Source-Critical Reinforcement Learning for Transferring Spoken Language Understanding to a New Language

He Bai[1,2], Yu Zhou[1,2], Jiajun Zhang[1,2], Liang Zhao[3], Mei-Yuh Hwang[3] and Chengqing Zong[1,2,4]
[1] National Laboratory of Pattern Recognition, Institute of Automation, CAS, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing, China
[3] Mobvoi AI Lab, Seattle, US
[4] CAS Center for Excellence in Brain Science and Intelligence Technology
{he.bai, yzhou, jjzhang, cqzong}@nlpr.ia.ac.cn, {liangzhao, mhwang}@mobvoi.com

Abstract

To deploy a spoken language understanding (SLU) model to a new language, language transferring is desired to avoid the trouble of acquiring and labeling a new big SLU corpus. Translating the original SLU corpus into the target language is an attractive strategy. However, SLU corpora consist of plenty of semantic labels (slots), which general-purpose translators cannot handle well, not to mention additional culture differences. This paper focuses on the language transferring task given a tiny in-domain parallel SLU corpus. The in-domain parallel corpus can be used as the first adaptation on the general translator. But more importantly, we show how to use reinforcement learning (RL) to further finetune the adapted translator, where translated sentences with more proper slot tags receive higher rewards. We evaluate our approach on Chinese to English language transferring for SLU systems. The experimental results show that the generated English SLU corpus via adaptation and reinforcement learning gives us over 97% in the slot F1 score and over 84% accuracy in domain classification. It demonstrates the effectiveness of the proposed language transferring method. Compared with naive translation, our proposed method improves domain classification accuracy by relatively 22%, and the slot filling F1 score by relatively more than 71%.

1 Introduction

Spoken language understanding (SLU) is a key technique in today's conversational systems such as Apple Siri, Amazon Alexa and Microsoft Cortana. To make these conversational systems support multiple languages over different markets, collecting and annotating a large SLU training corpus per language are tedious and costly, and thus hinders the scalability of these systems. It would be greatly helpful if the efforts taken to develop one SLU system could be reused for other languages.

For such a purpose, much work has been reported to explore language transferring of SLU systems or multilingual SLU systems (García et al., 2012; Calvo et al., 2013; Calvo et al., 2016; Jabaian et al., 2016). These systems can be grouped into two categories: test-on-source-model vs. train-on-target-language. Test-on-source-model is to translate the test sentence in the 2nd language (referred to as L2 from now on) into the first SLU system's language (as L1), and then process it with the L1 SLU system. Train-on-target-language is to translate the L1 training corpus into L2, and then train an SLU system in L2. The train-on-target-language strategy allows tuning and adaptation of the models in the target language directly, and it avoids an overhead of machine translation during real-time execution. The test-on-source-model implies that the final search engine has to deal with L1 while the target answer database might be still in L2. Either the slot value needs to be translated again back to L2, or the database needs to be pre-translated into L1. Hence train-on-target-language is our preferred approach.

Each sample of the SLU training corpus consists of a query and its semantic annotation, for example,"Play <song>Sorry</song> in <album>They Don't Know</album>". In the L1 SLU training corpus, both query and its semantic annotation need to be transferred properly to the target language. Literally sending the annotated sentence to general-purpose machine translator or web translator may

Proceedings of the 27th International Conference on Computational Linguistics, pages 3597–3607
Santa Fe, New Mexico, USA, August 20-26, 2018.

lose or screw up the annotation information. Two schemes (Jabaian et al., 2013) have been proposed to solve this problem: transferring source language annotation indirectly through word alignment, vs. adapting the translation model so that it learns how to translate text with slot labels. Each scheme has its own limitations. The first scheme is weak for distant language pairs (e.g. Chinese and English) and the errors are accumulative after translation and word alignment. The second scheme needs extra data for adapting the translation model. In addition, some previous works (Servan et al., 2010; Misu et al., 2012; Jabaian et al., 2013) only focused on transferring slot labels but ignored the fact that the slot values should be adjusted properly to the target culture. For example, people in London will probably say "call a taxi to Tower of London" rather than "call a taxi to Forbidden City". It's better to make some adaptations on such culture difference for the L2 SLU corpus. In this paper, we will address culture adaptation when we transfer Chinese corpora to English and we will demonstrate its importance via experimental results. Given that neural machine translation (NMT) is the state-of-the-art translator, NMT model is applied in this paper for language transferring.

The main contribution of this paper is the proposal of using reinforcement learning to improve translation for language transferring. We name it source-critical reinforcement translation (SCRT). We first define the slot keeping ratio (SKR) as a metric for evaluating the performance of slot transferring. SCRT adapts NMT models via rewarding those translation candidates with higher SKR. By doing so, we can obtain target language translations which maintain both the semantics and slot information of the SLU labeled sentences in the source language. We evaluate our method on domain classification and slot filling tasks. The results show that our RL method improves the slot F1 score from 93% to 97% and domain accuracy from 82% to 84%, on top of an already adapted NMT.

2 Related Work

Language transferring for SLU systems has been an active research topic in recent years and much work has been reported on both the test-on-source-model and train-on-target-language two strategies. In Jabaian et al. (2013), different approaches based on test-on-source-model and train-on-target-language strategies are compared.

The simplest way for the test-on-source-model scheme is to translate L2 target sentences with a web translator into L1, and process these translations with the L1 SLU system. In He et al. (2013) the Microsoft Bing translator is used for this purpose. This approach provides translations of the user input at a very low economic and time cost. Moreover, Stepanov et al. (2013) have demonstrated that application of language-style and domain adaptation techniques to the "off-the-shelf and out-of-domain" SMT system could yield improved translation and thus obtain better SLU performance. For adapting MT, Garcıa et al. (2014) translated the L1 SLU training corpus into L2, with multiple web translators to obtain a large parallel dataset, and then trained their own L2-to-L1 SMT model to reduce translation errors. Calvo et al. (2013; 2016) proposed to build graphs of words from different translations and then to parse the translation graph with the SLU model.

The train-on-target-language approach relies on the accurate transferring of semantic annotations from L1 to L2. García et al. (2012) translates each segment within one slot separately with a web translator, and then joins these pieces together into a sentence. This method doesn't need additional procedures to transfer slot labels. However, this method misses the context information that is crucial to translation quality. Moreover, it's difficult to determine the ordering of the translated segments. Misu et al. (2012) trains an SMT model using conversational in-domain parallel data, and then translates the entire L1 SLU training corpus to L2. However, bilingual in-domain data is scarce and costly, making it difficult to deliver both quality and low cost. Jabaian et al. (2010; 2013) propose two language transferring schemes. One is transferring source language annotation indirectly through word alignment. The other one is forcing the SMT model to translate the segmentation and slot labels simultaneously. The authors report that the indirect alignment gives the best performance. However, they also point out that distant language pairs suffer severely in word alignment. Finally rather than relying on automatic machine translation, Stepanov et al. (2014) prefer using human professional translation services. In Stepanov et al. (2017), they extend their work via crowdsourcing for semantic annotation.

Methods	**Source input**	**Translation result**
Naive Translation	我想打个电话给白晓霞	I would like to make a call to telephone number of white sunshine
Token-added Translation	我想打个电话给（a 白晓霞）	I would like to make a call to (a white sunshine) 's telephone number please
Class-based / SCRT	我想打个电话给 $contact_name	I would love to make a call to $contact_name 's number please

Table 1: Translation examples by different translation systems.

3 Translation Systems

This section describes all the translation systems conducted in our experiments. Each system has its own approach of transferring slot labels to the target sentence. They all start from the same general-purpose NMT, trained on a huge L1:L2 parallel corpus without SLU annotation. Hence the general translator usually doesn't do well with input containing slot labels.

3.1 Naive Translation

The naive translation system uses the given well-trained general-purpose translator as it is. The slot labels in the L1 SLU corpus are stripped off before entering the general translator. We then align the words between the source sentence and the translated sentence, in order to add back slot labels into the translated sentence. This is an indirect slot transferring approach.

Some limitations of the naive translation approach come from both translation and alignment. There are plenty of slot values like song names and contact names in the original SLU corpus. Many of these words are out-of-vocabulary (OOV) for the translation model. Although the translation model can handle these OOV words with sub-word modeling techniques (Sennrich et al., 2016), there are still many slot values remain to be OOV or mistranslated. For example, the Chinese name "白晓霞" in Table 1 is literally translated to "white sunshine". Furthermore, wrong translations might also result in wrong alignments, which will yield inaccurate positions for slot labels.

3.2 Token-added Translation

To make the translator be aware of slots, we then propose a token-added translation approach. This approach uses some special tokens to mark the segmentation boundary for the slot value in the source sentence. These special tokens are common in both the source vocabulary and target vocabulary of the general translator and their translation is unique and easy to spot. For example, parentheses and double quotes are good candidates as the special tokens. Enclosing slot values in source sentences by these special tokens can help identify slot boundaries in the translation outputs. In our example in Table 1, the special tokens we choose is a pair of parentheses, where the beginning parenthesis is followed by a single character "a", representing the slot label (in this case contact_name). We choose a single English letter for Chinese-to-English transfer because we are almost certain that the English letter will be carried to the target side as it is. After translation, the slot values can then be easily identified.

In token-added translation, no additional word alignment process is required. However such approach relies heavily on the NMT general training data where the special tokens (e.g. parentheses or double quotation marks) are kept in both source and target data. For different language pairs, different special tokens might be chosen for the best translation quality. Empirically we find that parentheses with a single English character are highly effective for Chinese to English translation.

3.3 Class-based Translation

To better translate the slots, we further propose a class-based translation approach. It is to use a class symbol to replace both the slot label and its slot value in the source sentence, for example, "Play $song in $album". In other words, we generalize the source sentence into a pattern sentence.

The class symbol represents the slot label, but without any specific value, as shown in the last row in Table 1. Representing each slot segment with a single symbol has a great advantage of avoiding a multi-word segment to be translated into several non-consecutive segments and not enclosed by the correct slot-label pairs. In order to help the general translator understand these new words (class symbols), we need some new parallel sentences with class symbols to adapt the general translator. Then for each sentence in the L1 SLU corpus, we first transform it to use class symbols and then translate it with the adapted general translator. The translated sentence will contain class symbols as well. We then replace these class symbols with slot values appropriate to the target culture.

Therefore unlike the first two systems, the class-based translation model is an adapted translator. It requires a small parallel annotated SLU corpus for adaptation.

3.4 Source-Critical Reinforcement Translation

The performance of class-based model relies on whether parallel annotated SLU data is enough or not to translate class symbols correctly. In other words, slots will be missed or mistranslated without enough parallel data. However, in-domain SLU parallel data is scarce. It's best if we can take advantage of the existing L1 SLU corpus to solve the slots missing or mistranslated problem of class-based method. Assume the general translator has been adapted into a class-based translation model, we now propose another adaptation algorithm based on reinforcement learning that depends on monolingual instead of bilingual SLU corpora. Hence we name it source-critical reinforcement translation (SCRT).

Before formally introducing SCRT, it is necessary to first define SKR, a metric named slot keeping ratio for evaluating the slot transferring performance of translation models. For each (Chinese) sentence c_i, its SKR is defined as the number of slots in the translated (English) sentence, e_i, divided by the number of slots in c_i:

$$\text{SKR}(c_i, e_i) = \frac{\sum_s \min(g(c_i, s), g(e_i, s))}{\sum_s g(c_i, s)} \times 100\%$$

The function $g(c_i, s)$ is used to count the occurrences of slot s in sentence c_i. Minimum is taken at the numerator to avoid SKR to be over 100%.

It's easy to see that a high SKR is a critical condition for generating a high-quality training corpus for L2 SLU models. Our SCRT algorithm directly optimizes the SKR for each source sentence, and learns how to translate specific words according to specific rules. In this task, specific words are those class symbols, and specific rules are the correspondence of each class symbol in the two languages. This is inspired by Ranzato et al. (2015) and Rennie et al. (2017). Instead of promoting sequence-level training performance via directly optimizing BLEU scores that require ground truth target sentences, our algorithm focuses on the performance of specific words only, and therefore ground-truth target sentences are not required. The architecture of this model is depicted in Figure 1.

Since SCRT is based on REINFORCE algorithm (Williams, 1992; Zaremba and Sutskever, 2015), we first describe reinforcement learning from the perspective of sequence generation.

3.4.1 REINFORCE

In order to directly maximize SKR, we can cast our problem in the reinforcement learning framework (Sutton and Barto, 1998) as in Ranzato et al. (2015). Our NMT model can be viewed as an *agent* that interacts with an external *environment* (words). The parameters of neural network, θ, defines a *policy* p_θ, whose execution results in an *action*. The *action* is the prediction of the next word in the sequence at each time step. After taking an action, our agent will update its internal *state* (attention weights). Once our agent generates the end-of-sequence (EOS) token, the agent will observe a *reward*. We can choose different reward functions for various purposes. Here, we use the SKR value as the reward r.

During training, the agent chooses an action according to the current policy and observes a reward only when the end token is generated. This reward is computed according to the generated sentence and the input sentence. The goal of training is to maximize the expected reward, and therefore the loss function (negative of reward) for the encoder-decoder given an input source sentence, $x = x_1, x_2, ...$, is

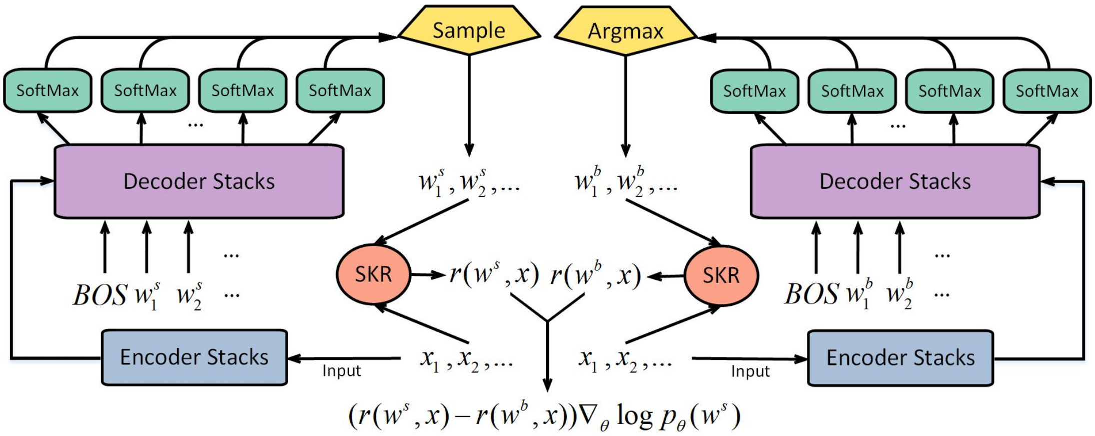


Figure 1: Architecture of Source-Critical Reinforcement Translation. $x_1, x_2, \ldots$ is the source input of the NMT model and $(r(w^s, x) - r(w^b, x))\nabla_\theta \log p_\theta(w^s)$ is the final gradient to optimize with policy gradient method. The encoder and decoder stacks represent inner layers of the Transformer general translator.

defined as:

$$\begin{aligned} L(\theta) &= -\sum_{w_1^g, \ldots, w_T^g} p_\theta(w_1^g, \ldots, w_T^g | x)\, r(w_1^g, \ldots, w_T^g, x) \\ &= -\mathbb{E}_{w^g \sim p(\theta)} r(w^g, x) \end{aligned} \tag{1}$$

where $w^g = (w_1^g, \ldots, w_T^g)$ is the generated sentence of length T. Notice the length of x is not necessarily T. The notion $w^g \sim p(\theta)$ means the generated sentence follows the distribution of the decoder softmax output $p(\theta)$.

The gradient of $L(\theta)$ can be computed as follow:

$$\nabla_\theta L(\theta) = -\mathbb{E}_{w^g \sim p(\theta)} r(w^g, x) \nabla_\theta \log p_\theta(w^g) \tag{2}$$

We approximate this expectation with Monte Carlo based sampling and a baseline b is used to reduce the variance in the Monte Carlo estimator of the gradient, following Weaver and Tao (2001):

$$\begin{aligned} \nabla_\theta L(\theta) &= -\mathbb{E}_{w^s \sim p(\theta)} (r(w^s, x) - b) \nabla_\theta \log p_\theta(w^s) \\ &\approx -(r(w^s, x) - b) \nabla_\theta \log p_\theta(w^s) \end{aligned} \tag{3}$$

where w^s is the sampled sentence. We use a single sample to approximate the expectation in the above formula. Using the chain rule of gradient computation, we have:

$$\nabla_\theta L(\theta) = \sum_t \frac{\partial L_\theta}{\partial y_t} \frac{\partial y_t}{\partial \theta} \tag{4}$$

where y_t is the input to the decoder softmax function at time stamp t. Following Zaremba's prior work, the gradient of loss $L(\theta)$ with respect to y_t is given by:

$$\frac{\partial L_\theta}{\partial y_t} = (r(w^s, x) - b)(p_\theta(w_t | h_t) - 1(w_t^s)) \tag{5}$$

where h_t is the input vector to the fully-connected layer before softmax, and $1(w_t^s)$ is the one hot vector for w_t^s. Notice y_t, $p_\theta(w_t | h_t)$, $1(w_t^s)$ are all vectors of length V, where V is the number of output units from the decoder.

3.4.2 Self-Critical and Source-Critical

The baseline reward b is obtained by the current model using the inference algorithm at test time (Rennie et al., 2017). This is called self-critical training. In this work, we choose greedy decoding at test time. That is, the most likely word from the decoder output $p(\theta)$ at each time stamp is selected as the baseline output, as shown on the right-hand side of Figure 1. The greedy translation sentence is denoted as w^b.

The above gradient can now be written as:

$$\frac{\partial L_\theta}{\partial y_t} = (r(w^s, x) - r(w^b, x))(p_\theta(w_t|h_t) - 1(w_t^s)) \tag{6}$$

Since the reward is based on SKR exclusively, no ground truth target sentence is needed. We call this source-critical learning.

3.4.3 Optimization

Equation (6) says that the sampled sentence w^s acts like a surrogate target for our output distribution, $p_\theta(w_t|h_t)$, at time t. Once the sampled sentence achieves a higher SKR than greedy decoding, a positive reward is used for parameter updating; otherwise, a negative reward is used.

The SKR criterion itself does not impose constraints on translation quality. Therefore it needs a well-trained general-purpose NMT system as the initial model. In this paper, we initialize the SCRT model with parameters from the adapted class-based model described in Section 3.3, which also offers good embedding features for the class symbols. Otherwise, the model would be difficult to converge, as the sampling space or action space is enormous.

After initialization, we then use policy gradient methods to find parameters that lead to a large expected reward. As our SCRT only optimizes the SKR score with monolingual data, the curriculum learning (Bengio et al., 2009) strategy is employed in which we begin with the class-based model, and gradually increase the number of SCRT training steps.

4 Experiments

4.1 Experimental Setup

We conduct experiments on Chinese to English language transferring task and evaluate the quality of the translated corpus via domain classification and slot filling tasks.

We annotated 3000 (over time this should be a very big number) Chinese dialogue sentences as the L1 SLU corpus and 1500 English dialogue sentences as the test set for the target language. Our goal is to transfer these 3000 Chinese sentences to English, and then train an English SLU model to handle these 1500 English test sentences.

Next we annotated additional 1500 pairs of parallel sentences for adapting the general translator into the class-based MT. This small parallel SLU corpus is called the adaptation corpus in this section. Finally, the Chinese side of the 3000 sentences of the L1 SLU corpus, together with the Chinese side of the 1500 sentences of the adaptation corpus is used in SCRT training, on top of the adapted class-based MT.

4.1.1 SLU model

The SLU annotated data are collected from communication, navigation and music three domains. It includes seven slots: contact name, contact type, address type, song name, album, feature and artist. As "others" domain is necessary for domain classification in practical applications, we collect additional 40,000 sentences from other scenarios as the fourth domain. For domain classification models, we use the SVM linear classifier with public toolkit LIBLINEAR (Fan et al., 2008). Stop words are removed from the training corpus. Simple N-gram features with a cut-off of 10 are used in domain classification. For slot filling experiments, the CRF++ toolkit (Kudo, 2005) is used. The slot labeling follows the IOB format.

Trans.	Culture A.	SKR	Slot_F1	Dom_Acc
Naive	No	57.13%	45.39	45.87%
Naive	Yes	57.13%	69.78	78.93%
TA	No	60.82%	24.87	55.07%
TA	Yes	60.82%	34.14	81.4%

Table 2: Slot F1 scores and domain accuracy, with naive translation vs. token-added (TA) translation using an unadapted general translator. The second column indicates whether culture adaptation is applied.

4.1.2 The general translator

We choose Transformer (Vaswani et al., 2017) as the general machine translation model. And for naive translation, we use Fast-Aligner (FA-IBM 2) (Dyer et al., 2013) to locate the slot labels for the translated queries. The architecture of our NMT model is the same as Vaswani et al. (2017).

Our training corpus is from AI Challenger: The English-Chinese Machine Translation track[1]. This competition provides over 10 million parallel English-Chinese sentences which were collected from English learning websites and movie subtitles. In our experiments, we extract nearly 8 million pairs of sentences from this corpus to train the baseline machine translation model. After sub-word (Sennrich et al., 2016) preprocessing, the source vocabulary size is 83,000 and the target part is 78,000. Our training batch size is 3,072 and we train the baseline model for a total of 300,000 steps with Adam optimizer (Kingma and Ba, 2014) on two GPUs. All decodings are conducted with a beam size of 4, and the top one translation is taken as the final translation output.

4.1.3 Culture adaptation

In our experiments, we select the culture-dependent slots such as contact name, song name, album and artist for target culture adaptation. We first collect thousands of English names, artists, song names and albums to build a database. The slot values in the test data are removed from the database. In the translated target queries, the corresponding slots are filled by randomly selected slot values from the database. The same random seed is used for all approaches.

4.2 Results

4.2.1 Without parallel adaptation corpus

We first compare the performance without the parallel adaptation data. Table 2 shows that the performance of naive translation method and token-added translation method. Note that the class-based model and the SCRT trained model are not included in this table, because the class-based translator needs the adaptation data to learn the translation of class symbols, and SCRT training is built on top of the class-based model.

In Table 2, compared with naive translation method, the token-added approach achieves a higher SKR score. However, keeping more slots doesn't yield good performance in the slot filling task. More mistranslations may happen in the token-added approach due to the fact that the added special tokens actually change the context of the original query and introduce noisy information. On the other hand, the token-added approach has a slightly better domain accuracy than the naive translation. This is probably because slot labels carry a certain degree of domain information.

In Table 2, the contribution of cultural adaptation is obvious. After filling the corresponding slots with culture appropriate slot values, substantial improvements are observed in this table. Adding culture adaptation gives naive translation method more than 50% relative improvement over slot filling F1 score and domain classification accuracy. The similar trend also holds for token-added translation.

4.2.2 With parallel adaptation corpus

In this section, we first compare different methods that use the adaptation corpus that consists of 1500 in-domain annotated sentences in communication, navigation and music domains. In the adaptation

[1] https://challenger.ai/datasets/translation

Trans.	Culture A.	SKR	Slot_F1	Dom_Acc
Naive	No	57.50%	56.79	68.73%
Naive	Yes	57.50%	70.79	81.4%
TA	No	**98.55**%	90.20	66.86%
TA	Yes	**98.55**%	91.91	82.87%
Class-based	Yes	97.03%	93.04	82.12%
+SCRT	Yes	98.08%	**97.19**	**84.2**%

Table 3: Results after adapting all translators with the additional 1200 parallel SLU sentences.

Trans.	Culture A.	SKR	Slot_F1	Dom_Acc
Naive	No	57.2%	50.15	76.73%
Naive	Yes	57.2%	70.13	**81.93**%
TA	No	88.22%	84.96	59.8%
TA	Yes	88.22%	85.21	78.53%
Class-based	Yes	85.07%	83.94	76.8%
+SCRT	Yes	**88.6**%	**91.07**	81.46%

Table 4: Results after adapting all translators with the additional 90 parallel SLU sentences.

corpus, 1200 sentences are randomly selected for training and the rest as the validation data to terminate training. Although 1200 sentences are tiny for model training, we also conduct the same experiments with a much small amount of data, 90 parallel sentence pairs which are also randomly selected from the 1500 in-domain annotated sentences, to test the performance of different methods under different condition.

In Table 3, comparing the naive translation with the token-added translation, we can find that the token-added method benefits more from adapting general purpose translator with in-domain parallel data. The SKR of naive translation barely increased, even using the adapted translator. It is obvious that the main limitation of naive translation is alignment rather than translation. The class-based model achieves 97.03% SKR after the convergence of supervised training with parallel adaptation data. Finally, SCRT training with 4500 monolingual annotated data is further applied to the adapted class-based translator. The SKR is increased to 98.08%. The F1 score of slot filling and accuracy of domain classification also jump significantly, which undoubtedly proves that our SCRT can generate better SLU training corpus for other languages, with the aid of monolingual annotated data exclusively.

In Table 4, the SCRT achieves the highest F1 score of slot filling and the slot transferring or SKR still benefits from monolingual data with SCRT incremental training. This result shows that our proposed method is very competitive even with an extremely small amount of parallel in-domain data. But the naive translation method with culture adaptation show a better performance on domain classification. This is because, with an extremely small amount of parallel training data, it is hard to achieve a high quality of translation for SCRT and class-based method, which is crucial for domain classification. Besides, the naive translation always benefits more from the culture adaptation in domain classification, as those mis-aligned segments will be replaced with correct slots, which makes such sentences more distinguish from the other.

4.2.3 Analysis of SCRT

In order to evaluate how much the proposed SCRT contributes to the class-based model, we analyze the systems at different iterations of training with 1200 in-domain parallel sentences. In the experiments, we find that the class-based model begins to converge after 800 steps with a batch size of 32 when adapted on the 1200 in-domain parallel sentence pairs. So this model at step 800 is chosen as a baseline and is labeled as SL800 in Figure 2. Based on such a baseline, we use the 4500 sentences of monolingual data to conduct 30 steps (SL800+RL30) and 60 steps (SL800+RL60) SCRT RL training. As shown in Figure 2, the SKR increases nearly one percent and the F1 score increases by over 7 percents for both models.

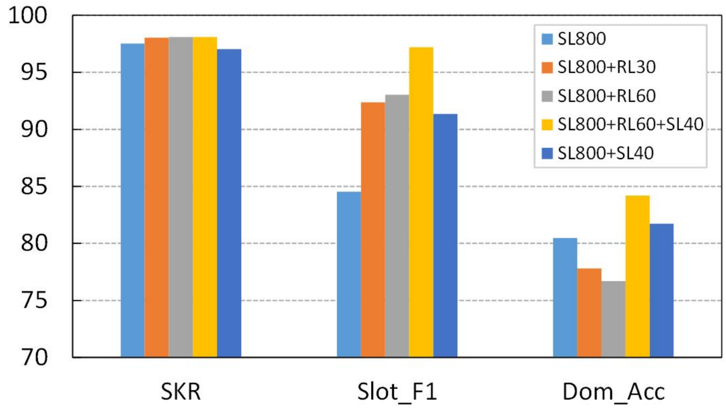
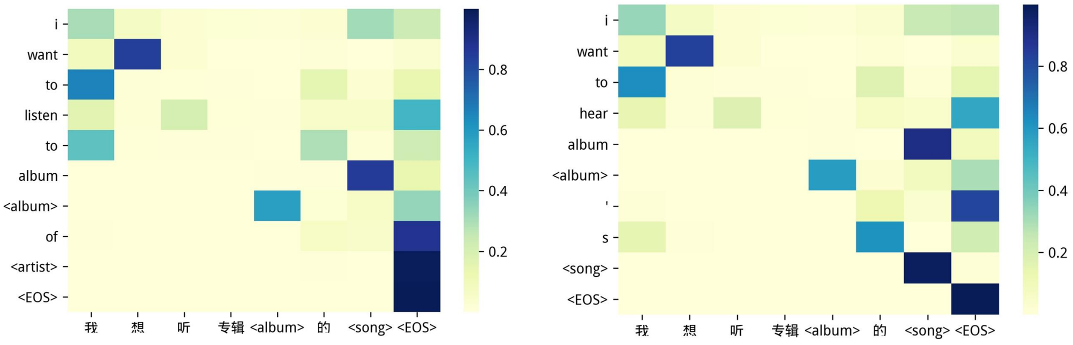


Figure 2: Performance on SKR, slot filling and domain classification, using translation models trained at different steps of supervised learning (SL) and reinforcement learning (RL).



(a) Class-based

(b) SCRT

Figure 3: Two heat maps of attention vectors between encoder and decoder in the translation model. The x-axis and y-axis of each plot correspond to the words in the source sentence (Chinese) and the generated translation (English), respectively.

However, we note that the accuracy of domain classification drops. This is because SKR imposes no constraints for general machine translation quality that is crucial for domain classification. The translation quality can be improved by additional steps of supervised training using in-domain parallel training data. For model SL800+RL60+SL40, another 40 steps of supervised training are conducted on top of SL800+RL60.

As we can see from Figure 2, SL800+RL60+SL40 achieves the highest scores on both slot filling and domain classification among all these models. The last model, SL800+SL40, is used for comparison with SL800+RL60+SL40. Although these two models are trained with the same steps of supervised learning, the model with SCRT achieves higher scores on all metrics. This result indicates the potential contribution of the monolingual data and the effectiveness of our proposed SCRT algorithm for language transferring. Furthermore, comparing SL800 and SL800+SL40, we can find that although the accuracy and F1 score goes up, the SKR goes down after additional training with parallel data. This is the shortcoming of maximum-likelihood estimation objective function for our task: treating all words equally without emphasizing important words like slot labels.

The proposed approach SCRT provides an intuitive way to guide the parameter updating for slots transferring with monolingual data. We visualize the attention weights between encoder and decoder of the translation model in Figure 3. The left plot corresponds to the model SL800 mentioned above and the right one is the model SL800+RL60. From the heat maps, we can see which positions in the source sentence were considered more important when generating the target word. Figure 3(a) shows that class-based model mistranslated the slot $song into $artist, and was corrected after additional SCRT training in Figure 3(b).

5 Conclusions

Our work is motivated by the practical demand in language transferring for SLU systems: the lack of large annotated in-domain parallel data, and the requirement of high-quality SLU corpus in the target language. To address this problem, we applied an adapted Neural Machine Translator to translate the SLU copora to other languages. A small in-domain parallel data is used to adapt the general purpose NMT firstly. Based on the adapted NMT, we proposed a reinforcement learning approach with a source-critical mechanism to do further adaptation using monolingual data exclusively. Our proposed method optimizes the slot keeping ratio directly and adapts slot values accordingly based on the target culture. The experiments showed that comparing with naive translation, the proposed method could improve domain classification accuracy by relatively 22%, and the slot filling F1 score by more than 71%.

Acknowledgements

The research work descried in this paper has been supported by the National Key Research and Development Program of China under Grant No. 2017YFC0822505 and the Natural Science Foundation of China under Grant No. 61673380.

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *International Conference on Machine Learning (ICML)*, pages 41–48.

Marcos Calvo, Fernando García, Lluís-F Hurtado, Santiago Jiménez, and Emilio Sanchis. 2013. Exploiting multiple hypotheses for multilingual spoken language understanding. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning (CoNLL)*, pages 193–201.

Marcos Calvo, Lluís-Felip Hurtado, Fernando Garcia, Emilio Sanchis, and Encarna Segarra. 2016. Multilingual spoken language understanding using graphs and multiple translations. *Computer Speech & Language*, 38:86–103.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of NAACL-HLT*, pages 644–648.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *Journal of Machine Learning Research (JMLR)*, 9(Aug):1871–1874.

Fernando García, Lluís F Hurtado, Encarna Segarra, Emilio Sanchis, and Giuseppe Riccardi. 2012. Combining multiple translation systems for spoken language understanding portability. In *Spoken Language Technology Workshop (SLT), 2012 IEEE*, pages 194–198.

Fernando Garcıa, Marcos Calvo, Emilio Sanchis, Lluıs-F Hurtado, and Encarna Segarra. 2014. Obtaining parallel corpora for multilingual spoken language understanding tasks. In *In Proceedings of the Iberspeech*.

Xiaodong He, Li Deng, Dilek Hakkani-Tur, and Gokhan Tur. 2013. Multi-style adaptive training for robust cross-lingual spoken language understanding. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8342–8346.

Bassam Jabaian, Laurent Besacier, and Fabrice Lefevre. 2010. Investigating multiple approaches for slu portability to a new language. In *Eleventh Annual Conference of the International Speech Communication Association (Interspeech)*, pages 2502–2505.

Bassam Jabaian, Laurent Besacier, and Fabrice Lefevre. 2013. Comparison and combination of lightly supervised approaches for language portability of a spoken language understanding system. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(3):636–648.

Bassam Jabaian, Fabrice Lefèvre, and Laurent Besacier. 2016. A unified framework for translation and understanding allowing discriminative joint decoding for multilingual speech semantic interpretation. *Computer Speech & Language*, 35:185–199.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Taku Kudo. 2005. Crf++: Yet another crf toolkit. *Software available at http://crfpp. sourceforge. net*, page 130.

Teruhisa Misu, Etsuo Mizukami, Hideki Kashioka, Satoshi Nakamura, and Haizhou Li. 2012. A bootstrapping approach for slu portability to a new language by inducting unannotated user queries. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4961–4964.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *Computer Science*.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1179–1195.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1715–1725.

Christophe Servan, Nathalie Camelin, Christian Raymond, Frédéric Béchet, and Renato De Mori. 2010. On the use of machine translation for spoken language understanding portability. In *2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 5330–5333.

Evgeny A Stepanov, Ilya Kashkarev, Ali Orkan Bayer, Giuseppe Riccardi, and Arindam Ghosh. 2013. Language style and domain adaptation for cross-language slu porting. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 144–149.

Evgeny A Stepanov, Giuseppe Riccardi, Ali Orkan Bayer, et al. 2014. The development of the multilingual luna corpus for spoken language system porting. In *LREC*, pages 2675–2678.

Evgeny A Stepanov, Shammur Absar Chowdhury, Ali Orkan Bayer, Arindam Ghosh, Ioannis Klasinas, Marcos Calvo, Emilio Sanchis, and Giuseppe Riccardi. 2017. Cross-language transfer of semantic annotation via targeted crowdsourcing: task design and evaluation. *Language Resources and Evaluation*, pages 1–24.

Richard S Sutton and Andrew G Barto. 1998. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*, pages 6000–6010.

Lex Weaver and Nigel Tao. 2001. The optimal reward baseline for gradient-based reinforcement learning. In *Proceedings of the Seventeenth conference on Uncertainty in Artificial Intelligence (UAI)*, pages 538–545.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*, pages 5–32.

Wojciech Zaremba and Ilya Sutskever. 2015. Reinforcement learning neural turing machines-revised. *Computer Science*.