A CLUSTER-TO-CLUSTER FRAMEWORK FOR NEURAL MACHINE TRANSLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

The quality of a machine translation system depends largely on the availability of sizable parallel corpora. For the recently popular Neural Machine Translation (NMT) framework, data sparsity problem can become even more severe. With large amount of tunable parameters, the NMT model may overfit to the existing language pairs while failing to understand the general diversity in language. In this paper, we advocate to broadcast every sentence pair as two groups of similar sentences to incorporate more diversity in language expressions, which we name as parallel cluster. Then we define a more general cluster-to-cluster correspondence score and train our model to maximize this score. Since direct maximization is difficult, we derive its lower-bound as our surrogate objective, which is found to generalize point-point Maximum Likelihood Estimation (MLE) and point-to-cluster Reward Augmented Maximum Likelihood (RAML) algorithms as special cases. Based on this novel objective function, we delineate four potential systems to realize our cluster-to-cluster framework and test their performances in three recognized translation tasks, each task with forward and reverse translation directions. In each of the six experiments, our proposed four parallel systems have consistently proved to outperform the MLE baseline, RL (Reinforcement Learning) and RAML systems significantly. Finally, we have performed case study to empirically analyze the strength of the cluster-to-cluster NMT framework.

1 INTRODUCTION

Recently, an encode-decoder neural architecture (Cho et al., 2014) has surged and gained its popularity in machine translation. In this framework, the encoder builds up a representation of the source sentence and the decoder uses its previous RNN hidden state and attention mechanism to generate target translation. In order to better memorize the input information, an attention mechanism (Bahdanau et al., 2014) has been exploited to further boost its performance. In order to train the attentive encoder-decoder architecture, Maximum Likelihood Estimation (MLE) algorithm has been widely used, which aims at maximizing the point-to-point (one sentence to one sentence) log-likelihood of data pairs in a given dataset. However, this algorithm has severely suffered from data sparsity problem, or in other word, maximizing only likelihood the existing language pairs might make the model blind to all the non-existing similar sentence pairs. Thus, the large neural model might overfit to certain prototypes existing in the training set while failing to generalize more unseen but similar scenarios in test time.

Recently, different approaches (Zhang & Zong, 2016; Sennrich et al., 2015a; Ma et al., 2017; Norouzi et al., 2016) have been proposed to tackle the data sparseness, which can be mainly classified into two categories: 1) Semi-Supervised Learning (Zhang & Zong, 2016; Sennrich et al., 2015a; He et al., 2016; Cheng et al., 2016) resorts to external monolingual (unpaired) data to augment the bilingual training pairs. 2) Pseudo-Learning (Ma et al., 2017; Norouzi et al., 2016; Ranzato et al., 2015; Bahdanau et al., 2016) directly generates pseudo training samples from the existing dataset. Though semi-supervised learning has achieved significant success in NMT, it's still heavily restricted by the quality and quantity of extern monolingual data. In this paper, we focus on pseudo-learning strategy to enlarge training samples directly from the existing dataset. The existing pseudo-learning algorithms can be categorized into two types: 1) Golden-Centroid Augmentation (RAML), Ma et al. (2017) and Norouzi et al. (2016) advocate to augment samples by sampling from a payoff distribution, which incorporates well-controlled modification to ground truth to enrich training data without

hurting its semantic meaning. 2) Model-Centroid Augmentation (RL), Bahdanau et al. (2016) and Ranzato et al. (2015) leverage model-generated candidates as pseudo training samples, which are weighted with rewards to enhance the model learning. By exploring self-generated candidates, the model is able to understand the diversity in the output space. In pseudo-learning algorithms, both

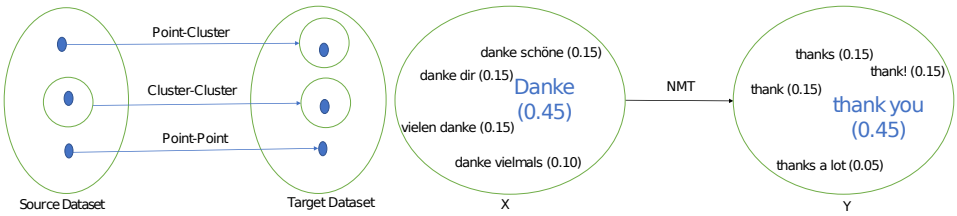

Figure 1: Left figure shows the three pseudo-learning algorithms, right figure shows the details of parallel cluster. X denotes the source language and Y denotes the target language.

RAML and RL can be interpreted as broadcasting a target ground truth as a cluster of analogues while leaving the source input untouched, which though helps the model understand target diversity, fails to capture the input diversity. In order to explore both sides' diversity, we advocate a novel and general cluster-to-cluster framework of pseudo learning, which first broadcasts both source and target sentence as clusters and then train the model to comprehend their correspondence, as described in Figure 1.

In this paper, we first introduce the concept of parallel cluster, then design the cluster-to-cluster correspondence score as our optimization objective, based on which, we derive its lower bound KL-divergence as our surrogate objective for model training. In order to realize our proposed framework, we design four parallel systems and apply them to three recognized machine translation tasks with both forward and reverse translation directions, these four systems have all demonstrated their advantages over the existing competing algorithms in six translation tasks. In the appendices, we draw samples from the parallel clusters and further analyze their properties to verify our motivation.

The contributions of our paper can be summarized as follows: 1) We are the first to propose the concept of cluster-to-cluster framework, which provides a novel perspective to current sequence-to-sequence learning problems. 2) We delineate the framework and arrive in a novel KL-divergence loss function and generalizes several existing algorithms as special cases, which provides a high-level understanding about the previous algorithms.

2 Related Literature

2.1 Reinforcement Learning for Sequence-to-Sequence Model

Exposure bias and train-test loss discrepancy are two major issues in the training of sequence prediction models. Many research works (Bahdanau et al., 2014; Shen et al., 2015; Ranzato et al., 2015; Norouzi et al., 2016; Lamb et al., 2016) have attempted to tackle these issues by adding reward-weighted samples drawn from model distribution into the training data via a Reinforcement Learning (Sutton & Barto, 1998) framework. By exposing the model to its own distribution, these methods are reported to achieve significant improvements. Bahdanau et al. (2016), Ranzato et al. (2015) and Shen et al. (2015) advocate to optimize the sequence model as a stochastic policy to maximize its expected task-level reward. Though RL is not initially designed to resolve data sparsity problem, the model-centroid training samples can indeed alleviate data sparseness by exposing the sequence-to-sequence model to more unseen scenarios. One problem of the previous RL works is that, the input information is still restricted to the dataset, which fails to teach model to comprehend source diversity. The cluster-to-cluster framework augments many similar input sentences to account for source language diversity.

2.2 REWARD AUGMENTED MAXIMUM LIKELIHOOD

One successful approach for data augmentation in neural machine translation system is Reward Augmented Maximum Likelihood (RAML) (Norouzi et al., 2016), which proposes a novel payoff distribution to augment training samples based on task-level reward (BLEU, Edit Distance, etc). In order to sample from this intractable distribution, they further stratify the sampling process as first sampling an edit distance, then performing random substitution/deletion operations. Following the work of RAML, Ma et al. (2017) introduces a novel softmax Q-Distribution to reveal RAML's relation with Bayes decision rule, and they also propose an alternative sampling strategy – first randomly replacing n-gram of the ground truth sentence and then using payoff distribution to compute corresponding importance weight with local normalization. These two approaches augment the target-side data by exposing the model to diverse scenarios and improve its robustness. We draw our inspiration from RAML, but with a difference that, instead of based on task-level reward, a learnable payoff function (cluster distribution) is used in our approach to take more latent structures into account, such as semantic meaning, language fluency, etc. From the cluster distribution, we can sample semantically and syntactically correct candidates to train the model. In addition, our more generalized bilateral data augmentation strategy also empowers our model more capability to generalize better.

2.3 SEMI-SUPERVISED LEARNING IN NMT

In order to utilize the large amount of monolingual data in current NMT framework, different strategies have been designed, the most common methods can be concluded into these categories: 1) using large monolingual data to train language model and integrates it to enhance language fluency (Brants et al., 2007). 2) using self-learning method to transform the monolingual data into bilingual form (Sennrich et al., 2015a; Zhang & Zong, 2016). 3) using reconstruction strategy to leverage monolingual data to enhance NMT training (He et al., 2016; Cheng et al., 2016). Although our motivation to augment training data is aligned with these semi-supervised algorithms, our proposed framework has substantial differences from them: 1) we don't rely on additional monolingual data to boost NMT performance; 2) Though we jointly train forward and backward translation models as advocated in He et al. (2016) and Cheng et al. (2016), our joint algorithm doesn't involve any interactions between these two models (they can be trained independently).

3 MODEL

3.1 PARALLEL CLUSTER

We define the parallel cluster as two groups of weighted sentences $C(Y^*)$ and $C(X^*)$, whose similarities (BLEU, METEOR, etc) with Y^* and X^* are above certain threshold M.

$$C(Y^*) = \underset{Y \in \mathcal{Y}}{\cup} \{Y : p(Y|Y^*)|R(Y, Y^*) > M\} \quad C(X^*) = \underset{X \in \mathcal{X}}{\cup} \{X : p(X|X^*)|R(X, X^*) > M\} \tag{1}$$

Every sample X or Y is associated with a normalized weight $p(X|X^*)$ or $p(Y|Y^*)$ to denote how much chance a sentence X or Y is sampled from the corresponding cluster, here we draw a schematic diagram to better visualize the parallel cluster in Figure 1. We will further talk about how we define and compute the weights in the following sections.

3.2 CLUSTER-TO-CLUSTER CORRESPONDENCE

Upon the definition of parallel cluster, we further design a cluster-to-cluster correspondence score $CR_{c \to c}(X^*, Y^*)$ as the log scaled expectation of likelihood of a random sentence X in source cluster $C(X^*)$ being translated to Y in target cluster $C(Y^*)$, which generally denotes the translatability of two clusters, formally, we define the cluster-to-cluster correspondence score $CR_{c \to c}(X^*, Y^*)$ as below:

$$CR_{c \to c}(X^*, Y^*) = \log \underset{X \sim C(X^*)}{E} \underset{Y \sim C(Y^*)}{E} p(Y|X) \tag{2}$$

Algorithm	Source $\rightarrow$ NMT $\rightarrow$ Target	Objective
MLE(P2P)	$\sigma(X\|X^*) \rightarrow \sigma(Y\|Y^*)$	$\text{argmin}\, KL(\sigma(Y\|Y^*)\|\|\sum_X \sigma(X\|X^*)p(Y\|X))$
RAML(P2C)	$\sigma(X\|X^*) \rightarrow q(Y\|Y^*)$	$\text{argmin}\, KL(q(Y\|Y^*)\|\|\sum_X \sigma(X\|X^*)p(Y\|X))$
Cluster-to-Cluster	$p(X\|X^*) \rightarrow p(Y\|Y^*)$	$\text{argmin}\, KL(p(Y\|Y^*)\|\|\sum_X p(X\|X^*)p(Y\|X))$

Table 1: Our proposed framework can generalize MLE as point-to-point case, while RAML as point-to-cluster case.

The higher correspondence score the more likely these two clusters correspond to each other. Note that the cluster-to-cluster correspondence score can reflect both NMT's and cluster's quality, assuming the cluster is ideal, then the correspondence score measures the translatability from a source sentence X to a target sentence Y, while assuming the NMT is ideal, then the correspondence score measures the quality of the cluster (the capability to rank paraphrases based on semantically similarity).

3.3 Maximizing cluster-to-cluster Correspondence

Based on the definition of parallel cluster and cluster-to-cluster correspondence score, we further design the cluster-to-cluster framework's objective function as maximizing the empirical correspondence score $CR_{c\rightarrow c}(X^*, Y^*; D)$ with the regularization of target cluster's entropy $H(p(Y|Y^*))$ in a dataset D, as described below:

$$\begin{aligned} Obj_{c\rightarrow c} =& CR_{c\rightarrow c}(X^*, Y^*; D) + H(p(Y|Y^*)) \\ =& \sum_{(X^*,Y^*)\in D} \log \mathop{E}_{Y\sim p(Y|Y^*)} \mathop{E}_{X\sim p(X|X^*)} p(Y|X) + H(p(Y|Y^*)) \end{aligned} \tag{3}$$

By applying Jensen's inequality to the objective function $Obj_{c\rightarrow c}$, we can further derive its lower-bound as:

$$\begin{aligned} Obj_{c\rightarrow c} =& CR_{c\rightarrow c}(X^*, Y^*; D) + H(p(Y|Y^*)) \\ \geq& \sum_{(X^*,Y^*)\in D} \mathop{E}_{Y\sim p(Y|Y^*)} \log \mathop{E}_{X\sim p(X|X^*)} p(Y|X) + H(p(Y|Y^*)) \\ =& \sum_{(X^*,Y^*)\in D} -\sum_{Y} p(Y|Y^*) \log \frac{p(Y|Y^*)}{\sum_X p(X|X^*)p(Y|X)} \\ =& \sum_{(X^*,Y^*)\in D} -KL(p(Y|Y^*)||\sum_X p(X|X^*)p(Y|X)) \\ =& \sum_{(X^*,Y^*)\in D} -KL(p(Y|Y^*)||p(Y|X^*)) \end{aligned} \tag{4}$$

From this, we notice that the cluster-to-cluster objective is lower bounded by a negative KL-divergence $-KL(p(Y|Y^*)||p(Y|X^*))$. Therefore, we can use this lower-bound to maximize correspondence score, by changing the sign of this lower-bound function, we further define the loss function as:

$$Loss_t = KL(p(Y|Y^*)||p(Y|X^*)) \tag{5}$$

We theoretically verify that this lower bound KL-divergence can generalize Maximum Likelihood (MLE) and Reward Augmented Maximum Likelihood (RAML) (Norouzi et al., 2016) as special cases when we instantiate cluster distribution as Kronecker-Delta function $\delta(Y|Y^*)$ and payoff distribution $q(Y|Y^*) = \frac{e^{R(Y,Y^*)/\tau}}{\sum_Y e^{R(Y,Y^*)/\tau}}$, as shown in Table 1.

4 Optimization

4.1 NMT

In this section, we try to minimize the proposed KL-divergence $KL(p(Y|Y^*)||p(Y|X^*))$ so as to raise the lower bound of the regularized cluster-to-cluster correspondence. We can write its deriva-

tives w.r.t to the NMT parameters in two forms, namely parallel sampling and NMT broadcasting modes, which differ in their Monte-Carlo proposing distribution.

- Parallel Sampling: sampling candidates independently from two clusters and then re-weighted pairwise samples with a translation confidence $w(Y|X, X^*)$.

$$\begin{aligned}\nabla Loss_t &= \sum_Y p(Y|Y^*) \sum_X \frac{-p(X|X^*)p(Y|X)}{\sum_{X'} p(X'|X^*)p(Y|X')} \nabla \log p(Y|X) \\ &= \sum_Y p(Y|Y^*) \sum_X p(X|X^*) \frac{-p(Y|X)}{\underset{X' \sim p(X|X^*)}{E} p(Y|X')} \nabla \log p(Y|X) \\ &= \underset{Y \sim p(Y|Y^*) X \sim p(X|X^*)}{E} \frac{-p(Y|X)}{p(Y|X^*)} \nabla \log p(Y|X) \\ &= \underset{Y \sim p(Y|Y^*) X \sim p(X|X^*)}{E} - w(Y|X, X^*) \nabla \log p(Y|X)\end{aligned} \tag{6}$$

- Translation Broadcasting: sampling candidates from one cluster and broadcasting them through the NMT to construct its opponents, and re-weighted by cluster confidence $c(Y|Y^*, X^*)$.

$$\begin{aligned}\nabla Loss_t &= \sum_Y p(Y|Y^*) \sum_X \frac{-p(X|X^*)p(Y|X)}{\sum_{X'} p(X'|X^*)p(Y|X')} \nabla \log p(Y|X) \\ &= \sum_X p(X|X^*) \sum_Y p(Y|X) \frac{-p(Y|Y^*)}{\underset{X' \sim p(X|X^*)}{E} p(Y|X')} \nabla \log p(Y|X) \\ &= \underset{X \sim p(X|X^*) Y \sim p(Y|X)}{E} \frac{-p(Y|Y^*)}{p(Y|X^*)} \nabla \log p(Y|X) \\ &= \underset{X \sim p(X|X^*) Y \sim p(Y|X)}{E} - c(Y|Y^*, X^*) \nabla \log p(Y|X)\end{aligned} \tag{7}$$

More specifically, translation broadcasting's samples are more NMT-aware in the sense that it incorporates NMT's knowledge to generate correspondents. The parallel sampling mode works like two-sided RAML (Norouzi et al., 2016) while translation broadcasting works more like mixed RAML-RL (Williams, 1992).

4.2 CLUSTER

In this paper, we design cluster distribution in two manners, namely inadaptive (pre-computed without training) and adaptive (trained during optimization) cluster. Both cluster designs meet the criterion of concentrating around the ground truth according to sentence similarity metric. In addition, a cutoff criterion is also leveraged to reject samples whose task-level score is lower than certain threshold M value as in Equation 1.

- Inadaptive Cluster: we use two non-parametric distributions $q(X|X^*)$ and $q(Y|Y^*)$ to denote source and target parallel clusters, based on the similarity score between sample X/Y and the ground truth X^*/Y^*. We follow the payoff distribution (Norouzi et al., 2016) to define our inadaptive cluster:

$$q(Y|Y^*) = \frac{exp\{R(Y, Y^*)/\tau\}}{\sum_{Y'} exp\{R(Y', Y^*)/\tau\}} = \tilde{R}(Y, Y^*) \tag{8}$$

  where $R(Y, Y^*)$ denotes the task-level reward (BLEU, CIDEr, METEOR, etc) and $\tilde{R}(Y, Y^*)$ denotes its normalization in the whole output space, τ is the hyper-parameter temperature to control the smoothness of the optimal distribution around correct target Y^*. Since the task-level reward only considers string-level matching (precision, recall, etc) while ignoring semantic coherence, the generated samples though lexically similar, prone to many semantical and syntactical mistakes, which might cause counter-effects to the NMT model.

- Adaptive Cluster: we use two parametric models $p(X|X^*)$ and $p(Y|Y^*)$ to denote the source and target adaptive cluster, which follow encoder-decoder neural architecture (Cho et al., 2014) but take ground truth X^*, Y^* as inputs. Adaptive cluster is designed to fulfill the following two requirements: 1) Proximity to ground truth: the randomly sampled candidates should have high similarity with the ground truth. 2) High correspondence score: parallel cluster should be highly correlated and translatable. Combining these two goals can guarantee mutual dependence between the source and target clusters and also retain its similarity to the original ground truth. Formally, we write the optimization target of the target cluster as:

$$Loss_l = KL(p(Y|X^*)||p(Y|Y^*)) + \underset{Y\sim p(Y|Y^*)}{E}[-R(Y, Y^*)] \tag{9}$$

  During optimization, we fix the forward NMT $p(Y|X)$ and target cluster $p(X|X^*)$ to update source cluster $p(Y|Y^*)$, and we fix the parameters of backward NMT $p(X|Y)$ and source cluster $p(Y|Y^*)$ to update target cluster $p(X|X^*)$. Here we write target cluster's derivative as following:

$$\nabla Loss_l = \underset{Y\sim p(Y|X^*)}{E} \nabla \log p(Y|Y^*) + \underset{Y\sim p(Y|Y^*)}{E}[-R(Y, Y^*)\nabla \log p(Y|Y^*)] \tag{10}$$

  Due to the mutual dependence between adaptive clusters and translation models, we advocate to alternately update the cluster and the translation models.

4.3 JOINT SYSTEMS

In this section, we advocate to combine both forward and backward translation directions in a joint system to simultaneously learn four models – forward NMT $p(Y|X)$, backward NMT $(X|Y)$, source cluster $p(X|X^*)$ and target cluster $p(Y|Y^*)$. We exploit different scenarios to combine these four models and then design four parallel systems, whose implementations are elaborated in Table 2. System-A and B use inadaptive (non-parametric) cluster, thus require optimizing only the two translation systems; system-A applies parallel sampling algorithm while B applies translation broadcasting algorithm. In contrast, system-C and D apply adaptive (parametric) cluster, thus require simultaneous optimization of both NMT and cluster, system-C applies parallel sampling while system-D applies translation broadcasting algorithm. These four systems exhibit different characteristics which are shown in details as below:

System	NMT: $p(X\|Y)$ and $p(Y\|X)$	Cluster: $p(X\|X^*)$ and $p(Y\|Y^*)$
A	Parallel Sampling	Inadaptive Model
B	Translation Broadcasting	Inadaptive Model
C	Parallel Sampling	Adaptive Model
D	Translation Broadcasting	Adaptive Model

Table 2: Four systems with different cluster models and different training methods for NMTs.

In a slight abuse of notation, we will denote $\sum_Y p_\eta(X|Y)q(Y|Y^*)$ as $\tilde{p}(X|Y^*)$ and $\sum_X p_\gamma(Y|X)q(X|X^*)$ as $\tilde{p}(Y|X^*)$.

System-A For system-A, we use inadaptive cluster with parallel sampling strategy to train the NMT model, and the forward-backward joint objective functions is defined as:

$$Loss_A(\beta, \alpha) = KL(q(X|X^*)||\sum_Y p_\eta(X|Y)q(Y|Y^*)) + KL(q(Y|Y^*)||\sum_X p_\gamma(Y|X)q(X|X^*)) \tag{11}$$

Formally, the derivative respect to η and γ are shown as:

$$\begin{aligned} \nabla_\eta Loss_A &= \underset{X\sim q(X|X^*)}{E}\underset{Y\sim q(Y|Y^*)}{E} - w(X|Y, Y^*)\nabla_\eta \log p_\eta(X|Y) \\ \nabla_\gamma Loss_A &= \underset{Y\sim q(Y|Y^*)}{E}\underset{X\sim q(X|X^*)}{E} - w(Y|X, X^*)\nabla_\gamma \log p_\gamma(Y|X) \end{aligned} \tag{12}$$

Parallel candidates are sampled from source and target cluster distributions are leveraged by scaled translation scores $w(X|Y, Y^*), w(Y|X, X^*)$ during optimization.

System-B With the same loss function in system-A, translation broadcasting is leveraged to compute derivatives in system-B, instead of parallel sampling, and the gradients is shown as:

$$\begin{aligned}
\nabla_\eta Loss_B &= \mathop{E}_{Y\sim p_\alpha(Y|Y^*)} \mathop{E}_{X\sim\tilde{p}(X|Y^*)} \frac{-\tilde{R}(X,X^*)}{\tilde{p}(X|Y^*)} \nabla_\eta \log p_\eta(X|Y) \\
&= \mathop{E}_{Y\sim p_\alpha(Y|Y^*)} \mathop{E}_{X\sim\tilde{p}(X|Y^*)} -c(X|X^*,Y^*)\nabla_\eta \log p_\eta(X|Y) \\
\nabla_\gamma Loss_B &= \mathop{E}_{X\sim p_\beta(X|X^*)} \mathop{E}_{Y\sim\tilde{p}(Y|X^*)} \frac{-\tilde{R}(X,X^*)}{\tilde{p}(X|Y^*)} \nabla_\gamma \log p_\gamma(Y|X) \\
&= \mathop{E}_{X\sim p_\beta(X|X^*)} \mathop{E}_{Y\sim\tilde{p}(Y|X^*)} -c(Y|Y^*,X^*)\nabla_\gamma \log p_\gamma(Y|X)
\end{aligned} \tag{13}$$

This system works similar as reinforcement Learning, where normalized environmental rewards $\tilde{R}(X,X^*), \tilde{R}(Y,Y^*)$ are leveraged to guide the model's policy search, and the gradients is interpreted as a form of Monte-Carlo Policy Gradient (Williams, 1992).

System-C Unlike System-A and system-B, two adaptive cluster distributions is used in system-C, thus the NMT and cluster are jointly optimized during training, and the loss function is defined as:

$$\begin{aligned}
Loss_C(\eta,\gamma,\beta,\alpha) =& KL(p_\beta(X|X^*)||\sum_Y p_\eta(X|Y)p_\alpha(Y|Y^*)) + \mathop{E}_{X\sim p_\eta(X|Y)}[-R(X,X^*)] + \\
& KL(p_\alpha(Y|Y^*)||\sum_x p_\gamma(Y|X)p_\beta(X|X^*)) + \mathop{E}_{Y\sim p_\gamma(Y|X)}[-R(Y,Y^*)]
\end{aligned} \tag{14}$$

we can get the derivatives as below:

$$\begin{aligned}
\nabla_\eta Loss_C &= \mathop{E}_{X\sim p_\beta(X|X^*)} \mathop{E}_{Y\sim p_\alpha(Y|Y^*)} -w(X|Y,Y^*)\nabla_\eta \log p_\eta(X|Y) \\
\nabla_\gamma Loss_C &= \mathop{E}_{X\sim p_\beta(X|X^*)} \mathop{E}_{Y\sim p_\alpha(Y|Y^*)} -w(Y|X,X^*)\nabla_\gamma \log p_\gamma(Y|X) \\
\nabla_\beta Loss_C &= \mathop{E}_{Y\sim\tilde{p}(Y|X^*)} \nabla \log p_\alpha(Y|Y^*) + \mathop{E}_{Y\sim p_\alpha(Y|Y^*)} -R(Y,Y^*)\nabla \log p_\alpha(Y|Y^*) \\
\nabla_\alpha Loss_C &= \mathop{E}_{X\sim\tilde{p}(X|Y^*)} \nabla \log p_\beta(X|X^*) + \mathop{E}_{X\sim p_\beta(X|X^*)} -R(X,X^*)\nabla \log p_\beta(X|X^*)
\end{aligned} \tag{15}$$

To train the NMT system, parallel sentence pairs (X,Y) are firstly sampled from two independent cluster distributions and then translation confidence scores $w(Y|X,X^*), w(X|Y,Y^*)$ are leveraged to guide the training. The derivatives w.r.t the cluster contain two elements, candidates sampled from translation system, and candidates sampled from cluster itself. The two components together ensure parallel cluster's translatability and the similarity to the ground truth.

System-D With the same loss function in system-C, translation broadcasting strategy is leveraged to compute derivatives, instead of parallel sampling, and the gradients is shown as:

$$\begin{aligned}
\nabla_\eta Loss_D &= \mathop{E}_{Y\sim p_\alpha(Y|Y^*)} \mathop{E}_{X\sim p_\eta(X|Y)} \frac{-p_\beta(X|X^*)}{\tilde{p}(X|Y^*)} \nabla_\eta \log p_\eta(X|Y) \\
&= \mathop{E}_{Y\sim p_\alpha(Y|Y^*)} \mathop{E}_{X\sim p_\eta(X|Y)} -c(X|X^*,Y^*)\nabla_\eta \log p_\eta(X|Y) \\
\nabla_\gamma Loss_D &= \mathop{E}_{X\sim p_\beta(X|X^*)} \mathop{E}_{Y\sim p_\gamma(Y|X)} \frac{-p_\alpha(Y|Y^*)}{\tilde{p}(Y|X^*)} \nabla_\gamma \log p_\gamma(Y|X) \\
&= \mathop{E}_{X\sim p_\beta(X|X^*)} \mathop{E}_{Y\sim p_\gamma(Y|X)} -c(Y|Y^*,X^*)\nabla_\gamma \log p_\gamma(Y|X) \\
\nabla_\beta Loss_D &= \nabla_\beta Loss_C \\
\nabla_\alpha Loss_D &= \nabla_\alpha Loss_C
\end{aligned} \tag{16}$$

System-D works quite similar as system-B but differs in that cluster confidence scores $c(X|X^*,Y^*), c(Y|Y^*,X^*)$ are leveraged in training NMT, hence it is more abundant than task-level rewards ($\tilde{R}(X,X^*)$ and $\tilde{R}(Y,Y^*)$). System-D adopts the same gradient formulas in system-C to update the clusters.

The details of the training algorithm for system-A,B,C,D are shown in Algorithm 1:

Algorithm 1 Cluster-to-Cluster Learning Framework

procedure PRE-TRAINING
  Initialize $p_\gamma(Y|X)$, $p_\eta(X|Y)$, $p_\beta(X|X^*)$, $p_\alpha(Y|Y^*)$ with random weights η, γ, β and α
  Pre-train the $p_\gamma(Y|X)$ and $p_\eta(X|Y)$ via Maximum Likelihood Estimation
  Generate Translations $\tilde{X}$ and $\tilde{Y}$ through $p_\eta(X|Y)$ and $p_\gamma(Y|X)$
  Pre-train $p_\beta(X|X^*)$ and $p_\alpha(Y|Y^*)$ with data pair $(X, \tilde{X})$ and $(Y, \tilde{Y})$
end procedure
procedure ALTERNATE ITERATIVE STRATEGY
  while Not Converged **do**
    Sample a random example (X^*, Y^*) from dataset D
    if System-A **then**
      Generate N sequence X, Y from $p_\beta(X|X^*)$ and $p_\alpha(Y|Y^*)$
      Evaluate X, Y with task-level reward and discard sequences (X, Y) below M
      Compute gradient for both $p_\eta(X|Y)$ and $p_\gamma(Y|X)$ based on Equation 12
    else if System-B **then**
      Generate sequence X, Y from $p_\beta(X|X^*)$ and $p_\alpha(Y|Y^*)$
      Evaluate X, Y with task-level reward and discard sequences (X, Y) below M
      Translate corresponding sequence $\hat{Y}$, $\hat{X}$ via translation system and sampled X, Y
      Compute gradient for both $p_\eta(X|Y)$ and $p_\gamma(Y|X)$ based on Equation 13
    else if System-C **then**
      Generate sequence X, Y from $p_\beta(X|X^*)$ and $p_\alpha(Y|Y^*)$
      Evaluate X, Y with task-level reward and discard sequences (X, Y) below M
      Compute gradient for $p_\eta(X|Y)$,$p_\gamma(Y|X)$,$p_\beta(X|X^*)$,$p_\alpha(Y|Y^*)$ based on Equation 15
    else if System-D **then**
      Generate N sequence X, Y from $p_\beta(X|X^*)$ and $p_\alpha(Y|Y^*)$
      Evaluate X, Y with task-level reward and discard sequences (X, Y) below M
      Translate corresponding sequence $\hat{Y}$, $\hat{X}$ via translation system and sampled X, Y
      Compute gradient for $p_\eta(X|Y)$,$p_\gamma(Y|X)$,$p_\beta(X|X^*)$,$p_\alpha(Y|Y^*)$ based on Equation 16
    end if
    Apply gradient descent: $\eta = \eta - lr\nabla_\eta Loss$ and $\gamma = \gamma - lr\nabla_\gamma Loss$
    if System-C or System-D **then**
      Apply gradient descent: $\beta = \beta - lr\nabla_\beta Loss$ and $\alpha = \alpha - lr\nabla_\alpha Loss$
    end if
  end while
end procedure

5 EXPERIMENT

5.1 MACHINE TRANSLATION RESULTS

To evaluate our cluster-to-cluster NMT framework on different-sized (small-data, medium-data and large-data) and different-lingual (German-English and Chinese-English) translation tasks, we conduct experiments on three datasets (IWSLT, LDC, WMT). For more details about the datasets, please refer to Appendix C. For comparability, we follow the existing papers to adopt similar network architectures, and apply learning rate annealing strategy described in Wu et al. (2016) to further boost our baseline NMT system. In our experiments, we design both the NMT and adaptive cluster models based on one-layered encoder-decoder network (Chung et al., 2014) with a maximum sentence length of 62 for both the encoder and decoder. During training, ADADELTA (Zeiler, 2012) is adopted with $\epsilon = 10^{-6}$ and $\rho = 0.95$ to separately optimize the NMT's and adaptive cluster's parameters. During decoding, a beam size of 8 is used to approximate the full search space. We compute the threshold similarity M via sentence-BLEU, some small-scaled experiments indicate $M = 0.5$ yields best performance, so we simply stick to this setting throughout all the experiments. To prevent too much hyper-parameter tuning in building the inadaptive cluster, we follow Norouzi et al. (2016) to select the best temperature $\tau = 0.8$ in all experiments. For comparison, RAML and RL systems are also implemented with the same sequence-to-sequence attention model,

following Norouzi et al. (2016) and Williams (1992). For more details of our RL's and RAML's implementations, please refer to Appendix A.

IWSLT2014 German-English We can see from Table 3 that our system-D achieves significant improvements on both directions. Though our baseline system is already extremely strong, using cluster-to-cluster framework can further boost the NMT system by over 1.0 BLEU point.

Direction	DE2EN		EN2DE	
Methods	Baseline	Model	Baseline	Model
MIXER (Ranzato et al., 2015)	20.10	21.81	-	-
BSO (Wiseman & Rush, 2016)	24.03	26.36	-	-
A-C (Bahdanau et al., 2016)	27.56	28.53	-	-
Softmax-Q (Ma et al., 2017)	27.66	28.77	-	-
Our implementation of RL (Williams, 1992)	29.10	29.70	24.40	24.75
Our implementation of RAML (Norouzi et al., 2016)	29.10	29.47	24.40	24.86
cluster-to-cluster NMT (System-A)	29.10	30.08	24.40	25.15
cluster-to-cluster NMT (System-B)		30.10		25.20
cluster-to-cluster NMT (System-C)		30.30		25.35
cluster-to-cluster NMT (System-D)		**30.50**		**25.46**

Table 3: Experimental results on IWSLT-2014 German-English Machine Translation Task

LDC Chinese-English We can see from Table 4 that System-D achieves the best performance in the CH2EN translation direction, while System-D achieves the best performance on most test data sets in the EN2CH translation direction. The average improvement over the baseline systems for both EN2CH and CH2EN direction are around 1.0 BLEU.

Direction	CH2EN (NIST03/05/06)		EN2CH (NIST03/05/06)	
Methods	Baseline	Model	Baseline	Model
Our RL	39.04/37.13/39.13	40.98/39.23/39.27	17.57/16.38/17.31	18.44/16.98/**17.99**
Our RAML	39.04/37.13/39.13	40.28/37.28/37.20	17.57/16.38/17.31	17.83/16.52/16.79
System-A	39.04/37.13/39.13	40.11/38.68/39.41	17.57/16.38/17.31	**18.81**/17.32/17.82
System-B		41.55/38.93/39.19		18.53/16.99/17.58
System-C		41.69/39.05/39.17		18.50/**17.36/17.88**
System-D		**41.78/39.15/39.48**		18.71/17.13/17.77

Table 4: Experimental results on NIST Chinese-English Machine Translation Task

WMT2014 German-English We can see from Table 5 that system-C achieves the strongest result on both WMT14 EN-DE and DE-EN tasks, which outperforms the baseline system by over 1.1 BLEU points. It's worth noting that our one-layer RNN model even outperforms the deep multi-layer RNN model of Zhou et al. (2016) and Luong et al. (2015), which contain a stack of 4-7 LSTM layers. By using cluster-to-cluster framework, our one-layer RNN model can fully exploit the dataset and learn to generalize better.

5.2 Result Analysis

From the above 24 parallel cluster-to-cluster experiments, we observe general improvements over the fine-tuned baseline systems as well as our implemented RL/RAML systems. To understand the strength of our cluster-to-cluster framework, we give more detailed comparisons with existing competing algorithms as below:

Comparison with RAML From the above three tables, we can observe general improvements yielded by RAML algorithm on different tasks (except LDC Chinese-English), but RAML still suffers from two problems: on one hand, RAML's benefits is restricted by its neglect of the input variabilities, and on the other hand, without considering semantic contexts and language fluency,

Direction	DE2EN		EN2DE	
Methods	Baseline	Model	Baseline	Model
Attention-NMT (Bahdanau et al., 2014)	-	-	16.46	18.79
Attention-NMT with LV (Jean et al., 2014)	-	-	16.95	19.40
Local-p Attention NMT (Luong et al., 2015)	-	-	19.0	20.90
Deep-Att (Zhou et al., 2016)	-	-	16.5	20.60
Our implementation of RL (Williams, 1992)	25.13	25.78	20.13	20.66
Our implementation of RAML (Norouzi et al., 2016)	25.13	25.54	20.12	20.54
cluster-to-cluster NMT (System-A)	25.13	25.94	20.12	20.90
cluster-to-cluster NMT (System-B)		25.94		20.71
cluster-to-cluster NMT (System-C)		**26.26**		**21.30**
cluster-to-cluster NMT (System-D)		26.09		20.06

Table 5: Experimental results on WMT-2014 German-English Machine Translation Task

RAML's random replacement strategy may introduce noisy and wrong bilingual pairs to hurt the translation performance (like in LDC Chinese-English translation task). Our adaptive cluster takes into account more semantic contexts to enclose more rational paraphrases, and the bilateral augmentation also empowers the model more chance to access various inputs.

Comparison with RL We can also observe prevalent improvements yielded by RL algorithm Bahdanau et al. (2016); Ranzato et al. (2015). Exposing the model to self-generated translation can improve the performance. Our methods inherit this merit and further enhance it with source and target clusters, which can improve the model with more sampled bilingual pairs from both source and target sides.

Comparison between four parallel systems Among our proposed four parallel systems, system-C and D achieve better performances than A and B throughout different experiments, which confirms the advantages of the adaptive clusters. The adaptive cluster is more flexible and target optimized than inadaptive cluster. Unlike the payoff distribution used in inadaptive cluster which only takes task-level reward into account, the adaptive cluster learns more sophisticated criterion and thus assigns more rational probability to sampled candidates. We give more detailed analysis and visualization in the appendices to demonstrate how the source and target clusters look like.

5.3 Learning curve and Case studies

We demonstrate the learning curves of four systems and visualize some adaptive clusters in Appendix D and Appendix E, which give a more intuition about cluster-to-cluster learning.

6 Conclusion

In this paper, we propose a cluster-to-cluster learning framework and incorporate this concept into neural machine translation. Our designed systems have proved to be efficient in helping current NMT model to generalize in both source and target sides. In the cluster-to-cluster framework, the cooperation of four agents can augment valuable samples and alleviate data sparsity, and achieve significant improvement compared with strong baseline systems. We believe the concept of cluster-to-cluster learning can be applicable to a wide range of natural language or computer vision tasks, which will be explored in the future.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 1, 2, 10, 12, 14

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. An actor-critic algorithm for sequence prediction. *arXiv preprint arXiv:1607.07086*, 2016. 1, 2, 9, 10

Thorsten Brants, Ashok C Popat, Peng Xu, Franz J Och, and Jeffrey Dean. Large language models in machine translation. In *In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Citeseer, 2007. 3

Yong Cheng, Wei Xu, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. Semi-supervised learning for neural machine translation. *arXiv preprint arXiv:1606.04596*, 2016. 1, 3

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. 1, 6, 12

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 8

Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. A character-level decoder without explicit segmentation for neural machine translation. *arXiv preprint arXiv:1603.06147*, 2016. 14

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tieyan Liu, and Wei-Ying Ma. Dual learning for machine translation. In *Advances in Neural Information Processing Systems*, pp. 820–828, 2016. 1, 3

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. On using very large target vocabulary for neural machine translation. *arXiv preprint arXiv:1412.2007*, 2014. 10, 14

Alex M Lamb, Anirudh Goyal ALIAS PARTH GOYAL, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. In *Advances In Neural Information Processing Systems*, pp. 4601–4609, 2016. 2

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015. 9, 10, 14

Xuezhe Ma, Pengcheng Yin, Jingzhou Liu, Graham Neubig, and Eduard Hovy. Softmax q-distribution estimation for structured prediction: A theoretical interpretation for raml. *arXiv preprint arXiv:1705.07136*, 2017. 1, 3, 9

Mohammad Norouzi, Samy Bengio, Navdeep Jaitly, Mike Schuster, Yonghui Wu, Dale Schuurmans, et al. Reward augmented maximum likelihood for neural structured prediction. In *Advances In Neural Information Processing Systems*, pp. 1723–1731, 2016. 1, 2, 3, 4, 5, 8, 9, 10, 13

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015. 1, 2, 9, 10

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*, 2015a. 1, 3

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015b. 14

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. Minimum risk training for neural machine translation. *arXiv preprint arXiv:1512.02433*, 2015. 2

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998. 2

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992. 5, 7, 9, 10

Sam Wiseman and Alexander M Rush. Sequence-to-sequence learning as beam-search optimization. *arXiv preprint arXiv:1606.02960*, 2016. 9

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016. 8, 14

Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012. 8

Jiajun Zhang and Chengqing Zong. Exploiting source-side monolingual data in neural machine translation. In *EMNLP*, pp. 1535–1545, 2016. 1, 3

Jie Zhou, Ying Cao, Xuguang Wang, Peng Li, and Wei Xu. Deep recurrent models with fast-forward connections for neural machine translation. *arXiv preprint arXiv:1606.04199*, 2016. 9, 10

Appendices

A SYSTEM-DESIGN

Sequence to sequence problem (machine translation) can be considered to produce an output sequence $Y = (y_1, y_2, \ldots, y_T), y_t \in A$ given an input X. Given input-target pairs (X, Y^*), the generated sequence Y on test is evaluated with task-specific score $R(Y, Y^*)$. Recurrent neural networks have been widely used in sequence to sequence prediction tasks. As proposed in Cho et al. (2014) and Bahdanau et al. (2014), the basic idea is to first encode the input sequence as a variable-length feature vectors, then apply attention mechanism to compute weighted average over the input vectors and summarize a context vector, with which, previous hidden states and previous label are fed into the decoder RNN to predict the next state and its label. In our approach, attention-based encoder-decoder (Bahdanau et al., 2014; Cho et al., 2014) is leveraged for both the translation and cluster models, shown as:

$$y_t \sim g(s_{t-1}, c_{t-1}) \tag{17}$$

$$s_t = f(s_{t-1}, c_{t-1}, e(y_t)) \tag{18}$$

$$\alpha_t = \beta(s_t, (h_1, \ldots, h_L)) \tag{19}$$

$$c_t = \sum_{j=1}^{L} \alpha_{t,j} h_j \tag{20}$$

A.1 RL NMT

In order to train our RL system as well as adaptive cluster, we need to define a task-level reward as driving signal. Instead of directly applying BLEU or other evaluation metric, we advocate to use a surrogate n-gram match interpolation, as shown as:

$$R(Y, Y^*) = 0.4 * N_4 + 0.3 * N_3 + 0.2 * N_2 + 0.1 * N_1 \tag{21}$$

where N_n denotes the number of n-gram match between Y and Y^*. In order to alleviate sequence-reward sparseness, we further split it as a series of local reward to drive model's policy search at every time step. Formally, we write the step-wise reward $r(y_t|y_{1:t-1}, Y^*)$ as following.

$$r(y_t|y_{1:t-1}, Y^*) = \begin{cases} 1.0; & 0 < N(Y^*, y_{t-3:t}) \leq N(Y, y_{t-3:t}) \\ 0.6; & 0 < N(Y^*, y_{t-2:t}) \leq N(Y, y_{t-3:t}) \\ 0.3; & 0 < N(Y^*, y_{t-1:t}) \leq N(Y, y_{t-3:t}) \\ 0.1; & 0 < N(Y^*, y_t) \leq N(Y, y_{t-3:t}) \\ 0.0; & otherwise \end{cases} \tag{22}$$

where $N(Y, \tilde{Y})$ represents the occurrence of n-gram $\tilde{Y}$ in sequence Y, specifically, if a certain n-sequence $y_{t-n+1:t}$ appears in reference and it's not repeating more than needed, then we assign a corresponding matching score to y_t, the policy gradient is described as:

$$\nabla_\theta = \sum_t r(y_t|y_{1:t-1}, Y^*) \nabla_\theta \log p_\theta(y_t|y_{1:t-1}, X) \tag{23}$$

A.2 RAML NMT

In order to sample from the intractable payoff distribution for system-A/B as well as our implemented RAML system, we adopt stratified sampling technique described in Norouzi et al. (2016). Given a sentence Y^*, we first sample an edit distance m, and then randomly select m positions to replace the original labels. For each sentence, we randomly sample four candidates to perform RAML training.

$$\nabla_\theta = \mathop{E}_{Y \sim q(Y|Y^*)} \nabla \log p_\theta(Y|X) \tag{24}$$

B MATHEMATICAL ANALYSIS

We optimize the model parameters of our cluster-to-cluster models by minimizing the lower-bound KL-divergence instead of maximizing the original correspondence score, to characterize the difference between the two objective function, we analyze the relationships between these two functions below:

Proposition 1. For any two non-zero distributions $p(Y|Y^*)$ and $p(Y|X^*)$, we can construct an auxiliary distribution $\tilde{p}(Y|X^*, Y^*)$ as $\frac{p(Y|Y^*)p(Y|X^*)}{\sum_{Y'} p(Y'|Y^*)p(Y'|X^*)}$ to get

$$CR_{c\to c}(X \to Y) + H(p(Y|Y^*)) = KL(p(Y|Y^*)||p(Y|X^*)) + CE(p(Y|Y^*), \tilde{p}(Y|X^*, Y^*)) \tag{25}$$

This reveals the relationship between our regularized cluster-to-cluster correspondence objective and the lower-bound KL-divergence, and their difference can be expressed as the cross-entropy between $p(Y|Y^*)$ and $\tilde{p}(Y|X^*, Y^*)$.

Proposition 2. The cross-entropy between $p(Y|Y^*)$ and $\tilde{p}(Y|X^*, Y^*)$ achieves minimum if and only if $p(Y|X^*) = Uniform(Y)$:

$$\mathop{\text{argmin}}_{p(Y|X^*)} CE(p(Y|Y^*), \frac{p(Y|Y^*)p(Y|X^*)}{\sum_{Y'} p(Y'|Y^*)p(Y|X^*))}) = Uniform(Y) \tag{26}$$

According to the above two propositions, the gap between two objective functions decreases as $p(Y|X^*)$ approaches uniform distribution, therefore, by introducing an entropy regularization $H(p(Y|X^*))$ during training, the gap can be further minimized. Although we have not yet incorporated this additional uniform regularization term into cluster-to-cluster framework, this is an interesting research direction.

Here is the proof of proposition 1:

$$\begin{aligned}
&CR_{c\to c}(X \to Y) + H(p(Y|Y^*)) - KL(p(Y|Y^*)||p(Y|X^*)) \\
=&\log \sum_X \sum_Y p(X|X^*)p(Y|Y^*)p(Y|X) + H(p(Y|Y^*)) - KL(p(Y|Y^*)||p(Y|X^*)) \\
=&\log \sum_Y p(Y|Y^*)p(Y|X^*) + H(p(Y|Y^*)) - KL(p(Y|Y^*)||p(Y|X^*)) \\
=&\sum_Y p(Y|Y^*) \log \sum_{Y'} p(Y'|Y^*)p(Y'|X^*) + H(p(Y|Y^*)) - KL(p(Y|Y^*)||p(Y|X^*)) \\
=&\sum_Y p(Y|Y^*) \log \frac{\sum_{Y'} p(Y'|Y^*)p(Y'|X^*)}{p(Y|X^*)} + H(p(Y|Y^*)) \\
=&\sum_Y p(Y|Y^*) \log \frac{\sum_{Y'} p(Y'|Y^*)p(Y'|X^*)}{p(Y|X^*)p(Y|Y^*)} \\
=&CE(p(Y|Y^*), \tilde{p}(Y|X^*, Y^*))
\end{aligned} \tag{27}$$

Here is the proof of proposition 2:

$$\mathop{\text{argmin}}_{\tilde{p}(Y|X^*,Y^*)} CE(p(Y|Y^*), \tilde{p}(Y|X^*, Y^*)) = p(Y|Y^*) \tag{28}$$

which can be further written as:

$$\frac{p(Y|Y^*)p(Y|X^*)}{\sum_{Y'} p(Y'|Y^*)p(Y'|X^*)} = p(Y|Y^*) \tag{29}$$

therefore, we can derive:

$$p(Y|X^*) = \sum_{Y'} p(Y'|Y^*)p(Y'|X^*) = Constant = \frac{1}{|Y|} \tag{30}$$

Since both cluster and translation confidence score $c(Y|Y^*, X^*)$ and $w(Y|X, X^*)$ require computing the marginalized probability $p(Y|X^*)$ known to be intractable for variable-length sequences, here we adopt different mechanisms to approximate them. In system-A and C, we simplify $\sum_X p_\gamma(Y|X)p_\beta(X|X^*)$ as $p_\gamma(Y|X^*)$ to approximate $w(Y|X, X^*)$ as $\frac{p_\gamma(Y|X)}{p_\eta(Y|X^*)}$. In system-B and D, since Y is broadcast through the translation system, the marginalized probability $\tilde{p}(Y|X^*)$ is close to one, we discard this factor and approximate $c(Y|Y^*, X^*)$ as $\hat{R}(Y, Y^*)/p_\alpha(Y|Y^*)$.

C Dataset Description

IWSLT2014 Dataset The IWSLT2014 German-English training data set contains 153K sentences while the validation data set contains 6,969 sentences pairs. The test set comprises dev2010, dev2012, tst2010, tst2011 and tst2012, and the total amount are 6,750 sentences. We adopt 512 as the length of RNN hidden stats and 256 as embedding size. We use bidirectional encoder and initialize both its own decoder states and coach's hidden state with the learner's last hidden state. The experimental results for IWSLT2014 German-English and English-German Translation Task are summarized in Table 3.

LDC Dataset The LDC Chinese-English training corpus consists of 1.25M parallel sentence, 27.9M Chinese words and 34.5M English words. We choose NIST 2003 as our development set and evaluate our results on NIST 2005, NIST2006. We adopt a similar setting as IWSLT German-English translation task, we use 512 as hidden size for GRU cell and 256 as embedding size. The experimental results for LDC Chinese-English translation task are listed in Table 4.

WMT2014 Dataset The WMT2014 German-English training data consists of 4.5M sentences pairs (116M English, 110M German words), we follow Sennrich et al. (2015b) to apply byte pair encoding (BPE) to deal with the massive number of rare words in this translation task. Newstest2012 and newstest2013 are used as development set to select best hyperparameters. Translation performance are reported in case-sensitive BLEU on newstest2014. Similar to Jean et al. (2014) and Bahdanau et al. (2014), we use 2014 as length of RNN hidden units and 512 as embedding size. Unlike the previous works (Luong et al., 2015; Chung et al., 2016; Wu et al., 2016), which uses multi-layer stacking LSTM/GRU to boost the NMT performance, our model only use one-layer encoder-decoder architecture. Thus, our results are not directly comparable with their results. The experimental results for WMT14 German-English Translation task are listed in Table 5.

D Learning Curve

Here we draw the learning curve on WMT2014 German-English translation task in Figure 2 to demonstrate a more intuitive comparison between system-A/B and system-C/D. From these curves, we can observe steady improvements for system-C/D and unstable vibration for system-A/B. Though system-C/D starts from a lower point, they gradually find better convergence points than system-A/B. We attribute the smoothness to adaptive cluster's capability to interact with the NMT models in system-C and D, in other words, the adaptive cluster can accordingly update its distribution to stabilize NMT's training.

We also demonstrate the learning curve of the adaptive cluster during the joint training in Figure 3. We use cluster distribution to perform beam-search and find out its most-likely sampled candidates and evaluate their BLEU against the ground truth for every iteration. We can observe that cluster's BLEU improvement is highly correlated with NMT's BLEU improvements. When the translation system becomes better, the adaptive cluster will concentrate more on the ground truth.

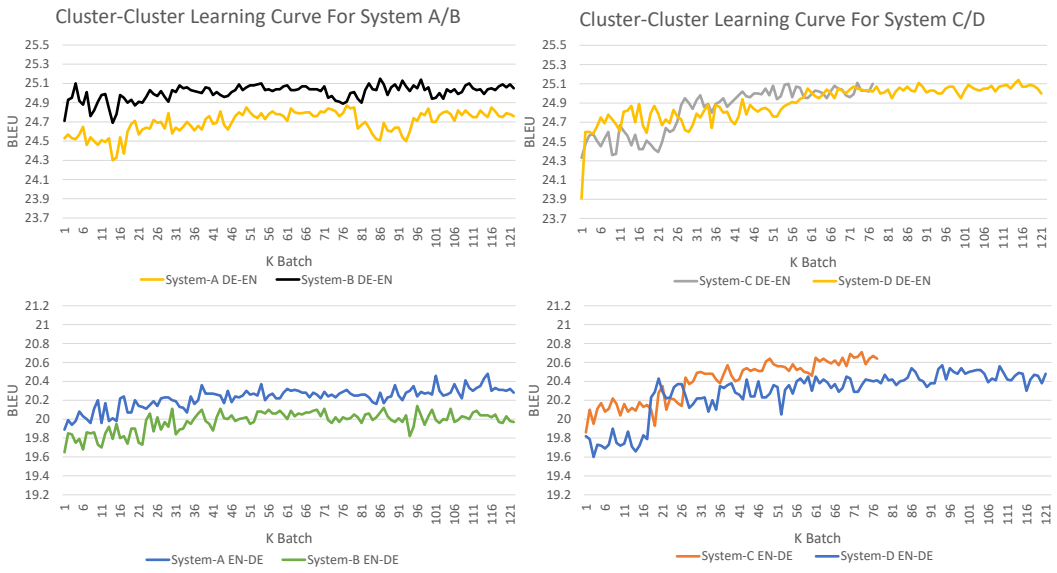

Figure 2: The trend of BLEU on WMT2014 Dev set for cluster-cluster system

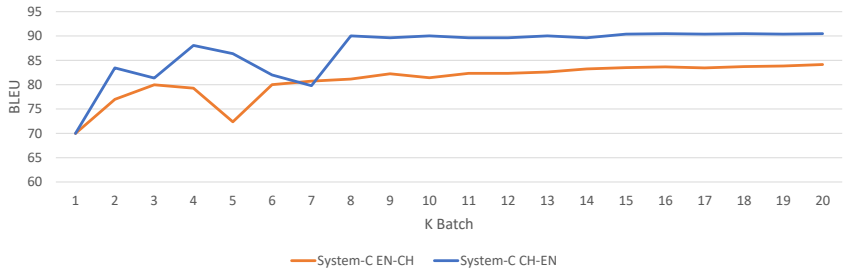

Figure 3: Adaptive cluster's BLEU trend on WMT2014 Dev set

E CASE STUDY

In order to give a more intuitive view about what the cluster distribution looks like, we draw some samples from the well-trained cluster distribution in LDC Chinese-English Translation task as shown in Table 6. we can observe that most of the paraphrases are based on three types of modification, namely form changing, synonym replacement as well as simplification. Most of the modifications does not alter the original meaning of the ground truth. Encompassing more expressions with close meanings can ease the data sparseness problem, and enhance its generalization ability. We here draw two samples from source and target clusters in Figure 4, which demonstrates how point-point correspondence can be expanded into cluster-to-cluster correspondence.

Property	Synonym Replacement
Reference	taihsi natives seeking work in other parts of the country are given a thorough UNK before being **hired**, and later their colleagues maintain a healthy distance at first .
Cluster	taihsi natives seeking work in other parts of the country are given a thorough UNK before being **employed**, and later their colleagues maintain a healthy distance at first .
Property	Simplification
Reference	i once took mr tung chee - hwa to a squatter area where he found beyond imagination that a narrow alley **could have accommodated so many people**.
Cluster	i once took mr tung chee - hwa to a squatter area where he fo und beyond imagination that a narrow alley **have a lot of people**.

Table 6: Samples drawn from cluster distribution in LDC Chinese-English Translation task

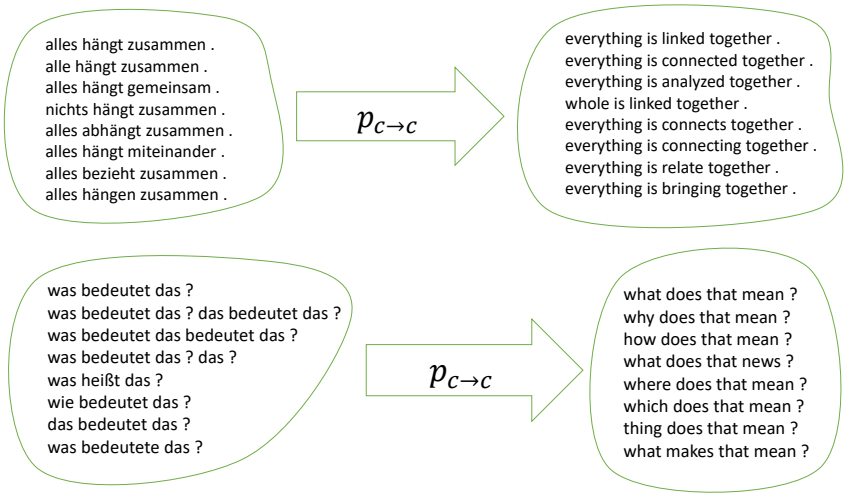

Figure 4: Cluster Visualization