Correct-and-Memorize: Learning to Translate from Interactive Revisions

Rongxiang Weng[1,2], Hao Zhou[3], Shujian Huang[1,2*], Lei Li[3], Yifan Xia[1,2], Jiajun Chen[1,2]
[1]**National Key Laboratory for Novel Software Technology, Nanjing, China**
[2]**Nanjing University, Nanjing, China**
[3]**ByteDance AI Lab, Beijing, China**
{wengrx, huangsj, xiayf, chenjj}@nlp.nju.edu.cn
{haozhou.nlp, lileilab}@bytedance.com

Abstract

State-of-the-art machine translation models are still not on a par with human translators. Previous work takes human interactions into the neural machine translation process to obtain improved results in target languages. However, not all model-translation errors are equal – some are critical while others are minor. In the meanwhile, same translation mistakes occur repeatedly in similar context. To solve both issues, we propose CAMIT, a novel method for translating in an interactive environment. Our proposed method works with critical revision instructions, therefore allows human to correct arbitrary words in model-translated sentences. In addition, CAMIT learns from and softly memorizes revision actions based on the context, alleviating the issue of repeating mistakes. Experiments in both ideal and real interactive translation settings demonstrate that our proposed CAMIT enhances machine translation results significantly while requires fewer revision instructions from human compared to previous methods.

1 Introduction

Recently, sequence-to-sequence models [Sutskever *et al.*, 2014; Cho *et al.*, 2014] have gain superior performance in machine translation [Bahdanau *et al.*, 2014; Vaswani *et al.*, 2017]. Yet these state-of-the-art neural machine translation (NMT) models still fail to generation target sentences with comparable quality as human translators.

To obtain high-quality translation, there are a few recent works attempt to incorporate human revision instructions into the algorithmic translation process. For example, previous interactive NMT [Sanchis-Trilles *et al.*, 2014; Álvaro Peris *et al.*, 2016; Knowles and Koehn, 2016] proposes to ask human to revise the translation output from the beginning of a sentence to the end (i.e. from the left to right), and regenerates the partial translation on the right side of the revised token. We refer to these models as *uni-directional* interactive models. The interactive steps can repeat multiple times until a satisfactory sentence is obtained. In this

---

*Corresponding author.
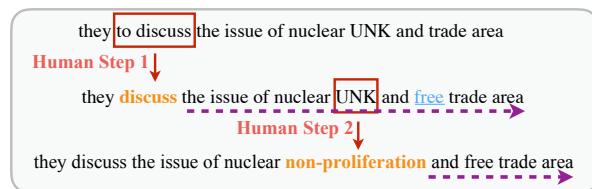
(a) The previous uni-directional interactive NMT method could fix the mistake of "trade" → "free trade" automatically after revising "to discuss" → " discuss".
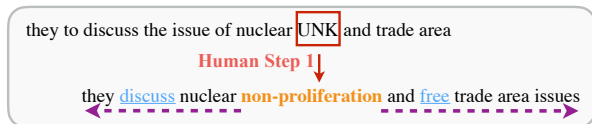
(b) Our proposed CAMIT takes the most critical revision "*UNK*" → "non-proliferation" and automatically fixes the rest mistakes across the whole sentence.

Figure 1: Illustration of the previous and our proposed interactive NMT methods. Words with bold and orange fonts are revised by human. Those with underlines and blue fonts are automatically fixed by models. Note our proposed CAMIT only needs one revision while the previous method requires two.

case, given human revisions for model-translation mistakes, the process could potentially fix some minor mistakes automatically, exploiting the base translation models' ability to save human efforts (See Figure 1a).

In this paper, we make a few observations about neural machine translation and note two issues in previous interactive NMT approaches. First, the previous uni-directional interactive NMT method could only possibly fix those errors to the right of human-revised words. Cheng *et al.* [2016] show that revising critical mistakes first could significantly reduces the number of revisions. However, their intuition could not be directly applied to current uni-directional NMT models, because after revising critical mistakes, uni-directional interactive model cannot correct minor mistakes to the left of the

revision, even with advanced decoding algorithm [Hokamp and Liu, 2017; Post and Vilar, 2018; Hasler *et al.*, 2018]. Since the uni-directional interactive model only regenerates the partial sentence to the right of revisions, instead of updating the whole sentence. This leads to the constraint that each human revision should be proposed at the left most errors. Thus, human annotators have to revise these remaining mistakes manually, which is not efficient enough.

Second, neural machine translation models often keep making same mistakes with similar discourse and domain context. Previous interactive NMT models rarely exploit the revision history to avoid similar mistakes. Our basic intuition is that previous sentences which have been corrected by human may be of greatly help to the translation of following sentences. Learning from past revisions could preventing NMT models from making similar mistakes.

To address the above issues, we propose a correct-and-memorize framework for interactive machine translation (CAMIT), a novel framework to perform efficient corrections and then softly memorize those corrections for enhancing translation results [1].

First of all, CAMIT contains a improved decoder given a sentence and one revised word. CAMIT performs the interactive NMT from both ends of the sentence, updating the whole sentence after getting a revision. Different from previous work about bi-directional decoder [Liu *et al.*, 2016; Mou *et al.*, 2016; Liu *et al.*, 2018], our method is designed to take human revisions for interactive NMT. In such case, human can revise the most critical mistake first in arbitrary position of the sentence; and after that the model will update the whole sentence, fixing minor mistakes automatically. Thus the efficiency of interaction could be significantly improved (Figure 1b).

Secondly, we propose to learn from and memorize interactive revisions to enhance the translation results. The revision history could be learned in the *word level* and the *sentence level*, respectively. At the word level, we propose a key-value memory mechanism [Weston *et al.*, 2014] called *revision memory*, to remember prior revisions in the interactive process. At the sentence level, we use the *online learning* approach to fine-tune our base translation model on prior revised sentences, adapting our model parameters to the specific discourse or domain. When translating a new sentence, our model avoids generating similar mistakes in the past by looking up the revision memory, which is especially helpful to the translation of rare words.

Experiments show that our proposed CAMIT outperforms the previous interactive methods in both ideal and real machine translation settings. Our method's decoder obtains the improvement of 17 BLEU points with merely two revisions in the ideal and 8 BLEU points with 1.81 revisions in the real environments, respectively. Using interaction history further helps improve the interactive efficiency and translation quality.
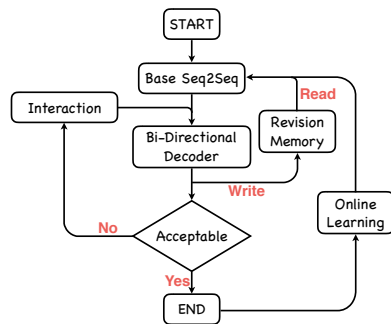

Figure 2: The procedure of our proposed interactive model.

2 Background

Neural machine translation (NMT) is based on a standard Seq2Seq model, which adopts an encoder-decoder architecture for sentence modeling and generation [Sutskever *et al.*, 2014; Cho *et al.*, 2014; Bahdanau *et al.*, 2014; Vaswani *et al.*, 2017]. The encoder summarizes the source sentence into an intermediate representation, and the decoder generates the target sentence from left to right.

Formally, let $\mathbf{x} = \{x_1, \cdots, x_i, \cdots\}$ be a given input sentence and $\mathbf{y} = \{y_1, \cdots, y_j, \cdots\}$ be the output sentence. The encoder transforms the sentence to a sequence of annotations $\mathbf{H}$, with each $\mathbf{h}_i$ being the annotation of input word x_i.

Based on the source annotations, the decoder generates the translation by predicting a target word y_j at each time step j:

$$P(y_j|y_{<j}, \mathbf{x}) = \text{softmax}(g(\mathbf{s}_j)), \quad (1)$$

where g is a non-linear activation function, and $\mathbf{s}_j$ is the decoding state for time step j, computed by

$$\mathbf{s}_j = \overrightarrow{f}(y_{<j}, \mathbf{c}_j). \quad (2)$$

Here $\overrightarrow{f}$ is a recurrent unit [Cho *et al.*, 2014] or self-attention network [Vaswani *et al.*, 2017]. $\mathbf{c}_j$ is a vector summarizing relevant source information. It is computed by the attention mechanism [Luong *et al.*, 2015b; Vaswani *et al.*, 2017].

$$\mathbf{c}_j = \text{ATT}(\mathbf{r}, \mathbf{H}), \quad (3)$$

where the $\mathbf{r}$ is a state from last time step or previous layer when the $\overrightarrow{f}$ is a recurrent unit or a self-attention network, respectively.

In the training stage, $\mathbf{y}$ is the gold sentence provided by training set. NMT models optimize the networks by maximizing the likelihood denoted by $\mathcal{L}_L$.

$$\mathcal{L}_L = \frac{1}{|\mathbf{y}|} \sum_{j=1}^{|\mathbf{y}|} \log P(y_j|y_{<j}, \mathbf{x}). \quad (4)$$

where $|\mathbf{y}|$ is the length of $\mathbf{y}$ and $P(y_j|y_{<j}, \mathbf{x})$ is defined in Equation 1.

3 The Proposed CAMIT

Practically, there may be multiple mistakes in the translation output. Prior work propose to incorporate human efforts to boost the translation performance from left to right,

both in the statistical machine translation (SMT) [Barrachina *et al.*, 2009; González-Rubio *et al.*, 2013] and the neural machine translation (NMT) [Knowles and Koehn, 2016; Álvaro Peris *et al.*, 2016; Wuebker *et al.*, 2016]. However, current interactive model is not efficient enough in both decoding and learning, as previously discussed.

In this section, we propose CAMIT, to improve the interactive efficiency in these two aspects. As shown in Figure 2, after each interaction from human, we propose a *sequential bi-directional decoder* for updating the whole sentence. All revisions are written into a *revision memory*, and the base Seq2Seq model will read from it latter for avoiding making same mistakes. Additionally, previously revised sentence will be learned by an *online learning* approach.

3.1 Preliminary

The interactive process always contains many sentences; and each sentence may need to be revised for many times. One *revision* is the process that human translator manually corrects one mistake in a sentence.

Commonly, the revision can include several operations: *replacement*, *insertion* and *deletion*. Following previous work [Álvaro Peris *et al.*, 2016; Knowles and Koehn, 2016; Cheng *et al.*, 2016], we focus on the replacement and others can be implemented with several replacement operation. For example, deletion could be converted to replace incorrect words with spacing, and the insertion could be converted to replace spacing with inserted words. Furthermore, we only apply the word level revision in this paper, other types of revision, such as phrase, subword, etc., can also be used in our interactive model.

Because human may need to give multiple revisions to one sentence, we refer to the multiple revisions for one sentence as one *round* of interaction. Commonly, adjacent sentences in the interactive process are related, and are usually in the same discourse or domain. We model the interactive process of many related sentences as one *session*. One session may contain many rounds, which could be accomplished by one or more human translators.

3.2 Sequential Bi-Directional Decoding

In the interactive process, let $\mathbf{y} = \{y_1, \cdots, y_j, \cdots\}$ be the initial sequence of words, output by the base Seq2Seq model. If human translator revises the word y_j, the output is divided into 2 parts:

$$\{\cdots, \underbrace{y_{j-2}, y_{j-1}}_{\text{left}}, y_j, \underbrace{y_j, y_{j+1}, \cdots}_{\text{right}}\}.$$

Uni-directional interactive model only updates the *right* part [Álvaro Peris *et al.*, 2016; Knowles and Koehn, 2016], ignoring potential mistakes in the *left* part, which means the revised word should always be the left most error, otherwise the errors in the left part will never be corrected.

We purpose a sequential *bi-directional decoder* for interactive NMT, which can update both parts of the sentence. The proposed model includes two decoders: a forward decoder $\overrightarrow{f}$ and a backward decoder $\overleftarrow{f}$. $\overrightarrow{f}$ and $\overleftarrow{f}$ share the same encoder, but working sequentially.

Formally, in the interactive process, the forward decoder $\overrightarrow{f}$ first acts as the basic model to generate the basic translation: $\mathbf{y} = \{y_1, \cdots, y_j, \cdots\}$. Given a *revision* $\{y_j \rightarrow y_j^r\}$, i.e. revising y_j to y_j^r, $\overrightarrow{f}$ then generates the *new right* part for the translation, the same as in the uni-directional model. The new decoding state $\mathbf{s}'_{j+1}$ for $\overrightarrow{f}$ at the position $j+1$ is computed as:

$$\overrightarrow{\mathbf{s}}'_{j+1} = \overrightarrow{f}(y_j^r, \mathbf{c}'_{j+1}), \tag{5}$$

and the new sentence is

$$\{\cdots, y_{j-2}, y_{j-1}, y_j^r, \underbrace{y'_{j+1}, y'_{j+2}, \cdots}_{\text{new right}}\}.$$

Here words before y_j^r are kept the same, the probabilities of words in the new right part is computed the same as in Equation 1.

Different from previous approaches, we then feed the inverted new right part $\{\cdots, y'_{j+2}, y'_{j+1}, y_j^r\}$ into $\overleftarrow{f}$, obtaining a new decoding state at the current revising position:

$$\overleftarrow{\mathbf{s}}'_{j-1} = \overleftarrow{f}(y_j^r, \mathbf{c}'_{j-1}). \tag{6}$$

Then starting from $\mathbf{s}'_{j-1}$, $\overleftarrow{f}$ generates the *new left* part $\{y'_{j-1}, y'_{j-2}, \cdots\}$ according to a reversed decoding probability:

$$P(y'_{j-1}|y'_{>j}, y_j^r, \mathbf{x}) = \text{softmax}(g(\mathbf{s}'_{j-1})). \tag{7}$$

Finally, the output is

$$\{\underbrace{\cdots, y'_{j-2}, y'_{j-1}}_{\text{new left}}, y_j^r, \underbrace{y'_{j+1}, y'_{j+2}, \cdots}_{\text{new right}}\}.$$

Note that, the length of the new left part may be not equal to the original left part.

In this way, the whole sentence is updated jointly by the two decoders. With the human revision, the new right part is expected to be better than the right part. Based on this better right part, the new left part is expected to become better as well.

The training stage is similar with multi-task models [Dong *et al.*, 2015; Luong *et al.*, 2015a], both decoders could be trained using cross-entropy as the objective:

$$\mathcal{L} = \mathcal{L}_L + \mathcal{L}_R, \tag{8}$$

where $\mathcal{L}_R$ is computed by:

$$\mathcal{L}_R = \frac{1}{|\mathbf{y}|} \sum_{j=1}^{|\mathbf{y}|} \log P(y_j|y_{>j}, \mathbf{x}). \tag{9}$$

The $P(y_j|y_{>j}, \mathbf{x})$ and $\mathcal{L}_L$ are defined in Equation 1 and Equation 4, respectively. $|\mathbf{y}|$ is the length of $\mathbf{y}$.

We have shown how to update the sentence after a single *revision*. However, another challenge for interactive NMT happens when human translator performs several revisions in one *round*, because the interactive process should regenerate the translation with all the revisions considered.

To solve the problem, we propose to combine the grid beam search [Hokamp and Liu, 2017] with our bi-directional decoder. Intuitively, the grid beam search adopts a grid to

store the partial translations that containing the previous revisions. After the gird search, all previous revisions will be included in the final translation output. We propose to perform the grid beam search twice in our bi-directional decoding framework.
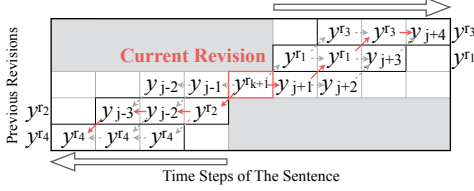

Figure 3: The restricted bi-directional grid beam search.

Specifically, given a current revision y_{k+1}^r and several previous revisions $\{y_1^r, y_2^r, y_3^r, y_4^r\}$, in which $\{y_1^r, y_3^r\}$ are in the right part of the current revision and $\{y_2^r, y_4^r\}$ are in the left part of the current revision, respectively. We will first perform the gird beam search starting from y_{k+1}^r with the forward decoder $\overrightarrow{f}$, obtaining the new right part containing previous revisions ($\{y_1^r, y_3^r\}$) right to y_{k+1}^r. Then given the new right part, we will perform the grid beam search the second time from right to left using the backward decoder $\overleftarrow{f}$, obtaining the new right part containing previous revisions ($\{y_2^r, y_4^r\}$) left to y_{k+1}^r. In such case, all previous revisions ($\{y_2^r, y_4^r, y_1^r, y_3^r\}$) are included in the final output, keeping their sequential order. Figure 3 gives an illustration of the bi-directional grid search process.

3.3 Learning from Interaction History

The interaction history is very helpful to improve the efficiency of interactive models. In this section, we propose methods to learn from the interactive history by online learning and adopting revision memory.

Revision Memory. We first propose to exploit the interaction history in the word level, using a key-value memory [Weston *et al.*, 2014], referred to as the *revision memory*, to memorize previously performed revisions of the session. This makes sense because some specific rare words, e.g. mostly out-of-vocabulary words (*OOV*), occur multiple times in a discourse. The proposed mechanism can help avoid making the same mistakes by memorizing and copying past revisions of the current session.

Specifically, as shown in Figure 4, the revision memory

$$\mathbf{M} = \{(<\mathbf{s}_1', \mathbf{c}_1'>, y_1^r), \cdots, (<\mathbf{s}_T', \mathbf{c}_T'>, y_T^r)\}$$

consists of many items, each of which has a key (*revision context*) and a value (*revised word*). The revised word is used to denote the revision itself. The revision context is used to determine whether the memorized revision should be performed again at the current decoding step. We define the revision context to include two parts: the decoding state $\mathbf{s}'$ (Equation 2) and the context vector $\mathbf{c}'$ (Equation 3) of the past revision, which represent the revision context in aspects of the language model and the translation model, respectively.

Every revision of the current session will be *written* into the memory. When generating a word in decoding, the model
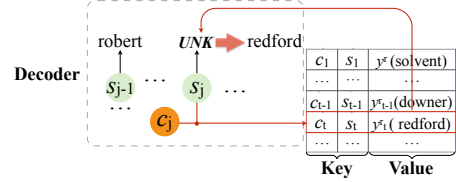

Figure 4: Using revision memory in the decoding stage.

will *read* the revision memory, trying to automatically fix mistakes with a *copy* mechanism [Gu *et al.*, 2016]. The final output distribution of word w is computed by distributions from the decoder and from the revision memory:

$$P_j(w) = (1 - \theta_j) * P_j(w) + \theta_j * r_j(w), \tag{10}$$

$$r_{j,t} = \text{softmax}(W_1 |\mathbf{s}_j \cdot \mathbf{s}_t'| + W_2 |\mathbf{c}_j \cdot \mathbf{c}_t'|), \tag{11}$$

where $r_{j,t}$ is the probability of copying word y_t^r in current position j. It is calculated by the revision context $<\mathbf{s}_j, \mathbf{c}_j>$ of current decoding state and revision contexts of items in the memory. θ_j is a weight computed at each decoding step:

$$\theta_j = \text{sigmoid}(W_s \mathbf{s}_j + W_c \mathbf{c}_j), \tag{12}$$

If the output word w is an unknown word (*UNK*), but has been corrected by human, our method could successfully generate w in future translation by copying it from the revision memory, which partially alleviate the problem of UNK. To keep the translation process going, the embedding of UNK is fed into the decoder, as the input for the next time step.

Online Learning. Adjacent sentences processed in one *session* are highly related, because they are always in the same discourse or domain. In this sense, former revised sentences may greatly help the translation for the following sentences. We use an online learning approach [Peris *et al.*, 2017; Peris and Casacuberta, 2018] to learn from previously corrected sentences in the same session.

The process starts with the basic model trained on the parallel training data. At the beginning of each session, the basic model is used for translation. The parameters of the basic model is fine-tuned after *each round* to maximize the generation probability of the revised sentence at current *round*. Formally, every time we obtain a correct translation pair $\{\mathbf{x}, \mathbf{y}'\}$ after interaction, we update the whole translation model for one step, according to Equation 4. By learning from the sentence level interaction history, our Seq2Seq model better fits sentences of current session. Especially for sessions containing large amounts of sentences, this continuous learning approach may significantly improve the interactive efficiency.

4 Experiment

4.1 Setup

Data-Set. We conduct experiments on standard Chinese-English (ZH-EN) and English-Chinese (EN-ZH) translation tasks. For both ZH-EN and EN-ZH, the training data consists of about 1.6 million sentence pairs.[2] We use NIST03

---

[2]includes LDC2002E18, LDC2003E14, LDC2004T08, LDC2005T06

Model	RN	03	04	05	Ave.	Δ
RNNSearch	–	37.27	37.25	33.64	35.45	–
UniDiR	1	39.35	39.01	37.22	38.12	+2.67
	2	42.09	42.46	40.64	41.55	+6.10
UniDiR$_G$	1	48.88	47.36	44.25	45.81	+10.36
	2	52.58	51.95	47.47	49.71	+14.26
BiDiR	1	49.05	49.22	45.32	47.27	+11.82
	2	**53.19**	**53.70**	**49.71**	**51.71**	**+16.26**

Table 1: The comparison of translation qualities between unidirectional and CAMIT's BiDiR decoder. Here RN is the revision number.

RN	1	2	3	4
RNNSearch	35.45			
BiDiR	47.27	51.71	55.84	59.91
Transformer	43.65			
BiDiR	50.58	53.66	55.73	57.21

Table 2: Average BLEU score of our CAMIT's bi-directional decoder after multiple interactions. RN: revision number.

Model	NIST04	NIST05	AVE.	Δ
RNNSearch	37.25	33.64	35.45	–
+OL	40.62	37.38	39.00	+3.56
+RM	37.66	34.42	36.04	+0.59
CAMIT	**41.13**	**37.95**	**39.54**	**+4.10**
BiDiR	53.70	49.71	51.71	–
+OL	54.75	49.50	52.13	+0.42
+RM	53.96	49.97	51.97	+0.26
CAMIT	**54.86**	**50.46**	**52.66**	**+0.95**

Table 3: The performance gain from online learning (OL) and revision memory (RM). BiDiR is CAMIT without OL and RM.

	Model	BLEU	#Revisions	Costs
ZH-EN	RNNSeach	38.23	–	–
	UniDiR$_G$	43.68	2.76	62.92
	CAMIT-OL-RM	45.82	2.19	62.34
	CAMIT	**46.26**	**1.81**	**51.48**
EN-ZH	RNNSearch	24.52	–	–
	UniDiR$_G$	29.55	2.82	36.60
	CAMIT-OL-RM	**30.36**	2.06	30.90
	CAMIT	30.01	**1.92**	**26.88**

Table 4: Results on the real environment. #Revisions: the average revision number. Costs: revising times, seconds per sentence. Note our proposed CAMIT achieves the best results.

as our validation set, and NIST04 and NIST05 as our test sets. These sets have 919, 1597 and 1082 source sentences, respectively, with 4 references for each sentence. In EN-ZH, we use ref0 of each data set as source sentences. We extract about 0.2 million sentence pairs[3] in our training set, which retains the discourse information for training parameters of the revision memory. We also sample 100 sentences randomly from test sets for the human evaluation.

Implementation Details. We test our proposed approach on both of the RNNSearch [Bahdanau *et al.*, 2014] and Transformer [Vaswani *et al.*, 2017] baselines. We train the bidirectional NMT model with the sentences of length up to 50 words. For the RNNSearch, vocabularies of both Chinese and English includes the most frequent 30K words for both Chinese and English. We map all out-of-vocabulary words to the special token *UNK*. The dimension of word embedding is 512, and the size of hidden layers is 1024. We use the gradient descent approach to update the parameters, with a batch size of 80. The learning rate is controlled by Adam [Kingma and Ba, 2014]. For the Transformer, we apply byte pair encoding (BPE) [Sennrich *et al.*, 2016] to all languages and limit the vocabulary size to 32K. We set the dimension of input and output of all layers as 512, and that of feed-forward layer to 2048. We employ 8 parallel attention heads. The number of layers for the encoder and decoder are 6. Other settings is same as Vaswani *et al.* [2017] . We use beam search for heuristic decoding, and the beam size is set as 4.

We measure the translation quality with the IBM-BLEU score [Papineni *et al.*, 2002]. We implement our interactive NMT model upon our in-house NMT system. The forward and backward decoders of the bi-directional model are trained together with a shared encoder. The learning rate of online learning is 10^{-5}. Following previous work [Tu *et al.*, 2017], we train the revision memory by randomly sampling words and contexts in the same discourse. The size of revision mem-

---

[3] LDC2003E14

ory is 100 and it will be used when revision number is more than 20. When the revision number is more than 100, the first part will be discarded until revision number is less than 100.

Ideal and Real Interactive Environments. Following previous work [Cheng *et al.*, 2016; Álvaro Peris *et al.*, 2016; Hokamp and Liu, 2017], we experiment on both the ideal and real environments. Because real-world human interactions are expensive and time-consuming to obtain, we first report results on the ideal environment, which generates simulated human interactions by identifying critical mistakes. The simulated critical mistake is those lead to the most significant BLEU score improvement after being corrected.

We also validate the interactive efficiency on the real environment, in which three human annotators are asked to revise the translation, until obtaining correct translations. The three human annotators are asked to work with all different interactive systems with no idea of which system they are working with.

4.2 Results on the Ideal Environment

We first evaluate CAMIT for ZH-EN on the ideal interactive environment. Tan *et al.* [2017] demonstrate that two revisions already can lead to very good translation outputs, which is also verified in our experiments on the real interactive environment. We also report the results of our model after multiple interactions as a complement.

Sequential Bi-directional Decoding. First, we valid our proposed bi-directional interactive model (BiDiR) in CAMIT on the RNN based NMT model (RNNSearch). As shown in Table 1, BiDiR significantly outperforms the unidirectional (UniDiR) counterpart, obtaining more than 10 absolute BLEU score improvements.

Model	Number	Δ
RNNSearch	2052	—
+RM	1723	-16.1 %

Table 5: *UNK* number after using the revision memory.

For fair comparison, we also report results of Hokamp and Liu [2017] (UniDiR$_G$), which works in a uni-directional fashion, but using a grid beam search. This enables the uni-directional decoder to revise the critical mistake first as in our bi-directional decoder, but the generation process is also uni-direction. The results for their model still fall behind ours in performance, and the gap may increase on the real environment. Because mistakes left to the revision still exist, and could not be fixed automatically. This harms their interactive efficiency.

We also apply BiDiR decoder on Transformer with the same setting of RNNSearch. By revision once, BiDiR achieves an absolute BLEU improvement of 6.93, upon a strong Transformer baseline (43.65 to 50.58), significantly outperforming UniDiR (+2.04) and UniDiR$_G$ (+4.57). We also compare RNNSearch and Transformer with our proposed bidirectional decoder in Table 2. Results show that although the baseline of Transformer is significantly better than RNNSearch, both of them give pretty high BLEU scores (around 60) after 4 revisions. RNNSearch even obtains higher BLEU scores. For convenience, we will only report results based on RNNSearch in followed experiments.

Learning from Interaction History. Table 3 shows that our baseline RNNSearch model obtains a significant BLEU score improvement of 4.1 on the ideal environment by learning from interaction history with both online learning (+OL) and revision memory (+RM). Additionally, we find that incorporating history learning into BiDiR can still give about 1 absolute BLEU score improvement (+0.95), which is achieved on a very strong baseline with over 50 BLEU score points. This experiment shows that learning from the interaction history is helpful to boost the translation performance. In such case, human will take less revision number for obtaining satisfactory translations.

4.3 Results on the Real Environment

We conduct experiments on the real environment for both EN-ZH and ZH-EN on the RNNSearch based system. We report the average results of 3 volunteers, who could revise arbitrary times until they feel satisfied for the translation. For better comparison, we include the results of Hokamp and Liu [2017] in Table 4. Our proposed model CAMIT gives 8.03 and 5.49 BLEU score improvements on ZH-EN and EN-ZH, respectively, with significantly[4] lower revision numbers than the uni-directional one, and taking less times.

We also compare the number of *UNK* in final translation outputs, between adopting revision memory or not. Table 5 shows that, the revision memory helps relatively decrease the *UNK* rates by 16.1%, reducing the number of *UNK* from 2052 to 1723. This indicates that the revision memory en-

---

[4]P-values are below 0.01 using pairwise t-test.

ables our model to remember the past revisions, which helps avoid making same mistakes, e.g., *UNK*.

4.4 Case study

We list two real cases of the interactive translation from UniDiR$_G$ [Hokamp and Liu, 2017] and our propsed model in Table 6. The two cases are related, and the second case is the next sentence of the first one in a discourse. For the first one, the baseline NMT system makes several mistakes. Our proposed bi-directional interactive model corrects these errors by two revisions, while the uni-directional method takes three revisions. After correcting the word "filmmakers", our bi-directional model automatically fixes the mistake of "*independent*" left to the revision position while the uni-directional model cannot do this.

The second case shows the translation output after using online learning and revision memory, which could reduce the revision number in practice. After using the first revised sentence to tune the model by online training, and remembering past revisions by the revision memory, our model have more prior knowledge that the session is related to the topic of *movie*. So the revised word "redford" and "filmmakers" can be fixed automatically.

5 Conclusion

In this paper, we propose CAMIT, to improve the interactive efficiency of NMT models in both decoding and learning aspects. For decoding, we propose an sequential bi-directional decoder, updating the whole sentence after each revision. For learning, we exploit the interaction history for the sentence level and word level. Experiments show that our model significantly improve the efficiency of interactive NMT.

References

[Bahdanau *et al.*, 2014] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, 2014.

[Barrachina *et al.*, 2009] Sergio Barrachina, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi, Antonio Lagarda, Hermann Ney, Jesús Tomás, Enrique Vidal, et al. Statistical approaches to computer-assisted translation. *CL*, 2009.

[Cheng *et al.*, 2016] Shanbo Cheng, Shujian Huang, Huadong Chen, Xin-Yu Dai, and Jiajun Chen. Primt: A pick-revise framework for interactive machine translation. In *NAACL*, 2016.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares,

1	Src.		独立制片人应当协助维护言论自由, 雷福说
	Ref.		independent filmmakers should help to protect freedom of speech, redford said
	RNNSearch		*independence and should to uphold* freedom of speech, *UNK*
	UniDiR$_G$	1	*independence* **filmmakers** <u>should help to protect</u> freedom of speech, *UNK*
		2	*independence* filmmakers should help to protect freedom of speech, **redford** <u>said</u>
		3	**independent** filmmakers should help to protect freedom of speech, redford said
	CAMIT	1	<u>independent</u> **filmmakers** <u>should help to protect</u> freedom of speech, *UNK*
		2	independent filmmakers should help to protect freedom of speech, **redford** <u>said</u>
2	Src.		雷福呼吁独立制片人协力防止言论自由遭到腐蚀
	Ref.		redford called on independent filmmakers to help prevent the freedom of speech from being eroded
	RNNSearch		*UNK* calls *for independence* to prevent the freedom of speech from being eroded
	CAMIT		<u>redford</u> calls *for* <u>independent filmmakers</u> to prevent the freedom of speech from being eroded

Table 6: Two real cases from the NIST ZH-EN test sets. Words with bold and orange fonts are revised by human. Words with underlines and blue fonts are corrected by the model automatically. Words with Italic fonts are mistakes.

Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, 2014.

[Dong *et al.*, 2015] Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. Multi-task learning for multiple language translation. In *ACL*, 2015.

[González-Rubio *et al.*, 2013] Jesús González-Rubio, Daniel Ortiz-Martínez, José-Miguel Benedí, and Francisco Casacuberta. Interactive machine translation using hierarchical translation models. In *EMNLP*, 2013.

[Gu *et al.*, 2016] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv*, 2016.

[Hasler *et al.*, 2018] Eva Hasler, Adrià De Gispert, Gonzalo Iglesias, and Bill Byrne. Neural machine translation decoding with terminology constraints. *arXiv*, 2018.

[Hokamp and Liu, 2017] Chris Hokamp and Qun Liu. Lexically constrained decoding for sequence generation using grid beam search. In *ACL*, 2017.

[Kingma and Ba, 2014] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, 2014.

[Knowles and Koehn, 2016] Rebecca Knowles and Philipp Koehn. Neural interactive translation prediction. *AMTA*, page 107, 2016.

[Liu *et al.*, 2016] Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. Agreement on target-bidirectional neural machine translation. In *NAACL*, 2016.

[Liu *et al.*, 2018] D. Liu, J. Lv, F. He, and Y. Pu. BFGAN: Backward and Forward Generative Adversarial Networks for Lexically Constrained Sentence Generation. *arXiv*, 2018.

[Luong *et al.*, 2015a] Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. Multi-task sequence to sequence learning. *CoRR*, 2015.

[Luong *et al.*, 2015b] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *EMNLP*, 2015.

[Mou *et al.*, 2016] Lili Mou, Yiping Song, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin. Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. *CoRR*, 2016.

[Papineni *et al.*, 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *ACL*, 2002.

[Peris and Casacuberta, 2018] Á. Peris and F. Casacuberta. Active Learning for Interactive Neural Machine Translation of Data Streams. *CoRR*, 2018.

[Peris *et al.*, 2017] Álvaro Peris, Luis Cebrián, and Francisco Casacuberta. Online learning for neural machine translation post-editing. *CoRR*, 2017.

[Post and Vilar, 2018] Matt Post and David Vilar. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. *arXiv*, 2018.

[Sanchis-Trilles *et al.*, 2014] Germán Sanchis-Trilles, Vicent Alabau, Christian Buck, Michael Carl, Francisco Casacuberta, Mercedes García-Martínez, Ulrich Germann, Jesús González-Rubio, Robin L Hill, Philipp Koehn, et al. Interactive translation prediction versus conventional post-editing in practice: a study with the casmacat workbench. *MT*, 2014.

[Sennrich *et al.*, 2016] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *ACL*, 2016.

[Sutskever *et al.*, 2014] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *In Proceedings of the NIPS*. 2014.

[Tan *et al.*, 2017] Yiming Tan, Zhiming Chen, Liu Huang, Lilin Zhang, Maoxi Li, and Mingwen Wang. Neural post-editing based on quality estimation. In *MT*, 2017.

[Tu *et al.*, 2017] Zhaopeng Tu, Yang Liu, Shuming Shi, and Tong Zhang. Learning to remember translation history with a continuous cache. *arXiv*, 2017.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[Weston *et al.*, 2014] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *CoRR*, 2014.

[Wuebker *et al.*, 2016] Joern Wuebker, Spence Green, John DeNero, Sasa Hasan, and Minh-Thang Luong. Models and inference for prefix-constrained machine translation. In *ACL*, 2016.

[Álvaro Peris *et al.*, 2016] Álvaro Peris, Miguel Domingo, and Francisco Casacuberta. Interactive neural machine translation. *Computer Speech and Language*, 2016.