.

Questions	Answers
ཇི་འདྲ་ཞིག་ལ་སྒེའུ་གར་ཞེས། (What is drama?)	ངག་ནས་གླུ་ཚིག་སྒེའུ་ལག་ལེན་བས་རོལ་འབྱེད་ལ་ཐུག་གཅིག་ཏུ་ཟིང་ལ་ཟེར། (The act of reciting a ballad or dancing.)
འགྲས་སྣོན་ཁྱད་ཆོད་སྣང་སེལ་འདྲ་བའི་ཚ་ནས་སྒེའུ་གར་རིགས་གང་དང་ཡོད་པ་ཡོད། (What are the different forms of drama?)	འགྲས་སྣོན་ཁྱད་ཆོད་སྣང་སེལ་བའི་ཚ་ནས་སྒེའུ་གར་ལ་ནང་གསེས་ལུ་གཏམ་བརྗོད་སྒེའུ་གར་དང་། རྩེའི་སྒེའུ་གར་ལུའི་སྒེའུ་གར་ཞུགས་པའི་སྒེའུ་གར་འབྲས་ཚན་གཅིག་མའི་སྒེའུ་གར། རོལ་མོའི་སྒེའུ་གར་ལ་སོགས་པོང་། (According to the different forms, drama can be divided into dancing, singing, monologue and musical.)

relevant answers retrieved by the model. When the MAP value is higher, the selected answer is better. The MAP is shown in Equation (14).

$$MAP = \frac{AP_i}{\sum_{i=0}^{n} AP_i} \tag{14}$$

where AP_i is the average accuracy of theme i shown in Equation (15). $P(j)$ is the fraction of the documents relevant to the users' needs, shown in Equation (16).

$$AP = \frac{\sum_{j=1}^{n_i} P(j) \cdot y_{i,j}}{\sum_{j=1}^{n_i} y_{i,j}} \tag{15}$$

$$P(j) = \frac{\sum_{k:\pi_i(k)\leq\pi_i(j)} y(i,k)}{\pi_i(j)} \tag{16}$$

$y_{i,j}$ refers to whether the j^{th} element in the document is relevant to the i^{th} element. If it is the relevant, $y_{i,j}$=1. Otherwise, $y_{i,j}$=0. $\pi_i(j)$ is the position of j. If the model does not return relevant answers, the accuracy is 0 by default.

Mean Reciprocal Rank (MRR) takes reciprocal of the ranking of the standard answers in the results as the accuracy, and outputs the average of all results. When the MRR value is higher, the selected answer is better. The MRR is shown in Equation (17).

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \tag{17}$$

$|Q|$ is the number of questions, $rank_i$ represents the i^{th} question.

Accuracy in top 1 (ACC@1) is mainly used to evaluate the overall performance of the model, shown in Equation (18). We rank the candidate answers and choose the highest score as the returned answer. If the returned answer is consistent with the standard answer, it is recorded as 1. Otherwise, it is 0.

$$ACC@1 = \frac{\sum_{i=0}^{n} \{value | \Pr e_{\max(D(a)_i)} == Sta_i\}}{n} \tag{18}$$

$\Pr e_{\max(D(a)_i)}$ is the highest score of candidate answers for question i. Sta_i is the standard answer for the question i. If $\Pr e_{\max(D(a)_i)}$ is equal to Sta_i, the *value* is 1. Otherwise, it is 0. n is the total number of questions.

TABLE 3. ACC@1 value and cost time of different layer setting.

1-1		2-1		2-3		3-3	
ACC@1	Time	ACC@1	Time	ACC@1	Time	ACC@1	Time
0.47	4.2h	0.35	4.7h	0.631	5.6h	0.637	10.1h

TABLE 4. Experimental results of N-gram.

N-gram Model	Numbers of Phrases Generated	Decoding Time	ACC@1
1-gram	53,005	5.4s	0.524
2-gram	529,208	6.1s	0.631
3-gram	1,688,624	12.2s	0.624
4-gram	2,622,891	11.1s	0.637

TABLE 5. Corpus size used to train LM and the size of phrases generated by LM in three languages.

Tibetan		Chinese		English	
Corpus	LM	Corpus	LM	Corpus	LM
11.23M	131.17M	11.53M	133.24M	12.43M	139.276M

TABLE 6. Main parameters setting.

Parameters	Values
EMB_DIM	50
HIDDEN_UNIT_NUM	64
SEQ_LENGTH	32
BATCH_SIZE	64
EPOCH	10
DROPOUT	0.75

We use MAP, MRR and ACC@1 to evaluate returned answers with standard answers.

C. EXPERIMENTAL RESULTS IN LAYER SETTING OF CNN AND LSTM

We use 4,000 Tibetan QA pairs to experiment on the layer setting of CNN and LSTM in the encoding layer, the results are shown in TABLE 3, where $n-m$ represents n CNN layers and m LSTM layers. TABLE 3 shows that the ACC@1 value is highest when setting 3-3, but the cost time is 10.1h. Although the ACC@1 value is 0.631 when setting 2-3, slightly lower than 3-3, the cost time is greatly reduced to 5.6h. So, we set two CNN layers and three LSTM layers, as mentioned in the part B of section IV.

D. EXPERIMENTAL RESULTS OF N-GRAM

We construct the Tibetan vocabulary which includes 7,781 words from 4,000 Tibetan QA pairs. TABLE 4 shows the results of using different n-gram models. The 2-gram model produces 529,208 phrases, and the ACC@1 value is 0.631, the decoding time in 32 characters is 6.1s. When using the 4-gram model, the ACC@1 value is the highest, but the number of phrases generated is 2,622,891 and the decoding time is 11.1s, which the time consumption is too

TABLE 7. Experimental results of different models in three languages.

Model	Tibetan			Chinese			English		
	MRR	MAP	ACC@1	MRR	MAP	ACC@1	MRR	MAP	ACC@1
LSTM	0.551	0.419	0.279	0.539	0.527	0.487	0.577	0.529	0.439
+CNN +Embedding	0.579	0.543	0.440	0.663	0.641	0.617	0.672	0.691	0.602
+CNN +Embedding +LM	0.744	0.741	**0.631**	0.711	0.671	**0.622**	0.754	0.722	**0.608**

high. Therefore, we select 2-gram model, as mentioned in the part C of section IV.

E. EXPERIMENTAL RESULTS OF MODELS

In order to prove the validity of our model, the size of the training corpus in Chinese (DoubanQA [19]) and English (SQuAD [14]) is consistent with the Tibetan corpus. The corpus size used to train LM, and the size of phrases generated by LM are shown in TABLE 5.

We experiment in three different languages and demonstrate the effectiveness of the model by adding different techniques. The parameters setting is shown in TABLE 6.

Also, we conduct the following three experiments, and results are shown in TABLE 7.

LSTM: we only use the LSTM model to the QA and use the one-hot representation [46] to assign each word a unique id. This is the baseline of our experiment.

LSTM + CNN + Embedding: based on the LSTM model, we use the embedding representation to each word, and add the CNN model, this is the second experiment.

LSTM + CNN + Embedding + LM: further, we add the LM, this is our model.

From TABLE 7, we can see that when adding the CNN and the word embedding to the LSTM, ACC@1 values increase by 57.7%, 26.7%, 55.4% in Tibetan, Chinese and English, respectively. It means that the word embedding has an effective impact on results. On this basis, we add the LM, ACC@1 values increase by 43.4%, 4.7%, 0.9% in Tibetan, Chinese and English, respectively. It means that the LM has greater influence on Tibetan than Chinese and English. Finally, ACC@1 values in Tibetan, Chinese and English are 0.631, 0.597, 0.608. Comparing with the baseline model, the proportion of improvement in Tibetan (126.2%) is higher than in Chinese (28.7%) and English (38.5%). So, our model achieves better results in Tibetan than in Chinese and English.

Meanwhile, we increase the corpus size to train LM in three languages, experimental results are shown in TABLE 8. ACC@1 values increase by 12.7%, 28.3%, 33.4% in Tibetan, Chinese and English, respectively. It means that the LM size can improve the model effectively. Since the data quality of English and Chinese is better than Tibetan, the promotion effect of Tibetan is not so obvious.

TABLE 8. Experimental results in the different corpus size for training LM.

Language	LM Corpus Size	MRR	MAP	ACC@1
Tibetan	11.23MB	0.744	0.741	0.631
	69.3MB	0.779	0.762	0.711
Chinese	11.53MB	0.711	0.671	0.622
	643.65MB	0.847	0.865	0.798
English	12.43MB	0.754	0.722	0.608
	40MB	0.892	0.834	0.811

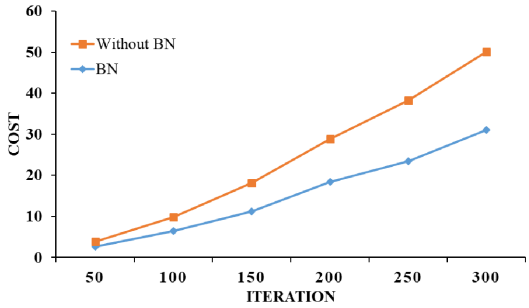

FIGURE 2. BN for speeding up.

Fig.2 shows the speed of the model. It is not difficult to find that with the number of iterations increasing, the optimization speed of BN is obviously faster than without BN. At about 200 iterations, the time is about 17.3h when adding the BN, which is much less than without BN (28.9h). With the increasing of iteration numbers, the time consumption is much higher without BN than using BN.

Finally, we show some QA examples of three languages using our model in the appendix. For each language, we choose three examples and show the top three answers in the order. The first column is questions, the second column is standard answers, the third column is the output candidate answers, and the last column is the scores of candidate answers, calculated by the Equation (11).

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose a hybrid network model to the Tibetan QA system. It not only proves the validity of the model in Tibetan, but also proves that the generalization ability of the model is good in Chinese and English. Meanwhile, we use the LM to the decoding layer which makes the answer

TABLE 9. English examples in SQuAD.

Questions	Standard Answers	Candidates Answers (The top three)	Score
What magazine rated Beyonce as the most powerful female musician in 2015?	Forbes.	Forbes.	0.97
		Stern.	0.02
		TIME.	0.01
What is the TV drama 24 designed on?	FBI Counterterrorism Division.	FBI Counterterrorism Division.	0.84
		Types of crime thrillers.	0.11
		Types of Action alarm.	0.04
In which papal document was the dogma of the Assumption defined?	Munificentissimus Deus.	Munificentissimus Deus.	0.74
		Fulgens Corona.	0.23
		Gratia Recordati.	0.01

TABLE 10. Chinese examples in DoubanQA.

Questions	Standard Answers	Candidates Answers (The top three)	Score
《超级勇者岛速升版》属于什么专题？ (What is the topic of *Super Brave Island Upgraded Version*?)	所属专题：勇士 (It belongs to Brave Subject.)	所属专题：勇士 (It belongs to Brave Subject.)	0.83
		超级勇者岛速升版 (Super Brave Island Upgraded Version.)	0.11
		挑战强者，成为勇者王。 (Challenge the strong and become the king of the brave.)	0.07
我很好奇乔万尼·薄伽丘的代表作是什么？ (I'm curious what is Giovanni Boccaccio's representative?)	代表作《十日谈》批判宗教守旧思想，主张“幸福在人间”，被视为文艺复兴的宣言。 (The representative *Decameron* criticizes religious conservatism and advocates "happiness in the world", which is regarded as a declaration of the Renaissance.)	代表作《十日谈》批判宗教守旧思想，主张“幸福在人间”，被视为文艺复兴的宣言。 (The representative *Decameron* criticizes religious conservatism and advocates "happiness in the world", which is regarded as a declaration of the Renaissance.)	0.74
		乔万尼·薄伽丘(1313—1375)，意大利文艺复兴运动的杰出代表，人文主义者。 (Giovanni Boccaccio (1313-1375), an outstanding representative of the Italian Renaissance movement, a humanist.)	0.17
		薄伽丘潜心研究古典文学，成为博学的人文主义者。 (Boccaccio became an erudite humanist by studying classical literature.)	0.04
百年战争中签订的《特鲁瓦条约》将法国怎么了？ (What happened to France when the *Treaty of Troyes* was signed in the Centennial war?)	这项条约实际上将法国分为由亨利五世、勃艮第公爵和法国王太子查理分别统辖的三个部分。 (The treaty divides France into three parts under the separate jurisdiction of Henry V, the Duke of Burgundy and Prince Charles of France.)	这项条约实际上将法国分为由亨利五世、勃艮第公爵和法国王太子查理分别统辖的三个部分。 (The treaty divides France into three parts under the separate jurisdiction of Henry V, the Duke of Burgundy and Prince Charles of France.)	0.75
		此后，法国人民抗英运动继续高涨，英军节节败退。 (Since then, the French people's resistance against the British movement continued to rise, British troops are losing.)	0.10
		1420 年，双方签订《特鲁瓦条约》，条约规定法国王太子的王位继承权转归英王亨利五世。 (In 1420, the two sides signed the *Treaty of Troyes*, which provided that the crown succession of the crown prince of France was transferred to king Henry V of England.)	0.09

more accurate and smoother. However, there are still many difficulties to be solved, such as the lack of Tibetan corpus, the processing of sentence length, and the difficulty of LM training.

In the future, we will continue to supplement Tibetan corpora. We are trying to expand question-answer pairs in other methods, such as knowledge base, Generative Adversarial Network, etc.

TABLE 11. Tibetan examples in the QA corpus.

Questions	Standard Answers	Candidates Answers (The top three)	Score
ཅི་ཞེས་ལ་སྦྱན་ཡིག་བརྩོད་པའི་རྒྱུན་ཟེར། (What is compositing writing?)	བརྩོད་པར་འདོད་པའི་དངོས་པོ་དེའི་ཡོན་ཏན་དང་ཐ་དད་པ་ཁོ་ནས་སྟན་ཡིག་ཆུ་ཚོགས་པའི་དངོས་པོ་གཞན་དང་མཉམ་དུ་བརྩོད་པ་ཞེས་ལ་སྦྱན་ཡིག་བརྩོད་པའི་རྒྱུན་ཟེར། (If you want to express the characteristics of something along with other things, it is called the assent.)	བརྩོད་པར་འདོད་པའི་དངོས་པོ་དེའི་ཡོན་ཏན་དང་ཐ་དད་པ་ཁོ་ནས་སྟན་ཡིག་ཆུ་ཚོགས་པའི་དངོས་པོ་གཞན་དང་མཉམ་དུ་བརྩོད་པ་ཞེས་ལ་སྦྱན་ཡིག་བརྩོད་པའི་རྒྱུན་ཟེར། (If you want to express the characteristics of something along with other things, it is called the assent.)	0.71
		མཁྱན་པ་སྤྲེན་བཞི་ནི་སྐང་ཚེན་དང་སྤྱིའུ་དེ་ཙོང་ཁུ་སྟེགས་པོ། (He is an elephant, monkey, rabbit and bird.)	0.02
		ཁང་ཚེན་བསོད་ནམས་རྒྱལ་པོ་ཡིན། (It's sonan jeb.)	0.01
ཆོས་རྒྱལ་མེས་དབོན་རྣམ་གསུམ་ནི་གང་དག་ཡིན། (Who are the three generation Kings in Tibetan?)	སྲོང་བཙན་སྒམ་པོ་ཁྲི་སྲོང་ལྡེ་བཙན།ཁངཊ་བདག་ཁྲི་རལ་བ་ཅན། (The three generation Kings of Tibetan are Songzan Ganb, Chisong Dezan and Chireba.)	སྲོང་བཙན་སྒམ་པོ་ཁྲི་སྲོང་ལྡེ་བཙན།ཁངཊ་བདག་ཁྲི་རལ་བ་ཅན། (The three generation Kings of Tibetan are Songzan Ganb, Chisong Dezan and Chireba.)	0.74
		ཨ་ཏིན་ཇོན་ཐྱི་མེས་པོ་ནི་དབྱིན་ཇི་མི་ཡིན། (Washington is the capital of Britain.)	0.04
		ཇོན་མི་རྒྱ་གར་དུ་ལོ་བདུན་ལ་བཞུགས། (Tommy spent seven years in India.)	0.01
འོག་གི་བཙན་པོ་ནི་ཞེས་པ་ནི་སྤྱ་རྒྱལ་ཐྱི་གདུང་རབས་ཉེར་བཞི་བ་ནས་ཉེར་བརྒྱད་པའི་བར་དུ་ཡོད། (Who are the five Zanpu from the 24th to the 28th generations?)	འོག་གི་བཙན་པོ་ནི་ཞེས་པ་ནི་སྤྱ་རྒྱལ་ཐྱི་གདུང་རབས་ཉེར་བཞི་བ་ནས་ཉེར་བརྒྱད་པའི་བར་དུ་ཡོད་བཙན་ནམ་སྤྱ་སྤས་ཇེ་ཁྲིས་སྤྱ་དཔྱང་བཙན་ཁྲི་ཙོག་རྩེ་ཡིན་བཙན་ནམ་སྤྱ་སྤས་ཇོ་ཙོ་རི་གཉན་བཙན་བཅས་སོ།། (The five Zanpu are Duoerlun Zanpu, Chi Zanpu, Chicha Zanpu, Chituo Zanpu and Tuotuori Zanpu.)	0.67
		སྐབ་ཞུའི་རྩོམ་རིག་ལ་དུས་ཀྱི་རང་བཞིན་དང་དོན་དངོས་ཀྱི་རང་བཞིན།སྤྱི་རང་བཞིན་བཅས་ཀྱི་ཡོད་པ་ནི། (The report literature has the characteristics of timeliness, authenticity and artistry.)	0.01
		རྩོམ་པ་པོས་གང་བརྩོད་པར་བྱ་བའི་དོན་གྱི་སྙིང་པོ་ལ་རྩོམ་གྱི་སྙིང་དོན་ཞེས་བརྩོད་དོ།། (The center of what the author wants to express is called the central idea.)	0.01

APPENDIX

See Tables 9–11.

REFERENCES

[1] P. Cimiano, C. Unger, and J. Mccrae, "Ontology-based interpretation of natural language," *Synth. Lect. Hum. Lang. Technol.*, vol. 7, no. 2, pp. 1–178, 2014.

[2] A. Bo, N. X. Han, and L. Sun, "Knowledge-representation-enhanced question-answering system," *Sci. Sinica Inf.*, vol. 48, no. 11, pp. 1521–1532, 2018.

[3] Z. Botong, C. Sun, L. Lin, and B. Liu, "LSTM based question answering for large scale knowledge base," *Acta Sci. Naturalium Univ. Pekinensis*, vol. 54, no. 2, pp. 286–292, 2018.

[4] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi. (2016). "Bidirectional attention flow for machine comprehension." [Online]. Available: https://arxiv.org/abs/1611.01603

[5] W. Wang, N. Yang, F. Wei, B. Chang, and M. Zhou, "Gated self-matching networks for reading comprehension and question answering," in *Proc. ACL*, Vancouver, BC, Canada, 2017, pp. 189–198.

[6] C. Xiong, V. Zhong, and R. Socher. (2016). "Dynamic coattention networks for question answering." [Online]. Available: https://arxiv.org/abs/1611.01604

[7] Y. Shen, P. S. Huang, J. Gao, and W. Chen, "ReasoNet: Learning to stop reading in machine comprehension," in *Proc. NIPS*, 2016, pp. 1047–1055.

[8] D. Chen, A. Fisch, J. Weston, and A. Bordes, "Reading wikipedia to answer open-domain questions," in *Proc. ACL*, Vancouver, BC, Canada, 2017, pp. 1–10.

[9] Y. Cui, Z. Chen, S. Wei, S. Wang, T. Liu, and G. Hu, "Attention-over-attention neural networks for reading comprehension," in *Proc. ACL*, Vancouver, BC, Canada, 2017, pp. 1–10.

[10] M. Hu, Y. Peng, Z. Huang, X. Qiu, F. Wei, and M. Zhou, "Reinforced mnemonic reader for machine reading comprehension," in *Proc. IJCAI*, Stockholm, Sweden, 2018, pp. 1–8.

[11] K. Tom *et al.* (2019). *Natural Questions: A Benchmark for Question Answer Research*. [Online]. Available: https://ai.google/research/pubs/pub47761

[12] W. Yih, X. He, and C. Meek, "Semantic parsing for single-relation question answering," in *Proc. ACL*, Baltimore, MD, USA, 2014, pp. 1–63.

[13] A. Bordes, S. Chopra, and J. Weston, "Question answering with subgraph embeddings," in *Proc. EMNLP*, Doha, Qatar, 2014, pp. 1–10.

[14] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," in *Proc. EMNLP*, Austin, TX, USA, 2016, pp. 1–10.

[15] E. M. Voorhees and D. M. Tice, "Building a question answering test collection," in *Proc. SIGIR*, Geneva, Switzerland, 2010, pp. 200–207.

[16] M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer, "TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension," in *Proc. ACL*, Vancouver, BC, Canada, 2017, pp. 1–11.

[17] P. Li *et al.* (2016). "Dataset and neural recurrent sequence labeling model for open-domain factoid question answering." [Online]. Available: https://arxiv.org/abs/1607.06275

[18] K. Heafield, I. Pouzyrevsky, J. H. Clark, and P. Koehn, "Scalable modified Kneser-Ney language model estimation," in *Proc. ACL*, Sofia, Bulgaria, 2013, pp. 690–696.

[19] L. Yu, K. M. Hermann, P. Blunsom, and S. Pulman. (2014). "Deep learning for answer sentence selection." [Online]. Available: https://arxiv.org/abs/1412.1632

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

[21] Y. Hao *et al.*, "An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge," in *Proc. ACL*, Vancouver, BC, Canada, 2017, pp. 221–231.

[22] W. Yin, M. Yu, B. Xiang, B. Zhou, and H. Schütze (2016). "Simple question answering by attentive convolutional neural work." [Online]. Available: https://arxiv.org/abs/1606.03391

[23] J. Yin, W. X. Zhao, and X.-M. Li, "Type-aware question answering over knowledge base with attention-based tree-structured neural networks," *J. Comput. Sci. Technol.*, vol. 32, no. 4, pp. 805–813, 2017.

[24] Z. Li, J. Huang, Z. Zhou, H. Zhang, S. Chang, and Z. Huang, "LSTM-based deep learning models for answer ranking," in *Proc. IEEE Conf. Data Sci. Cyberspace*, Changsha, China, Jun. 2016, pp. 90–97.

[25] M. Wang and C. D. Manning, "Probabilistic tree-edit models with structured latent variables for textual entailment and question answering," in *Proc. COLING*, Beijing, China, 2010, pp. 1164–1172.

[26] T. Rocktäschel, E. Grefenstette, K. M. Hermann, T. Kočiský, and P. Blunsom. (2015). "Reasoning about entailment with neural attention." [Online]. Available: https://arxiv.org/abs/1509.06664

[27] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. EMNLP*, Doha, Qatar, 2014, pp. 1–6.

[28] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proc. ACL*, Baltimore, MD, USA, 2014, pp. 1–11.

[29] M. Iyyer, J. L. Boyd-Graber, L. M. Claudino, R. Socher, and H. Daumé, III, "A neural network for factoid question answering over paragraphs," in *Proc. EMNLP*, Doha, Qatar, 2014, pp. 633–644.

[30] M. Tan, C. dos Santos, B. Xiang, and B. Zhou. (2015). "LSTM-based deep learning models for non-factoid answer selection." [Online]. Available: https://arxiv.org/abs/1511.04108

[31] Y. Li, "Two layers LSTM with attention for multi-choice question answering in exams," in *Proc. ICFMCE*, Abu Dhabi, United Arab Emirates, 2018, pp. 1–7.

[32] M. Feng, B. Xiang, M. R. Glass, L. Wang, and B. Zhou, "Applying deep learning to answer selection: A study and an open task," in *Proc. IEEE Workshop Autom. Speech. Recognit. Understanding*, Scottsdale, AZ, USA, Dec. 2015, pp. 813–820.

[33] W. Yih, M. Chang, C. Meek, and A. Pastusiak, "Question answering using enhanced lexical semantic models," in *Proc. ACL*, Sofia, Bulgaria, 2013, pp. 1744–1753.

[34] L. Santos, C. N. Barbosa, D. Bogdanova, and B. Zadrozny, "Learning hybrid representations to retrieve semantically equivalent questions," in *Proc. ACL*, Beijing, China, 2015, pp. 694–699.

[35] G. Zhou, Y. Liu, F. Liu, D. Zeng, and J. Zhao, "Improving question retrieval in community question answering using world knowledge," in *Proc. IJCAI*, Beijing, China, 2013, pp. 1–7.

[36] H. Schwenk, A. Rousseau, and M. Attik, "Large, pruned or continuous space language models on a GPU for statistical machine translation," in *Proc. NAACL-HLT*, Montreal, QC, Canada, 2012, pp. 11–19.

[37] Y. Bengio, R. Ducharme, and P. Vincent, "A neural probabilistic language model," in *Proc. NIPS*, Denver, CO, USA, 2000, pp. 1–7.

[38] S. Merity, N. S. Keskar, and R. Socher. (2017). "Regularizing and optimizing LSTM language models." [Online]. Available: https://arxiv.org/abs/1708.02182

[39] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling," in *Proc. INTERSPEECH*, Portland, OR, USA, 2012, pp. 194–197.

[40] H. Sun, H. Yu, and M. Su, "Tibetan question and answer system based on knowledge base," (in Chinese), *J. Northwest. Univ. Nat. (Nat. Sci.)*, vol. 36, no. 2, pp. 45–50, 2015.

[41] X. Y. Chen, "Design and research of Tibetan encyclopedia knowledge question answering system," (in Chinese), *Intell. Comput. Appl.*, vol. 7, no. 4, pp. 48–50, 2017.

[42] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. NIPS*, Lake Tahoe, Spain, 2013, pp. 3111–3119.

[43] T. Laurent and J. van Brecht, "The multilinear structure of ReLU networks," in *Proc. ICML*, Stockholm, Sweden, 2018, pp. 1–32.

[44] A. Graves, A. Mohamed, and G. E. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. ICASSP*, Vancouver, BC, Canada, May 2013, pp. 6645–6649.

[45] C. Laurent, G. Pereyra, P. Brakel, and Y. Bengio, "Batch normalized recurrent neural networks," in *Proc. ICASSP*, Shanghai, China, Mar. 2016, pp. 2657–2661.

[46] C. B. Ritesh, "Word representations for gender classification using deep learning," *Procedia Comput. Sci.*, vol. 132, pp. 614–622, Dec. 2018.

YUAN SUN received the M.S. and Ph.D. degrees from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2004 and 2007, respectively. She is currently an Associate Professor with the School of Information Engineering, Minzu University of China. She has more than 30 papers published in various journals and international conferences. Her current research interests include natural language processing and knowledge engineering.

TIANCI XIA received the B.E degree from Tianjin Polytechnic University, China, in 2015. He is currently pursuing the M.S. degree with the Minzu University of China. His current research interests include question answering and knowledge graph.