Synchronous Refinement for Neural Machine Translation

Kehai Chen[1], **Masao Utiyama**[2], **and Eiichiro Sumita**[2], **Rui Wang**[3], **and Min Zhang**[1*]
[1]Harbin Institute of Technology, Shenzhen, China
[2]National Institute of Information and Communications Technology, Kyoto, Japan
[3]Shanghai Jiao Tong University, Shanghai, China
{chenkehai,zhangmin2021}@hit.edu.cn
{mutiyama, eiichiro.sumita}@nict.go.jp
wangrui12@sjtu.edu.cn

Abstract

Machine translation typically adopts an encoder-to-decoder framework, in which the decoder generates the target sentence word-by-word in an auto-regressive manner. However, the auto-regressive decoder faces a deep-rooted *one-pass* issue whereby each generated word is considered as one element of the final output regardless of whether it is correct or not. These generated wrong words further constitute the target historical context to affect the generation of subsequent target words. This paper proposes a novel synchronous refinement method to revise potential errors in the generated words by considering part of the target future context. Particularly, the proposed approach allows the auto-regressive decoder to refine the previously generated target words and generate the next target word synchronously. The experimental results on three widely-used machine translation tasks demonstrated the effectiveness of the proposed approach.

1 Introduction

Recently, the *encoder-decoder* framework has obtained impressive results over various machine translation tasks (Barrault et al., 2020; Akhbardeh et al., 2021). Typically, *decoder* first represents those generated target words as a dependent-time target representation, and then uses an attention mechanism to summarize a dependent-time context from the source input for generating the next target word. Since this generated target word is conditioned on previously generated target words at each time step, the decoding process is often called auto-regressive decoding. Finally, *decoder* generates a target language sentence word-by-word in the auto-regressive decoding manner (Bahdanau et al., 2015; Vaswani et al., 2017).

However, the auto-regressive *decoder* often encounters an inherent *one-pass* issue whereby each generated target word is one element of the final output of the machine translation model regardless of whether it is correct or not. These generated wrong target words are further added to the target historical context to affect the generation of subsequent target words, which hinders the performance of machine translation. Take a Chinese-to-English translation case in Figure 1 generated by a trained neural machine translation (NMT) model (Vaswani et al., 2017), we illustrate the *once-generation* issue. In the generated target translation "**Tgt**", there is first an inappropriate translation "*clean up*" compared with "*monitor*" in the reference "***Ref***". The "*clean up*" is regarded as the final translation to confuse the understanding of the meaning of the source sentence. When these inappropriate or incorrect target words constitute part of the target historical context, the *once-generation* issue further affects the generation of subsequent target words, for example, "*drivers' mobile phone*" is far away from the meaning of source sentence "*the driver plays mobile phone while driving*". To verify this issue, we artificially revised the inappropriate translation "*clean up*" as "*monitor*" during the decoding, and observed that the subsequent translation "*driver plays mobile phone while driving*" in "***Revised***" almost completely expresses the corresponding Chinese meaning. We believe that correcting the potential errors in generated translations will improve the quality of the translations.

Many efforts have been initiated on revising potential errors in the generated target translation for machine translation, for example, automatic post-editing (Niehues et al., 2016; Zhou et al., 2017; Junczys Dowmunt and Grundkiewicz, 2017) and two-pass decoding (Xia et al., 2017; Geng et al., 2018; Nema et al., 2019; Ghazvininejad et al., 2019). Despite their success, most of these approaches **asynchronously** simulated the generation of the next target word and the revision

* Corresponding author

Findings of the Association for Computational Linguistics: ACL 2022, pages 2986 - 2996

Src: 上海市 松江区 交管部门 近期 使用 电子 警察
[*Shanghai*] [*Songjiang District*] [*Traffic Control Department*] [*recently*] [*used*] [*electronic*] [*police*]
整治 驾驶员 开车 玩 手机 ， 一个 星期 查获 30 多 起
[*monitor*] [*driver*] [*driving*] [*plays*] [*mobile phone*] [*in a week*] [*detected*] [*30*] [*more than*] [*cases*]

Tgt: traffic control department of Songjiang District in Shanghai recently used electronic police to clean up drivers' mobile phone, and find more than 30 seizures a week

Revised: traffic control department of Songjiang District in Shanghai recently used electronic police to *monitor driver plays mobile phone during the driving* , and find more than 30 cases a week

Ref: *Shanghai Songjiang District Traffic Control Department recently used electronic police to monitor whether the driver plays mobile phone while driving and detected more than 30 cases in a week*

Figure 1: Chinese-to-English translation cases generated by the standard decoder and the decoder with artificial revision. Note: English words in color are translations from the corresponding Chinese words with the same color.

of the generated target words or required a complex modification of the existing models. In this paper, we propose a novel method to refine the potential errors in the generated target words and generate the next target word **synchronously**. To this end, during the decoding, we consider their target future context for a part of previously generated target words, to synchronously obtain the refinement probabilities of the previously generated words and the generation probability of the next target word at each time step. When the refinement probability is greater than the previous generation probability on the same position, we replace the original generated target word with the revised target word. These refined target words together with the currently generated target word further provide an accurate target historical context for the generation of subsequent target words. Additionally, the proposed approach is easily introduced into the auto-regressive decoder without complex modification. We extensively evaluated it on three widely-used machine translation benchmarks, including WMT14 English-to-German, WMT14 English-to-French, and WMT17 Chinese-to-English, and the experimental results demonstrated the effectiveness of the proposed approach.

2 Background

In this paper, we use the advanced encoder-decoder framework, Transformer (Vaswani et al., 2017), to introduce the language generation models. To simplify the process, we simply format the main self-attention network (SAN) module and do not involve other modules (e.g., positional encoding, multiple stacked layers, and so on). *Encoder* represents the source input X=$\{x_1, \cdots, x_J\}$ as the source representation $\mathbf{H}$= $\{\mathbf{h}_1, \cdots, \mathbf{h}_J\}$ using SANs. *Decoder* then generates the target sentence word-by-word based on $\mathbf{H}$ with attention mechanism and the generated target fragment. Specifically, given a sequence of word vectors in the generated target fragment $\{\mathbf{E}[y_1], \cdots, \mathbf{E}[y_{i-1}]\}$ ($\mathbf{E}$ is the embedding matrix of the target language vocabulary), they are packed into key-value matrices $\mathbf{K}_{i-1}$ and $\mathbf{V}_{i-1}$ at the i-th time-step:

$$\mathbf{K}_{i-1} = \mathbf{V}_{i-1} = \mathrm{M}(\mathbf{E}[y_1], \cdots, \mathbf{E}[y_{i-1}]), \quad (1)$$

where the function $\mathrm{M}(\cdot)$ packs a sequence of word vectors into a matrix. Another $\mathrm{SelfATT}_s$ module is then used to learn the target representation $\mathbf{s}_i$:

$$\mathbf{s}_i = \mathrm{SelfATT}_s(\mathbf{c}_{i-1}, \mathbf{K}_{i-1}, \mathbf{V}_{i-1}), \quad (2)$$

where $\mathbf{c}_{i-1} \in \mathbb{R}^{d_{model}}$ is the previous context vector and d_{model} is the dimension of language generation model. $\mathbf{s}_i$ is then fed into another $\mathrm{SelfATT}_c$ to compute the dependent-time context vector $\mathbf{c}_i$:

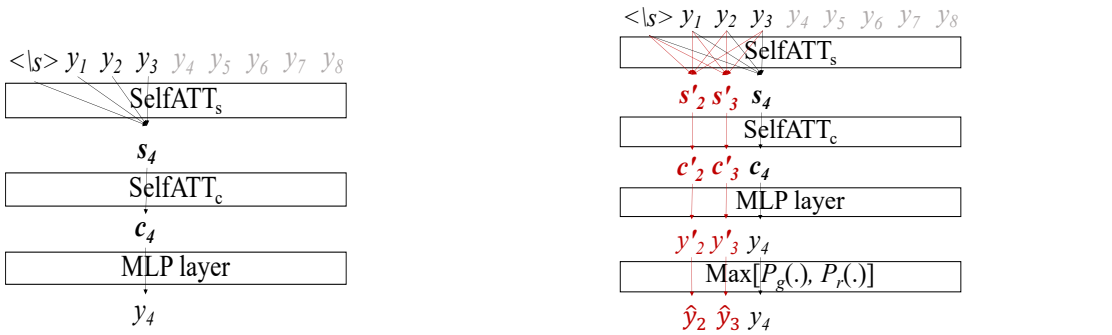
$$\mathbf{c}_i = \mathrm{SelfATT}_c(\mathbf{s}_i, \mathbf{K}_e, \mathbf{V}_e), \quad (3)$$

where $\mathbf{K}_e$ and $\mathbf{V}_e$ are key and value matrices, respectively, that are transformed from the source representation $\mathbf{H}$ according to Eq.(1). The probability distribution $P_g(y_i|y_{<i}, X)$ is then computed using the MLP layer:

$$P_g(y_i|y_{<i}, X) \propto \mathrm{MLP}(\mathbf{c}_i). \quad (4)$$

y_i with the maximum probability is selected as the output of *decoder* at the i-th time-step. To obtain the language generation model θ, the training objective maximizes the conditional generation probability over the training dataset $\{[\mathbf{X}, \mathbf{Y}]\}$:

$$\mathcal{J}(\theta) = P_g(\mathbf{Y}|\mathbf{X}; \theta). \quad (5)$$



(a) Auto-regressive decoder. (b) Auto-regressive decoder with synchronous refinement.

Figure 2: (a) The auto-regressive decoder; (b) The auto-regressive decoder with synchronous refinement where red and black arrows denote the refinement data flow and the generation data flow.

3 Methodology

Intuitively, when there is a target sentence with several incorrect target words, a native target language speaker often detects incorrect words according to the contextual information and tries to revise them. We infer that there are two key aspects in the artificial refinement process: i) How to identify those incorrect target words based on the context information; ii) How to revise the identified incorrect target words. To this end, we propose to simulate the above artificial refinement process, to refine the previously generated target words and generate the next target word synchronously.

3.1 Synchronous Refinement

Formally, at the i-th time step, given a key-value matrix pair $\{\mathbf{K}_{i-1}, \mathbf{V}_{i-1}\}$ (see Eq.(1)) of the generated target language fragment $\{\mathbf{E}[y_1], \mathbf{E}[y_2], \cdots, \mathbf{E}[y_{i-1}]\}$, we first pack the context vectors $\{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_{i-1}\}$ to generate the previous target words into a matrix $\mathbf{C}_{i-1}$ using Eq.(1):

$$\mathbf{C}_{i-1} = \mathrm{M}(\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_{i-1}). \quad (6)$$

We then use $\mathbf{C}_{i-1}$ instead of $\mathbf{c}_{i-1}$ in Eq. (2) to learn the target representation matrix $\mathbf{S}_i$:

$$\mathbf{S}_i = \mathrm{SelfATT}_s(\mathbf{C}_{i-1}, \mathbf{K}_{i-1}, \mathbf{V}_{i-1}), \quad (7)$$

where $\mathbf{S}_i \in \mathbb{R}^{i \times d_{model}}$ is a matrix which includes the updated target representations of previously generated target words $\{\mathbf{s}'_1, \mathbf{s}'_2, \cdots, \mathbf{s}'_{i-1}\}$ in addition to the current target representation $\mathbf{s}_i$. Here, for each of the previously generated target words, $\mathrm{SelfATT}_s$ considers a subset of the future target context to update its target representation. For example, $\mathbf{s}'_3$ of y_3 encodes its future target words $\{y_3, \cdots, y_{i-1}\}$ in addition to its previous target words $\{y_1, y_2\}$. The target future context, which has been shown to be useful for generating the target language in machine translation (Zhang et al., 2018; Zheng et al., 2018; Zhou et al., 2019; Zheng et al., 2019), provides more evidence information for correcting one among all the generated target words in this paper.

Then, $\mathbf{S}_i$ is fed into Eq. (3) to learn a sequence of the context vectors $\mathbf{C}_i$:

$$\mathbf{C}_i = \mathrm{SelfATT}_c(\mathbf{S}_i, \mathbf{K}_e, \mathbf{V}_e), \quad (8)$$

where $\mathbf{C}_i \in \mathbb{R}^{i \times d_{model}}$ is a matrix which includes the updated context vectors of previously generated target words $\{\mathbf{c}'_1, \cdots, \mathbf{c}'_{i-1}\}$ in addition to the current context vector $\mathbf{c}_i$. We then use $\mathbf{C}_i$ as the input to Eq.4 to obtain the combined probabilities:

$$P_r(y'_1, \cdots, y'_{i-1}, y_i | y_{<i}, X) \propto \mathrm{MLP}(\mathbf{C}_i), \quad (9)$$

where $P_r(\cdot)$ includes $i-1$ additional refinement probability distributions at each time step i. That is, $\{y'_1, \cdots, y'_{i-1}\}$ provides a potential error set for revising the generated target words. Furthermore, we select target words with max probabilities from refinement probability distributions as the target candidate words to be refined. When each refinement probability of $\{y'_1, \cdots, y'_{i-1}\}$ is greater than its counterpart in the generation probability $\{y_1, \cdots, y_{i-1}\}$ at previous time step, the previously generated target word y_k (0<k<i) will be replaced with the refined target word y'_k. Finally, the revised target fragment $\{\hat{y}_1, \cdots, \hat{y}_{i-1}, y_i\}$ are computed:

$$\begin{aligned} &P(\hat{y}_1, \cdots, \hat{y}_{i-1}, y_i | y_{<i}, X) = \\ &\quad \max[P_g(y_1, \cdots, y_{i-1}, 0 | y_{<i}, X), \\ &\qquad P_r(y'_1, \cdots, y'_{i-1}, y_i | y_{<i}, X)]. \end{aligned} \quad (10)$$

i-th time-step	full refinement mask							
1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	0
3	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	0
6	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1

(a) Full refinement mask.

i-th time-step	refinement mask							
1	1	1	1	1	0	0	0	0
2	1	1	1	0	0	0	0	0
3	1	1	1	1	1	0	0	0
4	1	1	1	1	1	1	1	0
5	1	1	1	1	1	1	1	0
6	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1

(b) Refinement mask for local constraint.

Figure 3: Full refinement mask and refinement mask with local constraint (i.e., N=3) for learning the target representation during the training, and red number denotes the used future context in the proposed synchronous refinement.

Note that during the inference, the greedy search was used to refine previously generated target words while beam search was only used to generate the next target word. This makes the search complexity of decoding with refinement as consistent as that of the original decoding with beam search, thereby efficiently performing the synchronous refinement in the existing auto-regressive decoding.

3.2 Local Constraint

For the proposed synchronous refinement, most of the generated target words will be refined many times, that is, the number of refinement operations is proportional to the length of the final target sentence. However, the native target language speaker may only revise the previously generated target words a few times. Thus, there may be a potential risk of "*over-refinement*" in the synchronous refinement, that is, excessive refinement operations may lead to new errors. To reduce the risk of "*over-refinement*", we further design a local constraint (see Figure 2) that focuses on refining part of the previously generated target words closest to the target word to be omitted at each time step, inspired by the local attention (Luong et al., 2015) and the fixed iteration prediction (Ghazvininejad et al., 2019). Specifically, in the i-th time step, we select N previously generated target words $\{\mathbf{E}[y_{i-N}], \mathbf{E}[y_{i-N+1}], \cdots, \mathbf{E}[y_{i-1}]\}$ closest to the target word to be omitted, and pack the context vectors $\{\mathbf{c}_{i-N}, \mathbf{c}_{i-N+1}, \cdots, \mathbf{c}_{i-1}\}$ into $\boldsymbol{\mathcal{C}}'_{i-1}$:

$$\boldsymbol{\mathcal{C}}'_{i-1} = \mathrm{M}(\mathbf{c}_{i-N}, \mathbf{c}_{i-N+1}, \cdots, \mathbf{c}_{i-1}). \quad (11)$$

Then, we feed the local target representation matrix $\boldsymbol{\mathcal{C}}'_{i-1}$ into Eq. (7) instead of $\mathbf{C}'_{i-1}$, and thereby efficiently focus on the revision of part of generated target words according to Eqs.8, 9, and 10 in turn.

3.3 Model Training

When the local constraint of the synchronous refinement is set to N, each generated target word will be refined at most N times. This may be inefficient for the training of machine translation models. To efficiently inject this synchronous refinement capability into the training of machine translation models, we introduce an additional refinement mask to select the target future context words under the local constraint. Compared with the existing lower triangle mask, the local constraint contains additional target future context words closest to the target word to be omitted, as shown in Figure 3. After a target word is generated, it will be refined in the subsequent N sequential steps, which indicates that the number of future target words is different. Therefore, the refinement mask with local constraint is to randomly select future target words not greater than N to cover a variety of different future contexts as blue parts in Figure 3.

Then, we use the proposed refinement mask to learn the generation and refinement context vectors at each time step. This allows the machine translation models to synchronously simulate the generation of generated target words and the refinement of the current target word during the training. Thus, the training objective maximizes the generation and the refinement probabilities over the training dataset $\{[\mathbf{X}, \mathbf{Y}]\}$:

$$\mathcal{J}(\theta) = P_g(\mathbf{Y}|\mathbf{X};\theta) + P_r(\mathbf{Y}|\mathbf{X};\theta). \quad (12)$$

Note that the proposed refinement mechanism retains the auto-regressive property of *decoder*

Methods	En-De				Zh-En	En-Fr
	BLEU	#Speed1.	#Speed2.	#Param.	BLEU	BLEU
Trans.base	27.67	13.2k	3.7k	65.0M	24.28	38.42
+Deliberation (Xia et al., 2017)	28.11	11.1k	3.1k	77.8M	24.62	38.94
+Two-stream (Song et al., 2020)	28.17	11.8k	3.4k	79.3M	24.76	39.17
+SynRefinement	28.37+	12.6k	3.5k	65.0M	24.98++	39.46++
Trans.big	28.57	11.2k	2.8k	221.2M	24.84	41.21
+Deliberation (Xia et al., 2017)	28.96	9.3k	2.2k	267.6M	24.97	41.59
+Two-stream (Song et al., 2020)	29.11	9.7k	2.3k	272.4M	25.06	41.55
+SynRefinement	29.22++	10.1k	2.5k	221.2M	25.18+	41.97+

Table 1: Main results of Trans.base/big, +SynRefinement, and comparison +Deliberation and +Two-stream models for standard machine translation tasks. "#Speed1." and "#Speed2." denote the training and decoding speeds (tokens/sec, k for thousand), and "#Param." denotes the size of model parameters (M for million). "+/++" after BLEU scores indicate that our approach was significantly better than Trans.base/big models at significance levels $p<0.05/0.01$ (Collins et al., 2005). Results were reported on average by conducting 3 runs of training.

to ensure the fluency of the target sentence. Meanwhile, the refinement of the generated target words is synchronized with the generation of the current target word at each time step. Particularly, the proposed approach can be easily introduced into the encoder-decoder machine translation models without complex modifications.

4 Experiments

4.1 Setting and Data set

We evaluated the proposed SynRefinement on three widely-used standard machine translation tasks: WMT14 En⇒De includes 4.43 million bilingual sentence pairs, and we used the *newstest2013* and *newstest2014* datasets as the dev set and test set, respectively; WMT14 En⇒Fr includes 36 million bilingual sentence pairs, and we used the *newstest2013* and *newstest2014* datasets as the dev set and test set, respectively; and WMT17 Zh⇒En includes 22 million bilingual sentence pairs, and we used the *newsdev2017* and *newstest2017* datasets as the dev set and the test set, respectively. The byte pair encoding algorithm (Sennrich et al., 2016) was adopted, and the vocabulary size was set to 40K. We set the dimension of all input and output layers to 512, the dimension of the inner feedforward neural network layer to 1024, and the total heads of all multi-head modules to 8 in both the encoder and decoder layers. Each training batch consisted of a set of sentence pairs that contained approximately 4000×8 source tokens and 4000×8 target tokens. To evaluate the test sets, following the training of 200,000 batches, we used a single model obtained by averaging the last five checkpoints, which validated the model with an interval of 2,000 batches on the dev set. We trained all models on eight V100 GPUs and evaluated them on a single V100 GPU. We chose the Transformer NMT model (Vaswani et al., 2017) as our baseline. For other configures of Transformer (e.g., Trans.base/big) models, we followed the settings in (Vaswani et al., 2017). We used the multi-bleu.perl script as the evaluation metric for the three translation tasks.

4.2 Translation Results

Table 1 showed BLEU scores of the baseline Trans.base/big models, +SynRefinement, +Deliberation (Xia et al., 2017) and +Two-stream (Song et al., 2020) attention models for comparison. First, +SynRefinement performed better than the baseline Trans.base/big models for three language pairs. This indicates that the proposed refinement mechanism improved the performance of NMT. Second, +SynRefinement was superior to the comparison +Deliberation network model, which confirms our hypothesis that jointly simulating generation and refinement of target sentence was better than the isolated multi-pass decoding way. Additionally, +SynRefinement outperformed the comparison +Two-stream attention model. Also, +SynRefinement did not increase any additional model parameters but +Two-stream attention increased about 19.7% model parameters compared to the baseline Trans.base model. Meanwhile, both training and decoding speeds of +SynRefinement were faster than those of +Deliberation and +Two-stream models. This means that the proposed refinement can more efficiently relieve the "*one-pass*" issue for the machine translation.

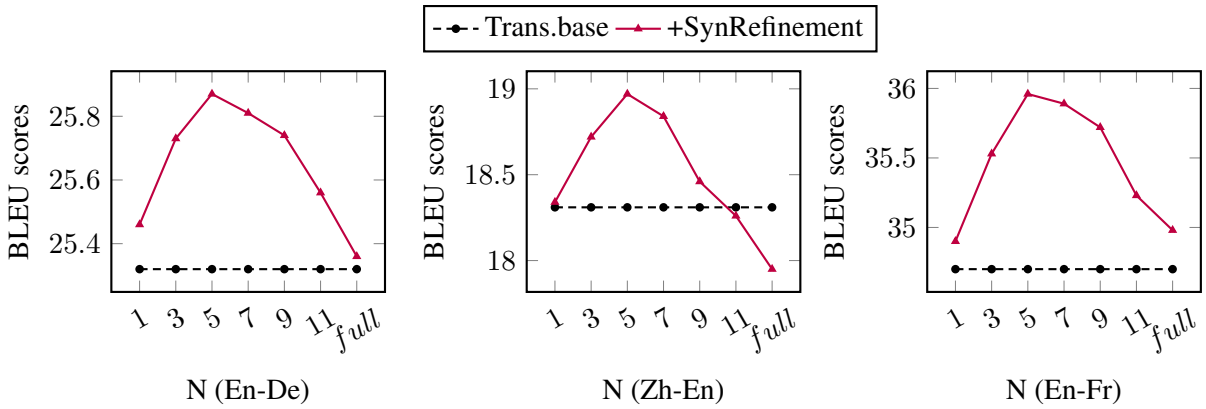


Figure 4: BLEU scores for Trans.base and +SynRefinement models for various hyperparameters N (x-axis) on three translation tasks.

4.3 Hyperparameter N of SynRefinement

In Section 3.3, we designed a local constraint to reduce the "*over-refinement*" risk. Thus, the hyperparameter N in Eq. (11) was used to control the range of refinement operations. Figure 4 shows BLEU scores for Trans.base and +SynRefinement models for various hyperparameter N (x-axis) on the WMT14 En-De, WMT17 Zh-En, and WMT14 En-Fr dev sets. +SynRefinement achieved the highest BLEU scores with N=5 for three dev sets. As a result, we set the hyperparameter N as five to conduct the main experiments shown in Table 1.

4.4 Ablation of Local Constraint and Refinement Mask

Methods	En-De	Zh-En	En-Fr
Trans.base	27.57	24.28	38.42
+Refinement Mask	27.73	24.39	38.78
+Local Constraint	27.89	24.62	38.95
+Both	28.37	24.98	39.46

Table 2: Ablation results of local constraint and refinement mask.

We incrementally added Refinement Mask and Local Constraint into the training and decoding passes of Trans.base model to evaluate their effectiveness. Table 3 showed the ablation results of Trans.base, +Refinement Mask, +Local Constraint, and Both (+Refinement Mask+Local Constraint) models. First, when Refinement Mask and Local Constraint were introduced to the training and the decoding, respectively, BLEU scores were higher than those of the Trans.base model on three translation tasks. The proposed approach was beneficial to the performance improvement of NMT. Second, when both Refinement Mask and Local Constraint were introduced into the training and decoding of the Trans.base model simultaneously, performance improved further. This means that maintaining consistent refinement operations in training and decoding helped NMT generate faithful and fluent target translation.

4.5 Investigation of SynRefinement Operation

Refinement times	En-De	Zh-En	En-Fr
#1	1,829	1,139	2,187
#2	1,121	621	1,431
#3	691	403	896
#4	277	189	382
#5	169	107	189
Total	4,087	2,459	5,085

Table 3: Statistical results for different refinement times on the same position for three translation tasks.

The proposed SynRefinement aims to revise potential errors in the generated target words. To investigate the effectiveness of synchronous refinement operations, we counted the number of different refinement times (e.g., replacement and deletion operation) on the same position during the inference. For example, "#2" denotes the number of BPE tokens that have been replaced (or refined) twice during the decoding. Table 3 showed statistical results for translations (include 74,487, 57,497 and 92,207 BPE tokens, respectively) of three translation tasks generated by Trans.base+SynRefinement models in Table 1. We observed that among the translations generated

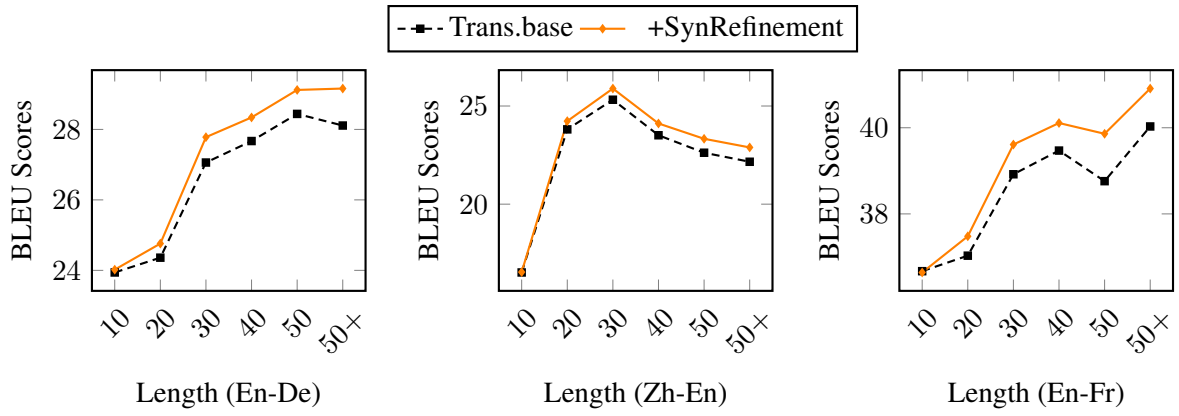


Figure 5: BLEU scores of various target translation lengths for the three translation tasks.

Src: 上海市 松江区 交管部门 近期 使用 电子 警察 整治 驾驶员
[*Shanghai*] [*Songjiang District*] [*Traffic Control Department*] [*recently*] [*used*] [*electronic*] [*police*] [*monitor*] [*driver*]
开车 玩 手机 , 一个 星期 查获 30 多 起
[*driving*] [*plays*] [*mobile phone*] [*in a week*] [*detected*] [*30*] [*more than*] [*cases*]

Trans.base: *traffic control department of Songjiang District in Shanghai recently used electronic police to clean up drivers' mobile phone, more than 30 seizures a week*

+SynRefinement: *traffic control department of Songjiang District in Shanghai recently used electronic police to monitor that the driver plays mobile phone, and discovered more than 30 cases in a week*

Ref: *Shanghai Songjiang District Traffic Control Department recently used electronic police to monitor whether the driver plays mobile phone while driving and detected more than 30 cases in a week*

Figure 6: Chinese-to-English translation cases generated by Trans.base and +SynRefinement models. Note: English words in color are translations from the corresponding Chinese words with the same color.

on the three tasks, total refinement operations of the same position occurred in 4,087, 2,459, and 5,058 positions, respectively. This indicates that the proposed SynRefinement worked during the decoding. When refinement times gradually increased from #1 to #5 on the same position, the number of such BPE tokens was greatly reduced.

4.6 Effect of Different Target Lengths

In the proposed SynRefinement, the number of refined words increased as the length of the generated target translation increased. To investigate the effect of SynRefinement on translations with different lengths, we divided each test set into six groups according to the length of the target translations. For example, "20" indicates that the length of target translations was between twenty and thirty. Figure 5 shows BLEU scores of the Trans.base and +SynRefinement models for the WMT14 En-De, WMT17 Zh-En, and WMT14 En-Fr test sets.

First, when the length of target translations was between zero and ten, BLEU scores of +SynRefinement were almost the same as those of Trans.base model. This reason may be that the refinement operation was performed from the fifth time step for three translation tasks. Second, when the length of the target translations was more than ten, BLEU scores of +SynRefinement were higher than those of Trans.base models for three translation tasks. Particularly, the extent of the improvement gradually increased as the length of the target translations increased. This means that +SynRefinement improved the quality of the target translations, especially long target translations.

4.7 Case Study

Figure 6 showed Chinese-to-English translation cases generated by Trans.base and +SynRefinement models. Trans.base first generated an inappropriate translation "*clean up*" and missed two key verbs "*plays*" and "*detected*", and thereby generated a incorrect translation "*seizures*" compared to the "***Ref***". Thus, the meaning in the target fragment was extremely confusing after "clean up" in the translation generated by the Trans.base model. +SynRefinement considered the future context "*that the driver mobile phone*" together

with the past context "*traffic ... police to*" to revise "*clean up*" as the correct "*monitor*". "*monitor*" constituted part of the target historical context which allowed +SynRefinement model to generate two missed key verbs "*plays*" and "*discovered*". +SynRefinement further generated the correct target word "*cases*" compared to the inappropriate "*seizures*". As a result, the translation generated by +SynRefinement was closer to the reference than that generated by the Trans.base model.

5 Related Work

5.1 Automatic Post-Editing

For the refinement of target output in the classical machine translation (MT), a direct method is automatic post-editing (APE) (Simard et al., 2007). APE is the process of automatic correction of raw MT output, so that a closer resemblance to human post-edited MT output is achieved. Béchara et al. (2011) proposed to automatically create a new joined MT output and source token pairs to improve the automatic post-editing results (Béchara et al., 2012; Pal et al., 2017). Also, the MT output is refined by humans or another model (Niehues et al., 2016; Junczys Dowmunt and Grundkiewicz, 2017), which indicates that the generating and refining are two separate processes in APE.

5.2 Two-Pass Auto-regressive Decoding

As the encoder-decoder framework becomes the dominant machine translation method, the generated potential errors caused by the "*one-pass*" issue still is a challenge. Many studies proposed two-pass decoding to revise the fixed potential errors in the auto-regressive machine translation (Yang et al., 2016; Xia et al., 2017; Zhang et al., 2018; Geng et al., 2018; Zhou et al., 2019; Nema et al., 2019; Song et al., 2020). For example, review network (Yang et al., 2016) was proposed to refine the source representation for the caption generation model. Compared with reviewing the source information, a deliberation network (Xia et al., 2017) proposed two levels of decoders which generate a draft of the target sentence and polish the draft of the target sentence for MT, respectively. Additionally, most relevant to our work is that Song et al. (2020) leveraged the scheduled sampling to simulate the prediction errors during training and designed an additional content-stream attention network to correct the generated error information, which requires complex two-stream attention (Yang et al., 2019) or dual attention (Novak et al., 2016). The refinement network (Nema et al., 2019) for the QA task used a dual attention network to refine the question generated by the first decoder, thereby making the answer correct in the second decoder.

5.3 Iterative Refinement for Non-autoregressive Decoding

Non-autoregressive decoding (Gu et al., 2018) was introduced to generate all words at once, but its performance was far away from that of the auto-regressive decoding due to lack of sufficient dependency modeling among target words. Lee et al. (2018) designed an iterative inference strategy to minimize the generation latency. Then, Ghazvininejad et al. (2019) proposed to first predict all of the target words non-autoregressively, and then repeatedly masked out and regenerated the subset of words for iterative refining the target translation. Different from the refinement of discrete target words, the iterative inference (Lee et al., 2020) was proposed to iterative perform refinement in the continuous space for enhancing dependency between target words.

Discussion: Inspired by iterative refinement for non-autoregressive decoding, we proposed a novel synchronous refinement for machine translation. The proposed approach differs from previous studies in two ways. First, our method allows the machine translation models to refine the previously generated target words and to generate the current target word synchronously instead of APE with separate refinement and asynchronous two-pass (or multi-pass) decoding. Second, the proposed SynRefinement can be introduced to the machine translation models efficiently without complex modification of the existing machine translation models. Additionally, the proposed SynRefinement can help the real-time machine translation scenarios to satisfy the practical application requirements.

6 Conclusion

This paper explored part of the target future context to revise fixed potential errors in the generated target fragment caused by the "*one-pass*" issue of the auto-regressive decoder. We proposed a novel SynRefinement approach to the machine translation, where the refinement of generated target words is synchronized with the

generation of the next target word at each time step. Meanwhile, the proposed SynRefinement can be easily introduced into the encoder-decoder framework without complex modifications. We evaluated the effectiveness of the proposed approach on three classical machine translation tasks. In the future, we will try to explore how to intelligently identify and correct generation errors.

Acknowledgments

We are grateful to the anonymous reviewers, area chair, senior area chair, and program committee for their insightful comments and suggestions. Min Zhang was partially supported by the National Natural Science Foundation of China (No. 62036004). Rui Wang is with MT-Lab, Department of Computer Science and Engineering, School of Electronic Information and Electrical Engineering, and also with the MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai 200204, China.

References

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

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

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Hanna Béchara, Raphaël Rubino, Yifan He, Yanjun Ma, and Josef van Genabith. 2012. An evaluation of statistical post-editing systems applied to RBMT and SMT systems. In *Proceedings of COLING 2012*, pages 215–230, Mumbai, India. The COLING 2012 Organizing Committee.

Hanna Béchara, Ma Yanjun, and Genabith Josef van. 2011. Statistical post-editing for a statistical mt system. In *the 13th Machine Translation Summit*, page 308–315, Xiamen, China.

Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 531–540, Ann Arbor, Michigan. Association for Computational Linguistics.

Xinwei Geng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2018. Adaptive multi-pass decoder for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 523–532, Brussels, Belgium. Association for Computational Linguistics.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6112–6121, Hong Kong, China. Association for Computational Linguistics.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*.

Marcin Junczys Dowmunt and Roman Grundkiewicz. 2017. An exploration of neural sequence-to-sequence architectures for automatic post-editing. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 120–129, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Brussels, Belgium. Association for Computational Linguistics.

Jason Lee, Raphael Shu, and Kyunghyun Cho. 2020. Iterative refinement in the continuous space for

non-autoregressive neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1006–1015, Online. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Preksha Nema, Akash Kumar Mohankumar, Mitesh M. Khapra, Balaji Vasan Srinivasan, and Balaraman Ravindran. 2019. Let's ask again: Refine network for automatic question generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3314–3323, Hong Kong, China. Association for Computational Linguistics.

Jan Niehues, Eunah Cho, Thanh-Le Ha, and Alex Waibel. 2016. Pre-translation for neural machine translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1828–1836, Osaka, Japan. The COLING 2016 Organizing Committee.

Roman Novak, Michael Auli, and David Grangier. 2016. Iterative refinement for machine translation. *CoRR*, abs/1610.06602.

Santanu Pal, Sudip Kumar Naskar, Mihaela Vela, Qun Liu, and Josef van Genabith. 2017. Neural automatic post-editing using prior alignment and reranking. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 349–355, Valencia, Spain. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Michel Simard, Cyril Goutte, and Pierre Isabelle. 2007. Statistical phrase-based post-editing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 508–515, Rochester, New York. Association for Computational Linguistics.

Kaitao Song, Xu Tan, and Jianfeng Lu. 2020. Neural machine translation with error correction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3891–3897. International Joint Conferences on Artificial Intelligence Organization.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

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 I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 1784–1794. Curran Associates, Inc.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Zhilin Yang, Ye Yuan, Yuexin Wu, William W. Cohen, and Ruslan R. Salakhutdinov. 2016. Review networks for caption generation. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, page 2369–2377, Barcelona, Spain. Curran Associates Inc.

Xiangwen Zhang, Jinsong Su, Yue Qin, Yang Liu, Rongrong Ji, and Hongji Wang. 2018. Asynchronous bidirectional decoding for neural machine translation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18)*, pages 5698–5705, New Orleans, Louisiana, USA. AAAI Press.

Zaixiang Zheng, Shujian Huang, Zhaopeng Tu, Xin-Yu Dai, and Jiajun Chen. 2019. Dynamic past and future for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 931–941, Hong Kong, China. Association for Computational Linguistics.

Zaixiang Zheng, Hao Zhou, Shujian Huang, Lili Mou, Xinyu Dai, Jiajun Chen, and Zhaopeng Tu. 2018. Modeling past and future for neural machine translation. *Transactions of the Association for Computational Linguistics*, 6:145–157.

Long Zhou, Wenpeng Hu, Jiajun Zhang, and Chengqing Zong. 2017. Neural system combination for machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*,

pages 378–384, Vancouver, Canada. Association for Computational Linguistics.

Long Zhou, Jiajun Zhang, and Chengqing Zong. 2019. Synchronous bidirectional neural machine translation. *Transactions of the Association for Computational Linguistics*, 7:91–105.