Trainable Greedy Decoding for Neural Machine Translation

Jiatao Gu[†], **Kyunghyun Cho**[‡] **and Victor O.K. Li**[†]

[†]The University of Hong Kong
[‡]New York University
[†]{jiataogu, vli}@eee.hku.hk
[‡]kyunghyun.cho@nyu.edu

Abstract

Recent research in neural machine translation has largely focused on two aspects; neural network architectures and end-to-end learning algorithms. The problem of decoding, however, has received relatively little attention from the research community. In this paper, we solely focus on the problem of decoding given a trained neural machine translation model. Instead of trying to build a new decoding algorithm for any specific decoding objective, we propose the idea of *trainable decoding algorithm* in which we train a decoding algorithm to find a translation that maximizes an arbitrary decoding objective. More specifically, we design an actor that observes and manipulates the hidden state of the neural machine translation decoder and propose to train it using a variant of deterministic policy gradient. We extensively evaluate the proposed algorithm using four language pairs and two decoding objectives, and show that we can indeed train a trainable greedy decoder that generates a better translation (in terms of a target decoding objective) with minimal computational overhead.

1 Introduction

Neural machine translation has recently become a method of choice in machine translation research. Besides its success in traditional settings of machine translation, that is one-to-one translation between two languages, (Sennrich et al., 2016; Chung et al., 2016), neural machine translation has ventured into more sophisticated settings of machine translation. For instance, neural machine translation has successfully proven itself to be capable of handling subword-level representation of sentences (Lee et al., 2016; Luong and Manning, 2016; Sennrich et al., 2015; Costa-Jussa and Fonollosa, 2016; Ling et al., 2015). Furthermore, several research groups have shown its potential in seamlessly handling multiple languages (Dong et al., 2015; Luong et al., 2015a; Firat et al., 2016a,b; Lee et al., 2016; Ha et al., 2016; Viégas et al., 2016).

A typical scenario of neural machine translation starts with training a model to maximize its log-likelihood. That is, we often train a model to maximize the conditional probability of a reference translation given a source sentence over a large parallel corpus. Once the model is trained in this way, it defines the conditional distribution over all possible translations given a source sentence, and the task of translation becomes equivalent to finding a translation to which the model assigns the highest conditional probability. Since it is computationally intractable to do so exactly, it is a usual practice to resort to approximate search/decoding algorithms such as greedy decoding or beam search. In this scenario, we have identified two points where improvements could be made. They are (1) training (including the selection of a model architecture) and (2) decoding.

Much of the research on neural machine translation has focused solely on the former, that is, on improving the model architecture. Neural machine translation started with with a simple encoder-decoder architecture in which a source sentence is encoded into a single, fixed-size vector (Cho et al., 2014; Sutskever et al., 2014; Kalchbrenner and Blunsom, 2013). It soon evolved with the attention mechanism (Bahdanau et al., 2014). A few variants of the attention mechanism, or its regularization, have been proposed recently to improve both the translation quality as well as the computational efficiency (Luong et al., 2015b; Cohn et al., 2016; Tu et al., 2016b). More recently, convolutional net-

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1957–1967
Copenhagen, Denmark, September 7–11, 2017. ©2017 Association for Computational Linguistics

works have been adopted either as a replacement of or a complement to a recurrent network in order to efficiently utilize parallel computing (Kalchbrenner et al., 2016; Lee et al., 2016; Gehring et al., 2016).

On the aspect of decoding, only a few research groups have tackled this problem by incorporating a target decoding algorithm into training. Wiseman and Rush (2016) and Shen et al. (2015) proposed a learning algorithm tailored for beam search. Ranzato et al. (2015) and (Bahdanau et al., 2016) suggested to use a reinforcement learning algorithm by viewing a neural machine translation model as a policy function. Investigation on decoding alone has, however, been limited. Cho (2016) showed the limitation of greedy decoding by simply injecting unstructured noise into the hidden state of the neural machine translation system. Tu et al. (2016a) similarly showed that the exactness of beam search does not correlate well with actual translation quality, and proposed to augment the learning cost function with reconstruction to alleviate this problem. Li et al. (2016) proposed a modification to the existing beam search algorithm to improve its exploration of the translation space.

In this paper, we tackle the problem of decoding in neural machine translation by introducing a concept of *trainable greedy decoding*. Instead of manually designing a new decoding algorithm suitable for neural machine translation, we propose to learn a decoding algorithm with an arbitrary decoding objective. More specifically, we introduce a neural-network-based decoding algorithm that works on an already-trained neural machine translation system by observing and manipulating its hidden state. We treat such a neural network as an agent with a deterministic, continuous action and train it with a variant of the deterministic policy gradient algorithm (Silver et al., 2014).

We extensively evaluate the proposed trainable greedy decoding on four language pairs (En-Cs, En-De, En-Ru and En-Fi; in both directions) with two different decoding objectives; sentence-level BLEU and negative perplexity. By training such trainable greedy decoding using deterministic policy gradient with the proposed critic-aware actor learning, we observe that we can improve decoding performance with minimal computational overhead. Furthermore, the trained actors are found to improve beam search as well, suggesting a future research direction in extending the proposed idea of trainable decoding for more sophisticated underlying decoding algorithms.

2 Background

2.1 Neural Machine Translation

Neural machine translation is a special case of conditional recurrent language modeling, where the source and target are natural language sentences. Let us use $X = \{x_1, \ldots, x_{T_s}\}$ and $Y = \{y_1, \ldots, y_T\}$ to denote source and target sentences, respectively. Neural machine translation then models the target sentence given the source sentence as: $p(Y|X) = \prod_{t=1}^{T} p(y_t|y_{<t}, X)$. Each term on the r.h.s. of the equation above is modelled as a composite of two parametric functions:

$$p(y_t|y_{<t}, X) \propto \exp\left(g\left(y_t, z_t; \theta_g\right)\right),$$

where $z_t = f(z_{t-1}, y_{t-1}, e_t(X; \theta_e); \theta_f)$. g is a read-out function that transforms the hidden state z_t into the distribution over all possible symbols, and f is a recurrent function that compresses all the previous target words $y_{<t}$ and the time-dependent representation $e_t(X; \theta_e)$ of the source sentence X. This time-dependent representation e_t is often implemented as a recurrent network encoder of the source sentence coupled with an attention mechanism (Bahdanau et al., 2014).

Maximum Likelihood Learning We train a neural machine translation model, or equivalently estimate the parameters θ_g, θ_f and θ_e, by maximizing the log-probability of a reference translation $\hat{Y} = \{\hat{y}_1, ..., \hat{y}_T\}$ given a source sentence. That is, we maximize the log-likelihood function:

$$J^{\mathrm{ML}}(\theta_g, \theta_f, \theta_e) = \frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T_n} \log p_\theta(\hat{y}_t^n | \hat{y}_{<t}^n, X^n),$$

given a training set consisting of N source-target sentence pairs. It is important to note that this maximum likelihood learning does not take into account how a trained model would be used. Rather, it is only concerned with learning a distribution over all possible translations.

2.2 Decoding

Once the model is trained, either by maximum likelihood learning or by any other recently proposed algorithms (Wiseman and Rush, 2016; Shen et al., 2015; Bahdanau et al., 2016; Ranzato et al., 2015), we can let the model translate a given sentence by

finding a translation that maximizes

$$\hat{Y} = \arg\max_{Y} \log p_\theta(Y|X),$$

where $\theta = (\theta_g, \theta_f, \theta_e)$. This is, however, computationally intractable, and it is a usual practice to resort to approximate decoding algorithms.

Greedy Decoding One such approximate decoding algorithm is greedy decoding. In greedy decoding, we follow the conditional dependency path and pick the symbol with the highest conditional probability so far at each node. This is equivalent to picking the best symbol one at a time from left to right in conditional language modelling. A decoded translation of greedy decoding is $\hat{Y} = (\hat{y}_1, \ldots, \hat{y}_T)$, where

$$\hat{y}_t = \arg\max_{y \in V} \log p_\theta(y|\hat{y}_{<t}, X). \quad (1)$$

Despite its preferable computational complexity $O(|V| \times T)$, greedy decoding has been over time found to be undesirably sub-optimal.

Beam Search Beam search keeps $K > 1$ hypotheses, unlike greedy decoding which keeps only a single hypothesis during decoding. At each time step t, beam search picks K hypotheses with the highest scores ($\prod_{t'=1}^{t} p(y_t|y_{<t}, X)$). When all the hypotheses terminate (outputting the end-of-the-sentence symbol), it returns the hypothesis with the highest log-probability. Despite its superior performance compared to greedy decoding, the computational complexity grows linearly w.r.t. the size of beam K, which makes it less preferable especially in the production environment.

3 Trainable Greedy Decoding

3.1 Many Decoding Objectives

Although we have described decoding in neural machine translation as a maximum-a-posteriori estimation in $\log p(Y|X)$, this is not necessarily the only nor the desirable decoding objective.

First, each potential scenario in which neural machine translation is used calls for a unique decoding objective. In simultaneous translation/interpretation, which has recently been studied in the context of neural machine translation (Gu et al., 2016), the decoding objective is formulated as a trade-off between the translation quality and delay. On the other hand, when a machine translation system is used as a part of a larger information extraction system, it is more important to correctly translate named entities and events than to translate syntactic function words. The decoding objective in this case must account for how the translation is used in subsequent modules in a larger system.

Second, the conditional probability assigned by a trained neural machine translation model does not necessarily reflect our perception of translation quality. Although Cho (2016) provided empirical evidence of high correlation between the log-probability and BLEU, a *de facto* standard metric in machine translation, there have also been reports on large mismatch between the log-probability and BLEU. For instance, Tu et al. (2016a) showed that beam search with a very large beam, which is supposed to find translations with better log-probabilities, suffers from pathological translations of very short length, resulting in low translation quality. This calls for a way to design or *learn* a decoding algorithm with an objective that is more directly correlated to translation quality.

In short, there is a significant need for designing multiple decoding algorithms for neural machine translation, regardless of how it was trained. It is however non-trivial to manually design a new decoding algorithm with an arbitrary objective. This is especially true with neural machine translation, as the underlying structure of the decoding/search process – the high-dimensional hidden state of a recurrent network – is accessible but not interpretable. Instead, in the remainder of this section, we propose our approach of *trainable greedy decoding*.

3.2 Trainable Greedy Decoding

We start from the noisy, parallel approximate decoding (NPAD) algorithm proposed in (Cho, 2016). The main idea behind NPAD algorithm is that a better translation with a higher log-probability may be found by injecting unstructured noise in the transition function of a recurrent network. That is,

$$z_t = f(z_{t-1} + \epsilon_t, y_{t-1}, e_t(X; \theta_e); \theta_f),$$

where $\epsilon_t \sim \mathcal{N}(0, (\sigma_0/t)^2)$. NPAD avoids potential degradation of translation quality by running such a noisy greedy decoding process multiple times in parallel. An important lesson of NPAD algorithm is that there exists a decoding strategy with the asymptotically same computational complexity that results in a better translation quality, and that such a better translation can be found by manipulating the hidden state of the recurrent network.

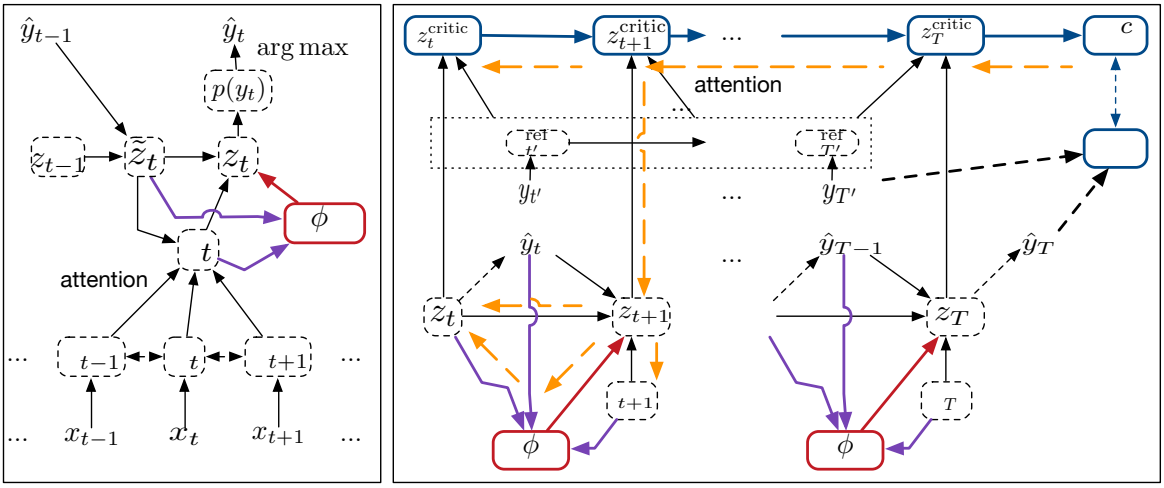


Figure 1: Graphical illustrations of the trainable greedy decoding. The left panel shows a single step of the actor interacting with the underlying neural translation model, and The right panel the interaction among the underlying neural translation system (dashed-border boxes), actor (red-border boxes), and critic (blue-border boxes). The solid arrows indicate the forward pass, and the dashed yellow arrows the actor's backward pass. The dotted-border box shows the use of a reference translation.

In this work, we propose to significantly extend NPAD by replacing the unstructured noise ϵ_t with a parametric function approximator, or an agent, π_ϕ. This agent takes as input the previous hidden state z_{t-1}, previously decoded word $\hat{y}_{t-1}$ and the time-dependent context vector $e_t(X;\theta_e)$ and outputs a real-valued vectorial action $a_t \in \mathbb{R}^{\dim(z_t)}$. Such an agent is trained such that greedy decoding with the agent finds a translation that maximizes any predefined, arbitrary decoding objective, while the underlying neural machine translation model is pretrained and fixed. Once the agent is trained, we generate a translation given a source sentence by greedy decoding however augmented with this agent. We call this decoding strategy *trainable greedy decoding*.

Related Work: Soothsayer prediction function Independently from and concurrently with our work here, Li et al. (2017) proposed, just two weeks earlier, to train a neural network that predicts an arbitrary decoding objective given a source sentence and a partial hypothesis, or a prefix of translation, and to use it as an auxiliary score in beam search. For training such a network, referred to as a Q network in their paper, they generate each training example by either running beam search or using a ground-truth translation (when appropriate) for each source sentence. This approach allows one to use an arbitrary decoding objective, but it still relies heavily on the log-probability of the underlying neural translation system in actual decoding. We expect a combination of these and our approaches may further improve decoding for neural machine translation in the future.

3.3 Learning and Challenges

While all the parameters—θ_g, θ_f and θ_e— of the underlying neural translation model are fixed, we only update the parameters ϕ of the agent π. This ensures the generality of the pretrained translation model, and allows us to train multiple trainable greedy decoding agents with different decoding objectives, maximizing the utility of a single trained translation model.

Let us denote by R our arbitrary decoding objective as a function that scores a translation generated from trainable greedy decoding. Then, our learning objective for trainable greedy decoding is

$$J^{\mathrm{A}}(\phi) = \mathbb{E}^{\hat{Y}=G_\pi(X)}_{X\sim D}\left[R(\hat{Y})\right],$$

where we used $G_\pi(X)$ as a shorthand for trainable greedy decoding with an agent π.

There are two major challenges in learning an agent with such an objective. First, the decoding objective R may not be differentiable with respect to the agent. Especially because our goal is to accommodate an arbitrary decoding objective, this becomes a problem. For instance, BLEU, a standard

quality metric in machine translation, is a piecewise linear function with zero derivatives almost everywhere. Second, the agent here is a real-valued, deterministic policy with a very high-dimensional action space (1000s of dimensions), which is well known to be difficult. In order to alleviate these difficulties, we propose to use a variant of the deterministic policy gradient algorithm (Silver et al., 2014; Lillicrap et al., 2015).

4 Deterministic Policy Gradient with Critic-Aware Actor Learning

4.1 Deterministic Policy Gradient for Trainable Greedy Decoding

It is highly unlikely for us to have access to the gradient of an arbitrary decoding objective R with respect to the agent π, or its parameters ϕ. Furthermore, we cannot estimate it stochastically because our policy π is defined to be deterministic without a predefined nor learned distribution over the action. Instead, following (Silver et al., 2014; Lillicrap et al., 2015), we use a parametric, differentiable approximator, called a critic R^c, for the non-differentiable objective R. We train the critic by minimizing

$$J^{\mathrm{C}}(\psi) = \mathbb{E}^{\hat{Y}=G_\pi(X)}_{X\sim D}\left[R^c_\psi(z_{1:T}) - R(\hat{Y})\right]^2.$$

The critic observes the state-action sequence of the agent π via the modified hidden states $(z_1, \ldots, z_T)$ of the recurrent network, and predicts the associated decoding objective. By minimizing the mean squared error above, we effectively encourage the critic to approximate the non-differentiable objective as closely as possible in the vicinity of the state-action sequence visited by the agent.

We implement the critic R^c as a recurrent network, similarly to the underlying neural machine translation system. This implies that we can compute the derivative of the predicted decoding objective with respect to the input, that is, the state-action sequence $z_{1:T}$, which allows us to update the actor π, or equivalently its parameters ϕ, to maximize the predicted decoding objective. Effectively we avoid the issue of non-differentiability of the original decoding objective by working with its proxy.

With the critic, the learning objective of the actor is now to maximize not the original decoding objective R but its proxy R^{C} such that

$$\hat{J}^{\mathrm{A}}(\phi) = \mathbb{E}^{\hat{Y}=G_\pi(X)}_{X\sim D}\left[R^{\mathrm{C}}(\hat{Y})\right].$$

Algorithm 1 Trainable Greedy Decoding

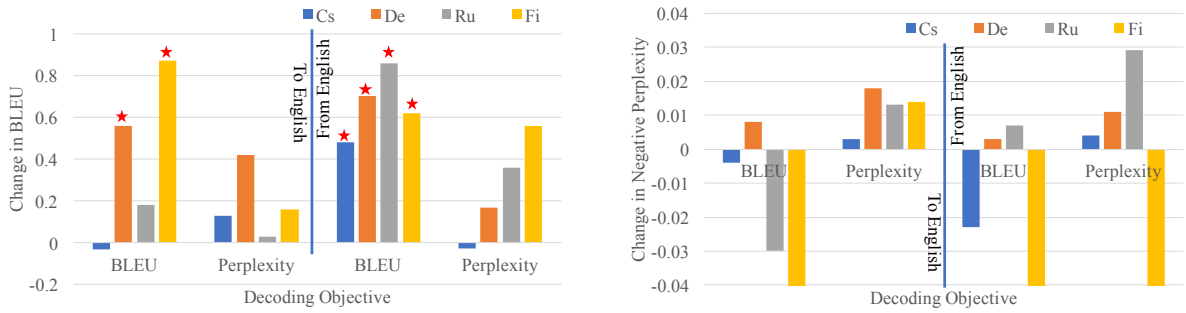
Require: NMT θ, actor ϕ, critic ψ, N_c, N_a, S_c, S_a, τ
Train θ using MLE on training set D;
Initialize ϕ and ψ;
Shuffle D twice into D_ϕ and D_ψ
while stopping criterion is not met **do**
  for $t = 1 : N_c$ **do**
    Draw a translation pair: $(X, Y) \sim D_\psi$;
    $r, r^c = \textsc{Decode}(S_c, X, Y, 1)$
    Update ψ using $\nabla_\psi \sum_k (r^c_k - r_k)^2/(S_c + 1)$
  for $t = 1 : N_a$ **do**
    Draw a translation pair: $(X, Y) \sim D_\phi$;
    $r, r^c = \textsc{Decode}(S_a, X, Y, 0)$
    Compute $w_k = \exp\left(-(r^c_k - r_k)^2/\tau\right)$
    Compute $\tilde{w}_k = w_k / \sum_k w_k$
    Update ϕ using $-\sum_k (\tilde{w}_k \cdot \nabla_\phi r^c_k)$

Function: $\textsc{Decode}(S, X, Y, c)$
$Y_s = \{\}, Z_s = \{\}, r = \{\}, r^c = \{\}$;
for $k = 1 : S$ **do**
  Sample noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$ for each action;
  Greedy decoding $\hat{Y}^k = G_{\theta,\phi}(X)$ with ϵ;
  Collect hidden states $z^k_{1:T}$ given $X, \hat{Y}, \theta, \phi$
  $Y_s \leftarrow Y_s \cup \{Y^k\}$
  $Z_s \leftarrow Z_s \cup \{z^k_{1:T}\}$
if $c = 1$ **then**
  Collect hidden states $z_{1:T}$ given X, Y, θ
  $Y_s \leftarrow Y_s \cup \{Y\}$
  $Z_s \leftarrow Z_s \cup \{z_{1:T}\}$
for $\hat{Y}, Z \in Y_s, Z_s$ **do**
  Compute the critic output $r^c \leftarrow R^c_\psi(Z, \hat{Y})$
  Compute true reward $r \leftarrow R(Y, \hat{Y})$
return r, r^c

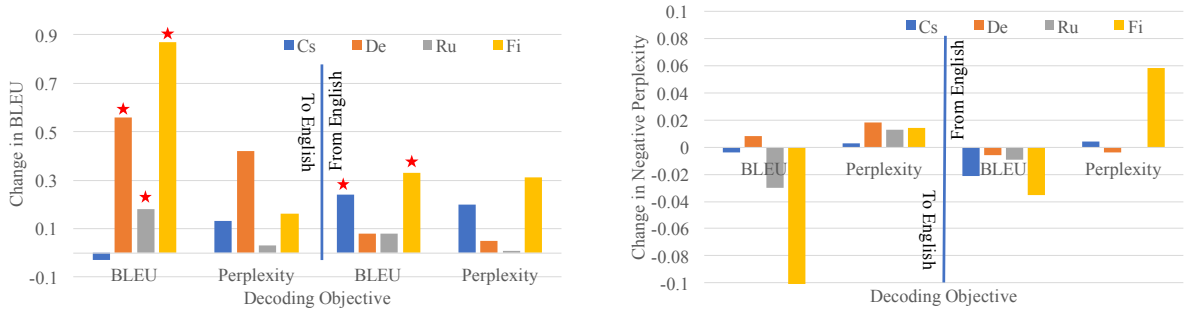
Unlike the original objective, this objective function is fully differentiable with respect to the agent π. We thus use a usual stochastic gradient descent algorithm to train the agent, while simultaneously training the critic. We do so by alternating between training the actor and critic. Note that we maximize the return of a full episode rather than the Q value, unlike usual approaches in reinforcement learning.

4.2 Critic-Aware Actor Learning

Challenges The most apparent challenge for training such a deterministic actor with a large action space is that most of action configurations will lead to zero return. It is also not trivial to devise an efficient exploration strategy with a deterministic actor with real-valued actions. This issue has however turned out to be less of a problem than in a usual reinforcement learning setting, as the state and action spaces are well structured thanks to pretraining by maximum likelihood learning. As observed by Cho (2016), any reasonable perturbation to the hidden state of the recurrent network generates a reasonable translation which would re-



(a) Trainable Greedy Decoding



(b) Beam Search + Trainable Greedy Decoding

Figure 2: The plots draw the improvements by the trainable greedy decoding on the test set. The x-axes correspond to the objectives used to train trainable greedy decoding, and the y-axes to the changes in the achieved objectives (BLEU for the figures on the left, and negative perplexity on the right.) The top row (a) shows the cases when the trainable greedy decoder is used on its own, and the bottom row (b) when it is used together with beam search. When training and evaluation are both done with BLEU, we test the statistical significance (Koehn, 2004), and we mark significant cases with red stars ($p < 0.05$.) The underlying neural machine translation models achieved the BLEU scores of 14.49/16.20 for En-Cs, 18.90/21.20 for Cs-En, 18.97/21.33 for En-De, 21.63/24.46 for De-En, 16.97/19.68 for En-Ru, 21.06/23.34 for Ru-En, 7.53/8.82 for En-Fi and 9.79/11.03 for Fi-En (greedy/beam).

ceive again a reasonable return.

Although this property of dense reward makes the problem of trainable greedy decoding more manageable, we have observed other issues during our preliminary experiment with the vanilla deterministic policy gradient. In order to avoid these issues that caused instability, we propose the following modifications to the vanilla algorithm.

Critic-Aware Actor Learning A major goal of the critic is not to estimate the return of a given episode, but to estimate the gradient of the return evaluated given an episode. In order to do so, the critic must be trained, or presented, with state-action sequences $z_{1:T'}$ similar though not identical to the state-action sequence generated by the current actor π. This is achieved, in our case, by injecting unstructured noise to the action at each time step, similar to (Heess et al., 2015):

$$\tilde{a}_t = \phi(z_t, a_{t-1}) + \sigma \cdot \epsilon, \tag{2}$$

where ϵ is a zero-mean, unit-variance normal variable. This noise injection procedure is mainly used when training the critic.

We have however observed that the quality of the reward and its gradient estimate of the critic is very noisy even when the critic was trained with this kind of noisy actor. This imperfection of the critic often led to the instability in training the actor in our preliminary experiments. In order to avoid this, we describe here a technique which we refer to as *critic-aware actor gradient estimation*.

Instead of using the point estimate $\frac{\partial R^c}{\partial \phi}$ of the gradient of the predicted objective with respect to the actor's parameters ϕ, we propose to use the expected gradient of the predicted objective with

respect to the critic-aware distribution Q. That is,

$$\mathbb{E}_Q\left[\frac{\partial R^c_\psi}{\partial \phi}\right], \quad (3)$$

where we define the critic-aware distribution Q as

$$Q(\epsilon) \propto \exp(\underbrace{-(R^c_\psi - R)^2/\tau}_{\text{Critic-awareness}}) \exp(\underbrace{-\frac{\epsilon^2}{2\sigma^2}}_{\text{Locality}}). \quad (4)$$

This expectation allows us to incorporate the noisy, non-uniform nature of the critic's approximation of the objective by up-weighting the gradient computed at a point with a higher critic quality and down-weighting the gradient computed at a point with a lower critic quality. The first term in Q reflects this, while the second term ensures that our estimation is based on a small region around the state-action sequence generated by the current, noise-free actor π.

Since it is intractable to compute Eq. (3) exactly, we resort to importance sampling with the proposed distribution equal to the second term in Eq. (4). Then, our gradient estimate for the actor becomes the sum of the gradients from multiple realizations of the noisy actor in Eq. (2), where each gradient is weighted by the quality of the critic $\exp(-(R^c_\phi - R)^2/\tau)$. τ is a hyperparameter that controls the smoothness of the weights. We observed in our preliminary experiment that the use of this critic-aware actor learning significantly stabilizes general learning of both the actor and critic.

Reference Translations for Training the Critic In our setting of neural machine translation, we have access to a reference translation for each source sentence X, unlike in a usual setting of reinforcement learning. By force-feeding the reference translation into the underlying neural machine translation system (rather than feeding the decoded symbols), we can generate the reference state-action sequence. This sequence is much less correlated with those sequences generated by the actor, and facilitates computing a better estimate of the gradient w.r.t. the critic.

In Alg. 1, we present the complete algorithm. To make the description less cluttered, we only show the version of minibatch size = 1 which can be naturally extended. We also illustrate the proposed trainable greedy decoding and the proposed learning strategy in Fig. 1.

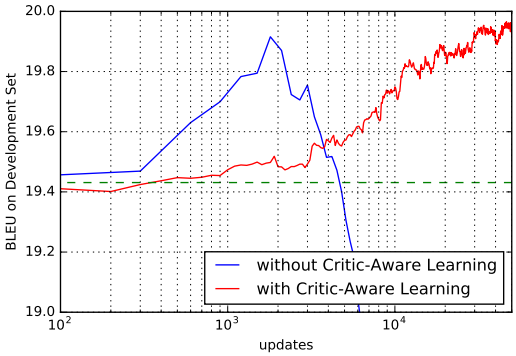


Figure 3: Comparison of greedy BLEU scores whether using the critic-aware exploration or not on Ru-En Dataset. The green line means the BLEU score achieved by greedy decoding from the underlying NMT model.

5 Experimental Settings

We empirically evaluate the proposed trainable greedy decoding on four language pairs – En-De, En-Ru, En-Cs and En-Fi – using a standard attention-based neural machine translation system (Bahdanau et al., 2014). We train underlying neural translation systems using the parallel corpora made available from WMT'15.[1] The same set of corpora are used for trainable greedy decoding as well. All the corpora are tokenized and segmented into subword symbols using byte-pair encoding (BPE) (Sennrich et al., 2015). We use sentences of length up to 50 subword symbols for MLE training and 200 symbols for trainable decoding. For validation and testing, we use newstest-2013 and newstest-2015, respectively.

5.1 Model Architectures and Learning

Underlying NMT Model For each language pair, we implement an attention-based neural machine translation model whose encoder and decoder recurrent networks have 1,028 gated recurrent units (GRU, Cho et al., 2014) each. Source and target symbols are projected into 512-dimensional embedding vectors. We trained each model for approximately 1.5 weeks using Adadelta (Zeiler, 2012).

Actor π We use a feedforward network with a single hidden layer as the actor. The input is a 2,056-dimensional vector which is the concatenation of the decoder hidden state and the time-dependent context vector from the attention mech-

[1] http://www.statmt.org/wmt15/

```
(a) S:  Главное зеркало инфракрасного космического телескопа имеет диаметр 6,5 метров
    T:  The primary mirror of the infrared space telescope has a diameter of 6.5 metres .
    G:  The main mirror of the infrared spaceboard has a diameter 6.5 m .
    A:  The main mirror of the infrared space-type telescope has a diameter of 6.5 meters .

(b) S:  Еще один пункт - это дать им понять , что они должны вести себя онлайн так же , как делают это оффлайн .
    T:  Another point is to make them see that they must behave online as they do offline .
    G:  Another option is to give them a chance to behave online as well as do this offline .
    A:  Another option is to give them to know that they must behave online as well as offline .

(c) S:  Возможен ли долговременный мир между арабами и израильтянами на Ближнем Востоке ?
    T:  Can there ever be a lasting peace between Arabs and Jews in the Middle East ?
    G:  Can the Long-term Peace be Out of the Middle East ?
    A:  Can the Long-term Peace be between Arabs and Israelis in the Middle East ?
```

Figure 4: Three Ru-En examples in which the difference between the trainable greedy decoding (A) and the conventional greedy decoding (G) is large. Each step is marked with magenta, when the actor significantly influenced the output distribution.

anism, and it outputs a 1,028-dimensional action vector for the decoder. We use 32 units for the hidden layer with $\tanh$ activations.

Critic R^c The critic is implemented as a variant of an attention-based neural machine translation model that takes a reference translation as a source sentence and a state-action sequence from the actor as a target sentence. Both the size of GRU units and embedding vectors are the same with the underlying model. Unlike a usual neural machine translation system, the critic does not language-model the target sentence but simply outputs a scalar value to predict the true return. When we predict a bounded return, such as sentence BLEU, we use a sigmoid activation at the output. For other unbounded return like perplexity, we use a linear activation.

Learning We train the actor and critic simultaneously by alternating between updating the actor and critic. As the quality of the critic's approximation of the decoding objective has direct influence on the actor's learning, we make ten updates to the critic before each time we update the actor once. We use RMSProp (Tieleman and Hinton, 2012) with the initial learning rates of 2×10^{-6} and 2×10^{-4}, respectively, for the actor and critic.

We monitor the progress of learning by measuring the decoding objective on the validation set. After training, we pick the actor that results in the best decoding objective on the validation set, and test it on the test set.

Decoding Objectives For each neural machine translation model, pretrained using maximum likelihood criterion, we train two trainable greedy decoding actors. One actor is trained to maximize BLEU (or its smoothed version for sentence-level scoring (Lin and Och, 2004)) as its decoding objective, and the other to minimize perplexity (or equivalently the negative log-probability normalized by the length.)

We have chosen the first two decoding objectives for two purposes. First, we demonstrate that it is possible to build multiple trainable decoders with a single underlying model trained using maximum likelihood learning. Second, the comparison between these two objectives provides a glimpse into the relationship between BLEU (the most widely used automatic metric for evaluating translation systems) and log-likelihood (the most widely used learning criterion for neural machine translation).

Evaluation We test the trainable greedy decoder with both greedy decoding and beam search. Although our decoder is always trained with greedy decoding, beam search in practice can be used together with the actor of the trainable greedy decoder. Beam search is expected to work better especially when our training of the trainable greedy decoder is unlikely to be optimal. In both cases, we report both the perplexity and BLEU.

5.2 Results and Analysis

We present the improvements of BLEU and perplexity (or its negation) in Fig. 2 for all the language pair-directions. It is clear from these plots that the best result is achieved when the trainable greedy decoder was trained to maximize the target decoding objective. When the decoder was trained to maximize sentence-level BLEU, we see the improvement in BLEU but often the degradation in the perplexity (see the left plots in Fig. 2.) On the other hand, when the actor was trained to minimize the perplexity, we only see the improvement in per-

plexity (see the right plots in Fig. 2.) This confirms our earlier claim that it is necessary and desirable to tune for the target decoding objective regardless of what the underlying translation system was trained for, and strongly supports the proposed idea of trainable decoding.

The improvement from using the proposed trainable greedy decoding is smaller when used together with beam search, as seen in Fig. 2 (b). However, we still observe statistically significant improvement in terms of BLEU (marked with red stars.) This suggests a future direction in which we extend the proposed trainable greedy decoding to directly incorporate beam search into its training procedure to further improve the translation quality.

It is worthwhile to note that we achieved all of these improvements with negligible computational overhead. This is due to the fact that our actor is a very small, shallow neural network, and that the more complicated critic is thrown away after training. We suspect the effectiveness of such a small actor is due to the well-structured hidden state space of the underlying neural machine translation model which was trained with a large amount of parallel corpus. We believe this favourable computational complexity makes the proposed method suitable for production-grade neural machine translation (Wu et al., 2016; Crego et al., 2016).

Importance of Critic-Aware Actor Learning In Fig. 3, we show sample learning curves with and without the proposed critic-aware actor learning. Both curves were from the models trained under the same condition. Despite a slower start in the early stage of learning, we see that the critic-aware actor learning has greatly stabilized the learning progress. We emphasize that we would not have been able to train all these 16 actors without the proposed critic-aware actor learning.

Examples In Fig. 4, we present three examples from Ru-En. We defined the influence as the KL divergence between the conditional distributions without the trainable greedy decoding and with the trainable greedy decoding, assuming the fixed previous hidden state and target symbol. We colored a target word with magenta, when the influence of the trainable greedy decoding is large (> 0.001). Manual inspection of these examples as well as others has revealed that the trainable greedy decoder focuses on fixing prepositions and removing any unnecessary symbol generation. More in-depth analysis is however left as future work.

6 Conclusion

We proposed trainable greedy decoding as a way to learn a decoding algorithm for neural machine translation with an arbitrary decoding objective. The proposed trainable greedy decoder observes and manipulates the hidden state of a trained neural translation system, and is trained by a novel variant of deterministic policy gradient, called critic-aware actor learning. Our extensive experiments on eight language pair-directions and two objectives confirmed its validity and usefulness. The proposed trainable greedy decoding is a generic idea that can be applied to any recurrent language modeling, and we anticipate future research both on the fundamentals of the trainable decoding as well as on the applications to more diverse tasks such as image caption generating and dialogue modeling.

Acknowledgement

KC thanks the support by TenCent, eBay, Facebook, Google (Google Faculty Award 2016) and NVidia. This work was partly supported by Samsung Advanced Institute of Technology (Next Generation Deep Learning: from pattern recognition to AI). We sincerely thank Martin Arjovsky, Zihang Dai, Graham Neubig, Pengcheng Yin and Chunting Zhou for helpful discussions and insightful feedbacks.

References

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

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

Kyunghyun Cho. 2016. Noisy parallel approximate decoding for conditional recurrent language model. *arXiv preprint arXiv:1605.03835* .

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

Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. Nyu-mila neural machine translation systems for wmt16. In *Proceedings of the First Conference on Machine Translation, Berlin, Germany. Association for Computational Linguistics*.

Trevor Cohn, Cong Duy Vu Hoang, Ekaterina Vymolova, Kaisheng Yao, Chris Dyer, and Gholamreza Haffari. 2016. Incorporating structural alignment biases into an attentional neural translation model. *arXiv preprint arXiv:1601.01085* .

Marta R Costa-Jussa and José AR Fonollosa. 2016. Character-based neural machine translation. *arXiv preprint arXiv:1603.00810* .

Josep Crego, Jungi Kim, Guillaume Klein, Anabel Rebollo, Kathy Yang, Jean Senellart, Egor Akhanov, Patrice Brunelle, Aurelien Coquard, Yongchao Deng, et al. 2016. Systran's pure neural machine translation systems. *arXiv preprint arXiv:1610.05540* .

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. ACL.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016a. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *NAACL*.

Orhan Firat, Baskaran Sankaran, Yaser Al-Onaizan, Fatos T Yarman Vural, and Kyunghyun Cho. 2016b. Zero-resource translation with multi-lingual neural machine translation. In *EMNLP*.

Jonas Gehring, Michael Auli, David Grangier, and Yann N Dauphin. 2016. A convolutional encoder model for neural machine translation. *arXiv preprint arXiv:1611.02344* .

Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor OK Li. 2016. Learning to translate in real-time with neural machine translation. *arXiv preprint arXiv:1610.00388* .

Thanh-Le Ha, Jan Niehues, and Alexander Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. *arXiv preprint arXiv:1611.04798* .

Nicolas Heess, Gregory Wayne, David Silver, Tim Lillicrap, Tom Erez, and Yuval Tassa. 2015. Learning continuous control policies by stochastic value gradients. In *Advances in Neural Information Processing Systems*. pages 2944–2952.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *EMNLP*. pages 1700–1709.

Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. 2016. Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099* .

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *EMNLP*. pages 388–395.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2016. Fully character-level neural machine translation without explicit segmentation. *arXiv preprint arXiv:1610.03017* .

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. A simple, fast diverse decoding algorithm for neural generation. *arXiv preprint arXiv:1611.08562* .

Jiwei Li, Will Monroe, and Dan Jurafsky. 2017. Learning to decode for future success. *arXiv preprint arXiv:1701.06549* .

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* .

Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, page 605.

Wang Ling, Isabel Trancoso, Chris Dyer, and Alan W Black. 2015. Character-based neural machine translation. *arXiv preprint arXiv:1511.04586* .

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015a. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114* .

Minh-Thang Luong and Christopher D Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. *arXiv preprint arXiv:1604.00788* .

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

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

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

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Edinburgh neural machine translation systems for wmt 16. *arXiv preprint arXiv:1606.02891* .

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

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. 2014. Deterministic policy gradient algorithms. In *ICML*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *NIPS* .

Tijmen Tieleman and Geoffrey Hinton. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning* 4(2).

Zhaopeng Tu, Yang Liu, Lifeng Shang, Xiaohua Liu, and Hang Li. 2016a. Neural machine translation with reconstruction. *arXiv preprint arXiv:1611.01874* .

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016b. Modeling coverage for neural machine translation. *arXiv preprint arXiv:1601.04811* .

Fernanda Viégas, Greg Corrado, Jeffrey Dean, Macduff Hughes, Martin Wattenberg, Maxim Krikun, Melvin Johnson, Mike Schuster, Nikhil Thorat, Quoc V Le, et al. 2016. Google's multilingual neural machine translation system: Enabling zero-shot translation .

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

Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Ł. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *ArXiv e-prints* .

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