TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214
DOI: 10.26599/TST.2020.9010029
Volume 26,

Enriching the Transfer Learning with Pre-Trained Lexicon Embedding for Low-Resource Neural Machine Translation

Mieradilijiang Maimaiti, Yang Liu*, Huanbo Luan, and Maosong Sun

Abstract: Most State-Of-The-Art (SOTA) Neural Machine Translation (NMT) systems today achieve outstanding results based only on large parallel corpora. The large-scale parallel corpora for high-resource languages is easily obtainable. However, the translation quality of NMT for morphologically rich languages is still unsatisfactory, mainly because of the data sparsity problem encountered in Low-Resource Languages (LRLs). In the low-resource NMT paradigm, Transfer Learning (TL) has been developed into one of the most efficient methods. It is difficult to train the model on high-resource languages to include the information in both parent and child models, as well as the initially trained model that only contains the lexicon features and word embeddings of the parent model instead of the child languages feature. In this work, we aim to address this issue by proposing the language-independent Hybrid Transfer Learning (HTL) method for LRLs by sharing lexicon embedding between parent and child languages without leveraging back translation or manually injecting noises. First, we train the High-Resource Languages (HRLs) as the parent model with its vocabularies. Then, we combine the parent and child language pairs using the oversampling method to train the hybrid model initialized by the previously parent model. Finally, we fine-tune the morphologically rich child model using a hybrid model. Besides, we explore some exciting discoveries on the original TL approach. Experimental results show that our model consistently outperforms five SOTA methods in two languages Azerbaijani (Az) and Uzbek (Uz). Meanwhile, our approach is practical and significantly better, achieving improvements of up to 4.94 and 4.84 BLEU points for low-resource child languages Az $\rightarrow$ Zh and Uz $\rightarrow$ Zh, respectively.

Key words: artificial intelligence; natural language processing; neural network; machine translation; low-resource languages; transfer learning

1 Introduction

The end-to-end framework is a common Neural Network (NN) architecture. Neural Machine Translation (NMT), which uses neural networks to model the translation process of natural languages in an end-to-end manner, has attracted intense attention from the community[1–4]. Excelling in learning representations and capturing long-distance dependencies by exploiting gating[5,6], attention[2] mechanisms, and pre-training[7] method, NMT has shown significant superiority over conventional Statistical Machine Translation (SMT)[8–10] for a number of natural languages[11]. The noticeable performance increases over traditional SMT on many language pairs. NMT has recently made an attractive approach for real-world Machine Translation (MT) systems.

• Mieradilijiang Maimaiti, Yang Liu, Huanbo Luan, and Maosong Sun are with the Institute for Artificial Intelligence, Beijing National Research Center for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China. E-mail: meadljmm15@mails.tsinghua.edu.cn; {liuyang.china, luanhuanbo}@gmail.com; sms@tsinghua.edu.cn.
• Yang Liu is also with the Beijing Academy of Artificial Intelligence, Beijing Advanced Innovation Center for Language Resources, Beijing 100084, China.
* To whom correspondence should be addressed.
Manuscript received: 2020-08-08; revised: 2020-08-23; accepted: 2020-08-31

Training the NMT model on Low-Resource Languages (LRLs) is a challenge because LRLs frequently bring a data scarcity issue. Therefore, we consider this problem and aim to enrich the performance of NMT for LRLs in our proposed model. The primary issue of the NMT is the model quality which heavily relies on the accessibility of large amounts of parallel data, which is frequently hard to obtain. Reference [12] showed that if the NMT models are trained without large-scale parallel corpora, i.e., less than one million sentence pairs, it is hard to significantly outperform SMT since neural network tends to learn badly on low-resource events.

NMT requires large amounts of parallel corpora, and it normally learns poorly on LRLs. Training the NMT model on low-resource corpora is a challenge because LRLs are highly agglutinative[13], and hence can add various affixes at the beginning or the end of words, to form a new word. In MT approaches, morphologically rich low-resource language pairs frequently bring about a data scarcity issue[14]. Many LRLs are Morphologically Rich Languages (MRLs)[15], but suffer from their prominent data sparsity. Therefore, we can maintain that in the low-resource NMT data sparsity issue is one of the difficult tasks. The primary issue of the NMT is the model quality which heavily relies on the accessibility of large amounts of parallel data, which is frequently hard to obtain for many language pairs. For instance, among the 20 language pairs of the Europarl dataset, only the French-English parallel corpus includes more than two million sentence pairs[16]. However, in the Europarl dataset, the corpus size of only ten language pairs contains less than one million sentence pairs. Reference [12] showed that if the NMT models are trained without large-scale parallel corpora, i.e., less than one million sentence pairs, it is hard to significantly outperform SMT since NN tends to learn the model parameters poorly on low-resource events.

In MT scenario, the translation of LRLs is one of the most challenging tasks. Intrinsically, the difficulty of obtaining large amounts of high-quality parallel data with good coverage constrains the performance of NMT. Nevertheless, various methods that may help to reduce these complexities have been presented. Such as "teacher-student network" which was proposed in Refs. [17] and [18] is also motivated by zero-shot learning. Additionally, the "semi-supervised"[19] and "pivot-based"[20] models have also attracted some attentions in this field. In the MT community of LRLs, Transfer Learning (TL) has become one of the main and most efficient methods[12,21–23]. The TL is one of the essential elements for coping with the data sparsity problem in low-resource NMT. Reference [12] obtained better translation quality of NMT systems by leveraging TL on a low-resource NMT task. Reference [24] explored how different choices of parent models affect the performance of child models. Reference [23] trained the parent languages number of iterations and switched the training corpus to the child language pairs for the rest of training by using of TL in NMT.

However, many existing approaches which take advantages of TL suffer from a core disadvantage: they are incapable of containing lexicon information as well as lexicon embeddings of low-resource child languages. The core idea of Ref. [12] has the drawback of only exploiting the vocabulary of one high-resource model (parent) to enhance one low-resource model (child) at a time, rather than using the lexicon information of the child model. By contrast, as we train the parent model on a combined training set with shared vocabularies, we leverage both the vocabulary of the parent and child, which may produce a better translation performance of the child model.

In this paper, in contrast with the aforementioned approaches we focus on addressing the problem of lexicon embedding and vocabulary information of MRLs (child) in the training step. In order to make our model stronger, we regard that the parent model is more helpful for the child model if it contains more information about the child pairs before initializing. If the parent model can be trained on the combined training set that combines parent and child language pairs, instead of only using the parent pair, the child model will work better than the aforementioned TL method. We introduce a practical and straightforward method of Hybrid Transfer Learning (HTL) for the LRLs in NMT. Our basic idea aims at sharing lexicon features between the parent and child model before fine-tuning. As illustrated in Fig.1, firstly, we train our parent model on a large-scale training set, then prepare a combined corpus to train the hybrid model rather than directly initialize the child model without using both the back translation and artificially injecting any noises[25]. Secondly, we combine the large-scale parent pairs and small-scale child pairs using an oversampling method[26]. We build a shared vocabulary based on a combined dataset and train hybrid model. Finally, we initialize the child model with the hybrid model, which was trained using

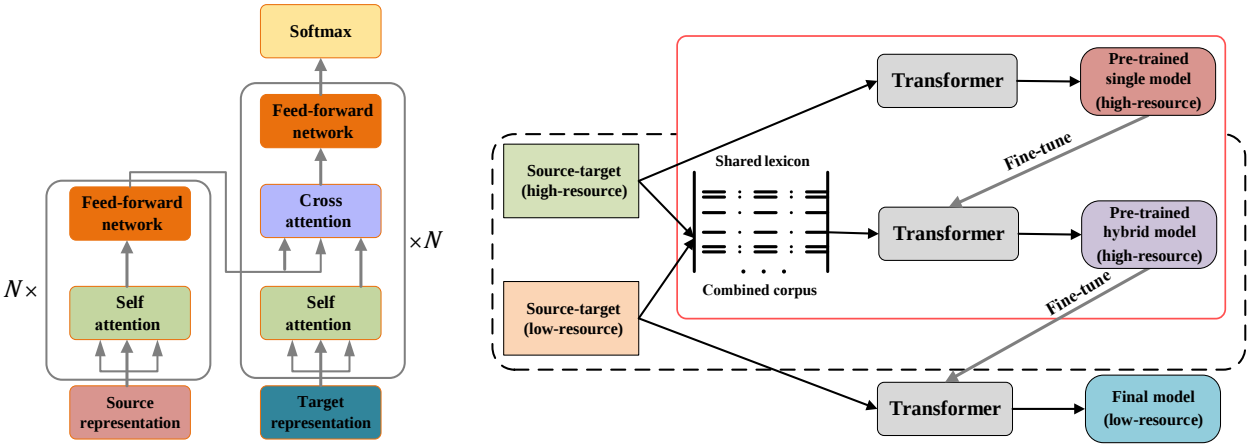

Fig. 1 (a) SOTA NMT model transformer and (b) illustration of HTL. Here the dotted rectangle denotes the training of hybrid model and the red rectangle represents pre-trained model. Here, N represents the number of identical layers.

shared vocabularies in the combined training corpus via an oversampling technique. Our hybrid model should include more lexicon information before fine-tuning the child model so that it can help the child model to achieve better improvements than five baseline systems. Our main contributions are as follows:

- We offer a solution for the disadvantages of the original TL, which is unable to learn the lexicon features of the child model before fine-tuning.
- We propose an efficient and simple HTL training approach for MRLs in NMT.
- We provide a model transparent training approach for any NMT architecture.
- We made interesting discoveries by taking advantage of the original TL.

2 Background

2.1 Neural machine translation

Obviously, we can regard X as a source language sentence and Y as a target language sentence. Given a source sentence $x = x_1, \ldots, x_i, \ldots, x_I$ and a target sentence $y = y_1, \ldots, y_j, \ldots, y_J$, standard NMT models[1–3] usually factorize the sentence-level translation probability as a product of word-level probabilities:

$$P(y|x;\theta) = \prod_{j=1}^{J} P(y_j|x, y_{<j};\theta) \quad (1)$$

where θ is a set of model parameters, and $y_{<j}$ is a partial translation.

Let $\langle X, Y\rangle = \{\langle x^{(n)}, y^{(n)}\rangle\}_{n=1}^{N}$ be a training corpus. The log-likelihood of the training parallel data is maximized by the standard training objective:

$$\hat{\theta} = \text{argmax}_\theta \Big\{ \sum_{n=1}^{N} \log P(y^{(n)}|x^{(n)};\theta) \Big\} \quad (2)$$

Given learned model parameters $\hat{\theta}$, the translation decision rule for unseen source sentence x is given by

$$\hat{y} = \text{argmax}_y \{P(y|x;\hat{\theta})\} \quad (3)$$

Meanwhile, calculating the highest probability $\hat{y} = \hat{y}_1, \ldots, \hat{y}_j, \ldots, \hat{y}_J$ of the target sentence can be separated at the word level:

$$\hat{y}_j = \text{argmax}_y \{P(y|x, \hat{y}_{<j};\hat{\theta})\} \quad (4)$$

2.2 Transfer learning

As described in the Machine Learning (ML) community, TL[27] contains the concepts of a domain and a task. A domain $\mathcal{D}$ comprises a feature space $\mathcal{X}$ and marginal probability distribution $P(X)$ over the feature space, where $X = x_1, \ldots, x_n \in \mathcal{X}$. TL exploits knowledge[28] from the existing model to improve the performance of related work. TL varies from ML in that it learns from learned-models, while ML learns from data. In both image processing[29] and Domain Adaptation (DA) tasks[30], TL has been widely used in Computer Vision (CV). The main point of the TL is to pre-train the NN or to pre-initialize the weights of the NN. The weight updating process after training the child language pair datasets is called fine-tuning.

As illustrated in Fig. 2, we take $L_3 \rightarrow L_2$ as a parent pair High-Resource Languages (HRLs) and $L_1 \rightarrow L_2$ as a child pair Low-Resource Languages (LRLs). L_3 and L_1 are source languages of parent and child pairs, respectively, while L_2 is the target language for both. Moreover, we set the dataset of parent pair as D_{L_3,L_2}, while the dataset of the child pair is D_{L_1,L_2}. We also set

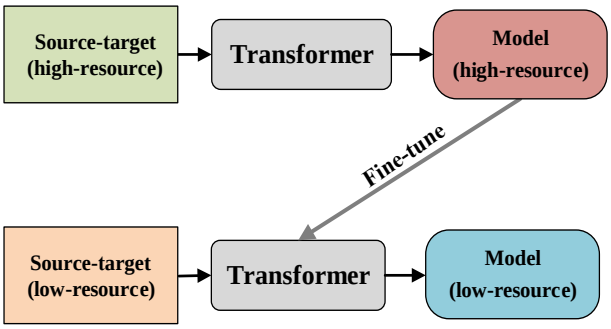

Fig. 2 Architecture of the original TL for NMT.

$M_{L_3 \to L_2}$ as the parent model which has been learned using its own dataset D_{L_3,L_2}. Generally, we initialize the child model $M_{L_1 \to L_2}$ using a parent model, and a corresponding parameter of the parent model $M_{L_3 \to L_2}$:

$$\theta_{L_3,L_2} = \{\langle e_{L_3}, W, e_{L_2} \rangle\} \tag{5}$$

where e_{L_3} and e_{L_2} are both source and target embeddings, and W is the model parameter. We also encourage the training objective to maximize the likelihood of the dataset D_{L_3,L_2}:

$$\hat{\theta}_{L_3 \to L_2} = \text{argmax}_{\theta_{L_3 \to L_2}} \left\{ L(D_{L_3,L_2}, \theta_{L_3,L_2}) \right\} \tag{6}$$

Then, the child model $M_{L_1 \to L_2}$ will be fine-tuned using the parent model $M_{L_3 \to L_2}$:

$$\theta_{L_1 \to L_2} = f(\hat{\theta}_{L_3 \to L_2}) \tag{7}$$

where $\hat{\theta}_{L_3 \to L_2}$ denotes the learned parameters of the parent model $M_{L_3 \to L_2}$ that are transferred to child model $M_{L_1 \to L_2}$ using the initialization function f.

2.3 Transformer

The transformer is one of the most popular neural machine translation architectures in the NMT community. The architecture of the transformer also consists of two parts, such as encoder and decoder. There are stacks of N identical layers in the encoder. Each of them is composed of two sub-layers, such as the multi-head self-attention layer and the simple feed-forward network. The self-attention sub-layer first runs to interact information between different words. It employs h attention heads, which allows the model to jointly attend to capturing features from different representation subspaces and different positions. A single attention head is calculated on a query Q, a key K, and a value V:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right)V \tag{8}$$

After that, the self-attention layer concatenates h different representations of (Q, K, V) and maps the concatenation by using a feed-forward layer to generate the output:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{9}$$

where each head in MultiHead is noted as below:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{10}$$

Moreover, the feed-forward sub-layer connected behind the self-attention mechanism runs as the following way:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{11}$$

We also utilize a residual connection[31] between each of the two sub-layers, following the idea of layer normalization[32]. Consequently, the output which is generated from each sub-layer is LayerNorm(x + Sublayer(x)), where Sublayer(x) is representation implemented by the sub-layer itself.

3 Methodology

Hybrid transfer learning. As depicted in Fig.1, we take $L_1 \to L_2$ as the child LRLs, while L_1 and L_2 represent the source language and target language. $D_{L_1 \to L_2}$ is a dataset of the child pairs, while $\theta_{L_1 \to L_2}$ is the model parameter, e_{L_1} and e_{L_2} are the source and target side embeddings, respectively. Similarly, all of the corpora $L_3 \to L_2, \ldots, L_{k+1} \to L_2$ act as parent language pairs, and $D_{L_3 \to L_2}$, …, $D_{L_{k+1} \to L_2}$ are datasets.

Reference [12] proposed the idea of transferring parameters of $M_{L_3 \to L_2}$ to $M_{L_1 \to L_2}$, as well as initialized the child with parent parameters directly. Generally, the parameters of $M_{L_3 \to L_2}$ are composed of embeddings e_{L_3} and e_{L_2} and the matrices W (see Eq. (5)). We maximize the log-likelihood of the $D_{L_3 \to L_2}$ as the objective of the standard training process (see Eq. (6)). $M_{L_3 \to L_2}$ fine-tunes $M_{L_1 \to L_2}$ using $D_{L_3 \to L_2}$ refer to Eq. (7), where f is the initialization function, which helps the $M_{L_1 \to L_2}$ to transfer the already learned parameters $\hat{\theta}_{L_3 \to L_2}$ from the parent model $M_{L_3 \to L_2}$.

The proposed model is one of the revised versions of the original TL. We were motivated by the domain adaptation problem in the computer vision field. As shown in Algorithm 1, we do not directly fine-tune the child model using a pre-trained parent model. Instead, we focus on vocabulary sharing between the parent and child models and combine the dataset after oversampling the LRLs. First, we oversample the child language pairs by duplicating each sentence of the child pair (it was updated from $D_{L_1 \to L_2}$ to $D'_{L_1 \to L_2}$). Then, we combine

Algorithm 1 Hybrid transfer learning

Input: high-resource dataset $D_{L_3 \to L_2} = \{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$ and low-resource dataset $D_{L_1 \to L_2} = \{(x^{(m)}, y^{(m)})\}_{m=1}^{M}$;
Output: translation results y on a low-resource testset;
1: Read the high-resource dataset $D_{L_3 \to L_2}$;
2: Read the low-resource dataset $D_{L_1 \to L_2}$;
3: $D'_{L_1 \to L_2}$ is the over-sampled version of $D_{L_1 \to L_2}$;
4: cps-size$_{\text{high}}$ $\leftarrow$ the corpus size of the $D_{L_3 \to L_2}$; ▷ Calculate corpus size
5: cps-size$_{\text{low}}$ $\leftarrow$ the corpus size of the $D_{L_1 \to L_2}$;
6: **for** each sen-pair in $D_{L_3 \to L_2}$ **do**
7: Train the parent model $M_{L_3 \to L_2}$ on $D_{L_3 \to L_2}$; ▷ Pre-train the parent model on the high-resource dataset
8: **end for**
9: smp$_{\text{num}}$ = cps-size$_{\text{high}}$ / cps-size$_{\text{low}}$; ▷ Get oversampling size
10: **for** each sen-pair in $D_{L_1 \to L_2}$ **do**
11: Duplicate each sentences smp$_{\text{num}}$ times; ▷ Copy each sentence
12: cps-size$_{\text{low}}$ $\leftarrow$ update the corpus size of $D_{L_1 \to L_2}$;
13: **end for**
14: $D_{L_1 \to L_2} \leftarrow D_{L_1 \to L'_2}$; ▷ Obtain oversampled data
15: $D_{L_m \to L_2} = D_{L_3 \to L_2} + D_{L_1 \to L'_2}$ ▷ Build combined data
16: Voc$_{\text{shared}}$ $\leftarrow$ build shared vocabulary on $D_{L_m \to L_2}$;
17: $M_{L_m \to L_2} \leftarrow$ initialize the hybrid model with $M_{L_3 \to L_2}$; ▷ Train the hybrid model
18: **for** each sen-pair in $D_{L_m \to L_2}$ **do**
19: Train the $M_{L_m \to L_2}$ on $D_{L_m \to L_2}$ with Voc$_{\text{shared}}$;
20: **end for**
21: $M'_{L_m \to L_2} \leftarrow M_{L_m \to L_2}$; ▷ Update the hybrid model
22: $M_{L_1 \to L_2} \leftarrow$ initialize the child model with $M'_{L_m \to L_2}$;
23: **for** each sen-pair in $D_{L_1 \to L_2}$ **do**
24: Train the $M_{L_1 \to L_2}$ after fine-tuned by $M'_{L_m \to L_2}$;
25: **end for**
26: Translate the testset by fine-tuned $M_{L_1 \to L_2}$;

$D_{L_3 \to L_2}$ with oversampling of $D'_{L_1 \to L_2}$, and train the new hybrid model on the new dataset $D_{L_m \to L_2}$ with new shared vocabulary. Namely, the main difference between the shared and non-shared vocabulary is building the vocabulary on merged and without merged training data. However, the hybrid model was also initialized using the previously pre-trained parent model $M_{L_3 \to L_2}$ on $D'_{L_1 \to L_2} + D_{L_3 \to L_2}$. The parameters of the hybrid model would then conform to

$$\theta_{L_m \to L_2} = \{\langle e_{L_m}, W, e_{L_2} \rangle\} \tag{12}$$

where m stands for the hybrid model, with the source side embeddings e_{L_m} and the target side embeddings e_{L_2}. The standard training objective is to maximize the log-likelihood of the training data $D_{L_m \to L_2}$, as well as the combined dataset originating from the oversampled low-resource and high-resource datasets:

$$\hat{\theta}_{L_m \to L_2} = \underset{\theta_{L_m \to L_2}}{\text{argmax}} \{L(D_{L_m \to L_2}, \theta_{L_m \to L_2})\} \tag{13}$$

We then initialize the $M_{L_1 \to L_2}$ after it was using the hybrid model $M_{L_m \to L_2}$ on its own dataset, instead of oversampled the big data $D_{L_1 \to L_2}$.

$$\theta_{L_1 \to L_2} = f(\hat{\theta}_{L_m \to L_2}) \tag{14}$$

where f is the same initialization function as in the previous step, which helps the child model $M_{L_1 \to L_2}$ transfer the already learned parameters $\hat{\theta}_{L_m \to L_2}$ from the hybrid model $M_{L_m \to L_2}$.

4 Experiment

4.1 Setup

4.1.1 Data preparation

Both the parent and child languages are MRLs. The data were sourced from open source platforms, and all of the training corpora which were used in our experiment are publicly available on Open Subtitle2016*, Tanzil corpora†, and Chinese-LDC (CLDC) corpus‡. The previous two corpora are the free resources, we can use the academic research without any charge, but the last one is not free and we can buy this corpus from the given urls. The specifications of corpora are listed in Table 1. In the experiment, we set the target side to Chinese, while the source sides differed from each other

*http://opus.nlpl.eu/OpenSubtitles2016.php
†http://opus.nlpl.eu/Tanzil.php
‡http://www.chineseldc.org/resource˙info.php?rid=156

Table 1 Characteristics of our corpora. "Dev." indicates the development set, "Word Type" and "Word Token" indicate vocabulary (non-repeated) and all tokens (includes repeated), respectively. Here, K represents 10^3 and M represents 10^6.

Language	Train	Dev.	Test	Source		Target	
				Word type	Word token	Word type	Word token
Ar → Zh	5.1M	2.0K	2.0K	1.0M	32.2M	0.5M	37.4M
Fa → Zh	1.4M	2.0K	1.0K	0.2M	10.4M	0.2M	10.0M
Az → Zh	20.1K	0.5K	0.5K	25.1K	0.6M	0.2M	10.0M
Tr → Zh	4.4M	2.0K	1.0K	0.7M	30.6M	0.5M	35.9M
Ug → Zh	10.9K	0.5K	0.5K	18.3K	0.4M	12.5K	0.4M
Uz → Zh	10.5K	0.5K	0.5K	26.5K	0.5M	16.3K	0.3M

and included low-resource parent or child languages. The parent language pairs Arabic (Ar) → Zh, Farsi (Fa)→ Zh, and Turkish (Tr) → Zh were obtained from the Open Subtitle2016 corpora, while another parent pair, Uyghur (Ug) → Zh is gained from Chinese-LDC (CLDC), and the child pairs Azerbaijani (Az) → Zh and Uzbek (Uz) → Zh were originated from Tanzil corpora, same as the parent language Tr → Zh. We also preprocessed our datasets in this step using the preprocessing *script*[§] to clean up the data and to eliminate some invalid Chinese sentences from our target sides in the original corpus. We also provide several pre-processing *scripts* for both the source and the target side. Each of these scripts works for parent and child language pairs and the Chinese corpus.

Moreover, we filter some language pairs from both the parent and child after removing illegal symbols, double-checked non-Chinese characters, and designed a converter that interconverts simplified and traditional Chinese. Furthermore, we take advantage of the open source Chinese word stemmer system THULAC[¶] [33]. We exploit the tokenizer toolkit[‖][9] which was provided with the State-Of-The-Art (SOTA) phrase-based SMT system MOSES[9] for word tokenization. The results from all experiment have not used any UNK-replacement techniques[34]. We also leveraged the BPE[**] method[35] for both the source and target sides of the parent and child language pairs. Analogously, the latinization of our parent languages Ar[††], Fa[‡‡], and the child language Uz[§§] were conduct based on their corresponding comparison tables, publicly available on Wikipedia.

Additionally, we use the open-source toolkit THUMT[36] for the NMT system[¶¶], together with the TRANSFORMER architecture to train and evaluate all the models. The reason we used the TRANSFORMER is that the proposed HTL method is model transparent and it is the SOTA architecture in NMT, and we can also use any other NMT model architecture, even newer architectures. Likewise, we ran all the experiments (including fine-tuning) for less than 3 days on 4 GPUs (TITAN X) with default parameters of THUMT, whereby we slightly modified some of dimensions (see Table 2). While building multilingual models, we exploited oversampling and so that data from all language pairs will be the same size as that of the largest language pair, ensuring an equal amount of data per language pair. During the evaluation, we used the case-insensitive BLEU[***] scores[37]. We will release the corresponding components in the future, i.e., all the datasets after preprocessing and all the scripts used for preprocessing and romanization.

4.1.2 Baseline

It stands to reason to compare our proposed method with highly similar approaches. Therefore, we select similar approaches highly as follows:

- TRANSFORMER: The state-of-the-art NMT system which is introduced in Ref. [3]; in this work, we take it as strong baseline without transfer from any parent models.
- MANY-to-ONE: Training a many-to-one model by

Table 2 Hyper-parameter settings.

Parameter	Value	Parameter	Value
Iter time	200K	Word length	50.0
Word embedding	620	Beam size	4.0
Hidden state	1000	Dropout	0.1
Vocabulary size	30K	Learning rate	1.0
Batch size	80		

[§]http://www.nlplab.com/niuplan/NiuTrans.YourData.html
[¶]https://github.com/thunlp/THULAC-Python
[‖]https://github.com/moses-smt/mosesdecoder/tree/master/scripts/tokenizer/tokenizer.perl
[**]https://github.com/rsennrich/subword-nmt
[††]https://en.wikipedia.org/wiki/Romanization_of_Arabic
[‡‡]https://en.wikipedia.org/wiki/Romanization_of_Persian
[§§]https://en.wikipedia.org/wiki/Uzbek_alphabet
[¶¶]https://github.com/thumt/THUMT
[***]https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

taking advantage of several parent language datasets, which was proposed by Google[26]. In the training procedure they do not transfer from any parent models, however, their model is a trained multilingual model with many source languages to one target language.

• ORIGINAL-TL: A general TL method for low-resource NMT[12]. It trains a parent model without considering the lexicon embedding of the child language pair in training.

• REVISED-TL: This method also follows the key idea of ORIGINAL-TL. It is a revised TL method for LRLs NMT[21]. Precisely, it exploits the different domain data in the same parent languages by employing Ref. [12] without sharing lexicon.

• TRIVIAL-TL: The key idea of this approach also further validates the major idea of Ref. [12], and trains the parent languages number of iterations by using the TL in NMT between low-resource languages[23]. Exactly, this approach is also similar to TL proposed in Ref. [12], but uses the shared vocabulary whose main novelty is that they remove the restrictions of language relatedness.

In the zero-shot multilingual field, Google provides the MANY-TO-ONE approach which aims to train a multilingual model with several source languages to one target language. It works well with a zero-shot or few-shot approach. However, it ignores the lexicon features of the child languages. It focuses on leveraging various source languages without considering similar lexicon features among these languages, which are used in both training and testing. Reference [12] presented the initializing TL method, with the idea of choosing a bigger parent model and then training the model on selected parent languages using only parent vocabulary. By contrast, Ref. [21] also introduced the similar TL method, REVISED-TL, by following the work of ORIGINAL-TL. While we can take REVISED-TL as the revised version of ORIGINAL-TL. The main contributions of REVISED-TL are the substitution of word stem and double fine-tuning with various sizes of the same parent language pairs. Both of the TL methods (ORIGINAL-TL and REVISED-TL) have achieved good results, but both of them have their disadvantages: they neglect the shared lexicon between the parent pairs and child pairs.

We were also motivated by the work of ORIGINAL-TL and REVISED-TL, but we regard that no matter if initializing the child model directly with one parent language or with a different size corpus of the same parent language twice sequentially, most parent languages are HRLs, and the child languages are LRLs in which they are MLRs. Therefore, we can feed some lexicon features into training the parent model before initializing the child model, by sharing the vocabulary which was built from parent and child language pairs. If we design an intermediate model that was trained on a combined corpus with a shared vocabulary similar to the DA task, we can obtain more adaptable knowledge from out-domain to in-domain. We also hope that this approach will work well among adaptation tasks such as the transfer learning NMT from HRLs to LRLs.

4.2 Effect of shared lexicon to hybrid TL

As shown in Table 3, we compare our HTL method using two different modes: shared vocabulary and non-shared vocabulary. The parent languages Ar and Fa are similar to the child language Uz, while the parent language sets Tr_a (2.4M) and Tr_b (4.4M) whose corpus size is different but have the same language family, language group, and language branch with the child language Az, such as the same syntactic order "SOV", the same language unit "word", and the same language inflection "moderate". The various parents were chosen for the two-child languages because only Tr has two different sources in our dataset, and we followed the idea of REVISED-TL choosing Tr_a and Tr_b as parents for Az. Furthermore, we pre-trained the HRL parent models on an own dataset (see Fig. 1), and then created the combined corpus and built the vocabulary on a combined corpus. Finally, we fine-tune the child models using pre-trained hybrid models. Obviously, the HTL with

Table 3 Effect of shared lexicon to Hybrid TL. Tr_a and Tr_b represent the different corpus size of 2.4M and 4.4M for Tr. "shared" and "Non-shared" denote with and without shared lexicon. "++" indicates a significantly better than TRANSFORMER ($p < 0.01$).

Method	Parent	Child	BLEU
TRANSFORMER[3]	N/A	Az → Zh	43.68
		Uz → Zh	40.99
HYBRID-TL$_{\text{Non-shared}}$	Tr_a → Zh	Az → Zh	45.97^{++}
	Fa → Zh	Uz → Zh	42.15^{++}
	Tr_b → Zh	Az → Zh	46.81^{++}
	Ar → Zh	Uz → Zh	42.64^{++}
HYBRID-TL$_{\text{shared}}$	Tr_a → Zh	Az → Zh	$\mathbf{46.44}^{++}$
	Fa → Zh	Uz → Zh	$\mathbf{42.53}^{++}$
	Tr_b → Zh	Az → Zh	$\mathbf{47.32}^{++}$
	Ar → Zh	Uz → Zh	$\mathbf{42.89}^{++}$

the non-shared or the shared hybrid model can help the child models gain significantly better improvements than TRANSFORMER ($p < 0.01$).

4.3 Effect of original TL model to hybrid TL

We further confirm the shared hybrid model is better than the non-shared hybrid model. Therefore, we choose the shared hybrid model as our HTL model in the following experiments. As shown in Table 4, the performance of HTL is better than TRANSFORMER and ORIGINAL-TL. Our models $M_{Tr_a \to Ch}$ and $M_{Tr_b \to Ch}$ with shared vocabulary yield better results than both the baselines of TRANSFORMER ($p < 0.01$) and ORIGINAL-TL($p < 0.05$) on the child language pairs Az $\to$ Zh. While, the models $M_{Ar \to Zh}$ and $M_{Fa \to Ch}$ yield better results than TRANSFORMER ($p < 0.01$). We also investigate some interesting settings, such as transliterate both the child pair Uz $\to$ Zh and the parent pairs Ar $\to$ Zh and Fa $\to$ Zh into the roman script before training all the models. Namely, before training the original pre-training model $M_{Ar \to Zh}$ (with its vocabularies) or training the hybrid pre-training model (with a shared vocabulary) on a combined corpus, we convert both the two-parent languages and the child language into a roman version. Then we repeated the previous steps entirely based on Algorithm 1. We found that the HTL with roman versions and use of shared vocabularies achieve a better result than without romanization.

4.4 Effect of revised TL model to hybrid TL

As shown in Table 5, based on the main idea of REVISED-TL, we try to exploit our HTL method in different manners with shared vocabularies and roman shape to train various models. Furthermore, we explore whether HTL can enable the performance of child models to outperform the baselines of TRANSFORMER and REVISED-TL. Clearly, our HTL method yields significantly better results than baseline systems. Both models $M_{Tr_a \to Zh}$ and $M_{Tr_b \to Zh}$ have been trained with REVISED-TL yielded results better than TRANSFORMER. Contrarily, We also ran the model repeatedly and the HTL also obtain even significantly better results than both the baselines TRANSFORMER ($p < 0.01$) and REVISED-TL ($p < 0.05$). Besides, we also explore the performance of the model training on a combined roman corpus for the $M_{Uz \to Zh}$. Since, as given in Table 4, we have not converted the Az $\to$ Zh into Latin script, the results from HTL with or without roman shape were identical.

Table 4 Effect of the original TL to hybrid mode. The subscript "shared+latin" represents exploit shared lexicon and roman form. "*" significantly better than original TL ($p < 0.05$).

Method	Parent	Child	BLEU
ORIGINAL-TL[12]	$Tr_a \to Zh$	Az $\to$ Zh	45.82^{++}
	Fa $\to$ Zh	Uz $\to$ Zh	42.03^{++}
	$Tr_b \to Zh$	Az $\to$ Zh	46.49^{++}
	Ar $\to$ Zh	Uz $\to$ Zh	42.51^{++}
HYBRID-TL_{shared}	$Tr_a \to Zh$	Az $\to$ Zh	$\mathbf{46.44}^{++*}$
	Fa $\to$ Zh	Uz $\to$ Zh	42.53^{++}
	$Tr_b \to Zh$	Az $\to$ Zh	$\mathbf{47.32}^{++*}$
	Ar $\to$ Zh	Uz $\to$ Zh	42.89^{++}
HYBRID-$TL_{shared+latin}$	$Tr_a \to Zh$	Az $\to$ Zh	46.44^{++*}
	Fa $\to$ Zh	Uz $\to$ Zh	$\mathbf{42.82}^{++*}$
	$Tr_b \to Zh$	Az $\to$ Zh	47.32^{++*}
	Ar $\to$ Zh	Uz $\to$ Zh	$\mathbf{43.11}^{++*}$

Table 5 Effect of the revised TL to hybrid model. "†" indicates a significantly better result than revised TL ($p < 0.05$).

Method	Parent	Child	BLEU
REVISED-TL[21]	$Tr_{b,a} \to Zh$	Az $\to$ Zh	47.55^{++}
		Uz $\to$ Zh	44.21^{++}
HYBRID-TL_{shared}	$Tr_{b,a} \to Zh$	Az $\to$ Zh	$\mathbf{48.62}^{++}$
		Uz $\to$ Zh	$\mathbf{45.41}^{++}$
HYBRID-$TL_{shared+latin}$		Az $\to$ Zh	$48.62^{++\dagger}$
		Uz $\to$ Zh	$\mathbf{45.83}^{++\dagger}$

4.5 Comparison of different TL methods

In the LRLs MT community, several methods have been proposed, and some baseline studies share significant similarities with this work. As shown in Table 6, our baselines TRANSFORMER, MANY-to-ONE, ORIGINAL-TL, REVISED-TL, and REVISED-TL obtained remarkably better results. Therefore, in this experiment, we further confirm their qualities and explore the effectiveness of our HTL method on child pairs Az$\to$Zh and Uz$\to$Zh. Precisely, the model trained on the combined corpus using the oversampling approach also yields better results than TRANSFORMER ($p < 0.01$ and $p < 0.05$) and ORIGINAL-TL ($p < 0.05$). The strategy for building a combined corpus of MANY-to-ONE is somewhat similar to our work, but it trains the parent model on the combined corpus first using the oversampling method, and then directly initializes the target languages without considering relatedness or lexicon features between the parent and child pairs. The model $M_{Tr_{b,a} \to Ch}$ significantly outperformed both the two baselines of TRANSFORMER and ORIGINAL-TL with $p < 0.01$ on the two models $M_{Az \to Zh}$ and

Table 6 Comparison of different TL methods. "⋆": significantly better than Ref. [12] ($p < 0.01$), "⋄" and "‡": significantly better than Ref. [21] with $p<0.05$ and $p < 0.01$, respectively.

Method	Parent	Child	BLEU
TRANSFORMER[3]	N/A	Az → Zh	43.68
		Uz → Zh	40.99
MANY-to-ONE[26]		Az → Zh	46.74^{++}
		Uz → Zh	43.67^{++*}
ORIGINAL-TL[12]	Tr_a → Zh	Az → Zh	45.82^{++}
	Fa → Zh	Uz → Zh	42.03^{++}
	Tr_b → Zh	Az → Zh	46.49^{++}
	Ar → Zh	Uz → Zh	42.51^{++}
REVISED-TL[21]	$Tr_{b,a}$ → Zh	Az → Zh	$47.55^{++\star\diamond}$
		Uz → Zh	$44.21^{++\star\diamond}$
TRIVIAL-TL[23]	$Tr_{b,a}$ → Zh	Az → Zh	$47.91^{++\star\diamond}$
		Uz → Zh	$44.73^{++\star\diamond}$
HYBRID-TL_{shared}	$Tr_{b,a}$ → Zh	Az → Zh	$\mathbf{48.62}^{++\star\ddagger\dagger}$
		Uz → Zh	$45.41^{++\star\ddagger}$
HYBRID-$TL_{shared+latin}$		Az → Zh	$48.62^{++\star\ddagger\dagger}$
		Uz → Zh	$\mathbf{45.83}^{++\star\ddagger\dagger}$

$M_{Uz\to Zh}$. Likewise, it is better than another baseline MANY-to-ONE with $p < 0.05$. We also investigate the effectiveness of HTL in a different manner, by training the combined model with or without a roman shape.

Moreover, our HTL method with a romanized model also significantly outperforms all the five baselines. Obviously, the motivation for developing our HTL method with or without training on a combined roman corpus is to share more lexicon features between hybrid parent models and child models, the improvements on child model $M_{Uz\to Zh}$ with the hybrid model "HYBRID-$TL_{shared+latin}$" yield better results than the hybrid model "HYBRID-TL_{shared}". Overall, the proposed HTL method, no matter if trained in the same manner with or without the roman shape form of the combined corpus, achieves comparable and significantly improved results compared to all five baselines on the two-child low-resource MRLs Az and Uz. Consequently, this experiment also further validates, and the results demonstrate that HTL is effective in MRLs NMT.

4.6 Discoveries related to the child models of HTL

Unexpectedly, as given in Table 7, we find some interesting discoveries by switching the position of parent and child models. The corpus size of the child language pair Uz→Zh is 10.5K, and we select another smaller child pair (Ug→Zh) which has a similar corpus size (10.9K), it also stems from Tanzil corpora. They have sane language features and same as Uz→Zh. Both

Table 7 Discovery on child models. "⋆" denotes significantly better than Ref. [3] ($p < 0.05$).

Method	Parent	Child	BLEU
TRANSFORMER[3]	N/A	Az → Zh	43.68
		Ug → Zh	21.70
		Uz → Zh	40.99
HYBRID-TL_{mutual}	Uz → Zh	Ug → Zh	$22.12^{\star}$
	Ug → Zh	Uz → Zh	$41.68^{\star}$
HYBRID-$TL_{inverted}$	Uz → Zh	Az → Zh	$44.04^{\star}$

Ug→Zh and Uz→Zh belong to the same language family, group, and branch, so that both of them can improve quality by mutual initializing (see Fig. 3a), we take this as MUTUAL TRANSFER. This demonstrates that LRLs can also improve each other in the same domain, and with similar corpus size.

Besides, we discover that low-resource language pairs can also be used to improve the quality of HRL models (see Fig. 3b). Here, the corpus size of the parent pair Uz → Zh is smaller than that of the child pair Az → Zh (20.1K), but it was able to improve the quality of the Az → Zh. This shows that even a smaller model can also be used to improve the quality of bigger child models. We regard this as INVERTED TRANSFER. As long as low-resource languages share similar syntactic and semantic features, share many common words, and belong to the same domain. Smaller parent model also helps the child model, which is trained on even bigger corpus size than the parent model trained on smaller corpus size.

5 Case Study

As given in Table 8, we can illustrate how the proposed HTL method improves translation quality by leveraging an example. The result obtained from the SOTA baseline TRANSFORMER merely translated a few words from the source sequence and skipped several words. At the beginning of first sub-sentence, the word "fanzuizhe" (perpetrators) translated into "fanzui" (offend), the previous one is a name and the next one is the verb. Besides, the pronoun "tamen" (their), the name "e'fa" (forelock), and the adverb "jiang" (in the future) are dropped. The second baseline MANY-to-ONE obtained better result, but still dropped the word "yin" (because) and "bei" (by) in the sub-sentence. Besides, it translated the word "xingji" (trail) into "xingwei" (action) and the clause "bei renshi" (be known) into "bei xielou" (be revealed).

ORIGINAL-TL also obtains even better result, but still exists some errors that dropped the word "yin" (because) and "bei" (by). Besides, translate "e'fa" (forelock) into

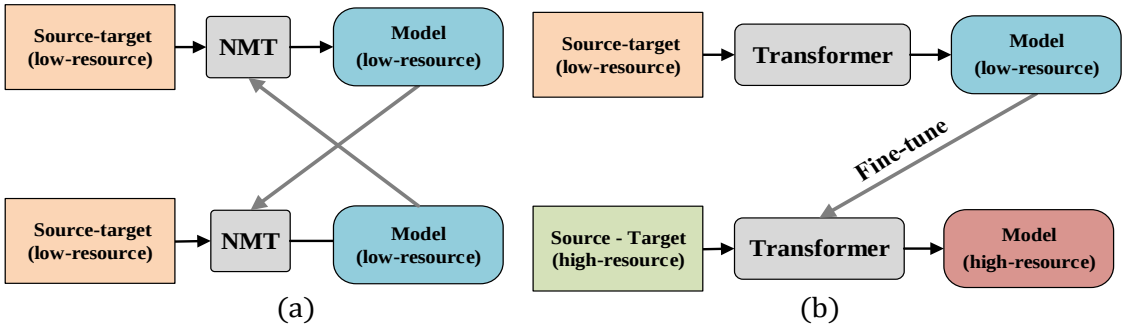

Fig. 3 (a) MUTUAL TRANSFER; (b) INVERTED TRANSFER.

Table 8 Translation example of Uz → Zh between various methods, while the source sentence was romanized.

Method	Translation result
Source	*jinayot@@?mlar siymalaridan bilinib turur. va ularning pe?ana sa?lari va ayoqlaridan tutilur.*
Reference	*fanzuizhe jiang yin* ***tamen*** *de xingji er bei renshi , tamen de* ***e'fa*** *jiang bei* ***ji*** *zai jiaozhang shang .* 犯罪者将因他们的形迹而被认识，他们的额发将被系在脚掌上。
TRANSFORMER[3]	*fanzui de* ***xingwei*** *bei renshi,* ***tamen*** *de kun zai jiao* ***shang****.* 犯罪的行为被认识，他们的困在脚上。
MANY-to-ONE[26]	*fanren de* ***xingwei*** *jianglai bei xielou,* ***tamen*** *tou jianglai bang zai jiao* ***shang****.* 犯人的行为将来被泄漏，他们头将来绑在脚上。
ORIGINAL-TL[12]	*fanzuiren* ***jianglai*** *yinji bei renshi,* ***jianglai*** *tamen tou bang zai tui* ***shang****.* 犯罪人将来印迹会被认识，将来他们头绑在腿上。
REVISED-TL[21]	***jianglai*** *fanzuizhe ta de xingji* ***bei*** *renshi chulai, tamen* ***toufa*** *ji zai jiaozhang* ***shang****.* 将来犯罪者他的形迹被认识出来，他们头发系在脚掌上。
TRIVIAL-TL[23]	*fanzuizhe* ***jianglai*** *ta de xingji* ***bei*** *renshi chu, tamen* ***toufa*** *bang zai jiao* ***shang****.* 犯罪者将来他的形迹被认识出，他们头发绑在脚上。
HYBRID-TL$_{\text{shared}}$	***jianglai*** *fanzuizhe tamen de xingji* ***bei*** *renshi chu, tamen* ***e'fa*** *jiang ji zai jiaozhang* ***shang****.* 将来犯罪者他们的形迹被认识出，他们额发将系在脚掌上。
HYBRID-TL$_{\text{shared+latin}}$	***jianglai*** *yinwei fanzuizhe tamen de xingji* ***bei*** *renshi, tamen* ***e'fa*** *jianglai ji zai jiaozhang* ***shang****.* 将来因为犯罪者他们的形迹被认识，他们额发将来系在脚掌上。

"tou" (head). REVISED-TL attained more reasonable results except for dropping the "yin" (because) and "jiang" (in the future) in first and second sub-sentence. By contrast, HTL$_{\text{sh}}$ also gained even better and similar result to reference, but just dropped the word "yin" (because). HTL$_{\text{sh+lt}}$ achieved highly analogous result to ground truth, but just translated "yin" into "yinwei" (because) and "jiang" into "jianglai" (in the future).

6 Related Work

As an essential potential strategy for overcoming the great problem posed by the shortage of large-scale parallel corpora, NMT has gained increasing attention in the machine translation community[3, 12, 35, 36, 38–40]. Several methods have been presented that focus on handling the deficiency of a parallel training corpora problem. Most of the present literatures on low-resource NMT can be classified into transfer learning[12, 30, 41], domain adaptation[42], pivot learning[20], and zero-shot learning[17]. To take the advantages of HTL into account, it is essential to consider rare studies that we investigate the NMT of LRLs by leveraging shared lexicon features or shared vocabulary between TL. To some extent, HTL differs from others, and it uses vocabulary sharing between the parent model and child model before fine-tuning the MRLs and child model. The NMT systems have been developed rapidly in recent years. Reference [43] improved the performance of the NMT using word level domain context in the multi-domain scenario. Besides, Ref. [44] proposed the iterative dual training method for domain adaptation task in NMT. Moreover, Ref. [45] commited to distinguishing and exploiting different word-level domain contexts for multi-domain NMT, and enriched the NMT model performance by adopting multi-task learning to jointly model NMT and monolingual attention-based domain classification tasks. In the past two years, many researchers proposed new approaches[46] in the NMT community for LRLs.

Reference [47] introduced the meta-learning for low-resource NMT and achieved remarkably better result. Reference [48] also took advantage of ORIGINAL-TL to fine-tune the child model $M_{Uy\rightarrow En}$ with $M_{Uz\rightarrow En}$ (Uzbek–English), which is similar to the ORIGINAL-TL used in the NMT of LRLs. The main research direction is to exploit a parent model similar to the child language. Furthermore, some of the indispensable factors of NN, as well as some parameters of the encoder, decoder, and attention were transferred from the parent model to the child language pairs[49].

We ponder over the key idea to share some parameters and share some features between the parent models and the child model. The main differences from existing literature are (1) we train the parent model on training dataset with its vocabularies; (2) we compare the parent and child language pairs, and we made the child language training corpus the same size as those of the parent language pairs using the oversampling method; (3) we combine the original parent and expended child language pairs to create a combined corpus, then train the hybrid model using the shared vocabularies after using the previously trained parent model; (4) we initialize the child model with a fine-tuned and trained hybrid model, which is derived from the previous step. One aim of this work was to make the parent model more efficient. Another aspect of our work focuses on how to make the child model learn more parameters and obtain more lexicon features from the parent model. Although it was not our direct aim, the method proposed in this work can be seen as a revised version of REVISED-TL.

Since we also train two models differently to that of Ref. [21]. By contrast, we do not use the same language with various corpus sizes, and instead used our hybrid model that was fine-tuned by the previous model with shared vocabularies and trained on the combined corpus. We do not need to consider stem substitution. However, if the parent and child language pairs do not belong to the same language branch, this does not work well as we expected. Therefore, no matter if the parent and child language pairs belong to the same language family, group, and branch, as long as we can find some relatedness (syntactically and semantically), our training method may make the performance of the child model better than others. Moreover, Refs. [12, 21, 50] ignored shared lexicon features of parent models to child models. In this paper, we further validate the proposed approach on two morphologically rich LRLs Az and Uz and achieve a better result than previous studies.

7 Conclusion and Future Perspective

In this paper, we introduce a rather straightforward and effective method that can feed certain lexicon features into the TL via shared vocabulary. First, We propose the HTL method for low-resource languages. Then, with the intent to make the child model obtain more lexicon features from the parent model, we design the hybrid transfer mode to be used before fine-tuning the child model. It should be noted that we leveraged the oversampling method, and made the child corpus size equal to parent corpus size, then shuffled them and created the bigger mixed corpus, rather than directly mixed the parent and child language pairs. Besides, we find sometimes the low-resource child languages can also help the bigger corpus size language pairs to achieve a better performance. We named this discovery INVERTED TRANSFER. Furthermore, another interesting finding is that both the child and parent models gain a better generalization when we apply mutual fine-tuning, sometimes it works unexpectedly well. We named this finding MUTUAL TRANSFER.

The HTL is transparent to network architectures and also can be used in other NLP tasks or in the field of computer vision. Likewise, we also leveraged this method on another child language pair, which demonstrated that our training approach is language independent. At the same time, we address the disadvantage of TL that it only exploits the original parent to fine-tune the child model without considering the lexicon features shared between the parent and child models. By contrast, our HTL method can cope with this drawback adequately. In future work, we aim to further verify the effectiveness of HTL on many NLP tasks and try to apply it to morphologically poor languages. Furthermore, it is also useful to apply our approach in Pos tagging, sentiment analysis, and domain adaptation tasks.

Acknowledgment

We would like to thank all anonymous reviewers for their valuable comments and suggestions during both the major revision and minor revision for this work. Besides, we also would like to thank Dr. Ivan Hajnal and Leah who is the assistant researcher at multi-lingual team of DAMO academy in Alibaba Group, for supporting proofread our revised paper after major and minor revision

patiently. This work was supported by the National Key R&D Program of China (No. 2017YFB0202204), the National Natural Science Foundation of China (Nos. 61925601, 61761166008, and 61772302), Beijing Advanced Innovation Center for Language Resources (No. TYR17002), and the NExT++ project which supported by the National Research Foundation, Prime Ministers Office, Singapore under its IRC@Singapore Funding Initiative.

References

[1] I. Sutskever, O. Vinyals, and V. Le Quoc, Sequence to sequence learning with neural networks, in *Proc. 27^{th} Int. Conf. Neural Information Processing Systems*, Cambridge, MA, USA, 2014, pp. 3104–3112.

[2] D. Bahdanau, K. Cho, and Y. Bengio, Neural machine translation by jointly learning to align and translate, in *Proc. 3^{rd} Int. Conf. Learning Representations*, San Diego, CA, USA, 2015.

[3] 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. 31^{st} Int. Conf. Neural Information Processing Systems*, Red Hook, NY, USA, 2017, pp. 6000–6010.

[4] Y. H. Wu, M. Schuster, Z. F. Chen, V. Le Quoc, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al., Google's neural machine translation system: Bridging the gap between human and machine translation, arXiv preprint arXiv: 1609.08144, 2016.

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

[6] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, in *Proc. 2014 Conf. Empirical Methods in Natural Language Processing*, Doha, Qatar, 2014, pp. 1724–1734.

[7] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv: 1810.04805, 2018.

[8] P. F. Brown, V. J. Della Pietra, S. A. Della Pietra, and R. L. Mercer, The mathematics of statistical machine translation: Parameter estimation, *Computational Linguistics*, vol. 19, no. 2, pp. 263–311, 1993.

[9] P. Koehn, F. J. Och, and D. Marcu, Statistical phrase-based translation, in *Proc. 2003 Conf. North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Stroudsburg, PA, USA, 2003, pp. 48–54.

[10] D. Chiang, A hierarchical phrase-based model for statistical machine translation, in *Proc. 43^{rd} Annu. Meeting on Association for Computational Linguistics*, Stroudsburg, PA, USA, 2005, pp. 263–270.

[11] M. Junczys-Dowmunt, T. Dwojak, and H. Hoang, Is neural machine translation ready for deployment? A case study on 30 translation directions, arXiv preprint arXiv: 1610.01108, 2016.

[12] B. Zoph, D. Yuret, J. May, and K. Knight, Transfer learning for low-resource neural machine translation, in *Proc. 2016 Conf. Empirical Methods in Natural Language Processing*, Austin, Texas, 2016, pp. 1568–1575.

[13] C. Vania and A. Lopez, From characters to words to in between: Do we capture morphology? in *Proc. 55^{th} Annu. Meeting of the Association for Computational Linguistics*, Vancouver, Canada, 2017, pp. 2016–2027.

[14] Y. Chen, Y. Liu, and V. O. K. Li, Zero-resource neural machine translation with multi-agent communication game, arXiv preprint arXiv: 1802.03116, 2018.

[15] A. Karakanta, J. Dehdari, and J. van Genabith, Neural machine translation for low-resource languages without parallel corpora, *Machine Translation*, vol. 32, nos. 1&2, pp. 167–189, 2018.

[16] P. Koehn, EuroParl: A parallel corpus for statistical machine translation, http://homepages.inf.ed.ac.uk/pkoehn/publications/europarl-mtsummit05.pdf, 2005.

[17] Y. Chen, Y. Liu, Y. Cheng, and V. O. L. Li, A teacher-student framework for zero-resource neural machine translation, in *Proc. 55^{th} Annu. Meeting of the Association for Computational Linguistics*, Vancouver, Canada, 2017, pp. 1925–1935.

[18] H. Zheng, Y. Cheng, and Y. Liu, Maximum expected likelihood estimation for zero-resource neural machine translation, in *Proc. 26^{th} Int. Joint Conf. Artificial Intelligence*, Melbourne, Australia, 2017, pp. 4251–4257.

[19] Y. Cheng, W. Xu, Z. J. He, W. He, H. Wu, M. S. Sun, and Y. Liu, Semi-supervised learning for neural machine translation, in *Proc. 54^{th} Annu. Meeting of the Association for Computational Linguistics*, Berlin, Germany, 2016, pp. 1965–1974.

[20] Y. Cheng, Q. Yang, Y. Liu, M. S. Sun, and W. Xu, Joint training for pivot-based neural machine translation, in *Proc. 26^{th} Int. Joint Conf. Artificial Intelligence*, Melbourne, Australia, 2017, pp. 3974–3980.

[21] P. Passban, Q. Liu, and A. Way, Translating low-resource languages by vocabulary adaptation from close counterparts, *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 16, no. 4, p. 29, 2017.

[22] M. Maimaiti and X. H. Zou, Discussion on bilingual cognition in international exchange activities, presented at International Conference on Intelligence Science (ICIS2018), Beijing, China, 2018, pp. 167–177.

[23] T. Kocmi and O. Bojar, Trivial transfer learning for low-resource neural machine translation, in *Proc. 3^{rd} Conf. Machine Translation: Research Papers*, Brussels, Belgium, 2018, pp. 244–252.

[24] R. Dabre, T. Nakagawa, and H. Kazawa, An empirical study of language relatedness for transfer learning in neural machine translation, https://www.aclweb.org/anthology/Y17-1038.pdf, 2017.

[25] Y. Kim, Y. B. Gao, and H. Ney, Effective cross-lingual transfer of neural machine translation models without shared vocabularies, in *Proc. 57^{th} Annu. Meeting of the Association for Computational Linguistics*, Florence, Italy, 2019, pp. 1246–1257.

[26] M. Johnson, M. Schuster, V. Le Quoc, M. Krikun, Y. H. Wu, Z. F. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, et al., Google's multilingual neural machine translation system: Enabling zero-shot translation, *Transactions of the Association of Computational Linguistics*, vol. 5, pp. 339–351, 2017.

[27] S. J. Pan and Q. Yang, A survey on transfer learning, *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[28] S. Hong, J. Oh, H. Lee, and B. Han, Learning transferrable knowledge for semantic segmentation with deep convolutional neural network, presented 2016 IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 3204–3212.

[29] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, Learning and transferring mid-level image representations using convolutional neural networks, presented at 2014 IEEE Conf. Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014, pp. 1717–1724.

[30] B. Tan, Y. Zhang, S. J. Pan, and Q. Yang, Distant domain transfer learning, in *Proc. 31^{st} AAAI Conf. Artificial Intelligence*, San Francisco, CA, USA, 2017, pp. 2604–2610.

[31] K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun, Deep residual learning for image recognition, in *Proc. 2016 IEEE Conf. Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016, pp. 770–778.

[32] J. L. Ba, J. R. Kiros, and G. E. Hinton, Layer normalization, arXiv preprint arXiv: 1607.06450, 2016.

[33] Z. G. Li and M. S. Sun, Punctuation as implicit annotations for Chinese word segmentation, *Computational Linguistics*, vol. 35, no. 4, pp. 505–512, 2009.

[34] T. Luong, I. Sutskever, V. Le Quoc, O. Vinyals, and W. Zaremba, Addressing the rare word problem in neural machine translation, in *Proc. 53^{rd} Annu. Meeting of the Association for Computational Linguistics and the 7^{th} Int. Joint Conf. Natural Language Processing*, Beijing, China, 2015, pp. 11–19.

[35] R. Sennrich, B. Haddow, and A. Birch, Neural machine translation of rare words with subword units, arXiv preprint arXiv: 1508.07909, 2016.

[36] J. C. Zhang, Y. Z. Ding, S. Q. Shen, Y. Cheng, M. S. Sun, H. B. Luan, and Y. Liu, THUMT: An open source toolkit for neural machine translation, arXiv preprint arXiv: 1706.06415, 2017.

[37] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, Bleu: A method for automatic evaluation of machine translation, in *Proc. 40^{th} Annu. Meeting of the Association for Computational Linguistics*, Philadelphia, PA, USA, 2002, pp. 311–318.

[38] D. X. Dong, H. Wu, W. He, D. H. Yu, and H. F. Wang, Multi-task learning for multiple language translation, in *Proc. 53^{rd} Annu. Meeting of the Association for Computational Linguistics and the 7^{th} Int. Joint Conf. Natural Language Processing*, Beijing, China, 2015, pp. 1723–1732.

[39] O. Firat, B. Sankaran, Y. Al-Onaizan, F. T. Y. Vural, and K. Cho, Zero-resource translation with multi-lingual neural machine translation, in *Proc. 2016 Conf. Empirical Methods in Natural Language Processing*, Austin, TX, USA, 2016, pp. 268–277.

[40] O. Firat, K. Cho, and Y. Bengio, Multi-way, multilingual neural machine translation with a shared attention mechanism, in *Proc. 2016 Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, CA, USA, 2016, pp. 866–875.

[41] B. Zoph, V. Vasudevan, J. Shlens, and V. Le Quoc, Learning transferable architectures for scalable image recognition, presented at 2018 IEEE/CVF Conf. Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 8697–8710.

[42] C. H. Chu, R. Dabre, and S. Kurohashi, An empirical comparison of simple domain adaptation methods for neural machine translation, arXiv preprint arXiv: 1701.03214, 2017.

[43] J. L. Zeng, J. S. Su, H. T. Wen, Y. Liu, J. Xie, Y. J. Yin, and J. Q. Zhao, Multi-domain neural machine translation with word-level domain context discrimination, in *Proc. 2018 Conf. Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018, pp. 447–457.

[44] J. L. Zeng, Y. Liu, J. S. Su, Y. B. Ge, Y. J. Lu, Y. J. Yin, and J. B. Luo, Iterative dual domain adaptation for neural machine translation, in *Proc. 2019 Conf. Empirical Methods in Natural Language Processing and the 9^{th} Int. Joint Conf. Natural Language Processing*, Hong Kong, China, 2019, pp. 845–855.

[45] J. S. Su, J. L. Zeng, J. Xie, H. T. Wen, Y. J. Yin, and Y. Liu, Exploring discriminative word-level domain contexts for multi-domain neural machine translation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, doi: 10.1109/TPAMI.2019.2954406.

[46] F. Marzieh, A. Bisazza, and C. Monz, Data augmentation for low-resource neural machine translation, in *Proc. 55^{th} Annu. Meeting of the Association for Computational Linguistics*, Vancouver, Canada, 2017, pp. 567–573.

[47] J. T. Gu, Y. Wang, Y. Chen, V. O. K. Li, and K. Cho, Meta-learning for low-resource neural machine translation, in *Proc. 2018 Conf. Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018, pp. 3622–3631.

[48] H. Setiawan, Z. Q. Huang, and R. Zbib, BBN's low-resource machine translation for the LoReHLT 2016 evaluation, *Machine Translation*, vol. 32, no. 1, pp. 45–57, 2018.

[49] M. Maimaiti, Y. Liu, H. B. Luan, and M. S. Sun, Multi-round transfer learning for low-resource NMT using multiple high-resource languages, *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 18, no. 4, p. 38, 2019.

[50] T. Q. Nguyen and D. Chiang, Transfer learning across low-resource, related languages for neural machine translation, in *Proc. 8^{th} Int. Joint Conf. Natural Language Processing*, Taipei, China, 2017. pp. 296–301.

Mieradilijiang Maimaiti is a PhD candidate at the Department of Computer Science and Technology, Tsinghua University, China. He received the BS and MS degrees from Xinjiang University, China in 2012 and 2015, respectively. He is broadly interested in machine learning and natural language processing, especially neural machine translation for low-resource languages and multi-lingual processing.

Yang Liu is a professor at the Department of Computer Science and Technology, Tsinghua University, China. He received the PhD degree from Chinese Academy of Sciences, China in 2007. His research interests include natural language processing and machine translation.

Huanbo Luan is the deputy executive director of NExT Search Center at both Tsinghua University, China and National University of Singapore, Singapore. He received the PhD degree in computer science from Chinese Academy of Sciences, China in 2008. His research interests include multimedia information retrieval, social media, and big data analysis.

Maosong Sun is a professor at the Department of Computer Science and Technology, Tsinghua University, Beijing. He received the PhD degree in computational linguistics from City University of Hong Kong, Hong Kong, China in 2004. His research interests include natural language processing, Web intelligence, and machine learning.