Almost Free Semantic Draft for Neural Machine Translation

Xi Ai
College of Computer Science
Chongqing University
barid.x.ai@gmail.com

Bin Fang
College of Computer Science
Chongqing University
fb@cqu.edu.cn

Abstract

Translation quality can be improved by global information from the required target sentence because the decoder can understand both past and future information. However, the model needs additional cost to produce and consider such global information. In this work, to inject global information but also save cost, we present an efficient method to sample and consider a semantic draft as global information from semantic space for decoding with almost free of cost. Unlike other successful adaptations, we do not have to perform an EM-like process that repeatedly samples a possible semantic from the semantic space. Empirical experiments show that the presented method can achieve competitive performance in common language pairs with a clear advantage in inference efficiency. We will open all our source code on GitHub.

1 Introduction

Successful NMT (Neural Machine Translation) (Vaswani et al., 2017; Bahdanau et al., 2015; Johnson et al., 2017; Ng et al., 2019) can translate sentences through left to right or through right to left. However, there is one critical limitation in this diagram. That is, the decoder can only have access to directional information (left-to-right or right-to-left) when processing auto-regressive (Graves, 2013).

To alleviate this pain, there have been three successful lines. **1)** *Generative NMT*: (Zheng et al., 2020; Shah and Barber, 2018; Su et al., 2018; Zhang et al., 2016; Eikema and Aziz, 2019) adapt VAE (variational auto-encoder) (Altieri and Duvenaud, 2015; Kingma and Ba, 2015; Bowman et al., 2016) for NMT that is trained in generative model settings, modeling the semantics of the source and target sentences in latent space. **2)** *Deliberation*: since the problem is caused by the one-pass process of decoding in the auto-regression process, (Xia et al., 2017) present a framework to predict a guess target sentence in the first-pass and jointly considers the encoding and the guess target sentence in the second-pass. **3)** *Soft-prototype*: (Wang et al., 2019) present a framework to generate a prototype on the encoder side and then the decoder can jointly use the encoding and the prototype. Although empirical results show the previous methods can successfully inject global information into the decoder, these methods either introduce computational complexity to the encoder-decoder architecture or employ an EM-like process in inferring, thus requiring even more than 100% additional time to produce and consider global information in inferring.

In this work, we present an efficient method to sample and consider a semantic draft as global information for decoding with almost free of cost, following the line of generative NMT. Concretely, we sample the semantic draft from semantic space that is a Gaussian inference model with learnable parameters. In the classic utilization of the semantic space, e.g., generative NMT, inferring needs to work with the EM-like process that could degrade the inference efficiency significantly. To mitigate the degradation but still use the semantic space, we train the encoder of NMT in multilingual settings and simultaneously train a cross-lingual generator to obtain an approximation of the target-sentence semantic, hence modeling the required semantic space from the approximation and the source-sentence semantic. In inferring, based on the source-sentence semantic and an approximation made by the cross-lingual generator, the semantic draft can be sampled from the semantic space in a one-shot style. Once the semantic draft has been sampled, we aggregate the semantic draft and the encoding so that the variational decoder can simply decompose the aggregation.

We train the model in generative settings with additional loss of KL-divergence that is used to optimize the semantic space, similar to generative NMT training (Zheng et al., 2020; Shah and Barber,

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3931–3941
June 6–11, 2021.

2018; Su et al., 2018; Zhang et al., 2016; Eikema and Aziz, 2019) and VAE training (Altieri and Duvenaud, 2015; Kingma and Ba, 2015; Bowman et al., 2016). Our work can build upon Transformer (Vaswani et al., 2017), LSTM/GRU (Hochreiter and Schmidhuber, 1997; Chung et al., 2014) and Convolutional sequence (Gehring et al., 2017). In this work, we use Transformer as an example to present our idea, evaluating our method on common translation tasks and 5 more comprehensive experiments.

Our empirical study shows that, compared to previously successful methods, our method can achieve competitive performance and has a clear advantage in inference efficiency. Since we do not change the architecture of the NMT model, our model is compatible with common technics in NMT.

2 Background

Notation x and y denotes word embeddings in the source language L_1 and the target language L_2, respectively. $X = (x_0, x_2, ..., x_n) \in R^{N \times d}$ and $Y = (y_0, y_2, ..., y_m) \in R^{M \times d}$ are the sentences sampled from corpora in L_1 and L_2 respectively, where N and M are the sequence length and d is the word embedding dimension. X and Y are parallel sentences that are used in our supervised training. The translation task $X \rightarrow Y$ is denoted as $Y = Dec(Enc(X))$, where Dec and Enc jointly construct an encoder-decoder model. s and t represent the source-sentence semantic for X and the target-sentence semantic for Y in translation, respectively. z is a latent variable to represent a semantic draft, sampled from the semantic space.

NMT (Vaswani et al., 2017; Bahdanau et al., 2015; Johnson et al., 2017; Ng et al., 2019) utilizes $seq2seq$ learning (Sutskever et al., 2014) and auto-regressive (Graves, 2013) to facilitate training and inferring. Concretely, the current translation y_j at time-step j is conditional on $Enc(X)$ and $y_{<j}$, where $y_{<j}$ is the previous translation before j. The intrinsic problem is caused because the translation y_j can only consider $y < j$ without considering $y > j$. Intuitively, a semantic draft or global information including $y < j$ and $y > j$ can benefit the translation y_j because the translation can be consistent with neighboring information.

Some impressive methods have been proposed to produce and consider a draft providing global information for better translation quality. *1)* Generative NMT (including variational NMT) (Shah and Barber, 2018; Zheng et al., 2020; Su et al., 2018; Zhang et al., 2016; Eikema and Aziz, 2019) study latent and continuous space of semantic (Bowman et al., 2016) for NMT, which can sample z. These methods inject z into NMT to provide global information for better translation. Meanwhile, the encoder is encouraged to consider z. In this manner, generative NMT models the joint probability $P_{nmt}(X, Y, z) = p(z)p(X|z)p_{nmt}(Y|X, z)$ in training. For inferring, the model utilizes the EM-like process to maximize a lower bound on $log(p(X, Y))$ by repeatedly guessing or predicting possible Y and resampling z. However, compared to NMT without z, generative NMT costs over 100% additional time in inferring typically. *2)* Sharing the same idea of the reconsideration of the current translation, *Deliberation* (Xia et al., 2017) is proposed to deliberate the complete output of the first-pass decoding as the attention context of the second-pass decoding. With the *Deliberation*, the final translation is based on the understanding of a possible translation in the target language. Although *Deliberation* is employed without the EM-like process, which is more efficient than generative NMT in inferring, the doubled pass increases the time of auto-regressive in decoding that costs 80% additional time in inferring. *3)* (Wang et al., 2019) further consider the inference efficiency and the storage cost, proposing *Soft-prototype* framework to use a prototype. The prototype is an approximation of the target sentence $Y' = (y'_0, ..., y'_i)$, produced by a probability generator R that accepts any x to generate a probability $p(y')$ over the target vocabulary to search y'.

These successful methods, although using different settings and frameworks, share the same idea to inject a draft of the required target sentence and introduce global information to the decoder. Therefore, the decoder can understand the target globally. Concretely, such an idea can be formulated into a framework as:

$$Y = Dec(Enc(X), draft) \tag{1}$$

However, these successful methods either introduce computational complexity to NMT (Wang et al., 2019; Xia et al., 2017) or employ the EM-like process, showing significant degradation in inference efficiency, e.g., GNMT(Shah and Barber, 2018) needs 110% additional inferring time. Intuitively, a high-quality draft should include two main aspects:

1) a good draft should include a global semantic for the target sentence; **2)** a draft should not degrade inference efficiency significantly.

3 NMT with Semantic Draft

In this section, we present our framework and method. We then discuss how to train the model in generative settings and how to tackle optimization challenges in practice.

3.1 Framework

Inspired from previously successful models, we employ the general framework $Y = Dec(Enc(X), draft)$ for our model, presenting the high-level architecture in Figure 1. Concretely, $draft$ is instantiated to z that the general framework is modified to $Y = Dec(Enc(X), z)$. Since z is sampled from the semantic space, our decoder is a *variational* decoder (Altieri and Duvenaud, 2015; Kingma and Ba, 2015; Bowman et al., 2016).

3.1.1 Generative Semantic Draft

To obtain z, we leverage a similar generative process of GNMT (Shah and Barber, 2018), sampling z from the semantic space that is a Gaussian inference model trained by s and t or approximations of s and t at the very least. Typically, s and t are obtained by modeling the semantics of X and Y with the same parameters.

Semantic for Source Sentence $s \in R^d$ is computed by averaging a set of vector representation. Specifically, we first process X to the NMT encoder before averaging, obtaining $Enc(X)$. Then, we compute $s = \frac{1}{N}\sum_{k=0}^{n} Enc(X)_k$.

Semantic for Target Sentence We encourage the model to learn an approximation of t instead of the "ground-truth target semantic". We assume $G(s) \approx t$, where G is a two-layer cross-lingual generator. In other words, we compute a *dummy* target-sentence semantic $G(s)$ based on s. We will discuss this assumption in §4 *Multilingual Encoder and Cross-lingual Generator* and how to train the cross-lingual generator G in §3.2 *Encoder and Generator Tweaking*.

Semantic Space Typically, a Gaussian inference model is used for the semantic space, representing a variational distribution $q_z(z|s,t)$ for sampling (Shah and Barber, 2018; Zhang et al., 2016; Zheng et al., 2020). It serves as an approximate posterior. Instead of $q_z(z|s,t)$, in our model, we use $q_z(z|s,G(s))$ for our required *semantic space* because $G(s)$ is encouraged to learn an approximation of t. Specifically, we concatenate s and $G(s)$ to compute the mean and variance of the diagonal Gaussian as:

$$\begin{aligned} &S = [s, G(s)] \\ &q_z(z|s,G(s)) = \mathcal{N}(W^{\mu}S, diag(exp(W^{\sigma}S))) \end{aligned} \tag{2}$$

3.1.2 Decoding with Draft

As aforementioned, z is sampled from the semantic space $q_z(z|s,G(s))$. We then aggregate z and $Enc(X)$, processing the aggregation to the decoder for decoding. In other words, we add generative context to the encoding for the encoder-decoder attention in the decoder. Therefore, the decoder is a variational decoder that is conditional on z and X.

3.2 Training

NMT Training To train the parameters of both NMT and the semantic space in generative settings, we follow the successful training strategy in previous works (Bowman et al., 2016; Zhang et al., 2016), using SGVB (stochastic gradient variational Bayes) (Kingma and Welling, 2014; Rezende et al., 2014) to perform approximate maximum likelihood estimation:

$$\begin{aligned} &\mathcal{L}(Y|X) = \mathbb{E}_{q_z(z|s,G(s))}[\log p_{nmt}(Y|X,z)] - \\ &\lambda D_{KL} q_z(z|s,G(s))||p(z) \end{aligned} \tag{3}$$

where λ weighs the KL divergence term and $p(z) = \mathcal{N}(0, I)$.

Encoder and Generator Tweaking Intuitively, the semantic space should consider the shared semantics between s and t. Ideally, s and t should be obtained from a shared model by processing X and Y, which is discussed in generative NMT (Shah and Barber, 2018; Zheng et al., 2020; Eikema and Aziz, 2019). In spired by this idea, we use the same NMT encoder to compute $Enc(Y)$, obtaining the "ground-truth target semantic" $t = \frac{1}{M}\sum_{k=0}^{m} Enc(Y)_k \in R^d$. As aforementioned, we do not directly use t for our semantic space, which is different from generative NMT. Instead, we only use t to enforce and regularize $G(s)$ in training. Concretely, we train the cross-lingual generator G to restore t from s so that $G(s) \approx t$.

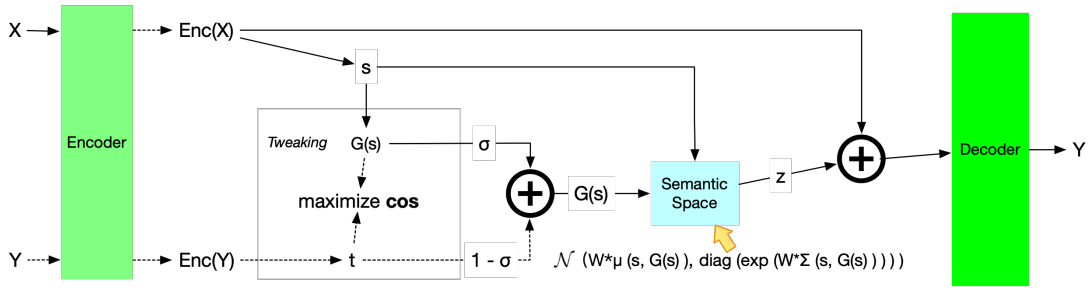

Figure 1: High-level view of NMT with a semantic draft. *Note that the "dotted line" is only used in training.* s and t represent the sentence semantics. The semantic draft z is sampled from the semantic space that is a parameterized space to model the Gaussian inference distribution $q_z(z|s, G(s))$, where G is our cross-lingual generator. σ linearly increases over the course of training so that the model learns to predict without t. cos denotes the similarity between $G(s)$ and t that we encourage $G(s) \approx t$. The variational decoder decomposes the sum of a draft and the encoding.

4 Discussion

4.1 Inferring with Almost Free Draft

Costly Draft In traditionally generative NMT, based on a random target sentence, the inference mode or the process of translation generating makes an initial guess z_{init} from the semantic space or the variational distribution $q_z(z|s, t_{random})$, where s is computed by X and t_{random} is obtained from a random Y_{random}. Then, it can generate a possible translation Y' and its semantic t'. To obtain a good translation, based on the last translation, the inference mode can re-sample a better semantic from the semantic space and regenerate a new translation to maximize a lower bound on $log(p(X, Y))$ in the EM-like process. Readers can also refer to Algorithm 1 in GNMT (Shah and Barber, 2018) for more details.

Almost Free Draft Unlike traditionally generative NMT, we do not need to make an initial guess and also do not employ the EM-like process to sample z for inferring, which improves the inference efficiency. In our model, $G(s)$, which is the *dummy* target semantic, plays a prominent role that aims to approximate t instead of making an initial guess. Therefore, we do not have to make an initial guess, and we can also eliminate the whole EM-like process because z is not randomly sampled, which results in a one-shot sampling. Since G is a simple generator, sampling z from $q_z(z|s, G(s))$ does not hurt the inference efficiency significantly and is almost free of cost.

4.2 Multilingual Encoder and Cross-lingual Generator

Approximation of t In *Encoder and Generator Tweaking* operation, we jointly train the encoder and the cross-lingual generator G to make $G(s)$ and t as similar as possible. Since we input parallel sentences to the encoder, the encoder is encouraged to search multilingual properties. Specifically, we notice that $s \approx t$ potentially[1], which is studied and reported in previous works of multilingual BERT empirically (Devlin et al., 2019; Karthikeyan et al., 2020; Wu and Dredze, 2019). Meanwhile, *Soft-prototype* (Wang et al., 2019) and multilingual NMT (Wu et al., 2016; Johnson et al., 2017) also explore this aspect in NMT scenario. We further introduce the cross-lingual generator G to tweak/finetune the property, observing the significant benefits of regularizing. Most importantly, with the cross-lingual generator G, the model can greedily gain a *dummy* t by $G(s)$ so that the semantic draft can be sampled in a one-shot generative style without the EM-like process.

Potential of s and $G(s)$ Besides, we are aware that only injecting s or $G(s)$ without processing to the semantic space may also provide global information or the shared semantic for decoding because $s \approx t$ and $G(s) \approx t$ potentially. We will present an ablation study in one of our comprehensive experiments §6.5 *Necessity of Semantic Space and Multilingual Encoder* to show the significance of G, the semantic space and their combination.

Semantic in Encoder and Decoder On the other hand, compared to generative NMT, which employs an auxiliary network to help the semantic space by feeding parallel sentences, our method simply processes the parallel sentences to the NMT

[1]There is a difference between s or t and the output of multilingual BERT. Specifically, s and t are sentence representations, whereas multilingual BERT outputs a sequence of the word representation.

encoder that is equivalent to the auxiliary network in generative NMT. In this way, there is no need to pass z to the encoder to model a joint probability $P_{nmt}(X, Y, z) = p(z)p(X|z)p_{nmt}(Y|X, z)$. Specifically, as discussed in VAE (Altieri and Duvenaud, 2015; Kingma and Ba, 2015; Bowman et al., 2016; Zhang et al., 2016), if z involves in the process of encoding, z can guide and regularize the encoder to consider the shared semantic. Therefore, generative NMT models the joint probability in training, encouraged to consider z in both the encoder and the decoder. However, in our model, we let the multilingual encoder consider the implicitly shared semantic itself, and we inject z into the decoder that is encouraged to consider the shared semantic.

4.3 Comparison

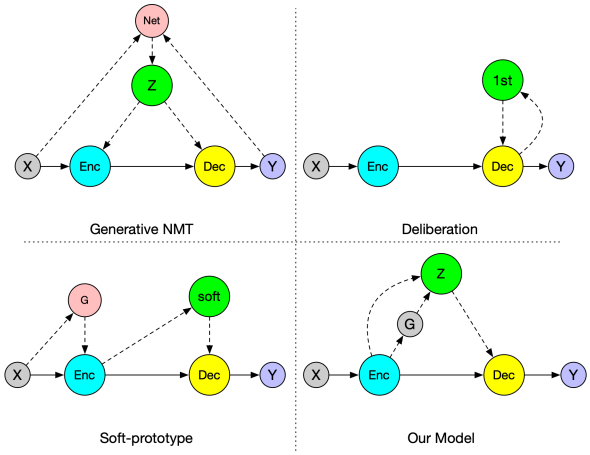

Figure 2: Comparison between our model and previous models. The "dotted line" indicates the flow of global information. *Z* denotes Gaussian semantic space. *Net* denotes an auxiliary network. *G* is a generator. *1st* denotes first-pass decoding. *soft* denotes soft-prototype.

In Figure 2, we compare our framework with previous successful models: GNMT (Shah and Barber, 2018), *Deliberation* (Xia et al., 2017) and *Soft-prototype* (Wang et al., 2019). We observe some significant differences from the perspective of our design:

- **vs GNMT** *1)* The semantic space is built upon the multilingual encoder and the cross-lingual generator in our model; *2)* the semantic/global information is only used in the decoder.
- **vs Deliberation** The global information comes from semantic space instead of the first-pass decoding.
- **vs Soft-prototype** The global information is sampled from the semantic space instead of target prototypes.
- Additionally, we notice an optimization solution for the EM-like process. (Eikema and Aziz, 2019) study an approximating method to maximize the lower bound on $log(p(X, Y))$ by employing an auxiliary distribution with only using source s, which boosts the inference efficiency with a single call (without the EM-like process) to an $argmax$ solver. Compared to their work, our model has three major differences: *1)* our model depends on both s and $G(s)$; *2)* an auxiliary distribution is not necessary in our model; *3)* we focus on the process of draft generating.

4.4 Optimization Challenges

Collapse of $\mathbf{D_{KL}}$ (Bowman et al., 2016) report the collapse of D_{KL} term in the objective function Eq.3. Following the instructions of (Bowman et al., 2016; Shah and Barber, 2018), we apply two common strategies: *1)* λ linearly increases from 0 to 1 over the initial 50k steps during training; *2)* we randomly drop a constant of 30% words when encoding X.

Warm-up of Generator Training is somewhat tricky when using the cross-lingual generator G. We apply a weight $\sigma \in [0, 1]$ for $G(s)$ and a weight $1 - \sigma$ for t, as presented in Figure 1. σ linearly increases from 0 to 1 over 50k steps after $\lambda = 1$. By this strategy, the semantic space is encoruaged to rely on t in warm-up. Significantly, it avoids that $cos(G(s), t)$ is close to 0 at the beginning of training. After warm-up, i.e., $G(s) \approx t$, we use $G(s)$ for the rest of training.

5 Experiment Settings

5.1 Dataset

To be comparable, we train our model on language pairs $\{French, German\} \leftrightarrow English$ and a relative low-resource language pair $Romanian \leftrightarrow English$ which are commonly used in previous work (Shah and Barber, 2018; Vaswani et al., 2017; Bahdanau et al., 2015; Zheng et al., 2020). Concretely, we download parallel corpora $\{French, German, \} \leftrightarrow English$ from WMT 2014 [2] (Bojar et al., 2014). For $Romanian \leftrightarrow$

[2] http://www.statmt.org/wmt14/translation-task.html

$English$, we retrieve parallel corpora from WMT 2016 [3] (Bojar et al., 2016). The preprocess is simple in our case that we only remove sentences with over 50-word length in our training datasets. Following standard evaluation, the model is evaluated on *newstest2014* for $\{French, German\} \leftrightarrow English$ and *newstest2016* for $Romanian \leftrightarrow English$. Case-sensitive BLEU score is computed by *multi-BLEU.perl*[4] to report the performance. We also employ beam search with beam size 4 and length penalty 0.6.

5.2 Model Settings

We implement presented model on Tensorflow 2.0 (Abadi et al., 2016). To be comparable with other models and baselines, the NMT settings are identical to big-Transformer (Vaswani et al., 2017). Specifically, we set model dimension, word embedding, head, encoder layer, decoder layer and FFN filter to 1024, 1024, 16, 6, 6 and 4096. Adam optimizer (Kingma and Ba, 2015) is employed with parameters $\beta_1 = 0.9$,$\beta_2 = 0.98$ and $\epsilon = 10^{-9}$. We use a dynamic learning rate over the course of NMT training (Smith, 2017; Vaswani et al., 2017)[5]. The dropout rate is set to $rate = 0.1$, and label smoothing is used with $gamma = 0.1$ (Mezzini, 2018). Parallel corpora for one translation task (e.g., $Romanian \leftrightarrow English$) are concatenated to train BPE (Sennrich et al., 2016b) with a balance strategy (Lample and Conneau, 2019) that forms a shared vocabulary with $40,000$ subtokens. For data feeding efficiency, each mini-batch of similar-length sentences are padded to the same length and may have a different number of elements in each mini-batch, such that $batch\_size \times padded\_length <= 3000$.

5.3 Reimplementation and Reconfiguration

To be fair, we reimplement some models on our machine with the same mini-batch size. We compare the reimplemented results to the reported results on the same test set to ensure the difference less than 5% (or 1.5) in BLEU. Then, we can confirm the reimplementation and reconfiguration.

[3] http://www.statmt.org/wmt16/translation-task.html

[4] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-BLEU.perl

[5] $lr = peak\_lr \times min(1, step/warm\_up) \times (max(step, warm\_up))^{-0.5}$, where $warm\_up = 3000$ and $peak\_lr = 0.05$.

6 Performance

6.1 Translation Task

We study the methods of how to produce and consider global information for NMT. Since we have discussed three successful directions, we compare our method with the baselines of Transformer (Vaswani et al., 2017), generative NMT including GNMT (Shah and Barber, 2018) and Mirror-GNMT (Zheng et al., 2020), *Deliberation* (Xia et al., 2017) and *Soft-prototype* (Wang et al., 2019). Meanwhile, we have introduced some additional parameters to the model, which is the same as the comparable models. Therefore, we evaluate not only the performance but also the inference efficiency. The comparison of the inference efficiency is based on the inference speed of the vanilla big-Transformer. Besides, we reconfigure Mirror-GNMT and GNMT to big-Transformer settings, and we additionally reimplement *Soft-prototype* on $English \rightarrow Romanian$. Table 1 presents the performance of our model and the baselines on the training dataset. We summarize the results that:

- **Competitive Translation Quality** Our method outperforms the baselines of big-Transformer and GNMT on all the language pairs. Compared to state-of-the-art models, our model gains competitive performance on all the language pairs.

- **Clear Advantage in Inference Efficiency** Besides competitive performance on all the language pairs, our model has a clear advantage in the comparison of inference efficiency. Specifically, GNMT, Mirror-GNMT and *Deliberation* introduce computational complexity to the decoder that needs more than 1 iteration[6] to consider a translation (+ 80% additional time at least), and *Soft-prototype* increases the computational complexity on the encoder side (+ 34% additional time). However, our method only introduces a generator to the model so that the computational complexity in the encoder and the decoder is the same as in vanilla big-Transformer, which results in an efficient inferring and an almost free draft (only + 5% additional time).

- **Improvement from EM-like process** We re-

[6] During our test, generative NMT including GNMT and Mirror-GNMT always need 2-3 iterations for the EM-like process, and *Deliberation* needs a constant of 2 iterations.

	newstest2014				newstest2016		Speed
Model	$Fr \rightarrow En$	$En \rightarrow Fr$	$De \rightarrow En$	$En \rightarrow De$	$Ro \rightarrow En$	$En \rightarrow Ro$	
Transformer (Vaswani et al., 2017), *DM baseline*	42.21	41.85	32.11	28.40	34.01	32.83	1 ×
GNMT (Shah and Barber, 2018), *GM baseline*	42.67	42.22	32.54	28.81	34.52	33.34	2.1 ×
Transformer + *Deliberation* (Xia et al., 2017)		42.58		29.11			1.8 ×
Mirror-GNMT (Zheng et al., 2020)			33.11	29.22	34.91	33.87	2.7 ×
Transformer + *Soft-prototype* (Wang et al., 2019)		42.99		29.46		34.12	1.34 ×
our method	42.94	42.73	33.03	29.20	34.89	33.82	1.05×
our method with EM-like process	43.05	42.97	33.31	29.49	35.09	34.18	2.42×

Table 1: Performance of our method. Our method is competitive on translation quality and has a clear advantage in inference efficiency. *DM baseline*: discriminative model baseline. *GM baseline*: generative model baseline.

port a result obtained by employing the EM-like process for our model in the last row. Although there is noticeable room for improvement, it degrades the inference efficiency significantly so that we do not suggest such a combination. We will discuss this result and integration in §6.2 *Drafting with EM-like process*.

6.2 Drafting with EM-like process

In the most discussion of this work, we sample z from $q_z(z|s, G(s))$ in a one-shot generative style for the sake of inference efficiency. The previous evaluation shows that such an idea is feasible. Meanwhile, our model shares some properties with generative NMT, which makes us interested in the integration with the EM-like process for the sake of the best translation quality only.

In this scenario, we have two steps to translate X:

1. We sample a semantic draft z from $q_z(z|s, G(s))$ and gain a possible translation Y'.

2. We then sample a new semantic draft z' from $q_z(z|s, t')$ to predict a possible and new translation Y'', where $t' = \frac{1}{M'} \sum_{k=0}^{M'} Enc(Y')_k$ and M' is the length of Y'.

The second step can be repeated to maximize a lower bound on $log(p_{nmt}(Y|X))$. We observe some improvements from employing the EM-like process, reporting the result in the last row of Table 1 that we achieve the best performance on all the language pairs. However, most significantly, the translation converges at $2 \sim 3$ iterations that increase the inference time by 137% (from $1.05\times$ to $2.42\times$). Concretely, the model needs to re-encode the last translation to obtain a new draft and re-decode the new draft to generate a new translation, e.g., re-encode Y' to obtain $Enc(Y')$ and its t', re-sample the draft z' from $q_z(z|s, t')$ and re-decode the aggregation of $Enc(X)$ and z'. Thus, we suggest the one-shot generative style in practice.

Additionally, we realize that in this case the improvement may come from not only the re-sampled draft but also the adaptation of two ideas: *1)* "double encoding" in *Soft-prototype* (Wang et al., 2019) because we encode the previously complete translation/prototype for the next translation; *2)* "double decoding" in *Deliberation* (Xia et al., 2017) because we make more than one complete translation. We will justify the significance of the draft in §6.3 *Test for Draft* and §6.4 *Draft Reliance Test*.

6.3 Test for Draft

We are interested in whether the draft does indeed provide useful semantics/global information. In the last section, the improvement from the EM-like process can intuitively show the effect of the draft because a better-quality draft re-sampled from the last translation continuously improves the performance, but the improvement may only come from "double encoding" and "double decoding". Therefore, we conduct a test to demonstrate that the generative draft learns the desired semantics.

In this test, we share the same *missing word translation* task with GNMT (Shah and Barber, 2018). Concretely, the model is forced to give a translation based on the draft heavily. We share the same settings that each word has a 30% chance of being missing independently. Note that we do not conduct this experiment for *Deliberation* (Xia et al., 2017) and *Soft-prototype* (Wang et al., 2019) because such discriminative models do not sample semantics from the semantic space. Table 2 shows the test result on training dataset $German \leftrightarrow English$ and test dataset *newstest2014*. We observe that our model outperforms GNMT and achieves competitive performance to Mirror-GNMT (Zheng et al., 2020). Specifically, compared to GNMT, our method trains a multilingual encoder and a cross-lingual generator to encourage shared semantics for the semantic space.

	newstest2014			
Model	$De \rightarrow En$	$noisy\ De \rightarrow En$	$En \rightarrow De$	$noisy\ En \rightarrow De$
GNMT (Shah and Barber, 2018)	32.54	23.28	28.81	19.93
Mirror-GNMT (Zheng et al., 2020)	33.11	24.37	29.22	20.74
our method	33.03	23.93	29.20	20.35
our method with EM-like process	33.31	24.21	29.49	20.52

Table 2: Performance of *missing word translation*. Our method is competitive in this scenario even without integrating language modeling to recover noisy input.

	GNMT	Mirror-GNMT	our method	+ EM-like process
D_{KL}	5.73	6.92	6.65	7.03

Table 3: Draft Reliance Test. The average value of $D_{KL} = D_{KL}q_z(z|X,Y)||p(z)$ on test set $German \leftrightarrow English$. Higher value indicates higher reliance on the semantic draft or the latent variable.

Compared to Mirror-GNMT, which gains the improvement from the simultaneously used LM (language model) and back-translation technic (Sennrich et al., 2016a), our model is not integrated with LM to counter noisy input so that Mirror-GNMT gains slightly better performance. We leave the integration with denoising language modeling (Vincent, 2010) for future experiments.

6.4 Draft Reliance Test

We have demonstrated that the semantic draft is useful for the translation task. We further indicate how much the model relies on the semantic draft. Since the objective function Eq.3 is the same as in GNMT (Shah and Barber, 2018) and Mirror-GNMT (Zheng et al., 2020), we report a comparison on the term of $D_{KL} = D_{KL}q_z(z|X,Y)||p(z)$, presenting the result in Table 3. The test is conducted on training dataset $German \leftrightarrow English$ and test dataset *newstest2014* by averaging the value of $D_{KL} = D_{KL}q_z(z|X,Y)||p(z)$. Our method relies on the semantic draft (or the latent variable from the semantic space) heavier than GNMT does. With the EM-like process, the reliance is higher than Mirror-GNMT.

6.5 Necessity of Semantic Space and G

Although the semantic draft does indeed provide useful global information in §6.3 *Test for Draft* and §6.4 *Draft Reliance Test*, we still question the necessity of the semantic space because $G(s) \approx t$ and $s \approx t$. In other words, we can simply process $G(s)$ or s to the decoder, which can provide global information for decoding potentailly. To justify, we train the model on training dataset $German \leftrightarrow English$ and test dataset *newstest2014* with 4 different types of $draft$ based on the framework $Dec(Enc(X), draft)$:

- We use our full-packaged model $draft = z$, where z comes from $q_z(z|s, G(s))$.
- $draft = G(s)$ is set for translation to test the significance of the semantic space.
- To test the significance of G, we set $draft = z'$, where z' comes from $q_{z'}(z'|s)$.
- We test both the significance of G and the semantic space by setting $draft = s$ for translation.

Besides the difference of $draft$, all the other configurations are the same for this test. We report the result in Table 4, and our observations are that:

- According to "row 2 vs row 4", we can see the significance of the cross-lingual generator G.
- "row 3 vs row 4" indicates the significance of the semantic space.
- When focusing on "row 2 vs row 3", G improves general translation performance (column 2&4), and the semantic space improves noisy translation (column 3&5)

We intuitively conclude that the semantic space and the cross-lingual generator G can further smooth and regularize the semantic for decoding, similar to that is found in GNMT (Shah and Barber, 2018) and (Bowman et al., 2016). Moreover, the cross-lingual generator G can only restore a coarse semantic so that the model cannot only rely on $G(s)$ to maintain translation quality when testing in the *missing word translation* task generally.

6.6 Improvement from Non-parallel Data

We have mentioned the multilingual property of the encoder in our design, using the NMT encoder to process X and Y . As reported in multilingual BERT (Devlin et al., 2019; Karthikeyan et al., 2020;

	newstest2014			
$draft$ type	$De \rightarrow En$	$noisy\ De \rightarrow En$	$En \rightarrow De$	$noisy\ En \rightarrow De$
$q_z(z\|s, G(s))$	33.03	23.93	29.20	20.35
$G(s)$	32.85	21.82	29.03	17.97
$q_{z'}(z'\|s)$	32.74	22.34	28.91	19.14
s	32.15	20.92	28.49	17.11

Table 4: Performance with/without semantic space or/and generator.

	newstest2016	
Model	$Ro \rightarrow En$	$En \rightarrow Ro$
Transformer + XLM + non-parallel (Lample and Conneau, 2019)	35.30	34.11
Mirror-GNMT + non-parallel(Zheng et al., 2020)	37.54	35.93
Transformer + *Soft-prototype* + non-parallel (Wang et al., 2019)	38.05	36.41
our method	34.89	33.82
our method + non-parallel	37.42	35.77
our method with EM-like process + non-parallel	38.19	36.53

Table 5: Performance of training with additional non-parallel data. The performance of our method is competitive, significantly improved by non-parallel data.

Wu and Dredze, 2019), sharing encoder for *non-parallel* sentences in different languages can still build shared semantic space implicitly. This leads us to experiment with that we can jointly train the encoder with the objective of multilingual BERT. We then train on a relative low-resource language pair $Romanian \leftrightarrow English$, and we use additional monolingual data *News Crawl articles 2015* from WMT 2016 to jointly train the multilingual encoder with the objective of multilingual BERT. In Table 5, we report competitive results, and the performance is significantly improved by simultaneously using non-parallel data. Note that, when training on non-parallel data, we can pre-train the multilingual encoder with the BERT objective instead of joint training. We leave this idea for further experiments.

7 Conclusion and Future Work

Translation quality can be further improved by global information from the target sentence. Although there have been three feasible solutions, successful methods do not consider inference efficiency carefully, which leads to high cost in inferring. In this work, we present a method/framework to improve the performance of NMT. We sample a semantic draft from semantic space that the decoder can consider the semantic draft to obtain the required global information with high efficiency in inferring. Our empirical study shows that, compared to previously successful methods, our method can achieve competitive performance and has a clear advantage in inference efficiency. Since we do not change the architecture of the NMT model, our model can be further improved by employing pre-training (Lample and Conneau, 2019; Devlin et al., 2019; Radford et al., 2018), back-translation (Sennrich et al., 2016a) and other finetuning methods with non-parallel data. And, our model can also be used in unsupervised NMT (Artetxe et al., 2018; Lample et al., 2018). We leave all these experiments for future work.

References

Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283.

Nicholas Altieri and David Duvenaud. 2015. Variational Inference with Gradient Flows. *NIPS Workshop*, 37:3–6.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*.

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

Ond rej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias

Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Ale s Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bryan Eikema and Wilker Aziz. 2019. Auto-encoding variational neural machine translation. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 124–141, Florence, Italy. Association for Computational Linguistics.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252. PMLR.

Alex Graves. 2013. Generating Sequences With Recurrent Neural Networks. *arXiv preprint arXiv:1308.0850*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

K Karthikeyan, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual bert: An empirical study. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*.

Diederik P Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

Diederik P Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. In *Advances in neural information processing systems*.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*.

Mauro Mezzini. 2018. Empirical study on label smoothing in neural networks. In *WSCG 2018 - Short papers proceedings*.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286, Bejing, China. PMLR.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

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

Harshil Shah and David Barber. 2018. Generative neural machine translation. In *Advances in Neural Information Processing Systems*, volume 2018-Decem, pages 1346–1355.

Leslie N Smith. 2017. Cyclical learning rates for training neural networks. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pages 464–472. IEEE.

Jinsong Su, Shan Wu, Deyi Xiong, Yaojie Lu, Xianpei Han, and Biao Zhang. 2018. Variational recurrent neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 4, pages 3104–3112.

Ashish Vaswani, Google Brain, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in neural information processing systems*, pages 5998–6008.

Pascal Vincent. 2010. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *Journal of Machine Learning Research*, 11:3371–3408.

Yiren Wang, Yingce Xia, Fei Tian, Fei Gao, Tao Qin, CengZiang Zhai, and Tie-Yan Liu. 2019. Neural Machine Translation with Soft Prototype. In *Advances in Neural Information Processing Systems*.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 833–844, Hong Kong, China. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv*, abs/1609.08144.

Yingce Xia, Fei Tian, Lijun Wu, Jianxin Lin, Tao Qin, Nenghai Yu, and Tie-Yan Liu. 2017. Deliberation Networks: Sequence Generation Beyond One-Pass Decoding. In *Advances in Neural Information Processing Systems*, pages 1784–1794. Curran Associates, Inc.

Biao Zhang, Deyi Xiong, Jinsong Su, Hong Duan, and Min Zhang. 2016. Variational neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 521–530, Austin, Texas. Association for Computational Linguistics.

Zaixiang Zheng, Hao Zhou, Shujian Huang, Lei Li, Xin-Yu Dai, and Jiajun Chen. 2020. MIRROR-GENERATIVE NEURAL MACHINE TRANSLATION. In *8th International Conference on Learning Representations, ICLR 2020 - Conference Track Proceedings*.