Shen SQ, Liu Y, Sun MS. Optimizing non-decomposable evaluation metrics for neural machine translation. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 32(4): 796–804 July 2017. DOI 10.1007/s11390-017-1760-9

Optimizing Non-Decomposable Evaluation Metrics for Neural Machine Translation

Shi-Qi Shen[1,2,3], *Student Member, CCF*, Yang Liu[1,2,3,4,*], *Senior Member, CCF*
and Mao-Song Sun[1,2,3,4], *Senior Member, CCF*

[1] *Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China*
[2] *State Key Laboratory of Intelligent Technology and Systems, Tsinghua University, Beijing 100084, China*
[3] *Tsinghua National Laboratory for Information Science and Technology, Tsinghua University, Beijing 100084, China*
[4] *Jiangsu Collaborative Innovation Center for Language Ability, Jiangsu Normal University, Xuzhou 221009, China*

E-mail: vicapple22@gmail.com; liuyang2011@tsinghua.edu.cn; sms@mail.tsinghua.edu.cn

Received December 20, 2016; revised May 20, 2017.

Abstract While optimizing model parameters with respect to evaluation metrics has recently proven to benefit end-to-end neural machine translation (NMT), the evaluation metrics used in the training are restricted to be defined at the sentence level to facilitate online learning algorithms. This is undesirable because the final evaluation metrics used in the testing phase are usually non-decomposable (i.e., they are defined at the corpus level and cannot be expressed as the sum of sentence-level metrics). To minimize the discrepancy between the training and the testing, we propose to extend the minimum risk training (MRT) algorithm to take non-decomposable corpus-level evaluation metrics into consideration while still keeping the advantages of online training. This can be done by calculating corpus-level evaluation metrics on a subset of training data at each step in online training. Experiments on Chinese-English and English-French translation show that our approach improves the correlation between training and testing and significantly outperforms the MRT algorithm using decomposable evaluation metrics.

Keywords neural machine translation, training criterion, non-decomposable evaluation metric

1 Introduction

The past several years have witnessed the rapid development of end-to-end neural machine translation (NMT)[1-3]. Unlike conventional statistical machine translation (SMT) models that rely on the linear combination of hand-crafted features[4-5], NMT directly learns distributed representations from data and translates between natural languages via non-linear transformations using neural networks. Due to the recent introduction of LSTMs (long short-term memory)[6], GRUs (gated recurrent units)[7] and the attention mechanism[3] to handle non-local dependencies, NMT systems have begun to deliver translation performance superior to SMT.

Recently, several researchers have endeavored to improve NMT by introducing new training criteria that take evaluation metrics into consideration[8-9]. [8] identifies two drawbacks of the standard maximum likelihood criterion: 1) the models are not exposed to model predictions during training and 2) loss functions are defined only at the word level. It proposes the Mixed Incremental Cross-Entropy Reinforce (MIXER) algorithm to enable incremental learning and the combination of both REINFORCE[10] and cross-entropy loss functions. [9] introduces minimum risk training[4,11-13]

Regular Paper

Special Issue on Deep Learning

This work is supported by the National Natural Science Foundation of China under Grant Nos. 61522204, 61432013, and the National High Technology Research and Development 863 Program of China under Grant No. 2015AA015407, also supported by the Singapore National Research Foundation under Its International Research Centre@Singapore Funding Initiative, and administered by the IDM (Interactive Digital Media) Programme.

[*] Corresponding Author

into NMT to optimize model parameters with respect to evaluation metrics such as BLEU (bilingual evaluation understudy)[14] and TER[15]. Their experiments show that optimizing NMT models with respect to evaluation metrics leads to significant improvements over maximum likelihood estimation.

Despite the advantages of introducing evaluation metrics into training, existing work still faces a major challenge: evaluation metrics are usually defined at the corpus level and thus non-decomposable over sentences. More specifically, for most evaluation metrics such as BLEU, the loss on a corpus (i.e., a set of sentences) cannot be calculated as the sum of losses on individual sentences. This is problematic for training NMT models because online learning algorithms that are widely used in NMT only consider single sentences rather than the entire corpus. To alleviate this problem, a conventional solution is to use sentence-level evaluation metrics instead[8-9]. Apparently, the downside is that optimizing with respect to sentence-level evaluation metrics in the training does not necessarily maximize the corpus-level counterparts in the testing. Therefore, the discrepancy between the training and the testing potentially deteriorates the translation performance of NMT.

In this work, we propose to optimize NMT model parameters with respect to non-decomposable evaluation metrics. We extend the minimum risk training algorithm[9] to include non-decomposable evaluation metrics. Approximate non-decomposable loss functions are used in this work to eliminate the discrepancy between training and testing. A major advantage of our approach is that online training can still be used. Although approximations of non-decomposable evaluation metrics for online training have been widely used in traditional SMT[16-19], to the best of our knowledge, this work is the first effort to introduce non-decomposable evaluation metrics into end-to-end NMT training. Experiments on Chinese-English, English-French translation tasks show that our approach leads to significant improvements over optimizing models with respect to sentence-level evaluation metrics.

2 Background

2.1 Maximum Likelihood Estimation

Given a source-language sentence $\boldsymbol{x} = x_1, \ldots, x_m, \ldots, x_M$ that contains M words and a target-language sentence $\boldsymbol{y} = y_1, \ldots, y_n, \ldots, y_N$ that contains N words, end-to-end NMT[1-3] directly models the translation probability with a single neural network:

$$P(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta}) = \prod_{n=1}^{N} P(y_n|\boldsymbol{x}, \boldsymbol{y}_{<n};\boldsymbol{\theta}),$$

where $\boldsymbol{\theta}$ is a set of model parameters and $\boldsymbol{y}_{<n} = y_1, \ldots, y_{n-1}$ is a partial translation.

Given a set of training examples $\{(\boldsymbol{x}^{(s)}, \boldsymbol{y}^{(s)})\}_{s=1}^{S}$, the standard training objective is to find the model parameters that maximize the log-likelihood of the training data:

$$\hat{\boldsymbol{\theta}}_{\mathrm{MLE}} = \underset{\boldsymbol{\theta}}{\mathrm{argmax}} \Big\{\mathcal{L}(\boldsymbol{\theta})\Big\},$$

where

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &= \sum_{s=1}^{S} \log P(\boldsymbol{y}^{(s)}|\boldsymbol{x}^{(s)};\boldsymbol{\theta}) \\ &= \sum_{s=1}^{S}\sum_{n=1}^{N^{(s)}} \log P(y_n^{(s)}|\boldsymbol{x}^{(s)}, \boldsymbol{y}_{<n}^{(s)};\boldsymbol{\theta}). \qquad (1)\end{aligned}$$

We use $N^{(s)}$ to denote the length of the s-th target sentence $\boldsymbol{y}^{(s)}$.

[8] indicates that the maximum likelihood criterion suffers from the exposure bias problem: the models are only exposed to the ground-truth training data rather than model predictions. Moreover, loss functions are defined only at the word level instead of the sentence level.

2.2 Minimum Risk Training with Decomposable Evaluation Metrics

As maximum likelihood estimation does not consider evaluation metrics that quantify translation quality, [9] introduces minimum risk training to optimize model parameters with respect to sentence-level evaluation metrics:

$$\hat{\boldsymbol{\theta}}_{\mathrm{sMRT}} = \underset{\boldsymbol{\theta}}{\mathrm{argmin}} \Big\{\mathcal{R}(\boldsymbol{\theta})\Big\},$$

where

$$\begin{aligned}&\mathcal{R}(\boldsymbol{\theta}) \\ &= \sum_{s=1}^{S} \mathbb{E}_{\tilde{\boldsymbol{y}}^{(s)}|\boldsymbol{x}^{(s)};\boldsymbol{\theta}}(\delta(\tilde{\boldsymbol{y}}^{(s)}, \boldsymbol{y}^{(s)})) \\ &= \sum_{s=1}^{S} \sum_{\tilde{\boldsymbol{y}}^{(s)} \in \mathcal{Y}(\boldsymbol{x}^{(s)})} P(\tilde{\boldsymbol{y}}^{(s)}|\boldsymbol{x}^{(s)};\boldsymbol{\theta}) \times \delta(\tilde{\boldsymbol{y}}^{(s)}, \boldsymbol{y}^{(s)}) \\ &\approx \sum_{s=1}^{S} \sum_{\tilde{\boldsymbol{y}}^{(s)} \in \mathcal{S}(\boldsymbol{x}^{(s)})} Q(\tilde{\boldsymbol{y}}^{(s)}|\boldsymbol{x}^{(s)};\boldsymbol{\theta}, \alpha) \times \delta(\tilde{\boldsymbol{y}}^{(s)}, \boldsymbol{y}^{(s)}), \quad (2)\end{aligned}$$

where $\mathcal{Y}(\boldsymbol{x}^{(s)})$ denotes the set of all translation candidates for the s-th source sentence $\boldsymbol{x}^{(s)}$ and $\delta(\tilde{\boldsymbol{y}}, \boldsymbol{y}^{(s)})$ is a sentence-level loss function that measures the discrepancy between model prediction $\tilde{\boldsymbol{y}}^{(s)}$ and ground truth $\boldsymbol{y}^{(s)}$. As the search space $\mathcal{Y}(\boldsymbol{x}^{(s)})$ is exponential, [9] approximates the posterior probability with a Q distribution defined over a subspace $\mathcal{S}(\boldsymbol{x}^{(s)})$ and uses a hyper-parameter α to control the smoothness of the Q distribution:

$$Q(\tilde{\boldsymbol{y}}^{(s)}|\boldsymbol{x}^{(s)};\boldsymbol{\theta},\alpha) = \frac{P(\tilde{\boldsymbol{y}}^{(s)}|\boldsymbol{x}^{(s)};\boldsymbol{\theta})^{\alpha}}{\sum_{\tilde{\boldsymbol{y}}\in\mathcal{S}(\boldsymbol{x}^{(s)})} P(\tilde{\boldsymbol{y}}|\boldsymbol{x}^{(s)};\boldsymbol{\theta})^{\alpha}}.$$

Although introducing evaluation metrics into the training proves to significantly boost translation quality for NMT[9], there remains a major challenge: the evaluation metrics used in the training and the testing phases are different. While sentence-level metrics are used in training, their corpus-level counterparts are used in testing. These corpus-level evaluation metrics are usually non-decomposable because they cannot be calculated as the sum of sentence-level metrics. This discrepancy might hinder translation performance. [9] shows that optimizing models with respect to sentence-level TER does not lead to the lowest corpus-level TER on the validation set.

Therefore, it is important to include corpus-level metrics in training to improve the correlation between training and testing.

3 Minimum Risk Training with Non-Decomposable Evaluation Metrics

3.1 Optimizing Non-Decomposable Metrics

Given a training set $\{(\boldsymbol{x}^{(s)}, \boldsymbol{y}^{(s)})\}_{s=1}^{S}$, we use $\boldsymbol{X} = \{\boldsymbol{x}^{(s)}\}_{s=1}^{S}$ and $\boldsymbol{Y} = \{\boldsymbol{y}^{(s)}\}_{s=1}^{S}$ to denote the source and the target parts, respectively. The new training objective that includes non-decomposable corpus-level evaluation metrics is defined as

$$\hat{\boldsymbol{\theta}}_{\mathrm{cMRT}} = \underset{\boldsymbol{\theta}}{\mathrm{argmin}} \Big\{ \mathcal{R}(\boldsymbol{\theta}) \Big\},$$

where

$$\begin{aligned}\mathcal{R}(\boldsymbol{\theta}) &= \mathbb{E}_{\tilde{\boldsymbol{Y}}|\boldsymbol{X};\boldsymbol{\theta}}(\Delta(\tilde{\boldsymbol{Y}}, \boldsymbol{Y})) \\ &= \sum_{\tilde{\boldsymbol{Y}}\in\mathcal{Y}(\boldsymbol{X})} P(\tilde{\boldsymbol{Y}}|\boldsymbol{X};\boldsymbol{\theta})\Delta(\tilde{\boldsymbol{Y}}, \boldsymbol{Y}). \qquad (3)\end{aligned}$$

Note that $\tilde{\boldsymbol{Y}} = \{\tilde{\boldsymbol{y}}^{(s)}\}_{s=1}^{S}$ is a set of model predictions for the training data and $\mathcal{Y}(\boldsymbol{X})$ is a set of all possible model prediction sets. Also note that the loss function $\Delta(\tilde{\boldsymbol{Y}}, \boldsymbol{Y})$ is defined at the corpus level and usually cannot be calculated as the sum of sentence-level losses (i.e., $\Delta(\tilde{\boldsymbol{Y}}, \boldsymbol{Y}) \neq \sum_{s=1}^{S} \delta(\tilde{\boldsymbol{y}}^{(s)}, \boldsymbol{y}^{(s)})$).

Assuming that translating an individual sentence is independent, we re-write (3) as

$$\mathcal{R}(\boldsymbol{\theta}) = \sum_{\tilde{\boldsymbol{Y}}\in\mathcal{Y}(\boldsymbol{X})} \prod_{s=1}^{S} P(\tilde{\boldsymbol{y}}^{(s)}|\boldsymbol{x}^{(s)};\boldsymbol{\theta})\Delta(\tilde{\boldsymbol{Y}}, \boldsymbol{Y}).$$

The partial derivative with respect to a model parameter $\boldsymbol{\theta}_i$ is calculated as

$$\begin{aligned}&\frac{\partial\mathcal{R}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}_i} \\ &= \mathbb{E}_{\tilde{\boldsymbol{Y}}|\boldsymbol{X};\boldsymbol{\theta}}\left\{\Delta(\tilde{\boldsymbol{Y}}, \boldsymbol{Y}) \times \sum_{s=1}^{S} \frac{\partial P(\tilde{\boldsymbol{y}}^{(s)}|\boldsymbol{x}^{(s)};\boldsymbol{\theta})/\partial\boldsymbol{\theta}_i}{P(\tilde{\boldsymbol{y}}^{(s)}|\boldsymbol{x}^{(s)};\boldsymbol{\theta})}\right\}. (4)\end{aligned}$$

Unfortunately, it is difficult to calculate partial derivatives using (4) because of the exponential space of $\mathcal{Y}(\boldsymbol{X})$. To make things worse, the non-decomposability of $\Delta(\tilde{\boldsymbol{Y}}, \boldsymbol{Y})$ makes online training algorithms such as Stochastic Gradient Descent (SGD) inapplicable.

3.2 Approximation

It is challenging to design an online learning framework for large-scale batch problems using non-decomposable loss functions. Besides approximating non-decomposable loss functions with decomposable variants[8-9], there also has been some recent progress[20-21] towards developing stochastic optimization methods for non-decomposable measures in the machine learning community. [20] proposes optimizing SVMperf-style objectives which requires to maintain large buffers. [21] considers optimizing for the performance measures that are concave or pseudo-linear in the canonical confusion matrix of the predictor. However, as these approaches focus on classification tasks, none of them can be readily applicable to NMT.

Another way to solve the problem is to use approximate corpus-level loss functions with a pseudo corpus, which has been widely used in SMT[16-19]. For example, [16] builds a pseudo corpus by using previous oracle translations, and k-best translations are generated to update the model. We borrow the idea from SMT and extend it to our NMT online training algorithm.

Our goal is to include corpus-level evaluation metrics in the training while retaining the benefits of online training. To do so, we have to achieve a compromise between sentence-level MRT ((2)) and corpus-level MRT ((3)). Our idea is to approximate optimizing models with respect to corpus-level evaluation metrics by replacing the full training set with a subset:

$$\begin{aligned}\mathcal{R}(\boldsymbol{\theta}) &= \sum_{s=1}^{S} \mathbb{E}_{\tilde{\boldsymbol{Y}}_s|\boldsymbol{X}_s;\boldsymbol{\theta}}(\Delta(\tilde{\boldsymbol{Y}}_s, \boldsymbol{Y}_s)) \\ &= \sum_{s=1}^{S} \sum_{\tilde{\boldsymbol{Y}}_s \in \mathcal{Y}(\boldsymbol{X}_s)} P(\tilde{\boldsymbol{Y}}_s|\boldsymbol{X}_s;\boldsymbol{\theta})\Delta(\tilde{\boldsymbol{Y}}_s, \boldsymbol{Y}_s),\end{aligned}$$

where $\boldsymbol{X}_s$ is a subset of the source part of the training set that contains the s-th source sentence (i.e., $\boldsymbol{X}_s \subset \boldsymbol{X} \wedge \boldsymbol{x}^{(s)} \in \boldsymbol{X}_s$), $\boldsymbol{Y}_s$ is a subset of the corresponding target part of the training set that contains the s-th target sentence (i.e., $\boldsymbol{Y}_s \subset \boldsymbol{Y} \wedge \boldsymbol{y}^{(s)} \in \boldsymbol{Y}_s$), and $\tilde{\boldsymbol{Y}}$ is a set of corresponding model predictions.

To facilitate the SGD algorithm, we use the last K sentences processed by SGD as the subset when dealing with the s-th sentence:

$$\begin{aligned}\boldsymbol{X}_s &= \{\boldsymbol{x}^{(s-K+1)}, \boldsymbol{x}^{(s-K+2)}, \ldots, \boldsymbol{x}^{(s)}\}, \\ \boldsymbol{Y}_s &= \{\boldsymbol{y}^{(s-K+1)}, \boldsymbol{y}^{(s-K+2)}, \ldots, \boldsymbol{y}^{(s)}\}, \\ \tilde{\boldsymbol{Y}}_s &= \{\tilde{\boldsymbol{y}}^{(s-K+1)}, \tilde{\boldsymbol{y}}^{(s-K+2)}, \ldots, \tilde{\boldsymbol{y}}^{(s)}\}.\end{aligned}$$

Therefore, the training objective can be written as

$$\begin{aligned}\mathcal{R}(\boldsymbol{\theta}) &= \sum_{s=1}^{S} \sum_{\tilde{\boldsymbol{Y}}_s \in \mathcal{Y}(\boldsymbol{X}_s)} \prod_{k=1}^{K} \\ &\quad P(\tilde{\boldsymbol{Y}}_s^{(s-k+1)}|\boldsymbol{X}_s^{(s-k+1)};\boldsymbol{\theta})\Delta(\tilde{\boldsymbol{Y}}_s, \boldsymbol{Y}_s).\end{aligned}$$

Note that the search space $\mathcal{Y}(\boldsymbol{X}_s)$ is exponential because it is the Cartesian product of the candidate translation sets of K source sentences:

$$\begin{aligned}\mathcal{Y}(\boldsymbol{X}_s) = \mathcal{Y}(\boldsymbol{x}^{(s-K+1)}) \times \cdots \times \\ \mathcal{Y}(\boldsymbol{x}^{(s-1)}) \times \mathcal{Y}(\boldsymbol{x}^{(s)}).\end{aligned}$$

For efficiency, we approximate $\mathcal{Y}(\boldsymbol{X}_s)$ with subset $\mathcal{S}(\boldsymbol{X}_s)$ by restricting that the first $K-1$ sets only contain 1-best candidates:

$$\begin{aligned}\mathcal{S}(\boldsymbol{X}_s) = \left\{\tilde{\boldsymbol{y}}_*^{(s-K+1)}\right\} \times \cdots \times \\ \left\{\tilde{\boldsymbol{y}}_*^{(s-1)}\right\} \times \mathcal{S}(\boldsymbol{x}^{(s)}),\end{aligned} \qquad (5)$$

where $\mathcal{S}(\boldsymbol{x}^{(s)})$ is a sampled subspace[9] and $\tilde{\boldsymbol{y}}_*^{(s-k+1)}$ is the 1-best candidate translation for the $(s-k+1)$-th source sentence:

$$\tilde{\boldsymbol{y}}_*^{(s-k+1)} = \underset{\boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x}^{(s-k+1)})}{\operatorname{argmax}} \left\{P(\boldsymbol{y}|\boldsymbol{x}^{(s-k+1)};\boldsymbol{\theta})\right\}.$$

Therefore, our final training objective is defined as follows:

$$\begin{aligned}\mathcal{R}(\boldsymbol{\theta}) &\approx \sum_{s=1}^{S} \sum_{\tilde{\boldsymbol{Y}}_s \in \mathcal{S}(\boldsymbol{X}_s)} \prod_{k=1}^{K} \\ &\quad P(\tilde{\boldsymbol{Y}}_s^{(s-k+1)}|\boldsymbol{X}_s^{(s-k+1)};\boldsymbol{\theta})\Delta(\tilde{\boldsymbol{Y}}_s, \boldsymbol{Y}_s) \\ &\approx \sum_{s=1}^{S} \sum_{\tilde{\boldsymbol{Y}}_s \in \mathcal{S}(\boldsymbol{X}_s)} \prod_{k=1}^{K} \\ &\quad Q(\tilde{\boldsymbol{Y}}_s^{(s-k+1)}|\boldsymbol{X}_s^{(s-k+1)};\boldsymbol{\theta},\alpha)\Delta(\tilde{\boldsymbol{Y}}_s, \boldsymbol{Y}_s) \\ &= \sum_{s=1}^{S} \sum_{\tilde{\boldsymbol{Y}}_s \in \mathcal{S}(\boldsymbol{X}_s)} Q(\tilde{\boldsymbol{y}}^{(s)}|\boldsymbol{x}^{(s)};\boldsymbol{\theta},\alpha)\Delta(\tilde{\boldsymbol{Y}}_s, \boldsymbol{Y}_s).\end{aligned} \qquad (6)$$

Note that $Q(\tilde{\boldsymbol{Y}}_s^{(s-k+1)}|\boldsymbol{X}_s^{(s-k+1)};\boldsymbol{\theta},\alpha) \equiv 1$ for $k \in [1, K-1]$, because we restrict that the first $K-1$ sets only contain 1-best candidates (see (5)).

It is clear that (6) is a more general form of the training objective in [9] because it can be reduced to (2) by setting $K = 1$.

3.3 Training Algorithm

Algorithm 1 shows one pass that scans the randomly shuffled full training set during online training. The algorithm first initializes a queue $q^{(0)}$ to store the 1-best candidates of last $K-1$ source sentences from the previous pass (line 1). The model parameters are also inherited from the previous pass (line 2). Then, the algorithm iteratively processes each sentence in the training set. It first computes the 1-best candidate translation $\tilde{\boldsymbol{y}}_*^{(s)}$ for the s-th source sentence using current model parameters $\boldsymbol{\theta}^{(s-1)}$ (line 5). Then, the algorithm builds the sampled subspace $\boldsymbol{x}^{(s)}$ using the procedure described in [9] (line 6). After that, the subset $\mathcal{S}(\boldsymbol{X}_s)$ can be constructed using $q^{(s-1)}$ and $\boldsymbol{x}^{(s)}$ according to (5) (line 7). After computing the gradients of the approximate expectations of $\Delta(\tilde{\boldsymbol{Y}}_s, \boldsymbol{Y}_s)$, the algorithm updates model parameters (line 9) and the queue of 1-best candidates (line 10). Note that q only needs to retain $K-1$ latest 1-best candidates during training.

4 Experiments

4.1 Setup

We evaluated our approach on two translation tasks: Chinese-English and English-French. The evaluation metrics are BLEU[14] and TER[15], which are calculated by the `multi-bleu.pl` and `tercom.7.25.jar` scripts, respectively.

Algorithm 1. One Pass During Online Training

```
Input a set of randomly shuffled training examples $\{(\boldsymbol{x}^{(s)}, \boldsymbol{y}^{(s)})\}_{s=1}^{S}$, a hyper-parameter $K$ that determines the number of sentences
    involved in calculating corpus-level metrics, a hyper-parameter $\alpha$ that controls the smoothness of the $Q$ distribution
Output optimized model parameters $\boldsymbol{\theta}^{(S)}$
Initialize a queue $q^{(0)}$ that contains 1-best candidates from the previous pass;
Initialize model parameters $\boldsymbol{\theta}^{(0)}$ from the previous pass;
$s \leftarrow 1$;
while $s \leqslant S$ do
    $\tilde{\boldsymbol{y}}_*^{(s)} = \mathrm{argmax}_{\boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x}^{(s)})} \left\{ P(\boldsymbol{y}|\boldsymbol{x}^{(s)}; \boldsymbol{\theta}^{(s-1)}) \right\}$;
    Sample $\mathcal{S}(\boldsymbol{x}^{(s)})$;
    Build $\mathcal{S}(\boldsymbol{X}_s)$ using $q^{(s-1)}$ and $\mathcal{S}(\boldsymbol{x}^{(s)})$;
    Compute the gradients of $\sum_{\tilde{\boldsymbol{Y}}_s \in \mathcal{S}(\boldsymbol{X}_s)} Q(\tilde{\boldsymbol{y}}^{(s)}|\boldsymbol{x}^{(s)}; \boldsymbol{\theta}^{(s-1)}, \alpha) \Delta(\tilde{\boldsymbol{Y}}_s, \boldsymbol{Y}_s)$ w.r.t. parameters;
    Update model parameters and get $\boldsymbol{\theta}^{(s)}$;
    Get $q^{(s)}$ by appending $\tilde{\boldsymbol{y}}_*^{(s)}$ to $q^{(s-1)}$ ;
end while
```

For Chinese-English translation tasks, to compare with the results reported by previous work[9], we used the same training data that consists of 2.56M pairs of sentences from Linguistic Data Consortium (LDC), which contains 67.5M Chinese words and 74.8M English words, respectively. The NIST 2006 dataset serves as the validation set for optimizing hyper-parameters and selecting models, and the NIST 2002, NIST 2003, NIST 2004, NIST 2005, and NIST 2008 datasets as test sets in our experiments[1].

For English-French translation tasks, to compare with the results reported by previous work on end-to-end NMT[2-3,9,22-24], we used the same subset of the WMT 2014 training corpus that contains 12M sentence pairs, which contains 304M English words and 348M French words, respectively. Following common practice, the concatenation of news-test 2012 and news-test 2013 serves as the validation set and news-test 2014 as the test set.

On top of RNNsearch[3], we compared the following three training criteria:

1) MLE: maximum likelihood estimation ((1)),

2) sMRT: minimum risk training with decomposable sentence-level evaluation metrics ((2)),

3) cMRT: minimum risk training with non-decomposable corpus-level evaluation metrics ((6)).

The three variants share the same setting of hyper-parameters: the vocabulary size is set to 30k for both source and target languages, the beam size for decoding is set to 10, α is set to 0.005 for two MRT variants, and K is set to 100 for cMRT which is tuned on the validation set. For each source sentence, 100 samples are sampled randomly to build the sampled subspace $\mathcal{S}(\boldsymbol{x}^{(s)})$, which is suggested by [9]. We removed duplicate candidates and added the gold reference when building the subspace. All the samples in the subspace and the corresponding source sentence compose one mini-batch to calculate the gradient according to (4)[2].

4.2 Effect of Evaluation Metrics

Table 1 shows the effect of evaluation metrics on the Chinese-English validation set. As MLE aims to maximize the likelihood of training data and does not consider any evaluation metric, it achieves significantly lower translation performance in terms of cBLEU and cTER than two MRT variants.

Table 1. Effect of Evaluation Metrics on Translation Quality on the Validation Set

Criterion	Metric	cBLEU	cTER
MLE	N/A	30.48	60.85
sMRT	-sBLEU	36.71	53.48
	sTER	30.14	53.83
cMRT	-cBLEU	38.26	54.76
	cTER	35.41	52.84

Note: "-sBLEU" denotes negative sentence-level BLEU, "-cBLEU" denotes negative corpus-level BLEU, "sTER" denotes sentence-level TER, and "cTER" denotes corpus-level TER.

By using decomposable evaluation metrics — sBLEU (i.e., negative sentence-level BLEU) and sTER (sentence-level TER) as loss functions, sMRT dramatically outperforms MLE in terms of both metrics. However, due to the discrepancy between training and testing (i.e., -sBLEU is used in training but -cBLEU is used

[1] LDC2002E18, LDC2003E07, LDC2003E14, LDC2004T07, LDC2004T08, LDC2005T06, LDC2005T10, LDC2007T23, LDC2008T06, LDC2008T08, LDC2008T18, LDC2009T02, LDC2009T06, LDC2009T15, LDC2010T03.

[2] We find that a pseudo corpus size larger than 100 (e.g., $K = 1\,000$) does not lead to significant improvements and increases memory requirement.

in testing), optimizing model parameters with respect to sTER during training fails to result in the lowest cTER in testing.

In contrast, our approach directly optimizes model parameters with respect to corpus-level metrics and enables training to correlate well with testing: optimizing model parameters with respect to cTER during training does result in the lowest cTER in testing.

4.3 Comparison of Training Time

We trained our NMT system on a cluster with 16 Tesla M40 GPUs. For sMRT, it takes the cluster about one hour to train 21 000 mini-batches. The training speed for cMRT is slightly slower than that for sMRT: 20 000 mini-batches can be processed in one hour on the same cluster. Note that each mini-batch only contains one source sentence and its corresponding target samples, thereby calculating pseudo corpus evaluation metrics on one mini-batch is difficult. Previous model predictions are needed to approximate the non-decomposable evaluation metrics.

As shown in Fig.1, we give the learning curves of sMRT and cMRT on the validation set to compare the model complexity and the real training time. Initializing with an MLE model, both sMRT and cMRT increase BLEU scores dramatically within about 20 hours. Afterwards, the BLEU score keeps improving gradually but there are slight oscillations. cMRT almost always achieves higher BLEU scores compared with sMRT with the same training time. Both sMRT and cMRT need about more than 270 hours to achieve the best results.

4.4 Comparison of BLEU Scores on Chinese-English Translation

Table 2 shows the case-insensitive BLEU scores on Chinese-English datasets. Statistical significance testing is performed with paired bootstrap resampling[25]. We follow [23] to handle rare words. We find that introducing both MRT variants into NMT leads to surprisingly substantial improvements over MLE (up to 7.20 and 8.53 BLEU points) across all test sets. On top of sMRT, our approach (cMRT) achieves consistent and statistically significant improvements (up to 1.88 BLEU points) across all test sets. On the concatenation of all test sets (i.e., “All”), our approach improves over MLE and sMRT by +7.89 and +1.44 BLEU points, respectively.

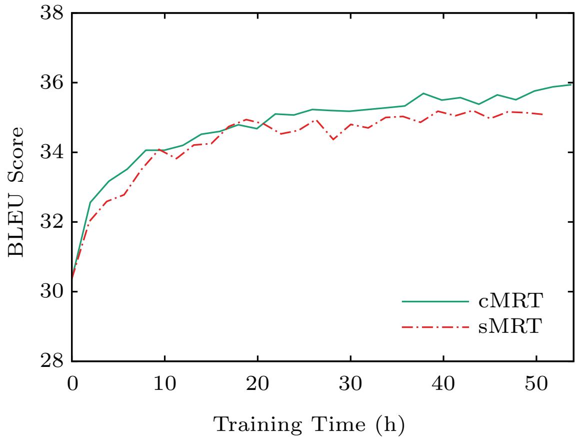

Fig.1. Comparison of training time on the Chinese-English validation set.

Fig.2 shows the BLEU scores on the Chinese-English test sets over various input sentences. While both MRT variants consistently improve over MLE for all lengths, cMRT outperforms sMRT for sentences longer than 40. One possible reason is that translations for long sentences have a more important effect on corpus-level BLEU score, especially on brevity penalty. Also, cMRT tends to produce longer translations compared with sMRT for long input sentences, as shown in Fig.3.

4.5 Comparison of TER Scores on Chinese-English Translation

Table 3 shows the case-insensitive TER scores on Chinese-English datasets. As optimizing with respect to -sBLEU leads to lower cTER than sTER (see Table 1), we used -sBLEU instead of sTER for sMRT.

Table 2. Case-Insensitive BLEU Scores on the Test Sets

Criterion	Metric	MT06	MT02	MT03	MT04	MT05	MT08	All
MLE	N/A	30.70	35.13	33.73	34.58	31.76	23.57	31.63
sMRT	-sBLEU	37.34**	40.36**	40.93**	41.37**	38.81**	29.23**	38.08**
cMRT	-cBLEU	38.95**††	41.65**††	41.99**††	42.64**††	40.29**††	31.11**††	39.52**††

Note: We use “**” to denote that the difference is statistically significant at $p < 0.01$ level compared with MLE, and “††” compared with sMRT. “All” denotes the concatenation of all test sets.

Our approach also significantly outperforms MLE and sMRT across all test sets. On the concatenation of all test sets, our approach improves over MLE and sMRT by −8.38 and −0.93 TER points, respectively.

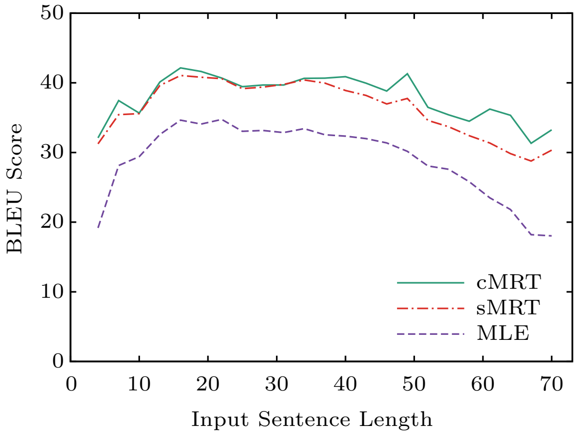

Fig.2. BLEU scores on the Chinese-English test sets over various input sentence lengths.

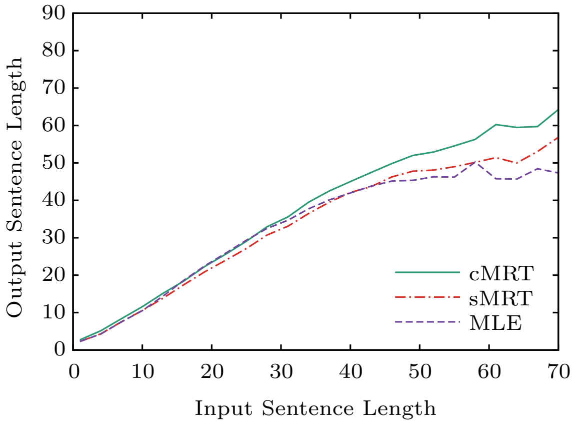

Fig.3. Comparison of output sentence lengths on the Chinese-English test sets.

Fig.4 shows the TER scores on the Chinese-English test sets over various input sentences. We find that cMRT systematically outperforms both sMRT and MLE for almost all lengths.

Table 3. Case-Insensitive TER Scores on the Test Sets

Criterion	Metric	MT06	MT02	MT03	MT04	MT05	MT08	All
MLE	N/A	60.74	58.94	60.10	58.91	61.74	64.52	60.90
sMRT	-sBLEU	52.86	52.87	52.17	51.49	53.42	57.21	53.45
cMRT	cTER	51.86	51.12	50.42	50.63	51.66	56.54	52.52

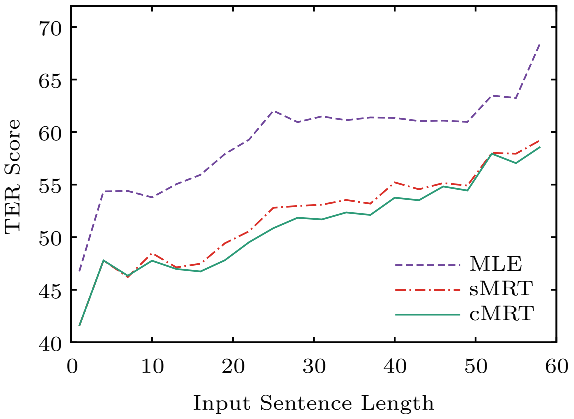

Fig.4. TER scores on the test sets over various input sentence lengths.

4.6 Results on English-French Translation

Table 4 shows the results on English-French translation. We list existing end-to-end NMT systems that are comparable to our system. All these systems use the same subset of the WMT 2014 parallel training corpus. They differ in network architectures, vocabulary sizes and training criterion. Note that [24] uses another monolingual dataset to train a language model, which is not used in other systems. RNNsearch-cMRT achieves the highest BLEU score in this setting even with a vocabulary size smaller than [23] and less data than [24]. Our approach does not assume specific architectures and can in principle be applied to any NMT systems[3].

5 Related Work

Our work is closely related to minimum risk training widely used in statistical machine translation. The minimum error rate training (MERT) algorithm[4] is a special form of MRT. Although MERT is capable of optimizing models with respect to non-decomposable evaluation metrics, it is restricted to optimizing linear models with tens of features on a small development set. [12] proposes an approach to maximizing expected BLEU for training phrase and lexicon translation models. It uses the extended Baum-Welch algorithm to efficiently update model parameters. These approaches cannot be directly applied to neural machine transla-

[3] Some results are from the arxiv version.

Table 4. Comparison with Previous Work on English-French Translation

System		Training	Vocabulary ($\times 10^3$)	BLEU
Existing end-to-end NMT systems	RNNSearch[3]	MLE	30	28.45
	LSTM with 4 layers[2]	MLE	80	30.59
	RNNSearch + PosUnk[22]	MLE	30	33.08
	LSTM with 6 layers + PosUnk[23]	MLE	40	32.70
	RNNSearch + PosUnk[9]	sMRT	30	34.23
	RNNSearch + monolingual data + PosUnk[24]	Dual learning	30	34.83
Our end-to-end NMT system	RNNSearch + PosUnk	cMRT	30	34.93

Note: The BLEU scores are case-sensitive. "PosUnk" denotes the technique of handling rare words in [23].

tion because of the non-linearity in neural networks.

Neural machine translation needs efficient online learning algorithms because the training dataset is always very large. As it is difficult to directly optimize models with respect to non-decomposable evaluation metrics in online learning frameworks, one possible solution is to maintain a large buffer to compute online gradient estimates that can be prohibitive[20]. [21] considers optimizing for the performance measures that are concave or pseudo-linear in the canonical confusion matrix of the predictor. A key limitation of these approaches is that they only focus on classification tasks. It is non-trivial to adapt these approaches to optimize non-decomposable evaluation metrics for neural machine translation.

Another possible solution is to optimize models parameters with respect to an approximation of non-decomposable evaluation metrics in online training framework, which is widely used in conventional SMT[16-19]. The basic idea is to build a pseudo corpus by using previous oracle translations, and k-best translations are generated to update the model[16]. An exponential decay is used in [18] to reduce dependence on the distant past. We borrow the idea from SMT and extend it to our NMT online training algorithm. In fact, different methods for the approximation of non-decomposable corpus-level evaluation metrics can also be used in our approach (e.g., using oracle translations[16] and exponential decay[18]).

Our work extends the approach in [9] by incorporating corpus-level evaluation metrics. We still maintain the online learning framework to minimize GPU memory requirements and build a subset of the training corpus on the fly. This can be seen as a balance between calculating metrics on individual sentences and that on the full training corpus. We show that this strategy effectively improves the correlation between training and testing and thus leads to better translation results.

6 Conclusions

In this work, we proposed an approach to training neural machine translation models with non-decomposable evaluation metrics. The basic idea is to calculate the expectations of corpus-level metrics on a subset of the training data to allow online training algorithms. Experiments showed that our approach is capable of improving the correlation between training and testing and significantly outperforms minimum risk training with decomposable evaluation metrics.

In the future, we plan to explore more methods for building subsets $\boldsymbol{X}_s$ and $\mathcal{S}(\boldsymbol{X}_s)$, which seem to have an important effect on translation performance. As our approach is transparent to network architectures and evaluation metrics, it can potentially benefit more natural language processing tasks.

References

[1] Kalchbrenner N, Blunsom P. Recurrent continuous translation models. In *Proc. the Conference on Empirical Methods in Natural Language Processing*, Oct. 2013, pp.1700-1709.

[2] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks. In *Proc. Advances in Neural Information Processing Systems*, Dec. 2014, pp.3104-3112.

[3] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. In *Proc. ICLR*, May 2015.

[4] Och F J. Minimum error rate training in statistical machine translation. In *Proc. the 41st Annual Meeting of the Association for Computational Linguistics*, July 2003, pp.160-167.

[5] Chiang D. A hierarchical phrase-based model for statistical machine translation. In *Proc. the 43rd Annual Meeting of the Association for Computational Linguistics*, June 2005, pp.263-270.

[6] Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Computation*, 1997, 9(8): 1735-1780.

[7] Chung J, Gulcehre C, Cho K, Yoshua B. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv:1412.3555*, 2014. https://arxiv.org/abs/1412.3555, May 2017.

[8] Ranzato M, Chopra S, Auli M, Zaremba W. Sequence level training with recurrent neural networks. In *Proc. ICLR*, May 2016.

[9] Shen S, Cheng Y, He Z, He W, Wu H, Sun M, Liu Y. Minimum risk training for neural machine translation. In *Proc. the 54th Annual Meeting of the Association for Computational Linguistics* (*Volume* 1: *Long Papers*), Aug. 2016, pp.1683-1692.

[10] Willams R J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 1992, 8(3/4): 229-256.

[11] Smith D A, Eisner J. Minimum risk annealing for training log-linear models. In *Proc. the COLING/ACL on Main Conference Poster Sessions*, July 2006, pp.787-794.

[12] He X, Deng L. Maximum expected BLEU training of phrase and lexicon translation models. In *Proc. the 50th Annual Meeting of the Association for Computational Linguistics* (*Volume* 1: *Long Papers*), July 2012, pp.292-301.

[13] Gao J, He X, Yih W, Deng L. Learning continuous phrase representations for translation modeling. In *Proc. the 52nd Annual Meeting of the Association for Computational Linguistics* (*Volume* 1: *Long Papers*), June 2014, pp.699-709.

[14] Papineni K, Roukos S, Ward T, Zhu W J. BLEU: A method for automatic evaluation of machine translation. In *Proc. the 40th Annual Meeting of the Association for Computational Linguistics*, July 2002, pp.311-318.

[15] Snover M, Dorr B, Schwartz R, Micciulla L, Makhoul J. A study of translation edit rate with targeted human annotation. In *Proc. the 7th Association for Machine Translation in the Americas*, Aug. 2006, pp.223-231.

[16] Watanabe T, Suzuki J, Tsukada H, Isozaki H. Online large-margin training for statistical machine translation. In *Proc. the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (*EMNLP-CoNLL*), June 2007, pp.764-773.

[17] Chiang D, Marton Y, Resnik P. Online large-margin training of syntactic and structural translation features. In *Proc. the Conference on Empirical Methods in Natural Language Processing*, Oct. 2008, pp.224-233.

[18] Chiang D. Hope and fear for discriminative training of statistical translation models. *The Journal of Machine Learning Research*, 2012, 13(1): 1159-1187.

[19] Neubig G, Watanabe T. Optimization for statistical machine translation: A survey. *Computational Linguistics*, 2016, 42(2): 1-54.

[20] Kar P, Narasimhan H, Jain P. Online and stochastic gradient methods for non-decomposable loss functions. In *Proc. the 27th Advances in Neural Information Processing Systems*, Dec. 2014, pp.694-702.

[21] Narasimhan H, Vaish R, Agarwal S. On the statistical consistency of plug-in classifiers for non-decomposable performance measures. In *Proc. the 27th Advances in Neural Information Processing Systems*, Dec. 2014, pp.1493-1501.

[22] Jean S, Cho K, Memisevic R, Bengio Y. On using very large target vocabulary for neural machine translation. In *Proc. the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing* (*Volume* 1: *Long Papers*), July 2015, pp.1-10.

[23] Luong M T, Sutskever I, Le Q V, Vinyals O, Zaremba W. Addressing the rare word problem in neural machine translation. In *Proc. the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing* (*Volume* 1: *Long Papers*), July 2015, pp.11-19.

[24] He D, Xia Y, Qin T, Wang L, Yu N, Liu T, Ma W Y. Dual learning for machine translation. In *Proc. the 30th Advances in Neural Information Processing Systems*, Dec. 2016, pp.820-828.

[25] Koehn P. Statistical significance tests for machine translation evaluation. In *Proc. the Conference on Empirical Methods in Natural Language Processing*, July 2004, pp.388-395.

Shi-Qi Shen is currently working toward his Ph.D. degree in computer science at Tsinghua University, Beijing. His current research interests include natural language processing and machine translation.

Yang Liu is an associate professor in the Department of Computer Science and Technology at Tsinghua University, Beijing. He received his Ph.D. degree from Institute of Computing Technology, Chinese Academy of Sciences, Beijing, in 2007. His research areas include natural language processing and machine translation.

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