

# Multi-agent Learning for Neural Machine Translation

Tianchi Bi, Hao Xiong, Zhongjun He, Hua Wu and Haifeng Wang

Baidu Inc. No. 10, Shangdi 10th Street

Beijing, 100085, China

{bitianchi, xionghao05, hezhongjun, wu\_hua, wanghaifeng}@baidu.com

## Abstract

Conventional Neural Machine Translation (NMT) models benefit from the training with an additional agent, e.g., dual learning, and bidirectional decoding with one agent decoding from left to right and the other decoding in the opposite direction. In this paper, we extend the training framework to the multi-agent scenario by introducing diverse agents in an interactive updating process. At training time, each agent learns advanced knowledge from others, and they work together to improve translation quality. Experimental results on NIST Chinese-English, IWSLT 2014 German-English, WMT 2014 English-German and large-scale Chinese-English translation tasks indicate that our approach achieves absolute improvements over the strong baseline systems and shows competitive performance on all tasks.

## 1 Introduction

Training with more than one agents has attracted intensive research interest in recent years, for example, dual learning (He et al., 2016; Xia et al., 2017, 2018) and bidirectional decoding (Liu et al., 2016; Zhang et al., 2019b). The former method leverages the duality between the two related agents as the feedback signal to regularize training, while the latter targets the agreement between one agent decoding from left to right (L2R) while the other decoding in opposite direction (R2L). Both methods enhance the translation models by introducing a regularization term into the training objective.

The effectiveness of these two methods lies in the fact that appropriate regularization can help each agent to learn from the superior models while integrating their advantages (e.g., good translations for prefixes for L2R, and good translation quality for suffixes for R2L). As shown in the Table 1, due to the *exposure bias* problem (Ranzato

Src	此次 豪雨 灾情 , 半数 遇难者 遭洪水 冲走 溺毙 ; 其他 则 因 房屋 倒塌 或 电线 走火 触电 致死 。
Ref	In this disaster caused by torrential rains, half of the victims were carried away and drowned in the floods; the others died because their houses collapsed or from electrocution caused by short circuits.
L2R	In the torrential rain, half of the victims were washed away by floods, while others <b>died of electricity caused by collapsed houses or electric wires</b> .
R2L	<b>Half of the victims were drowned by floods</b> , while others were killed by collapsed houses or electrocuted by sparkling wires.

Table 1: In this sample, NMT systems with different implementations present diverse translation errors for one long sentence. The agent decoding from left to right (L2R) tends to generate better prefixes and bad suffixes (bold letters), and the R2L agent has the opposite preference.

et al., 2016), the agent trained to decode from left to right tends to generate better prefixes and bad suffixes, and the agent decoding in the reverse direction demonstrates the opposite preference. By introducing additional Kullback-Leibler (KL) divergences between the probability distributions defined by L2R and R2L models into the NMT training objective (Zhang et al., 2019b), it is possible for two models to learn advantages from each other.

According to the empirical achievements of previous studies on training with two agents, it is natural to consider the training with more than two agents, and to extend our study to the multi-agent scenario. However, training with more than two agents is more complex, and we face two critical

problems. First, when deploying the multi-agent system, should we focus on the diversity or the strength of each agent? (Wong et al., 2005; Marcolino et al., 2013) Second, learning in multi-agent scenario is *many-to-many*, as opposed to the relatively simpler *one-to-one* learning in two-agent training, and requires an effective learning strategy.

There have been many alternatives to improve the diversity of models even based on the Transformer model (Vaswani et al., 2017). For example, decoding in the opposite direction usually results in different preferences: good prefixes and bad prefixes (Zhang et al., 2019b). Rather, self-attention with relative position representations enhances the generalization to sequence lengths unseen during training (Shaw et al., 2018). Furthermore, increasing the size of layers in the encoder is expected to specialize in word sense disambiguation (Tang et al., 2018; Domhan, 2018). In this paper, we investigate the effects of two teams of multi-agents: a team of alternative agents mentioned above, and a uniform team of different initialization for the same model.

To resolve the second problem, we simplify the *many-to-many* learning to the *one-to-many* (one teacher *vs.* many students) learning, extending ensemble knowledge distillation (Fukuda et al., 2017; Freitag et al., 2017; Liu et al., 2018; Zhu et al., 2018). During the training, each agent performs better by learning from the ensemble model (*Teacher*) of all agents integrating the knowledge distillation (Hinton et al., 2015; Kim and Rush, 2016) into the training objective. This procedure can be viewed as the introduction of an additional regularization term into the training objective, with which each agent can learn advantages from the ensemble model gradually. With this method, each agent is optimized not only to the maximization of the likelihood of the training data, but also to the minimization of the divergence between its own model and the ensemble model.

However, the above learning strategy converges on the local optimum rapidly in our empirical studies. It seems that each agent tends to focus on learning from the ensemble model while completely ignoring its own exploration. To alleviate this problem, we train each agent to learn from the ensemble models when necessary, and to distill the knowledge based on the translation quality

(BLEU score) of the ensemble model. This means we evaluate the quality of the ensemble model to see whether it is good enough to be studied. Consequently, the knowledge distillation from the ensemble model to the agent stems from the better translation generated by the ensemble model.

We evaluate our model on NIST Chinese-English Translation Task, IWSLT 2014 German-English Translation task, large-scale WMT 2014 English-German Translation task and large-scale Chinese-English Translation Task. Extensive experimental results indicate that our model significantly improves the translation quality, compared to the strong baseline systems. Moreover, our model also reports competitive performance on the German-English and English-German translation tasks, achieving 36.27 and 29.67 BLEU scores, respectively.

To the best of our knowledge, this is the first work on training NMT model with more than two agents<sup>1</sup>. The contributions of this paper are summarized as follows:

- We extend the study on training with two agents to the multi-agent scenario, and propose a general learning strategy to train multiple agents.
- We investigate the effects of the diversity and the strength of each agent in the multi-agent training scenario.
- We simplify complex *many-to-many* learning in multi-agent learning to *one-to-many* learning by forcing each agent learning knowledge from ensemble model as necessary.
- Extensive experiments on multiple translation tasks confirm that our model significantly improves the translation quality and reports the new state-of-the-art results on IWSLT 2014 German-English and competitive results on WMT 2014 English-German translation tasks.

## 2 Background

Conventional autoregressive NMT models (Bahdanau et al., 2015; Sutskever et al., 2014) decode target tokens sequentially, which indicates that the

---

<sup>1</sup>Although Wang et al. (2019) proposed a work named multi-agent dual learning contemporarily, the training in their work is conducted on the two agents with fixed parameters of other agents.

determination of the current token is conditioned by the previous generated sequence. Formally, at time step  $t$ , the generation of the current token  $y_t$  is determined by the following equation:

$$p(y_t) = p(y_t|y_{<t}, x; \theta) \quad (1)$$

where  $y_{<t}$  represents the previously generated sequence, and  $\theta$  are the parameters of the representation of the source sequence and the partial target sequence.

Furthermore, the usual training criterion is to minimize the negative log-likelihood for each sample from the training data,

$$\mathcal{L}_{NLL} = - \sum_{t=0}^T \sum_{k=1}^{|\mathcal{V}|} \mathbb{1}\{y_t = k\} \log p(y_t = k|y_{<t}, x; \theta) \quad (2)$$

where  $T$  is the length of the target sequence,  $|\mathcal{V}|$  is the size of vocabulary and  $\mathbb{1}\{\cdot\}$  is the indicator function. This objective can be viewed as minimizing the cross-entropy between the correct translation  $y^*$  and the model generation  $p(y_t|y_{<t}, x; \theta)$ .

### 3 Multi-agent Learning

Empirical studies indicate that one agent is trained to perform better through learning advantages from the other agent in the two-agent scenario, namely *one-to-one* learning (Figure 1.(a)). The training objective of the agent is regularized, which is to learn better models by leveraging the relationship between the two agents as feedback, i.e., duality in dual learning problem, and agreement in bidirectional decoding.

Extending the study to the multi-agent scenario is feasible and desirable, as multiple agents might supply more reliable and diverse advantages compared to the two-agents scenario. However, the agent is expected to learn advantages from each other in the multi-agent scenario, which results in a complex *many-to-many* learning problem (Figure 1.(b)).

Instead of tackling *many-to-many* learning, we force the agent to learn from a common *Teacher* by introducing ensemble knowledge distillation, thus reduce the learning to *one-to-many* (Figure 1.(c)). With this learning strategy, each agent can learn to improve the performance in an interactive updating process.

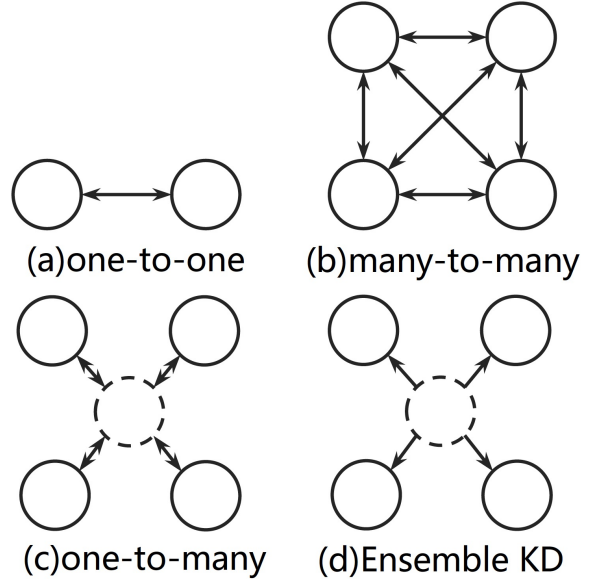


Figure 1: Illustration for different learning approaches.

As opposed to ensemble Knowledge Distillation (KD) (Figure 1.(d)), the important difference is that in the knowledge distillation of an ensemble of models, the *Teacher* network is fixed after pre-training, during the training process. While in our framework, the state of the *Teacher* network is updated at each iteration, and its performance can be further improved by the improvements of each agent explicitly, in an interactive updating process. In some ways the ensemble KD can be viewed as a particular case of our model, as we fix the update of the *Teacher* network in the training framework.

#### 3.1 Overall Framework

For the sake of simplicity, four variant agents are referred to as *Agent1*, *Agent2*, *Agent3* and *Agent4*. We begin by pre-training each agent independently (Figure 2.(a)), and then enhance the model in the multi-agent scenario in two steps: 1) Generating Ensemble Model (Figure 2.(b)), and 2) *One-to-Many* Learning (Figure 2.(c)). The performance of each agent is improved in an interactive updating process, through repeating the above two steps.

#### 3.2 Generate Ensemble Model

As pointed out in the work of Liu et al. (2018), ensemble models can empirically alleviate problems existing in the standard NMT model, such as ambiguities in the training data, and discrepancy between training and testing.

According to the practical advantages of ensem-

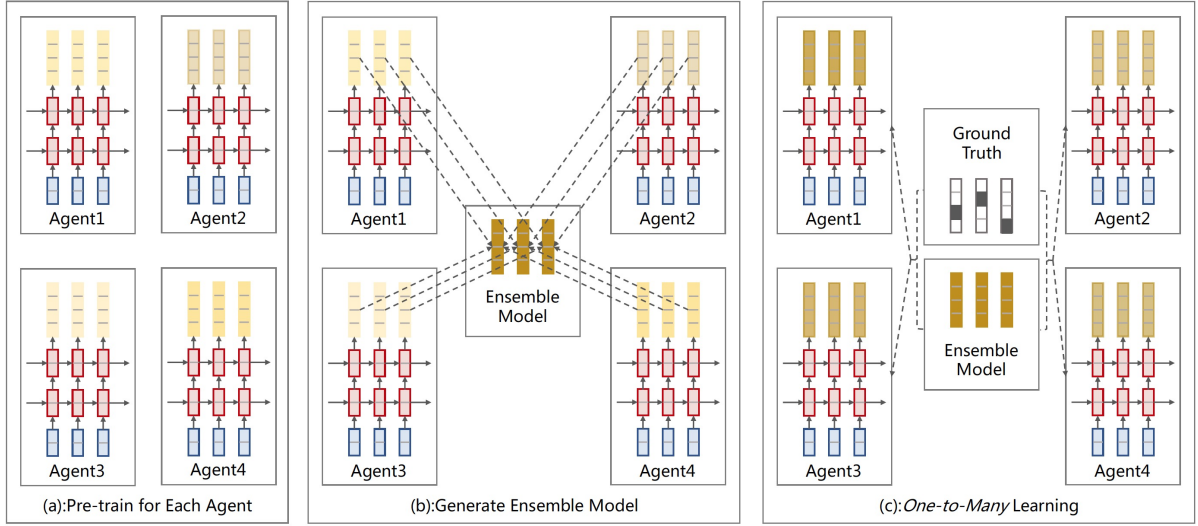


Figure 2: In this example, four agents decode the similar sentence with different model capacity. (a): At first, each agent is pre-trained to generate the translation independently. (b) The ensemble model is generated by the average prediction from each agent. (c): The *One-to-Many* learning distills the knowledge from the ensemble model to each agent as necessary. The performance of each agent is improved explicitly in an interactive updating process, through repeating the process (b) and (c).

ble models, it is relevant to force agents to learn from the ensemble model, instead of learning from each other separately. Following previous work, we develop our ensemble model by averaging the model distributions of all agents.

Formally, the model distribution of the  $i$ -th agent  $agent_i$  is defined as  $p(y_t^i | y_{<t}^i, x; \theta)$ . Assume we have  $\mathcal{N}$  agents, the model distribution of the ensemble model can be formulated as:

$$q(y_t | y_{<t}, x; \theta_t) = \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} p(y_t^i | y_{<t}^i, x; \theta_i) \quad (3)$$

where  $\theta_t$  are parameters for representing the ensemble model. Notably, we do not train it in the training process.

In the above formula, the probability  $q(y_t | y_{<t}, x; \theta_t)$  is one reliable estimator of the model distribution, as the majority of the agents are likely to generate correct sequence. From this perspective, we expect that more agents will lead to better and more robust performance.

### 3.3 One-to-Many Learning

In the *one-to-many* learning framework, the ensemble model acts as the *Teacher* network, which distills knowledge to each agent iteratively.

Rather than minimizing cross-entropy with the observed data, we minimize the cross-entropy with the probability distribution from the ensemble

model for each agent.

$$\mathcal{L}_{KD}^i = - \sum_{t=0}^T \sum_{k=1}^{|\mathcal{V}|} q(y_t = k | y_{<t}, x; \theta_t) \times \log p(y_t^i = k | y_{<t}^i, x; \theta) \quad (4)$$

With the above formula, the agent is optimized to the minimization of the model divergence between its own model and the ensemble model.

However, integrating the above regularization term into the training objective straightforwardly is problematic in practice. The agent tends to focus on learning from the ensemble model rather than exploring its own prediction to converge rapidly. Consequently, the model converges at a suboptimal point, and fails to enhance the performance by learning from each other due to the lack of evaluation of the ensemble model.

To alleviate this problem, we train each agent to learn from the ensemble model when necessary, distilling the knowledge conditioned by the translation quality of the ensemble model. We evaluate the quality of the ensemble model to see whether it is good enough to be studied. The knowledge distillation from the ensemble model to the agent comes from the better translation generated by the ensemble model. Otherwise, the agent is forced to learn its own distribution.

Let  $\langle X_g, Y_g \rangle$  be one sentence pair in the training corpus, the quality of the translation sequence generated by the ensemble model is mea-

sured by the BLEU score,

$$score(Y_t) = \text{BLEU}(Y_t, Y_g) \quad (5)$$

where  $Y_t$  is generated by the model distribution  $q(y_t|y_{<t}, x; \theta_t)$ . The quality of the translation sequence generated by each agent can also be defined using the similar metric,

$$score(Y_s^i) = \text{BLEU}(Y_s^i, Y_g) \quad (6)$$

where  $Y_s^i$  is generated by the model distribution  $p(y_t^i|y_{<t}^i, x; \theta)$  of each agent.

Conditioned by the translation quality of the ensemble model, we modify the training objective for each agent as follows:

$$\mathcal{L}_{KD}^i = - \sum_{t=0}^T \sum_{k=1}^{|\mathcal{V}|} S \times \log p(y_t^i = k | y_{<t}^i, x; \theta) \quad (7)$$

where  $S$  is defined as:

$$S = \begin{cases} q(y_t = k | y_{<t}, x; \theta_t) & score(Y_t) > score(Y_s^i) \\ \mathbb{1}_s\{y_t^i = k\} & otherwise \end{cases} \quad (8)$$

where  $\mathbb{1}_s\{\cdot\}$  is an indicator, conditioned by the existence of  $y_t^i = k$  in the sequence  $Y_s^i$ .

In practice, each agent is not only optimized to maximize the likelihood of the training data, but also to minimize the model divergence between its own model and the ensemble model:

$$\mathcal{L}_a^i = \lambda_i \mathcal{L}_{NLL} + (1 - \lambda_i) \mathcal{L}_{KD}^i \quad (9)$$

where  $\lambda_i$  is a hyperparameter, balancing the weight of two factors.

From the perspective of learning, the agent learns to minimize the training objective by generating competitive translations, which leads to a global improvement for all agents. On the other hand, a sequence-level training objective might alleviate the *exposure bias* problem implicitly (Ranzato et al., 2016).

### 3.4 Joint Learning

In the work of Zhang et al. (2019b), they proposed a relatively complex joint learning framework for training two agents. In this paper, according to the

---

### Algorithm 1: Multi-agent Learning

---

**Input:**  $N$  variant agents;  
1 Pre-train each agent  $agent_i$  independently;  
2 **repeat**  
3   **for** each training sample  $(X_g, Y_g)$  **do**  
4     Ensemble Model:  $q(y_t|y_{<t}, x; \theta_t)$ ;  
5     Generate sequence:  
6        $Y_t \leftarrow \arg \max_{0 \leq t < T} q(y_t|y_{<t}, x; \theta_t)$ ;  
7     **for** each agent  $agent_i$  **do**  
8       Generate sequence:  
9         $Y_s^i \leftarrow \arg \max_{0 \leq t < T} p(y_t^i|y_{<t}^i, x; \theta)$ ;  
10       Compute KD loss:  $\mathcal{L}_{KD}^i \leftarrow (q(y_t), p(y_t^i), Y_t, Y_s^i, X_g, Y_g)$ ;  
11       Compute NLL loss:  
12         $\mathcal{L}_{NLL} \leftarrow (p(y_t^i), X_g, Y_g)$  ;  
13       Compute Agent loss:  
14         $\mathcal{L}_a^i \leftarrow \lambda \mathcal{L}_{NLL} + (1 - \lambda) \mathcal{L}_{KD}^i$ ;  
15     **end**  
16     Model loss:  $\mathcal{L}_{final} = \sum_{i=1}^N \mathcal{L}_a^i$ ;  
17     Update gradients for each agent;  
18 **end**  
19 **until** convergence;

---

previous experiments, we find that a simple multi-task learning technique without sharing any modules presents promising performance,

$$\mathcal{L}_{final} = \sum_{i=1}^N \mathcal{L}_a^i \quad (10)$$

In Algorithm 1, we describe the overall procedure of our approach. It deserves noting that in line 3, we assume the model reads a single pair of training examples per timestep for simplistic description, while in practice the model reads a batch size of samples at each step.

## 4 Experiments

In this paper, we evaluate our model on four translation tasks: NIST Chinese-English Translation Task, IWSLT 2014 German-English Translation Task, WMT 2014 English-German Translation Task and large-scale Chinese-English Translation Task.

### 4.1 Data Preprocessing

To compare with previous studies, we conduct byte-pair encoding (Sennrich et al., 2016) for Chinese, English and German sentences, setting the

MODEL	MT02	MT03	MT04	MT08	AVERAGE	IMPROVEMENTS
	Results for Best Agent				Results for each Agent	
<i>Wang et al. (2018)</i>	-	46.60	47.73	-	-	-
<i>a.L2R</i>	48.53	47.07	48.43	42.21	46.56	-
<i>b.R2L</i>	47.06	45.58	47.14	41.04	45.20	-
<i>c.Enc</i>	48.86	47.54	48.57	42.93	<b>46.97</b>	-
<i>d.Rel</i>	48.12	48.19	48.33	42.51	46.78	-
<i>a+b</i>	48.82	47.65	48.45	42.49	<b>46.86/45.42</b>	<b>+0.33/+0.20</b>
<i>a+c</i>	48.79	48.30	49.32	43.44	<b>47.3/47.27</b>	<b>+0.80/+0.30</b>
<i>a+d</i>	48.76	48.40	48.74	43.27	<b>47.19/47.09</b>	<b>+0.63/+0.31</b>
<i>c+d</i>	49.45	49.01	49.52	43.71	47.62/ <b>47.79</b>	+0.65/ <b>+1.01</b>
<i>a×2</i>	48.64	47.98	49.08	43.07	46.96/ <b>47.10</b>	+0.40/ <b>+0.54</b>
<i>d×2</i>	48.23	48.78	48.90	43.85	<b>47.44/47.31</b>	<b>+0.66/+0.53</b>
<i>a+b+c</i>	49.32	48.72	49.32	44.34	47.69/45.62/ <b>47.74</b>	<b>+1.13/+0.42/+0.77</b>
<i>a+b+d</i>	49.29	48.90	49.52	44.21	<b>47.73/45.71/47.65</b>	<b>+1.17/+0.51/+0.87</b>
<i>a+c+d</i>	49.42	49.13	49.68	44.66	47.72/ <b>48.21/47.94</b>	+1.16/ <b>+1.24/+0.96</b>
<i>a×3</i>	48.75	48.09	49.19	43.18	47.07/ <b>47.28/47.20</b>	+0.51/ <b>+0.72/+0.64</b>
<i>d×3</i>	48.47	49.02	49.42	44.09	47.51/47.46/ <b>47.57</b>	+0.73/+0.68/ <b>+0.79</b>
<i>a+b+c+d</i>	49.52	48.94	49.61	44.70	47.95/46.10/ <b>48.3/47.95</b>	<b>+1.39/+0.90/+1.33/+1.17</b>

Table 2: BLEU score for the representative models in multi-agent training on NIST Chinese-English translation.  $X \times Y$  stands for  $Y$  agents with the identical model  $X$  but from different initializing seeds.

vocabulary size to 20K and 18K for Chinese-English, a joint 20K vocabulary for German-English, and a joint 32K vocabulary for English-German, respectively.

For Chinese-English task, the training data consists of about 1.5M sentence pairs extracted from LDC corpora <sup>2</sup>. We choose the NIST 2006 (MT06) dataset for validation, and NIST 2002-2004 (MT02-04), as well as NIST 2008 (MT08) datasets for testing. For large-scale Chinese-English task, the training data consists of about 40M sentence pairs extracted from web data.

## 4.2 Agent Variants

To increase the diversity of agents, we implement the following variants of the Transformer based system: *L2R*, the officially released open source toolkit for running Transformer model, *R2L*, the standard Transformer decodes in reversed direction, *Enc*, the standard Transformer with 30 layers in the encoder, and *Rel*, the reimplement of self-attention with relative position (Shaw et al., 2018).

<sup>2</sup>LDC2002E18, LDC2002L27, LDC2002T01, LDC2003E07, LDC2003E14, LDC2004T07, LDC2005E83, LDC2005T06, LDC2005T10, LDC2005T34, LDC2006E24, LDC2006E26, LDC2006E34, LDC2006E86, LDC2006E92, LDC2006E93, LDC2004T08(HK News, HK Hansards )

## 4.3 Training Details

We implement our models using PaddlePaddle <sup>3</sup>, an end-to-end open source deep learning platform developed by Baidu. It provides a complete suite of deep learning libraries, tools and service platforms to make the research and development of deep learning simple and reliable.

We use the hyperparameters of the *base* version for the standard Transformer model of NIST Chinese-English and IWSLT German-English translation tasks, except the smaller token size ( $batch\_size = 320$ ). For WMT English-German and large-scale Chinese-English translation tasks, we use the hyperparameters of the *big* version. As described in the previous section, we use a hyperparameter  $\lambda_i$  for each agent to balance the preference between learning from the observed data and the ensemble model. According to the performance of each agent after pre-training, we set this value as follows:

$$\lambda_i = 0.5 + \max(-0.5, \min(0.5, \frac{B_i - B_{avg}}{10})) \quad (11)$$

where  $B_i$  is the BLEU score obtained by the pre-training of the  $i$ -th agent, and the  $B_{avg}$  is the average BLEU score of all agents.

The above formula suggests the agent learns more from the ensemble model as its performance

<sup>3</sup><https://github.com/paddlepaddle/paddle>

Task	<i>L2R</i>	<i>Rel</i>	KD-4	Dual-5	Rel-4
De-En	33.63	34.91	35.53	34.70	<b>36.27</b>

Table 3: BLEU score on IWSLT 2014 German-English translation. KD-4 stands for ensemble knowledge distillation with four agents. Dual-5 is the SOTA model from the work of Wang et al. (2019). And Rel-4 is our best model (*Rel*) training with four diverse agents.

is worse than the majority vote, rather than focusing on exploring by its own prediction.

We train our model with parallelization at data batch level. For NIST Chinese-English task, it takes about 1.5 days to train models on 8 NVIDIA P40 GPUs, 5 days for WMT English-German task, 8 hours for IWSLT German-English task and 7 days for large-scale Chinese-English task. The detailed training process is as follows, first we train each agent until BLEU doesn't improve any more and then execute Algorithm 1 for one-to-many learning which takes 30K-40K steps to converge.

#### 4.4 Chinese-English Results

**Study on different numbers of agents.** We first assess the impact of diverse agents on translation quality. From Table 2, we see that the model's performance consistently improves as the number of agents increases, and we observe that 1) The four baseline systems with different implementations present diverse translation quality, in particular *Rel* with refined position encoding achieves the best performance. 2) The improvement of each agent after multi-agent learning is dependent on the performance of the co-trained agent (+0.33, +0.8, +0.63 improvements obtained by *L2R* when training with *R2L*, *Enc* and *Rel*). 3) Better results can be obtained by increasing the number of training agents (e.g., 47.3  $\rightarrow$  47.73  $\rightarrow$  47.95 for *L2R*).

From the overall results, increasing the number of agents in multi-agent learning significantly improves the performance of each agent (at most +1.39, +0.9, +1.33, +1.17 in  $a+b+c+d$ ), which suggests that each agent learns advantages from the other agents, and more agents might lead to further improvement. More importantly, our improvements are obtained from the advanced training strategy without any modification in decoding stage, which indicates its practicability in deployment.

**Study on uniform agents.** To measure the importance of diversity in multi-agents, we conduct mul-

Models	En-De
ConvS2S Gehring et al. (2017)	25.2
Transformer Vaswani et al. (2017)	28.4
<i>Rel</i> Shaw et al. (2018)	29.2
DynamicConv Wu et al. (2019)	29.7
Back-translation Sergey et al. (2018)	35.00
Dual-3 Wang et al. (2019)	29.92
Dual-3 + Mono Data	30.67
<i>L2R</i>	28.37
<i>Rel</i>	29.16
Rel-4	29.67

Table 4: BLEU score on newstest2014 for WMT English-German translation.

iple experiments by initializing the similar model with different initialization seeds to generate multiple agents. As reported in Table 2, although the single *Rel* model achieves the best performance compared to the other three single agents, the  $d \times 2$  (47.44) and  $d \times 3$  (47.57) perform worse than the best counterparts  $c+d$  (47.79) and  $a+c+d$  (47.94). Moreover, when training with more than two agents, the performance of  $d \times 3$  (47.57) is even lower than arbitrary diverse counterparts ( $a+b+c$  (47.74);  $a+b+d$  (47.73)).

These results suggest that multi-agent learning is better conducted with diverse agents instead of the uniform agents with excellent single performance. On the other hand, in our multi-agent learning, training with more agents brings consistent improvements even when deployed by identical models from different initialization seeds.

#### 4.5 German-English and English-German Results

We work with the IWSLT German-English translation to compare our model to the SOTA model. From Table 3, we observe that both KD-4 and Rel-4 models achieve the best result on this dataset (35.53 and 36.27), which manifests the effectiveness of using multiple agents.

Even when trained with four agents, our model outperforms the dual model trained with five agents (34.70), and reports a SOTA score on this translation task. Moreover, we argue that fine-tuning with *L2R* to obtain a better performance could further improve the performance of Rel-4, as there exists a large gap between *L2R* and *Rel*, which affects the learning efficiency.

We further investigate the performance of our

Models	AVERAGE
<i>L2R(baseline)</i>	33.74
<i>L2R+R2L+Enc+Rel</i>	<b>34.60</b>

Table 5: BLEU score for the *L2R* model in multi-agent training on large-scale corpus of Chinese-English translation. Because we only use *L2R* model in Baidu Translate, so part of results are presented.

model on the WMT English-German translation task, which achieves a competitive result of 29.67 BLEU score on newstest2014. From Table 4, we can see that another related work, multi-agent dual learning (Wang et al., 2019) achieves promising results. This confirms that training with more agents leads to better translation quality, and reveals the relevance of using multiple agents.

Although Sergey et al. (2018) reports a BLEU score of 35.0 on this dataset, they leverage refined training corpus and a large number of monolingual data. We argue that our model can bring further improvement using their back-translation technique. Moreover, the goal of this paper is introducing a general learning framework for multi-agent learning rather than exhaustively fine-tuning to report a SOTA results. We argue that the performance of our model can be further improved using an advanced single agent, such as DynamicConv (Wu et al., 2019) and Dual (Wang et al., 2019).

#### 4.6 Large-Scale Chinese-English Results

In order to prove that multi-agent learning can yield improvement in large-scale corpus, we conducted experiments on 40M Chinese-English sentence pairs with the same data processing method as NIST. We built test sets of more than 10K sentences which cover various domains such as news, spoken language and query logs. From Table 5, we can see that multi-agent learning can increase the average baseline BLEU score of several test sets by 0.86. This technique has already been applied to Baidu Translate.

#### 4.7 Contrastive Evaluation

In this paper, we evaluate our model on the two types of contrastive evaluation datasets: *Lingeval97*<sup>4</sup> and *ContraWSD*<sup>5</sup>. The former is utilized to measure the performance of different models on dealing with subject-verb agreement in

<sup>4</sup><https://github.com/rsennrich/lingeval97>

<sup>5</sup><https://github.com/a-rios/ContraWSD>

Models	SVA (%)	WSD (%)
<i>L2R</i>	89.06	78.05
<i>R2L</i>	88.92	77.97
<i>Enc</i>	89.13	<b>79.07</b>
<i>Rel</i>	<b>89.21</b>	78.75
<i>L2R+R2L</i>	89.28	77.94
<i>L2R+Enc</i>	<b>89.42</b>	78.29
<i>L2R+Rel</i>	89.39	<b>78.47</b>
<i>L2R+R2L+Enc</i>	89.26	78.36
<i>L2R+R2L+Rel</i>	<b>89.58</b>	78.28
<i>L2R+Enc+Rel</i>	89.39	<b>78.84</b>
<i>L2R+R2L+Enc+Rel</i>	89.47	78.62

Table 6: Accuracy of subject-verb agreement (SVA) and word sense disambiguation (WSD) for different models. We report the performance of *L2R* in different models for comparison of using different agents.

English-German translation, while the latter evaluates the performance on word sense disambiguation in German-English translation. We suggest the readers refer to the work (Sennrich, 2017; Tang et al., 2018) for detailed descriptions of the two datasets.

From Table 6, we can see that the *L2R* training with multiple agents improves the accuracy both on SVA and WSD, and we observe that: 1) The *L2R* improves its ability of SVA with the help of the other agents significantly. 2) The *Enc* presents better advantage in resolving WSD than in SVA, while *Rel* has the opposite preference. 3) Although the *R2L* presents ordinary BLEU score, it still does help the *L2R* resolve the SVA.

## 5 Related Work

The most related work is recently proposed by Wang et al. (2019), who introduced a multi-agent algorithm for dual learning. However, the major differences are 1) the training in their work is conducted on the two agents while fixing the parameters of other agents. 2) they use identical agent with different initialization seeds. 3) our model is simple yet effective. Actually, it is easy to incorporate additional agent trained with their dual learning strategy in our model, to further improve the performance. More importantly, both of our work and their work indicate that using more agents can improve the translation quality significantly.

Another related work is ensemble knowledge distillation (Fukuda et al., 2017; Freitag et al., 2017; Liu et al., 2018; Zhu et al., 2018), in which



the ensemble model of all agents is leveraged as the *Teacher* network, to distill the knowledge for the corresponding *Student* network. However, as described in the previous section, the knowledge distillation is one particular case of our model, as the performance of the *Teacher* in their model is fixed, and cannot be further improved by the learning process.

Our work is also motivated by the work of training with two agents, including dual learning (He et al., 2016; Xia et al., 2017, 2018), and bidirectional decoding (Liu et al., 2016; Zhang et al., 2018, 2019b,a). Our method can be viewed as a general learning framework to train multiple agents, which explores the relationship among all agents to enhance the performance of each agent efficiently.

## 6 Conclusions and Future Work

In this paper, we propose a universal yet effective learning method for training multiple agents. In particular, each agent learns advantages from the ensemble model when necessary. The knowledge distillation from the ensemble model to the agent stems the better translation generated by the ensemble model, which enables each agent to learn high-quality knowledge while retaining its own exploration.

Extensive experimental results prove that our model brings an absolute improvement over the baseline system, reporting SOTA results on IWSLT 2014 German-English and competitive results on WMT 2014 English-German translation tasks.

In the future, we will focus on training with more agents by translating apart of one sentence considering its advantage for each agent, rather than translating the whole sentence.

## References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*.

Tobias Domhan. 2018. How much attention do you need? a granular analysis of neural machine translation architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1799–1808.

Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation. *arXiv preprint arXiv:1702.01802*.

Takashi Fukuda, Masayuki Suzuki, Gakuto Kurata, Samuel Thomas, Jia Cui, and Bhuvana Ramabhadran. 2017. Efficient knowledge distillation from an ensemble of teachers. *Proc. Interspeech 2017*, pages 3697–3701.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proc. of ICML*.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. In *Advances in Neural Information Processing Systems*, pages 820–828.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. *Distilling the knowledge in a neural network*. In *NIPS Deep Learning and Representation Learning Workshop*.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Agreement on target-bidirectional neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 411–416.

Yijia Liu, Wanxiang Che, Huaipeng Zhao, Bing Qin, and Ting Liu. 2018. *Distilling knowledge for search-based structured prediction*. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1393–1402. Association for Computational Linguistics.

Leandro Soriano Marcolino, Albert Xin Jiang, and Milind Tambe. 2013. Multi-agent team formation: diversity beats strength? In *Twenty-Third International Joint Conference on Artificial Intelligence*.

Marc’Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *International Conference on Learning Representations*.

Rico Sennrich. 2017. How grammatical is character-level neural machine translation? assessing mt quality with contrastive translation pairs. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 376–382.

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 (Volume 1: Long Papers)*, volume 1, pages 1715–1725.
- Edunov Sergey, Ott Myle, Auli Michael, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the conference on empirical methods in natural language processing*, pages 489–500. Association for Computational Linguistics.
- Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, volume 2, pages 464–468.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.
- Gongbo Tang, Mathias Müller, Annette Rios, and Rico Sennrich. 2018. [Why self-attention? a targeted evaluation of neural machine translation architectures](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4263–4272. Association for Computational Linguistics.
- 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*, pages 5998–6008.
- Mingxuan Wang, Jun Xie, Zhixing Tan, Jinsong Su, Deyi Xiong, and Chao Bian. 2018. Neural machine translation with decoding history enhanced attention. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1464–1473.
- Yiren Wang, Yingce Xia, Tianyu He, Fei Tian, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2019. Multi-agent dual learning. In *International Conference for Learning Representation (ICLR)*.
- KY Michael Wong, SW Lim, and Zhuo Gao. 2005. Effects of diversity on multiagent systems: Minority games. *Physical Review E*, 71(6):066103.
- Felix Wu, Angela Fan, Alexei Baevski, Yann N. Dauphin, and Michael Auli. 2019. Pay less attention with lightweight and dynamic convolutions. In *ICLR*.
- Yingce Xia, Tao Qin, Wei Chen, Jiang Bian, Nenghai Yu, and Tie-Yan Liu. 2017. Dual supervised learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3789–3798. JMLR. org.
- Yingce Xia, Xu Tan, Fei Tian, Tao Qin, Nenghai Yu, and Tie-Yan Liu. 2018. Model-level dual learning. In *International Conference on Machine Learning*, pages 5379–5388.
- Jiajun Zhang, Long Zhou, Yang Zhao, and Chengqing Zong. 2019a. Synchronous bidirectional inference for neural sequence generation. *arXiv preprint arXiv:1902.08955*.
- Xiangwen Zhang, Jinsong Su, Yue Qin, Yang Liu, Rongrong Ji, and Hongji Wang. 2018. Asynchronous bidirectional decoding for neural machine translation. *arXiv preprint arXiv:1801.05122*.
- Zhirui Zhang, Shuangzhi Wu, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2019b. Regularizing neural machine translation by target-bidirectional agreement. In *AAAI*.
- Xiatian Zhu, Shaogang Gong, et al. 2018. Knowledge distillation by on-the-fly native ensemble. In *Advances in Neural Information Processing Systems*, pages 7528–7538.