IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 28, 2019, accepted August 13, 2019, date of publication August 19, 2019, date of current version November 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2936002

Hierarchical Transfer Learning Architecture for Low-Resource Neural Machine Translation

GONGXU LUO, YATING YANG, YANG YUAN, ZHANHENG CHEN, AND AIZIMAITI AINIWAER

Xinjiang Laboratory of Minority Speech and Language Information Processing, The Xinjiang Technical Institute of Physics and Chemistry, Chinese Academy of Sciences, Urumqi 830011, China
University of Chinese Academy of Sciences, Beijing 100049, China

Corresponding author: Yating Yang (yangyt@ms.xjb.ac.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant U1703133, in part by the Western Light of the Chinese Academy of Sciences under Grant 2017-XBQNXZ-A-005, in part by the Subsidy of the Youth Innovation Promotion Association of the Chinese Academy of Sciences under Grant 2017472, in part by the Major Science and Technology Project of Xinjiang Uygur Autonomous Region under Grant 2016A03007-3, and in part by the Xinjiang Uygur Autonomous Region Level Talent Introduction Project under Grant Y839031201.

ABSTRACT Neural Machine Translation(NMT) has achieved notable results in high-resource languages, but still works poorly on low-resource languages. As times goes on, It is widely recognized that transfer learning methods are effective for low-resource language problems. However, existing transfer learning methods are typically based on the parent-child architecture, which does not adequately take advantages of helpful languages. In this paper, inspired by human transitive inference and learning ability, we handle this issue by proposing a new hierarchical transfer learning architecture for low-resource languages. In the architecture, the NMT model is trained in the unrelated high-resource language pair, the similar intermediate language pair and the low-resource language pair in turn. Correspondingly, the parameters are transferred and fine-tuned layer by layer for initialization. In this way, our hierarchical transfer learning architecture simultaneously combines the data volume advantages of high-resource languages and the syntactic similarity advantages of cognate languages. Specially, we utilize Byte Pair Encoding(BPE) and character-level embedding for data pre-processing, which effectively solve the problem of out of vocabulary(OOV). Experimental results on Uygur-Chinese and Turkish-English translation demonstrate the superiorities of the proposed architecture over the NMT model with parent-child architecture.

INDEX TERMS Hierarchical transfer learning, low-resource problem, neural machine translation.

I. INTRODUCTION

Language is the most important human communication tools and the main way of expression for people to communicate [1]. There are 6809 different languages in the world, most of them are resource-poor languages [2]. Language diversity leads to communication barriers. Therefore, how to communicate effectively has always been an urgent and challenging problem, which has drawn great attentions from both research and industry communities in recent years. Machine Translation is an effective way to provide a bridge between different languages, where the sequence-to-sequence neural machine translation(NMT) [3]–[5] has achieved remarkable progress on resource-rich language pairs in the past few years [6]–[8]. But because of the complexity of the network and large number of parameters, the NMT models are highly depended on the quality and the availability of extensive parallel corpora. For this reason, NMT models still perform poorly on most low-resource languages compared with the Statistical Machine Translation(SMT) [9], [10]. Therefore, data scarcity is a huge challenge for NMT [11].

In order to deal with the problem of data scarcity in NMT, there are many strategies for low-resource languages. Using monolingual data to enrich parallel data is a simple and intuitive way such as back-translation [12], data augment [13], the self-learning algorithm [14], the semi-supervised method [15], the joint EM optimization method [16] and the dual learning method [17]. In [18], the model-agnostic meta-learning algorithm(MAML) was applied to the low-resource NMT. M-NMT used the memory augmented structure to

The associate editor coordinating the review of this article and approving it for publication was Mohsin Jamil.

solve the out of vocabulary problem for translation of Uygur-Chinese [19]. TA-NMT that uses the unified bidirectional EM algorithm leveraged bilingual data to improve the translation performance of low-resource languages [20]. The teacher-student architecture based on the assumption that parallel language pairs have close probabilities of generating a sentence in the third language improved the translation performance for low-resource languages with the help of high-resource language pairs [21]. Despite the success of TA-NMT and teacher-student architecture in low-resource languages, we argue that they are not suitable enough if some low-resource languages have parallel corpora with only one high-resource language.

Transfer learning is an effective method to solve the low-resource problem. There are four basic methods of transfer learning as follows: sample-based transfer learning, model-based transfer learning, feature-based transfer learning and relationship-based transfer learning [22]. Due to the improvement of computer performance, the deep learning methods are applied to various fields and have achieved excellent achievements in recent years [23], [24]. In [25], Jason Yosinski et al took the lead in conducting research on the mobility of deep neural network. Following their work, some studies are mainly about the fine-tuning and domain adaption for different tasks [26]–[28]. Since the model-based transfer learning method is perfectly combined with deep neural network and improves the exiting network structure conveniently [29]–[31]. Therefore, the model-based transfer learning is widely explored in many fields [32]–[36].

In this paper, inspired by human transitive inference and learning ability in languages, we propose a new hierarchical transfer learning architecture to make full use of helpful languages by adding an intermediate layer for the low-resource languages especially that have only one parallel corpus. In training process, the three-layer architecture transfers and fine-tunes the parameters layer by layer. The training process mimics the process of a person learning new languages as shown in Figure 1. For example, if a person has mastered English and Turkish, so the person knows the method about how to learn a language and their respective syntactic structure. It is intuitive that it is easier for him to master Uygur compared with people without language foundation. By using the architecture, we combine data volume advantage of high-resource language and linguistic similarity advantage of intermediate language that provides useful syntactic knowledge.

We evaluate the hierarchical transfer learning architecture on Uygur-Chinese and Turkish-English. Experimental results show that: Our hierarchical transfer learning architecture improves 1.15 BLEU scores compared with the NMT systems based on the transformer-big model, improves 0.58 BLEU scores compared with the NMT system with parent-child architecture [10], [37], and excesses the strong Phrase-based statistical machine translation model [9] 1.95 BLEU scores on Uygur-Chinese. Similarly on Turkish-English, our method outperforms the parent-child architecture 0.73 BLEU scores, which verifies the generalization of the hierarchical transfer learning architecture.

FIGURE 1. A demo example of hierarchical transfer learning architecture in the source side.

In summary, our contributions are as follows:

1) We propose a new hierarchical transfer learning architecture to combine data volume advantage of high-resource languages with syntactic similarity advantage of similar languages by adding a intermediate layer.
2) Based on the hierarchical architecture, we make full use of helpful language resources for low-resource languages and are more flexible in language choice.
3) we verify the generalization of the hierarchical transfer learning architecture by experimenting it on different low-resource languages.
4) Experimental results show that our architecture significantly improves the translation performance compared with the parent-child architecture, the NMT system based on transformer-big model and the phrase-based SMT model on low-resource languages.

Section II presents related work about transfer learning method on NMT for low-resource language problem. Section III describes the details of our hierarchical transfer learning architecture and the methods that are used to tackle training data. The details of our experiments and the introduction of the three baselines are described in Section IV. Section V reports the results of the comparative experiments and analyzes the process of our experiments. Finally, the conclusion is drawn in section VI.

II. RELATED WORK

In order to improve translation performance of low-resource languages, transfer learning methods were applied to NMT in recent years. Its purpose is to initialize the parameters with the trained models instead of random initialization, which transfers helpful information from the trained models. The previous work that uses transfer learning methods [22], [38] and similar methods to solve the low-resource problem is outlined. In [39], Dong et al proposed a multi-task learning model that shares encoder across different translation tasks for one target language to improve the performance of low-resource languages. In [40], Firat et al proposed a multi-way, multilingual NMT to share the attention mechanism. It used

different encoders and decoders for multiple language pairs. Majority of exiting transfer learning methods are based on the parent-child architecture that is proposed by Zoph et al [10], which pioneer to apply the transfer learning method to NMT. They trained the parent model on a high-resource language pair and saved some parameters, then transferred the parameters to initialize the child model and to constrain training for child model with freezing and fine-tuning. This method got significant performance on low-resource languages and even exceeded the Phrase-based SMT system on Hausa-English. Following Zoph et al's[10] work, Nguyen et al used BPE to increase vocabulary overlap to optimize the word embedding and chose related low-resource language as parent model [41]. Experimental results showed that similar languages are helpful for low-resource languages. In [42], Dabre et al explored the influence of language relevance on transfer learning. They concluded that it is better to choose the language that is closer to the low-resource language. In contrast, in [37], Kocmi et al compared the effect of data volume and language similarity on the transfer learning method. They concluded that the data volume of high resource language is more important than the relatedness of language. Despite the success of previous work for low-resource languages, we argue that they did not make good use of these advantages, which had been proved are both helpful for low-resource languages. In this paper, we propose the hierarchical transfer learning architecture, which adds the intermediate layer, to combine the data volume advantage with the language similarity advantage. DDTF [43] and TTL [44] also proposed similar three-layer's architectures, which choose the data that is useful for target domain in Computer Vision with the help of intermediate domain. But the difference is that our method transfers the parameters trained on the high-resource layer and intermediate layer in the architecture instead of increasing helpful data based on similarity measure algorithm.

III. HIERARCHICAL TRANSFER LEARNING ARCHITECTURE

In this section, we present the hierarchical transfer learning architecture for NMT. To start with, we describe the training process of the architecture. Afterwards, we introduce the details of the NMT model(Transformer).

The hierarchical transfer learning architecture consists of three layers. The training strategy of the architecture is shown in algorithm 1, while its flow chart and model structure are illustrated in Figure 2. In the process of training, considering the training time and efficiency, we train our model on the high-resource language pair(English-Chinese) for several steps and transfer the parameters to the intermediate model in the first layer. In the second layer, the model is trained on the intermediate language pair(Turkish-English) that is similar with Uygur on syntax; and the parameters are fine-tuned until converging. Finally, we transfer the parameters that are trained on the intermediate language pair to initialize the model of the low-resource language and train the model on the low-resource language pair(Uygur-Chinese) until converging in the third layer. We do not modify the framework of the NMT model but transfer the parameters to initialize the next model instead of initializing randomly.

Algorithm 1 Hierarchical Transfer Learning Architecture

Input: Three different parallel corpus, D(X,Y),D(X_1,Y_1),D(X_2,Y_2).

1: compute argmax P($Y|X$,θ_1) X is the source language of high-resource language pair, Y is target language of high-resource language pair, θ_1 are the parameters of the model.

2: compute argmax P($Y_1|X_1$,θ_2) $X_1$is the source language of intermediate language, Y_1 is target language of intermediate language, θ_2 is the parameters fine-tuned on θ_1.

3: compute argmax P($Y_2|X_2$,θ_3) X_2 is the source language of low-resource language, Y_2 is target language of low-resource language, θ_3 is the parameters fine-tuned on θ_2.

Output: The model of hierarchical transfer learning architecture with the parameters θ_3.

Our model is the transformer, which works by relying on self-attention mechanism completely [7]. In the part of data pre-processing, the transformer discards Recurrent Neural Network(RNN) that considers the time series information, but adds the relative position information. The relative position information is calculated in equation (4). The input embeddings add the relative position embeddings to compose new sub-word embeddings. The encoder is composed of six identical layers that each layer consists of multi-head self-attention mechanism and feed-forward network. The decoder is also composed of six identical layers that each layer consists of masked multi-head self-attention mechanism, multi-head self-attention mechanism and feed-forward network. The attention mechanism is computed as:

$$Attention\,(A, B, C) = softmax\left(\frac{AB^T}{\sqrt{d_b}}\right)C \tag{1}$$

One sentence can be presented by a [maxlength,sub-word dimension] matrix. A, B, C are sub-work embedding matrix of query, key, value with dimension d_a, d_b, d_c respectively. AB^T calculates the weights of matrix C. The multi-head attention is computed as:

$$\begin{aligned} MultiHead(A, B, C) &= Concat(head_1, \cdots, head_i)W^D \\ head_i &= Attention(AW_i^A, BW_i^B, CW_i^c) \end{aligned} \tag{2}$$

where each $head_i$ has its transformation matrix $W_i^A \in \Re^{d_{input} \times d_a}$, $W_i^B \in \Re^{d_{input} \times d_b}$, $W_i^C \in \Re^{d_{input} \times d_c}$. The three matrices transform the input sub-word embedding matrix into different dimensions for the calculation of self-attention mechanism. The $W^D \in \Re^{hd_c \times d_{input}}$ transform the concatenate heads into the dimension of input. The feed-forward network of different layers have different parameters. The attention mechanism links the encoder and decoder. The feed-forward network is computed as:

$$FFN(a) = max\,(0, aW_1 + b_1)\,W_2 + b_2 \tag{3}$$

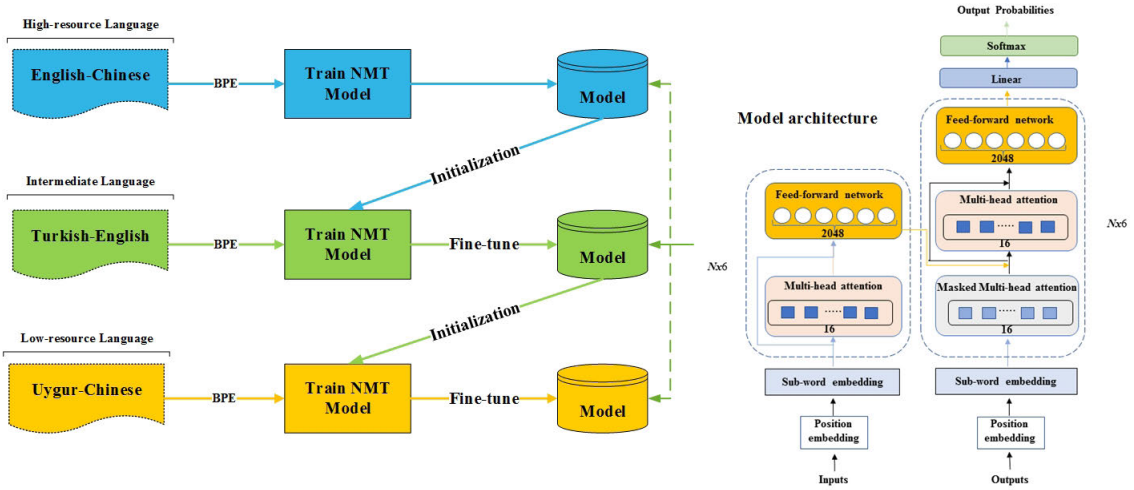


FIGURE 2. Hierarchical transfer learning architecture.

where a is the input embedding. $W_1 \in \Re^{d_{input} \times 2048}$, $b_1 \in \Re^{2048}$ is the parameters of the first linear transformation. $W_2 \in \Re^{2048 \times d_{input}}$, $b_2 \in \Re^{d_{input}}$ is the parameters of the second liner transformation. The position embedding is calculated as:

$$PE_{index,2j} = sin\left(index/10000^{2i/d_{input}}\right)$$
$$PE_{index,2j+1} = cos\left(index/10000^{2i/d_{input}}\right) \quad (4)$$

where the *index* shows the position of the word in the sentence. j represents the jth dimension of the input sub-word embedding vector. The *sine* and *cosine* functions can express each other through linear relationships.

IV. EXPERIMENT

A. EXPERIMENT SETTINGS

All the experiments about NMT systems, which are based on transformer, are implemented in tensor2tensor [45] version 1.11.0. Our GPU is NVIDIA Corporation GK210GL [Tesla k80] with 11GB RAM. For training the models, we set the hyper-parameters of the NMT model according to the training tips in [46], which explores the best performance for transformer. The hyper-parameters, which determine the structure of model, are as shown in table 1. Considering the data sparsity of low-resource languages, the dropout of 0.2 is applied to prevent over-fitting [47]. We use Adam [48] as the optimizer with learning rate constant of 2. In order to prevent diverged training, we set the learning-rate-warmup-steps of 16000. We also set the shared embedding and softmax weights is true. For decoding, the translation is generated by the 8 words with the highest probability of each position via setting beam search size is 8.

B. DATA SET AND PREPROCESSING

In order to testify our method is effective for the scenario that the low-resource language has parallel corpus only with one high-resource language such as Uygur-Chinese. We choose

TABLE 1. Hyper-parameters used to train transformer models, unless differently specified.

Model parameters	Transformer-base	Transformer-big
Embedding	512	1024
Encoder depth	6	6
Decoder depth	6	6
Head numbers	8	16
Feed-forward network	2048	4096
Batch size	2048	2048
Beam Size	8	8

TABLE 2. Data sets for the hierarchical transfer learning architecture.

Language Pair	Role	Train Size	Dev Size	Test Size
English-Chinese	High-resource language	15M	2K	2K
Turkish-English	Intermediate language	0.2M	3K	3K
Uygur-Chinese	Low-resource language	0.35M	1K	1K

the open Uygur-Chinese news corpus in CWMT as low-resource language. The dev set and test set are from the 2017 CWMT Uygur-Chinese evaluation campaign. Turkish-English parallel data that was published on WMT 2016 [49] is the intermediate language. The dev set and test set are the newstdev2016(Tr-En) and newstest2016(Tr-En). English-Chinese parallel corpus that is open on Union Corpus [50] is the high-resource language. The dev set and test set are form the newstest-2017 [51]. Table 2 shows the details of the used data.

In the process of data pre-processing, we applied word segmentation for English, Turkish and Uygur in each training condition by learning a sub-word vocabulary via BPE [52]. BPE breaks words into sub-words to solve the out-of-vocabulary(OOV) problem, which is used as the smallest unit

TABLE 3. Sub-word embedding and character-level embedding for data preprocessing on Chinese.

Data preprocessing	Model	Steps	BLEU Scores
BPE	Transformer(big)	0.3M	32.78
Character	Transformer(big)	0.3M	**34.25**

to present sentences instead of the whole word. Sub-word segmentation can harshly reduce the size of the vocabulary. By using this way, some OOV words can be presented by basic sub-word unit. Furthermore, the character-level embedding [53], which uses characters to present words, is applied to Chinese. Different from languages such as English, there are more characters in Chinese. According to the characteristics of Chinese, it is more suitable for Chinese in low-resource languages compared with BPE as we testified in table 3. Specially, Wolk et al divided polish text into the suffix prefix core and grammatical groups with POS tag for data sub-word division and augmentation, which is necessary to reduce the size of dictionary [54]. However, Uygur is a complex language, and there is not a public and accurate morphological segmentation tool. Therefore, we will study the morphological segmentation method for Uygur in future work. Then the shared vocabulary is created among all source languages and among all target languages respectively. The information of Sub-words can be shared by the overlap vocabulary.

C. VOCABULARY

Considering the fairness of the three language pairs, we respectively choose 0.2M parallel corpus of three language pairs to create shared vocabulary. The 0.2M high-resource parallel corpus is provided by selecting one of every five sentence on the full language pair. Because of the data scarcity of Turkish and Uygur, all Turkish-English parallel corpus and almost all Uygur-Chinese parallel corpus are used to get the 0.2M parallel corpus respectively. Then we mix the three parallel corpus to make the 32K shared vocabulary. The 36K shared vocabulary is also created by the mixed parallel corpus with 0.35M English-Chinese, 0.2M Turkish-English and 0.35M Uygur-Chinese parallel corpus to explore the impact of different vocabulary on results.

D. BASELINES

There are three baselines to be compared with our method. The first baseline is the Phrase-based SMT system that is based on Moses [9]. The second baseline is the NMT system that is based on the transformer [7], which is experimented with transformer-base model and transformer-big model respectively, where the big model doubles overall in model structure.

The third baseline is the NMT system with the transfer learning method, which follows the parent-child architecture proposed by Zoph et al [10], [37]. The parent model is trained on the high-resource language pairs (English-Chinese) and the child model is trained on the low-resource language pairs (Uygur-Chinese). In the process of data pre-processing, the 32k and 50K shared vocabulary are created respectively by mixing 0.35M English-Chinese parallel corpus and 0.35M Uygur-Chinese parallel corpus via word segmentation. In the training process, the parent model is trained several steps on English-Chinese, and transfers the parameters to the child model for initialization. For the transformer-base model, the parent model of the parent-child architecture trains 0.567M and 0.8M steps on the 32k and 50k shared vocabulary respectively. For the transformer-big model, the parent model trains 0.46M and 0.456M steps on the 32k and 50k shared vocabulary respectively. Finally, the child model is trained on Uygur-Chinese parallel corpus. The BLEU scores of the second baseline and the third baseline and experimental details are showed in Table 4. From Table 4, we can find that the performance of transformer-big model improves 1.14 BUEU scores compared to the transformer-base model. For the transformer-base model, the performance of the parent-child architecture with 32k shared vocabulary is 0.44 BLEU scores higher than that with 50k shared vocabulary. Similarly for the transformer-big model, the improvement is 0.17 BLEU scores compared with the 50K shared vocabulary. In general, the 32K shared vocabulary performs better than the 50K shared vocabulary and the transformer-big model is more effective than the transformer-base model.

TABLE 4. The BLEU scores of the baselines that are based on different models.

Vocabulary Size	Method	Steps	BLEU Scores
32K	Transformer-base	0.5M	33.22
32K	Transformer-big	0.5M	**34.36**
32K	Parent-child (base)	1M	**34.80**
50K	Parent-child (base)	1.459M	34.36
32K	Parent-child (big)	0.628M	**34.93**
50K	Parent-child (big)	0.696M	34.76

TABLE 5. The performance of the parent-child architecture that trains different steps on the parent model.

Vocabulary Size	Method	Steps(parent)	BLEU Scores
32K	Parent-child (big)	293K	34.85(160K)
32K	Parent-child (big)	460K	**34.93**(168K)

In the parent-child architecture, we also compare the performance of training different steps on the parent model with the 32K shared vocabulary as shown in table 5. The results show that when the parent model training is more convergent, the performance of the transfer learning is better.

E. DETAILS OF OUR EXPERIMENTS

The problems are created in tensor2tensor firstly. In the process of data pre-processing, we use BPE to segment words into sub-words and use character-level embedding for Chinese of the third layer. The two different mixed corpora are used to create shared vocabularies. One is mixing 350K

Uygur-Chinese, 200K Turkish-English and 350K English-Chinese that choose one in five sentence in all English-Chinese parallel corpus to get a 36K shared vocabulary, the other is mixing 200K Uygur-Chinese, 200K Turkish-English and 200K English-Chinese to get a 32K shared vocabulary for the sake of fairness. Then the English-Chinese training data and validation set are generated by the shared vocabulary. Subsequently, the tensor2tensor encodes training data and validation set to binary files. In the process of training, the model is trained 500K steps on the training data, it takes about 9 days. Then we continue to get the training data on Turkish-English parallel corpus. The parameters of intermediate model are initialized with the parameters that are trained on English-Chinese parallel corpus and the model is trained on the training data for 100k steps, it takes about 31 hours. Next, we get the training data on Uygur-Chinese parallel corpus. The model of low-resource language pair is initialized by the parameters that are fine-tuned on the Turkish-English parallel corpus and because of the small amount of data, the model is trained for 100K steps to converge, it takes one day. Other than this, different steps are experimented on the intermediate language and the low-resource language respectively to explore the best results. In order to explore the generalization of the hierarchical transfer learning architecture, we compare hierarchical transfer learning architecture with the parent-child architecture on Turkish-English. In training process of hierarchical transfer learning architecture, we set English-Chinese as the high-resource language and Uygur-Chinese as the intermediate language. In the process of parent-child architecture, the parent model is trained on English-Chinese. Finally, the test sets are decoded with the beam search size is 8, and the quality of the results is evaluated by the BLEU score [55] that is common evaluation method in the field of NMT. The BLEU score is computed as:

$$BLEU = BP * \left(\sum_{i=1}^{n} \alpha_i * log^{p_i} \right) \tag{5}$$

where BP is the sentence brevity penalty to punish a sentence that is too long or too short. BP is calculated in equation (6). p_i is the modified i-gram precision. α_i is the weight of each modified i-gram precision.

$$BP = \begin{cases} 1 & if\ a > b \\ e^{1-\frac{b}{a}} & if\ a \leq b \end{cases} \tag{6}$$

where a is the sentence that needs to be evaluated. b is the reference sentence.

V. RESULTS AND ANALYSIS

In this section, the performance of our hierarchical transfer learning architecture based on the transformer is compared with the three baselines. And numbers of experiments have been done to explore the agents that affect the results of our hierarchical transfer learning architecture.

TABLE 6. The BLEU Scores of our hierarchical transfer learning architecture are compared with the three baselines on Uygur-Chinese.

Vocabulary Size	Model	Steps	BLEU Scores
-	Phrase-based SMT	-	33.56
32K	Transformer(big)	0.5M	34.36
32K	Parent-child (base)	1M	34.80
32K	Parent-child (big)	0.69M	34.93
32K	Hierarchical transfer learning (big)	0.8M	**35.51**
36K	Hierarchical transfer learning (big)	0.8M	34.75

TABLE 7. The BLEU Score of the hierarchical transfer learning architecture is compared with parent-child architecture on Turkish–English.

Vocabulary Size	Model	BLEU Scores
32K	Parent-child (big)	17.73
32K	Hierarchical transfer learning (big)	**18.46**

Table 6 shows the BLEU scores of the three baselines and our method on Uygur-Chinese. We can find that our hierarchical transfer learning architecture improves 1.95 BLEU scores compared with the Phrase-Based SMT system, improves 1.15 BLEU scores compared with the transformer-big model and improves 0.58 BLEU scores compared with the parent-child architecture based on the Transformer-Big model. Experimental results show that our hierarchical outperforms the three baselines. The reasons are that the single transformer initializes the parameters of the model randomly. The previous work, which is based on parent-child architecture, only considers the impact of data volume advantage or language similarity advantage. However, both of these factors are proven to be effective for low-resource problem. Hence, in order to further improve the performance for low-resource languages, we use a three-layer architecture to combine these advantages. The hierarchical transfer learning architecture applies transfer learning method by setting the same hyper-parameters to maintain the consistency of the model structure. The parameters of the model such as the multi-head attention, the feed-forward network, the attention mechanism and the masked multi-head attention are transferred layer by layer for initialization. The first transference utilizes the data volume advantage of high-resource language. Furthermore, both Turkish and Uygur are based on the basic syntactic structures of subject-object-predicate as shown in Figure 3 and are also very similar in terms of word formation [56]. The second layer adds syntactic similarity information of the intermediate language. But we also notice that the improvement of our architecture is not particularly obvious compared with parent-child architecture on Uygur-Chinese. We speculate that the Turkish-English parallel corpus is more scarce than Uygur-Chinese parallel corpus, which leads to insufficient learning of synthetic information.

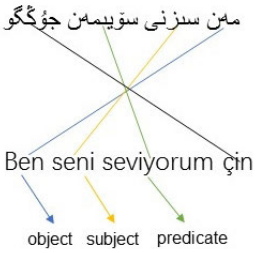


FIGURE 3. The basic syntactic structure of Uygur and Turkish.

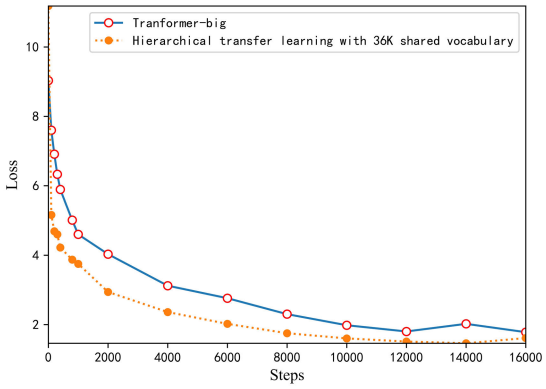


FIGURE 4. The loss curve of our hierarchical transfer learning architecture and the transformer-big model.

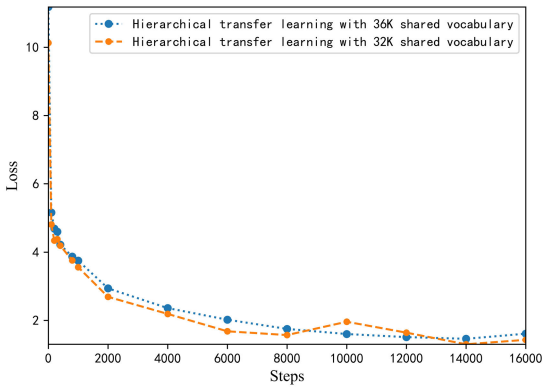


FIGURE 5. The loss curve of the hierarchical transfer learning architecture with different shared vocabularies.

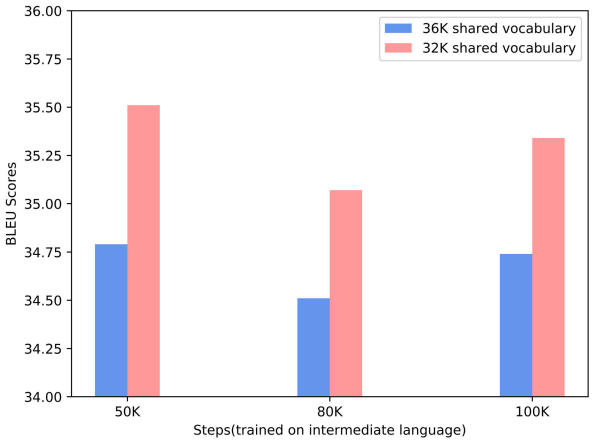


FIGURE 6. The BLEU scores of the hierarchical transfer learning architecture that trains 50k, 80k, 100k steps on the intermediate language with 32k and 36k shared vocabulary respectively.

Furthermore, we explore the generalization of the hierarchical transfer learning architecture on Turkish-English. From Table 7, we can find that the hierarchical transfer learning architecture significantly outperforms the parent-child architecture on Turkish-English, which gets helpful information from Uygur-Chinese. Besides, the improvements are more obvious compared with Uygur-Chinese. We speculate that more similar intermediate languages can better improve the performance of low-resource languages. Specially, the significant improvements on Turkish-English also shows that without setting same target languages relieves the constrain on languages selection, which increases the flexibility of the architecture. Due to the fact that most similar languages of low-resource languages are still low-resource, our hierarchical transfer learning architecture allows low-resource languages to help each other.

We also find that our hierarchical transfer learning architecture not only can improve the BLEU scores compared with the baselines but also can converge faster than the NMT system without transfer learning method on low-resource language pairs. It is obvious that the man who is adept at languages learns a new language faster than the man without language foundation. The parameters that are trained on the first two layers have learned the common information about languages and the similar syntax information of the intermediate language. Therefore, our architecture converges faster on low-resource language. The loss curves are shown in Figure 4. The results testified that our hierarchical transfer learning architecture can effectively initialize the parameters for the low-resource language pair compared with random initialization.

To explore the impact of the size of shared vocabulary on our hierarchical transfer learning architecture, Figure 5 shows the loss curve with different shared vocabularies. The size of the shared vocabulary has little impact on the loss curve, this further illustrates that effective parameter initialization through the hierarchical transfer learning architecture can make the model converge faster. However, the BLEU scores of our hierarchical transfer learning architecture with different shared vocabulary are significantly different. The BLEU scores of low-resource language that trained different steps on the intermediate language pair with different shared vocabulary are compared in Figure 6. Experimental results show that the BLEU scores of the hierarchical transfer learning architecture with the 32k shared vocabulary are outstandingly better than the 36k shared vocabulary.

To explore the impact of training steps of intermediate language on our architecture with 32K shared vocabulary, Figure 7 shows the fluctuation of the BLEU scores

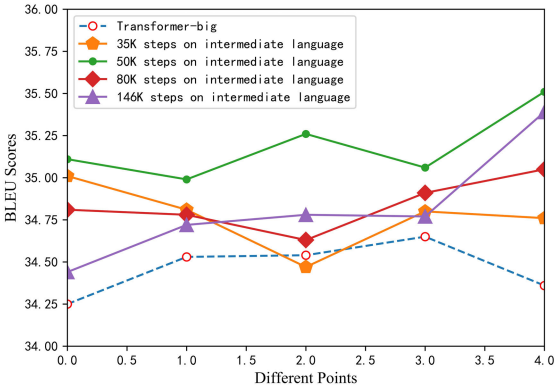


FIGURE 7. The performance of the hierarchical transfer learning architecture that is trained on the intermediate language with different steps is compared with the transformer-big model.

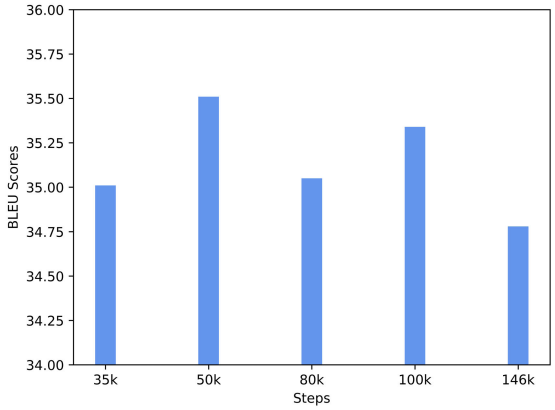


FIGURE 8. The best BLEU Scores of our hierarchical transfer learning architecture that is trained on the intermediate language with different steps.

of the hierarchical transfer learning architectures on Uygur-Chinese, which are trained with different steps on the intermediate corpus. Because of learning the prior knowledge of the high-resource language and the helpful knowledge such as syntactic information of the similar intermediate language, the Figure shows that our hierarchical transfer learning architecture that trains different steps on the intermediate language is outstandingly excesses the transform-big model.

The 32K shared vocabulary outstandingly excesses the 36K shared vocabulary. The different steps that are trained on the intermediate language are experimented to explore the best result with 32K shared vocabulary. The BLEU scores of the hierarchical transfer learning architecture that is trained on the intermediate language pair with different steps are shown in Figure 8. We can find that the model, which is trained on the intermediate language with 50K steps and continue to train the model on the low-resource language with 250K steps, gets the best result. The reasons are that the small number of steps that are trained on the intermediate language pair will result in under-fitting. Nevertheless the large number of training steps will lead to over-fitting. That will damage the quality of initialization for the low-resource language.

In general, experimental results show that the hierarchical transfer learning architecture is an effective method for the low-resource problems and converges faster than the single transformer. Specially, the architecture also has excellent generalization on other low-resource languages. We guess that the more similar the intermediate language is to the low-resource language, the better the initialization of the parameters, which is the hypothesis we are going to verified.

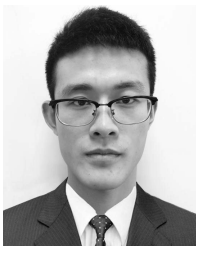
VI. CONCLUSION

We propose a hierarchical transfer learning architecture to handle low-resource problems in this paper. Different from majority of exiting studies that are constrained by the parent-child architecture, our hierarchical transfer learning architecture adds the intermediate layer to make full use of helpful languages. Based on this architecture, not only can our model combine the advantage of data volume on the high-resource language and the superiority of synthetic similarity on intermediate language, but also can increase model flexibility. Experimental results on Uygur-Chinese and Turkish-English translation show that our hierarchical transfer learning architecture achieve significant improvements over a variety of baselines. In the future, we are going to explore the effect of the difference and the size of the intermediate language on the performance of our hierarchical transfer learning architecture.

REFERENCES

[1] A. V. Lyovin, "An introduction to the languages of the world," *Amer. Anthropologist*, vol. 101, no. 4, pp. 856–858, 2010.
[2] S. R. Anderson, "How many languages are there in the world," Linguistic Soc. Amer., Tech. Rep., 2010.
[3] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.
[4] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. ICLR*, 2014, pp. 1–15.
[5] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *Comput. Sci.*, 2015.
[6] Y. Wu *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," 2016, *arXiv:1609.08144*. [Online]. Available: https://arxiv.org/abs/1609.08144
[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–11.
[8] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. Dauphin, "Convolutional sequence to sequence learning," in *Proc. ICML*, Aug. 2017, pp. 1243–1252.
[9] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proc. 45th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Jun. 2007, pp. 177–180.
[10] B. Zoph, D. Yuret, J. May, and K. Knight, "Transfer learning for low-resource neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2016, pp. 1568–1575.
[11] P. Koehn and R. Knowles, "Six challenges for neural machine translation," in *Proc. 1st Workshop Neural Mach. Transl., Meeting Assoc. Comput. Linguistics (ACL)*, 2017, pp. 28–39.
[12] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2016, pp. 86–96.

[13] M. Fadaee, A. Bisazza, and C. Monz, "Data augmentation for low-resource neural machine translation," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Jul./Aug. 2017, pp. 567–573.
[14] J. Zhang and C. Zong, "Exploiting source-side monolingual data in neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Nov. 2016, pp. 1535–1545.
[15] Y. Cheng, W. Xu, Z. He, W. He, H. Wu, M. Sun, and Y. Liu, "Semi-supervised learning for neural machine translation," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, Aug. 2016, pp. 1965–1974.
[16] Z. Zhang, S. Liu, M. Li, and E. Chen, "Joint training for neural machine translation models with monolingual data," in *Proc. AAAI*, Apr. 2018, pp. 1–8.
[17] Y. Xia, Y. Xia, T. Qin, L. Wang, N. Yu, T.-Y. Liu, and W.-Y. Ma, "Dual learning for machine translation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1–9.
[18] J. Gu, Y. Wang, Y. Chen, K. Cho, and V. O. K. Li, "Meta-learning for low-resource neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2018.
[19] S. Zhang, G. Mahmut, D. Wang, and A. Hamdulla, "Memory-augmented Chinese-Uyghur neural machine translation," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Dec. 2017, pp. 1092–1096.
[20] S. Ren, W. Chen, S. Liu, M. Li, M. Zhou, and S. Ma, "Triangular architecture for rare language translation," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2018, pp. 1–10.
[21] Y. Chen, Y. Liu, Y. Cheng, and V. O. K. Li, "A teacher-student framework for zero-resource neural machine translation," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, May 2017, pp. 1–11.
[22] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
[23] G. Luo, S. Dong, K. Wang, W. Zuo, S. Cao, and H. Zhang, "Multi-views fusion CNN for left ventricular volumes estimation on cardiac MR images," *IEEE Trans. Biomed. Eng.*, vol. 65, no. 9, pp. 1924–1934, Sep. 2018.
[24] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.
[25] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3320–3328.
[26] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *Comput. Sci.*, 2014.
[27] M. Ghifary, W. B. Kleijn, and M. Zhang, "Domain adaptive neural networks for object recognition," in *Proc. Pacific Rim Int. Conf. Artif. Intell.*, Cham, Switzerland: Springer, 2014, pp. 898–904.
[28] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. ICML*, 2015, pp. 97–105.
[29] M. Long, J. Wang, Y. Cao, J. Sun, and P. S. Yu, "Deep learning of transferable representation for scalable domain adaptation," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 8, pp. 2027–2040, Aug. 2016.
[30] M. Long, H. Zhu, J. Wang, and M. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, Aug. 2017, pp. 2208–2217.
[31] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4068–4076.
[32] Z. Zhao, Y. Chen, J. Liu, Z. Shen, and M. Liu, "Cross-people mobile-phone based activity recognition," in *Proc. IJCAI*, Jun. 2011, pp. 1–6.
[33] W.-Y. Deng, Q.-H. Zheng, and Z.-M. Wang, "Cross-person activity recognition using reduced kernel extreme learning machine," *Neural Netw.*, vol. 53, pp. 1–7, May 2014.
[34] S. J. Pan, J. T. Kwok, and Q. Yang, "Transfer learning via dimensionality reduction," in *Proc. AAAI*, Jul. 2008, pp. 677–682.
[35] F. Nater, T. Tommasi, H. Grabner, L. V. Gool, and B. Caputo, "Transferring activities: Updating human behavior analysis," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, Nov. 2011, pp. 1737–1744.
[36] Y. Wei, Y. Zhu, C. W. Leung, Y. Song, and Q. Yang, "Instilling social to physical: Co-regularized heterogeneous transfer learning," in *Proc. AAAI*, Feb. 2016, pp. 1–7.
[37] T. Kocmi and O. Bojar, "Trivial transfer learning for low-resource neural machine translation," in *Proc. 3rd Conf. Mach. Transl., Res. Papers*, Oct. 2018, pp. 244–252.
[38] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*. Philadelphia, PA, USA: IGI Global, 2010, pp. 242–264.
[39] D. Dong, H. Wu, W. He, D. Yu, and H. Wang, "Multi-task learning for multiple language translation," in *Proc. 53th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2015, pp. 1723–1732.
[40] O. Firat, K. Cho, and Y. Bengio, "Multi-way, multilingual neural machine translation with a shared attention mechanism," in *Proc. NAACL-HLT*, Jun. 2016, pp. 866–875.
[41] T. Q. Nguyen and D. Chiang, "Transfer learning across low-resource, related languages for neural machine translation," in *Proc. 8th Int. Joint Conf. Natural Lang. Process.*, Nov. 2017, pp. 296–301.
[42] R. Dabre, T. Nakagawa, and H. Kazawa, "An empirical study of language relatedness for transfer learning in neural machine translation," in *Proc. 31st Pacific Asia Conf. Lang., Inf. Comput.*, Nov. 2017, pp. 282–286.
[43] B. Tan, Y. Zhang, S. J. Pan, and Q. Yang, "Distant domain transfer learning," in *Proc. AAAI*, Feb. 2017, pp. 1–7.
[44] B. Tan, Y. Zhang, S. J. Pan, and Q. Yang, "Transitive transfer learning," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2015, pp. 1155–1164.
[45] A. Vaswani, S. Bengio, E. Brevdo, F. Chollet, A. Gomez, S. Gouws, L. Jones, L. Kaiser, N. Kalchbrenner, N. Parmar, R. Sepassi, N. Shazeer, and J. Uszkoreit, "Tensor2tensor for neural machine translation," in *Proc. 13th Conf. Assoc. Mach. Transl. Americas*, Boston, MA, USA, Mar. 2018, pp. 193–199.
[46] M. Popel and O. Bojar, "Training tips for the transformer model," *Prague Bull. Math. Linguistics*, vol. 110, no. 1, pp. 43–70, Apr. 2018.
[47] S. Nitish, G. Hinton, A. Krizhevsky, I. Sutskever, and T. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.
[48] K. Diederik and J. Ba, "Adam: A method for stochastic optimization," *Comput. Sci.*, 2014.
[49] J. Tiedemann and P. Data, "Tools and interfaces in OPUS," in *Proc. LREC*, 2012.
[50] M. Ziemski, M. Junczys-Dowmunt, and B. Pouliquen, "The united nations parallel corpus v1. 0," in *Proc. LREC*, 2016.
[51] O. Bojar, R. Chatterjee, and C. Federmann, "Findings of the 2017 conference on machine translation (wmt17)," in *Proc. 2nd Conf. Mach. Transl.*, 2017, pp. 169–214.
[52] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 1715–1725.
[53] J. Chung, K. Cho, and Y. Bengio, "A character-level decoder without explicit segmentation for neural machine translation," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Berlin, Germany, Aug. 2016, pp. 1693–1703.
[54] K. Wolk and K. Marasek, "Survey on neural machine translation into polish," in *Proc. Int. Conf. Multimedia Netw. Inf. Syst.* Cham, Switzerland: Springer, 2018, pp. 260–272.
[55] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, "BLEU: A method for automatic evaluation of machine translation.," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2002, pp. 311–318.
[56] G. Asli and K. Cella, *Turkish: A Comprehensive Grammar*. Evanston, IL, USA: Routledge, 2005.



GONGXU LUO was born in Penglai, China, in 1993. He received the B.E. degree in software engineering from Yantai University, Yantai, China, in 2012. He is currently pursuing the M.E. degree with The Xinjiang Technical Institute of Physics and Chemistry, Chinese Academy of Sciences, Urumqi, China. He is also with the University of Chinese Academy of Sciences. His research interests include NMT and deep learning.

YATING YANG received the Ph.D. degree in computer application technology from The Xinjiang Institute of Physics Chemistry, University of Chinese Academy of Sciences. She is currently an Associate Researcher with The Xinjiang Technical Institute of Physics and Chemistry, Chinese Academy of Sciences. She is currently involved in the research of key technologies in multilingual information processing. Her main research interests include key technologies in the fields of natural language processing such as multilingual speech recognition and machine translation.

YANG YUAN received the B.E. degree in computer science and technology from Xi'an Jiaotong University, China, in 2010. He is currently pursuing the Ph.D. degree in natural language processing with The Xinjiang Technical Institute of Physics and Chemistry, Chinese Academy of Sciences, Urumqi, China. His current research interests include machine translation, synonym extraction, and word sense disambiguation.

ZHANHENG CHEN is currently pursuing the Ph.D. degree with the University of Chinese Academy of Sciences, China. His current research interests include data mining, natural language processing, and pattern identification. He has several publications in journals such as *BMC Genomics*, *BMC Systems Biology*, *Frontiers in Genetics*, and *Applied Sciences*. He has published in international conferences such as RECOMB, ICIC, and ISBRA.

AIZIMAITI AINIWAER is currently pursuing the Ph.D. degree with the University of Chinese Academy of Sciences. He is currently an Assistant Researcher with The Xinjiang Technical Institute of Physics and Chemistry, Chinese Academy of Sciences. His research interests include natural language processing and knowledge graphs construction.