Learning to Copy for Automatic Post-Editing

Xuancheng Huang[†], **Yang Liu**[†‡*], **Huanbo Luan**[†], **Jingfang Xu**[§] **and Maosong Sun**[†]
[†]Institute for Artificial Intelligence
State Key Laboratory of Intelligent Technology and Systems
Department of Computer Science and Technology, Tsinghua University, Beijing, China
Beijing National Research Center for Information Science and Technology
[§]Sogou Inc., Beijing, China
[‡]Beijing Advanced Innovation Center for Language Resources
hxc17@mails.tsinghua.edu.cn, liuyang2011@tsinghua.edu.cn, luanhuanbo@gmail.com,
xujingfang@sogou-inc.com, sms@tsinghua.edu.cn

Abstract

Automatic post-editing (APE), which aims to correct errors in the output of machine translation systems in a post-processing step, is an important task in natural language processing. While recent work has achieved considerable performance gains by using neural networks, how to model the copying mechanism for APE remains a challenge. In this work, we propose a new method for modeling copying for APE. To better identify translation errors, our method learns the representations of source sentences and system outputs in an interactive way. These representations are used to explicitly indicate which words in the system outputs should be copied, which is useful to help CopyNet (Gu et al., 2016) better generate post-edited translations. Experiments on the datasets of the WMT 2016-2017 APE shared tasks show that our approach outperforms all best published results. [1]

1 Introduction

Automatic post-editing (APE) is an important natural language processing (NLP) task that aims to automatically correct errors made by machine translation systems (Knight and Chander, 1994). It can be considered as an efficient way to modify translations to a specific domain or to incorporate additional information into translations rather than translating from scratch (McKeown et al., 2012; Chatterjee et al., 2015, 2018).

Approaches to APE can be roughly divided into two broad categories: *statistical* and *neural* approaches. While early efforts focused on statistical approaches relying on manual feature engineering (Simard et al., 2007; Béchara et al., 2011), neural network based approaches capable of learning representations from data have shown remarkable superiority over their statistical counterparts (Varis and Bojar, 2017; Chatterjee et al., 2017; Junczys-Dowmunt and Grundkiewicz, 2017; Unanue et al., 2018). Most of them cast APE as a multi-source sequence-to-sequence learning problem (Zoph and Knight, 2016): given a source sentence (src) and a machine translation (mt), APE outputs a post-edited translation (pe).

src	I ate a cake yesterday
mt	**Ich** esse **einen** Hamburger
pe	**Ich** hatte gestern **einen** Kuchen gegessen

Table 1: Example of automatic post-editing (APE). Given a source sentence (src) and a machine translation (mt), the goal of APE is to post-edit the erroneous translation to obtain a correct translation (pe). Our work aims to explicitly model how to copy words from mt to pe (highlighted in bold), which is a common phenomenon in APE.

A common phenomenon in APE is that many words in mt can be **copied** to pe. As shown in Table 1, two German words "Ich" and "einen" occur in both mt and pe. Note that the positions of copied words in mt and pe are not necessarily identical (e.g., "einen" in Table 1). Our analysis on the datasets of the WMT 2016 and 2017 APE shared tasks shows that over 80% of words in mt are copied to pe. As APE models not only need to decide which words in mt should be copied correctly, but also should place the copied words in appropriate positions in pe, it is challenging to model copying for APE. Our experiments show that the state-of-the-art APE method (Junczys-Dowmunt and Grundkiewicz, 2018) only achieves a copying accuracy of 64.63% (see Table 7).

We believe that existing approaches to APE (Varis and Bojar, 2017; Chatterjee et al., 2017; Junczys-Dowmunt and Grundkiewicz, 2017; Unanue et al., 2018) suffer from two major

*Corresponding author: Yang Liu
[1]The source code is available at https://github.com/THUNLP-MT/L2Copy4APE

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 6122–6132,
Hong Kong, China, November 3–7, 2019.

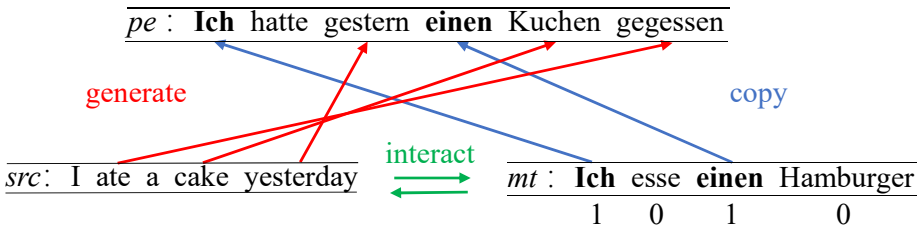

Figure 1: Learning to copy for APE. Our work is based on two key ideas. As both src and mt play important roles in APE, the first idea is that src and mt should "interact" with each other during representation learning to better generate words from src and copy words from mt during inference. The second idea is to predict which target words in mt should be copied since it is easy to obtain labeled data automatically by comparing mt and pe. The words to be copied (e.g., "Ich") in mt are labeled with 1's and other words (e.g., "esse") with 0's.

drawbacks when modeling the copying mechanism. First, the representations of src and mt are learned separately. APE is a two-source sequence-to-sequence learning problem in which both src and mt play important roles. On the one hand, if src is ignored, it is difficult to identify translation errors related to adequacy in mt, especially for fluent but inadequate translations (e.g., mt in Figure 1). On the other hand, mt serves as a major source for generating pe since many words (e.g., "Ich" and "einen" in Figure 1) are copied from mt to pe. Intuitively, it is likely to be easier to decide which words in mt should be copied if src and mt fully "interact" with each other during representation learning. Although CopyNet (Gu et al., 2016) can be adapted for explicitly modeling the copying mechanism in multi-source sequence-to-sequence learning, the lack of the interaction between src and mt still remains a problem.

Second, there is no explicit labeling that indicates which target words in mt should be copied. Existing approaches only rely on the attention between the encoder and decoder to implicitly choose target words to be copied. Given mt and pe, it is easy to decide whether a target word in mt should be copied or not. In Figure 1, the words in mt that should be copied are labeled with 1's. Other words are labeled with with 0's, which should be re-generated from src. These labels can served as useful supervision signals to help better copy words from mt to pe, even when CopyNet is used.

In this work, we propose a new method for modeling the copying mechanism for APE. As shown in Figure 1, our work is based on two key ideas. First, our method is capable of learning the representations of input in an interactive way by enabling src and mt to attend to each other during representation learning. This might be useful for deciding when to generate words from src and when to copy words from mt during post-editing. Second, it is possible to predict which words in mt should be copied because it is easy to automatically construct labeled data by comparing mt and pe. Such predictions can be combined with CopyNet to better model copying for APE. Experiments show that our approach outperforms the best published results on the datasets of the WMT 2016-2017 APE shared tasks.

2 Background

2.1 Multi-source Sequence-to-Sequence Learning

Multi-source sequence-to-sequence learning has been widely used in APE in recent years (Junczys-Dowmunt and Grundkiewicz, 2018; Pal et al., 2018; Tebbifakhr et al., 2018; Shin and Lee, 2018). The architecture of multi-source Transformer is shown in Figure 2(a). It can be equipped with CopyNet (see Section 2.2) to serve as a baseline in our experiments. It is worth noting that src and mt are encoded separately.

Let $\mathbf{x} = x_1 \dots x_I$ be a source sentence (i.e., src) with I words, $\tilde{\mathbf{y}} = \tilde{y}_1 \dots \tilde{y}_K$ be a translation output by a machine translation system (i.e., mt) with K words, and $\mathbf{y} = y_1 \dots y_J$ be the post-edited translation (i.e., pe) with J words. The APE model is given by

$$P(\mathbf{y}|\mathbf{x}, \tilde{\mathbf{y}}; \boldsymbol{\theta}) = \prod_{j=1}^{J} P(y_j|\mathbf{x}, \tilde{\mathbf{y}}, \mathbf{y}_{<j}; \boldsymbol{\theta}), \quad (1)$$

where y_j is the j-th target word in pe, $\mathbf{y}_{<j} = y_1 \dots y_{j-1}$ is a partial translation, $\boldsymbol{\theta}$ is a set of model parameters, and $P(y_j|\mathbf{x}, \tilde{\mathbf{y}}, \mathbf{y}_{<j}; \boldsymbol{\theta})$ is a word-level translation probability.

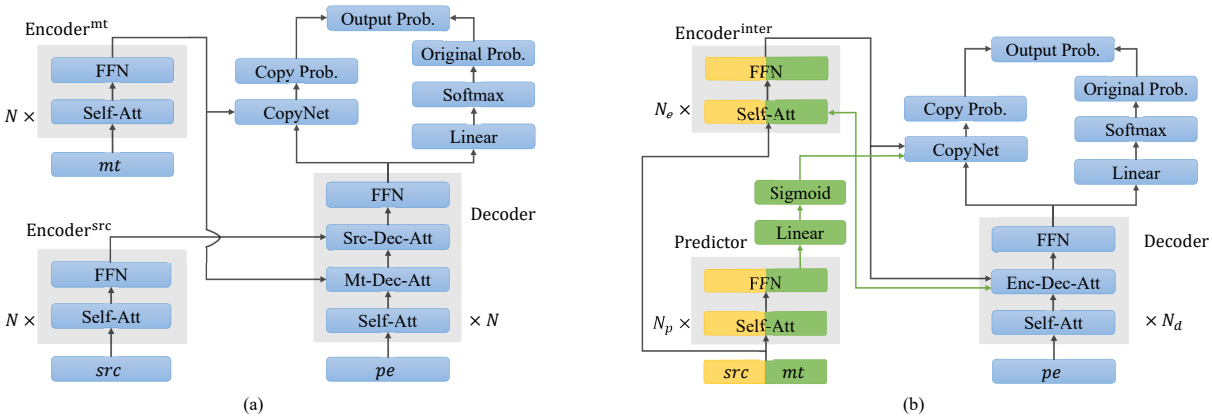

Figure 2: (a) The architecture of multi-source Transformer (Junczys-Dowmunt and Grundkiewicz, 2018) equipped with CopyNet (Gu et al., 2016) and (b) the architecture of our approach. While the existing work learns the representations of src and mt separately, our approach allows for learning the representations of src and mt in an interactive way by concatenating them as a single input. In addition, our approach introduces a Predictor module to explicitly indicate which words in mt should be copied.

The word-level translation probability in Eq. (1) is computed as

$$\mathbf{H}^{\text{src}} = \text{Encoder}^{\text{src}}(\mathbf{x}, \boldsymbol{\theta}), \quad (2)$$

$$\mathbf{H}^{\text{mt}} = \text{Encoder}^{\text{mt}}(\tilde{\mathbf{y}}, \boldsymbol{\theta}), \quad (3)$$

$$\mathbf{h}_j^{\text{pe}} = \text{Decoder}(\mathbf{y}_{<j}, \mathbf{H}^{\text{src}}, \mathbf{H}^{\text{mt}}, \boldsymbol{\theta}), \quad (4)$$

$$P(y_j|\mathbf{x}, \tilde{\mathbf{y}}, \mathbf{y}_{<j}; \boldsymbol{\theta}) \propto \exp(\mathbf{h}_j^{\text{pe}} \mathbf{W}_g), \quad (5)$$

where $\text{Encoder}^{\text{src}}(\cdot)$ is the encoder for src, $\mathbf{H}^{\text{src}}$ is the real-valued representation of src, $\text{Encoder}^{\text{mt}}(\cdot)$ is the encoder for mt, $\mathbf{H}^{\text{mt}}$ is the representation of mt, $\text{Decoder}(\cdot)$ is the decoder, $\mathbf{h}_j^{\text{pe}}$ is the representation of the j-th target word y_j in pe. $\mathbf{W}_g \in \mathbb{R}^{d \times \mathcal{V}_y}$ is a weight matrix, d is the dimension of hidden states, and $\mathcal{V}_y$ is the target vocabulary size.

A limitation of the aforementioned model is that src and mt are encoded separately without interacting with each other, which might lead to the inability to find which src word is untranslated and which mt word is incorrect. For example, the mt sentence in Figure 1 is fluent and meaningful. Without src, the APE system is unable to identify translation errors. In addition, the multi-source Transformer does not explicitly model the copying between mt and pe in neither the Encoder nor the Decoder.

2.2 CopyNet

CopyNet (Gu et al., 2016) is a widely used method for modeling copying in sequence-to-sequence learning. It has been successfully applied to single-turn dialogue (Gu et al., 2016), text summarization (See et al., 2017), and grammar error correction (Zhao et al., 2019).

It is possible to extend the multi-source Transformer with CopyNet to explicitly model the copying mechanism, as shown in Figure 2(a). CopyNet defines the word-level translation probability in Eq. (1) as a linear interpolation of copying and generating probabilities:

$$P(y_j|\mathbf{x}, \tilde{\mathbf{y}}, \mathbf{y}_{<j}; \boldsymbol{\theta}) = \gamma_j \times P^{\text{copy}}(y_j) + (1 - \gamma_j) \times P^{\text{gen}}(y_j), \quad (6)$$

where $P^{\text{copy}}(y_j)$ is the copying probability for y_j, $P^{\text{gen}}(y_j)$ is the generating probability for y_j, and γ_j is a gating weight. They are defined as follows:

$$P^{\text{copy}}(y_j) \propto \exp(g(\mathbf{H}^{\text{mt}}, \mathbf{h}_j^{\text{pe}})), \quad (7)$$

$$P^{\text{gen}}(y_j) \propto \exp(\mathbf{h}_j^{\text{pe}} \mathbf{W}_g), \quad (8)$$

$$\gamma_j = u(\mathbf{H}^{\text{mt}}, \mathbf{h}_j^{\text{pe}}), \quad (9)$$

where $g(\cdot)$ and $u(\cdot)$ are non-linear functions. See (Zhao et al., 2019) for more details.

Copying in APE involves two kinds of decisions: (1) choosing words in mt to be copied and (2) placing the copied words in appropriate positions in pe. CopyNet makes the two kinds of decisions simultaneously. We conjecture that if which words in mt should be copied can be explicitly indicated, it might be easier for CopyNet to copy words from mt to pe correctly. Therefore, it is necessary to design a new method for identifying words to be copied.

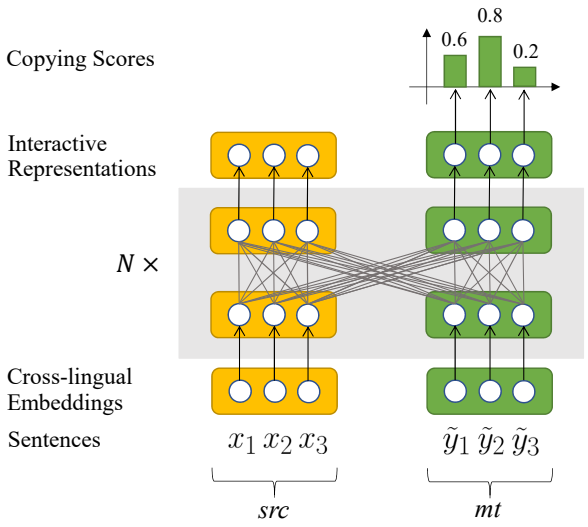

Figure 3: Interactive representation learning. Concatenated to serve as a single input, src and mt attend to each other during representation learning. Note that learnable weights of src and mt are shared. Interactive representation learning is used in both Predictor and Encoder. For example, copying scores are predicted based on the learned representations (see Eq. (13)).

3 Approach

Figure 2(b) shows the overall architecture of our approach. It differs from previous work in two aspects. First, we propose to let src and mt "interact" with each other to learn better representations (Section 3.1). Second, our approach introduces a Predictor module to predict words to be copied (Section 3.2). Section 3.3 describes how to train our APE model.

3.1 Interactive Representation Learning

We propose an interactive representation learning method by making src and mt attend to each other. Following Lample and Conneau (2019) and He et al. (2018), we concatenate src and mt in the dimension of sentence length with additional position and language embeddings:

$$\mathbf{X}_i = \mathbf{E}^{\text{token}}[x_i] + \mathbf{E}^{\text{pos}}[i] + \mathbf{E}^{\text{lang}}[0], \quad (10)$$

$$\tilde{\mathbf{Y}}_k = \mathbf{E}^{\text{token}}[\tilde{y}_k] + \mathbf{E}^{\text{pos}}[k] + \mathbf{E}^{\text{lang}}[1], \quad (11)$$

where $\mathbf{X}_i$ is the embedding of the i-th source word x_i, $\tilde{\mathbf{Y}}_k$ is the embedding of the k-th target word $\tilde{y}_k$, and $\mathbf{E}^{\text{token}}$, $\mathbf{E}^{\text{pos}}$, and $\mathbf{E}^{\text{lang}}$ are the token, position and language embedding matrices.

As shown in Figure 3, the representation of src and mt can be learned jointly:

$$\mathbf{H}^{\text{inter}} = \text{Encoder}^{\text{inter}}([\mathbf{X}; \tilde{\mathbf{Y}}], \boldsymbol{\theta}), \quad (12)$$

where $\text{Encoder}^{\text{inter}}(\cdot)$ is the interactive Encoder and $[\mathbf{X}; \tilde{\mathbf{Y}}]$ is the concatenation of $\mathbf{X}$ and $\tilde{\mathbf{Y}}$ in the dimension of sentence length.

As shown in Figure 2(b), the multi-source Encoders are replaced by the interactive Encoder, which enables src and mt to attend to each other.[2] We expect that enabling the interactions between them can help to strengthen the ability of the model to find which words in src is untranslated and which words in mt is correct. Note that interactive representation learning is used both in Predictor and Encoder. In the following, we will describe how to predict which words in mt should be copied based on these learned representations.

3.2 Predicting Words to be Copied

Given mt and pe, we can label each word in mt as 0 or 1. We use 1 to denote that the word is to be copied (e.g. "Ich" and "einen" in Figure 1) and 0 not to be copied (e.g. "esse" and "Hamburger" in Figure 1). It is possible to use the Longest Common Sequence (LCS) (Wagner and Fischer, 1974) algorithm to obtain common sequences between mt and pe. If the word in mt also appears in the common sequences, it will be labeled 1; otherwise, it will be labeled 0. We denote these labels as $l_1 \dots l_K$.

We propose a Predictor module to predict words to be copied. As discrete labels are non-differentiable during training, the Predictor module outputs **copying scores** instead for the target words in mt:

$$\mathbf{s} = \text{sigmoid}([\mathbf{H}^{\text{pred}}_{I+1}; \cdots; \mathbf{H}^{\text{pred}}_{I+K}]\mathbf{W}_s), \quad (13)$$

where $\mathbf{s} \in \mathbb{R}^{K\times 1}$ is a vector of copying scores corresponding to the K words in mt, $\mathbf{H}^{\text{pred}} \in \mathbb{R}^{(I+K)\times d}$ is the representation of src and mt:

$$\mathbf{H}^{\text{pred}} = \text{Predictor}([\mathbf{X}; \tilde{\mathbf{Y}}]; \boldsymbol{\theta}), \quad (14)$$

and $\mathbf{W}_s \in \mathbb{R}^{d\times 1}$ is a weight matrix. Only the representation of mt (i.e., $[\mathbf{H}^{\text{pred}}_{I+1}; \cdots; \mathbf{H}^{\text{pred}}_{I+K}]$) is used for calculating copying scores. [3]

As shown in Figure 2(b), copying scores can be incorporated into three parts of our model: the

[2] Note that the Predictor and Encoder do not share their weights because we found that sharing leads to degraded performance in experiments.

[3] It is also possible to predict which source words in src should be used to generate non-copied words in pe (e.g., "Kuchen" in Figure 1). This can be done by generating explicit labels using bilingual word alignment. We leave this for future work.

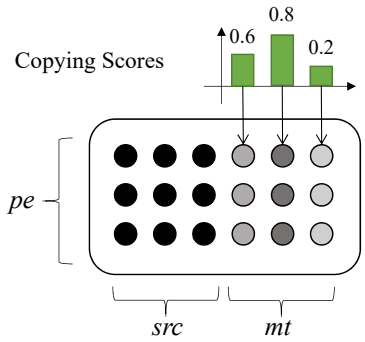

Figure 4: Copying scores as scaling masks. The copying scores are used to modify the attention layers of the Encoder, Decoder, and CopyNet.

Encoder, the Decoder, and the CopyNet. Inspired by Yang et al. (2018)'s strategy to integrate localness to self-attention, we propose to incorporate copying scores into our model by modifying attention weights involved in the aforementioned three modules.

The original scaled dot-product attention (Vaswani et al., 2017) is defined as

$$\text{energy} = \frac{\mathbf{q}\mathbf{K}^\top}{\sqrt{d}}, \tag{15}$$

$$\text{Att}(\mathbf{q}, \mathbf{K}) = \text{softmax}(\text{energy}), \tag{16}$$

where $\mathbf{q} \in \mathbb{R}^{1\times d}$ is the query vector, $\mathbf{K} \in \mathbb{R}^{(I+K)\times d}$ is the key matrix, and $\text{energy} \in \mathbb{R}^{1\times(I+K)}$ is the "energy" vector.

As shown in Figure 4, the copying scores can be used to form a scaling mask on the attention sub-layer:[4]

$$\text{Att}(\mathbf{q}, \mathbf{K}) = \text{softmax}\left(\text{energy} \odot [\mathbf{m}; \mathbf{s}]^\top\right), \tag{17}$$

where $\mathbf{m} = \{1.0\}^I$ is a masking vector corresponding to src and $\mathbf{s} \in \mathbb{R}^{K\times 1}$ is the vector of copying scores calculated by Eq. (13) corresponding to mt. Note that copying scores are used to only change the attention weights related to mt while the src part is unchanged.

3.3 Training

The training objective of our approach $L_{\text{all}}(\boldsymbol{\theta})$ consists of three parts:

$$L_{\text{all}}(\boldsymbol{\theta}) = (1-\alpha) \times (L_{\text{ape}}(\boldsymbol{\theta}) + \lambda \times L_{\text{copy}}(\boldsymbol{\theta})) + \alpha \times L_{\text{pred}}(\boldsymbol{\theta}), \tag{18}$$

where α and λ are hyper-parameters.

The first part is the original log-likelihood loss function of APE: [5]

$$L_{\text{ape}}(\boldsymbol{\theta}) = -\log P(\mathbf{y}|\mathbf{x}, \tilde{\mathbf{y}}; \boldsymbol{\theta}). \tag{19}$$

The second part is related to the CopyNet:

$$L_{\text{copy}}(\boldsymbol{\theta}) = \frac{1}{K}\sum_{k=1}^{K}(l_k - c_k)^2, \tag{20}$$

where l_k is the ground-truth label (see Section 3.2) and c_k is a quantity that measures how likely the k-th word in mt to be copied by CopyNet:

$$c_k = \sum_{j=1}^{J} \gamma_j \times P^{\text{copy}}(\tilde{y}_k). \tag{21}$$

Note that $\gamma \times P^{\text{copy}}(y)$ is the term related to copying the target word y in Eq. (6).

The third part is a cross-entropy loss related to the Predictor:

$$L_{\text{pred}}(\boldsymbol{\theta}) = -\sum_{k=1}^{K}\Big[l_k \log(s_k) + (1-l_k)\log(1-s_k)\Big], \tag{22}$$

where s_k is the copying score of the k-th word $\tilde{y}_k$ in mt.

Finally, we use an optimizer to find the model parameters that minimize the overall loss function:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\text{argmin}}\left\{L_{\text{all}}(\boldsymbol{\theta})\right\}. \tag{23}$$

4 Experiments

4.1 Setup

Datasets

We evaluated our approach on the WMT APE datasets, which often distinguish between two tasks: phrase-based statistical machine translation (i.e., PBSMT) and neural machine translation (i.e., NMT). All these APE datasets consist of English-German triplets containing source text (src), the translations (mt) from a "black-box" MT system and the corresponding human-post-edits (pe). The statistics for the WMT APE datasets are shown in Table 2. In addition to the official dataset, the organizers also recommend using additional datasets

[4] Actually, we let "energy" vector minus its minimum value to keep it non-negative.

[5] For simplicity, we use sentence-level loss functions here. In practice, we use corpus-level loss functions.

Category	Dataset	# Sent.
PBSMT	training set	23,000
	dev2016	1,000
	test2016	2,000
	test2017	2,000
NMT	training set	13,442
	dev2018	1,000
Additional	artificial-small	526,368
	artificial-big	4,391,180
	eSCAPE-PBSMT	7,258,533
	eSCAPE-NMT	7,258,533

Table 2: Statistics of the English-German datasets in the WMT APE task. Note that the NMT official data only contains training and development sets.

(Junczys-Dowmunt and Grundkiewicz, 2016; Negri et al., 2014).

We used the WMT official dataset for the PBSMT task and the NMT task separately. The artificial training data (Junczys-Dowmunt and Grundkiewicz, 2016) was also used for both tasks. More precisely, we used the concatenation of the official training data and the artificial-small data to learn a truecasing model (Koehn et al., 2007) and obtain sub-word units using byte-pair encoding (BPE) (Sennrich et al., 2015) with 32k merges. Then, we applied truecasing and BPE to all datasets. We oversampled the official training data 20 times and concatenated them with both artificial-small and artificial-big datasets (Junczys-Dowmunt and Grundkiewicz, 2018). Finally, we obtained a dataset containing nearly 5M triplets for both tasks. To test our approach on a larger PBSMT dataset, we used the eSCAPE synthetic dataset (Negri et al., 2014), which contains 7.2M sentences. By including the eSCAPE dataset, the training set is enlarged to nearly 12M sentences.

Hyper-Parameter Settings

For the original Transformer model, CopyNet and our approach, the hidden size was set to 512 and the filter size was set to 2,048. The number of individual attention heads was set to 8 for multi-head attention. We set $N = N_e = N_d = 6$, $N_p = 3$ and we tied all three src, mt, pe embeddings for saving memory. The embeddings and softmax weights were also tied. In training, we used Adam (Kingma and Ba, 2014) for optimization. Each mini-batch contains approximately 25K tokens. We used the learning rate decay policy described by (Vaswani et al., 2017). In decoding, the beam size was set to 4. We used the length penalty (Wu et al., 2016) and set the hyper-parameter to 1.0. The other hyper-parameter settings were the same as the Transformer model (Vaswani et al., 2017). We implemented our approach on top of the open-source toolkit THUMT (Zhang et al., 2017).[6]

α	λ	TER↓	BLEU↑
0.1	1.0	18.83	72.59
0.5	1.0	18.45	72.83
0.9	0.1	18.89	72.27
0.9	0.5	18.47	72.83
0.9	1.0	**18.38**	**72.99**

Table 3: Effect of α and λ. The TER and BLEU scores are calculated on the WMT 2016 APE official development set.

Evaluation Metrics

We used the same evaluation metrics as the official WMT APE task (Chatterjee et al., 2018): case-sensitive BLEU and TER. BLEU is computed by *multi-bleu.perl* (Koehn et al., 2007). TER is calculated using TERcom.[7]

Baselines

We compared our approach with the following seven baselines:

1. ORIGINAL: the original mt without any post-editing.
2. COPYNET (Gu et al., 2016; Zhao et al., 2019): the multi-source Transformer equipped with CopyNet (see Figure 2(a)).
3. NPI-APE (Vu and Haffari, 2018): a neural programmer-interpreter approach.
4. MS_UEDIN (Junczys-Dowmunt and Grundkiewicz, 2018): a multi-source Transformer-based APE system that shares the encoders of src and mt. It is the champion of the WMT 2018 APE shared task.
5. USAAR_DFKI (Pal et al., 2018): a multi-source Transformer-based APE system with a joint encoder that attends over a combination of two encoded sequences. It is a participant of the WMT 2018 APE shared task.

[6] https://github.com/THUNLP-MT/THUMT
[7] http://www.cs.umd.edu/snover/tercom/

System	TEST16		TEST17		TEST16+17	
	TER↓	BLEU↑	TER↓	BLEU↑	TER↓	BLEU↑
(1) ORIGINAL	24.76	62.11	24.48	62.49	24.62	62.30
WMT 2017 official + artificial (5M)						
(2) NPI-APE	22.07	66.67	22.58	65.52	–	–
(3) POSTECH	19.14	70.98	19.26	70.50	–	–
(4) FBK	18.79	71.48	19.54	70.09	–	–
(5) MS_UEDIN$_{\text{ensemble}}$	18.86	71.04	19.03	70.46	–	–
(6) COPYNET	18.91	71.64	19.47	70.75	19.19	71.19
(7) Ours	18.39	72.50	18.81	71.35	18.60	71.92
(8) Ours$_{\text{ensemble}}$	**17.77**	**73.19**	**18.41**	**72.09**	**18.09**	**72.62**
WMT 2017 official + artificial + eSCAPE (12M)						
(9) USAAR_DFKI	–	68.52	–	68.91	–	–
(10) MS_UEDIN$_{\text{ensemble}}$	17.34	73.43	17.47	72.84	–	–
(11) COPYNET	18.06	72.77	18.29	71.89	18.18	72.37
(12) Ours	17.45	73.51	17.77	72.98	17.61	73.24
(13) Ours$_{\text{ensemble}}$	**17.06**	**74.00**	**17.37**	**73.26**	**17.22**	**73.62**

Table 4: Results on the English-German PBSMT sub-task. "TEST16+17" is the concatenation of "TEST16" and "TEST17". MS_UEDIN$_{\text{ensemble}}$ and Ours$_{\text{ensemble}}$ used ensembles of four models.

6. POSTECH (Shin and Lee, 2018): a multi-source Transformer-based APE system with two encoders. It is a participant of the WMT 2018 APE shared task.

7. FBK (Tebbifakhr et al., 2018): a multi-source Transformer-based APE system with two encoders. It is a participant of the WMT 2018 APE shared task.

We implemented COPYNET also on top of THUMT (Zhang et al., 2017). The results of all other baselines were taken from the corresponding original papers.

4.2 Effect of Hyper-parameters

We first investigated the effect of the hyper-parameters α and λ in Eq. (18). As shown in Table 3, using $\alpha = 0.9$ and $\lambda = 1.0$ achieves the best performance in terms of TER and BLEU on the WMT 2016 development set, suggesting that both the Predictor and CopyNet play important roles in our approach. Therefore, we set $\alpha = 0.9$ and $\lambda = 1.0$ in the following experiments.

4.3 Main Results

Results on the PBSMT Sub-task

Table 4 shows the results of the PBSMT sub-task. We used the development set of the WMT 2016 APE PBSMT sub-task for model selection for our approach.

Under the small-data training condition (i.e., 5M), our single model (i.e., System 7) outperforms all single-model baselines on all test sets. The superiority over COPYNET (i.e., System 6) suggests that interactive representation learning and incorporating copying scores are effective in improving APE. Our approach that uses the ensemble of four models (i.e., System 8) also improves over the best published result (i.e., System 5).

Under the large-data training condition (i.e., 12M), we find that our single (i.e., System 12) and ensemble (i.e., System 13) models still outperform the best single (i.e., System 11) and ensemble (i.e., System 10) models of baselines.

System	TER↓	BLEU↑
ORIGINAL	15.08	76.76
POSTECH	14.94	77.26
COPYNET	15.12	77.05
Ours	**14.88**	**77.40**

Table 5: Experiments on the WMT 2018 English-German NMT sub-task. The TER and BLEU scores are calculated on the WMT 2018 APE NMT sub-task development set.

ID	Module				DEV16		TEST16+17	
	Interactive	Predictor	CopyNet	Joint Training	TER↓	BLEU↑	TER↓	BLEU↑
1	√	×	×	×	18.74	72.21	19.11	71.03
2	×	×	√	×	19.33	71.86	19.19	71.19
3	×	√	√	√	19.11	72.03	19.07	71.30
4	√	×	√	×	18.62	72.42	18.91	71.53
5	√	√	×	×	18.53	72.53	18.85	71.54
6	√	√	√	×	18.44	72.96	18.77	71.75
7	√	√	√	√	**18.38**	**72.99**	**18.60**	**71.92**

Table 6: Ablation study. "Interactive" denotes interactive representation learning (Section 3.1), "Predictor" denotes detecting correct words (Section 3.2), "CopyNet" denotes the CopyNet used in our approach (Section 2.2), and "Joint Training" denotes the combination of three loss functions (Section 3.3).

Results on the NMT Sub-task

Table 5 shows the results on the NMT sub-task. Besides ORIGINAL and COPYNET, we also compared our approach with POSTECH (Shin and Lee, 2018), which is the only participating system that released the results on the development set of the WMT 2018 NMT sub-task.[8] We find that our approach also outperforms all baselines.

4.4 Ablation Study

Table 6 shows the results of ablation study. It is clear that interactive representation learning plays a critical role since removing it impairs post-editing performance (line 3). As shown in line 4, the Predictor is also an essential part of our approach. CopyNet and joint training are also shown to be beneficial for improving APE (lines 5 and 6) but seem to have relatively smaller contributions than interactive representation learning and predicting words to be copied.

4.5 Results on Prediction Accuracy

The Predictor is an important module in our approach as it predicts which words in mt should be copied. Given the ground-truth labels, it is easy to calculate *prediction accuracy* by casting the prediction as a binary classification problem. We find that the Predictor achieves a prediction accuracy of 85.09% on the development set.

4.6 Comparison of Copying Accuracies

A target word y in a machine translation $\tilde{\mathbf{y}}$ is called to be *correctly* copied to an automatic edited translation $\hat{\mathbf{y}}$ if and only if the positions where y occurs in $\hat{\mathbf{y}}$ and $\mathbf{y}$ (i.e., the ground-truth edited translation) are identical. Therefore, it is easy to define copying accuracy to measure how well the copying mechanism works.

System	Accuracy
MS_UEDIN	64.63
COPYNET	64.72
Ours	**65.61**

Table 7: Comparison of copying accuracies.

Table 7 shows the comparison of copying accuracies between MS_UEDIN, COPYNET, and our approach. We find that our approach outperforms the two baselines. However, the copying accuracy of our approach is almost 20% lower than the prediction accuracy (i.e., 65.61% vs. 85.09%), indicating that it is much more challenging to place the copied words in correct positions.

4.7 Visualization

Figure 5 gives an example that illustrates how copying scores influence attention. It shows the heatmap of the Enc-Dec-Attention, which averages over 8 different heads. Only the attention weights beween pe and mt are included. We take the second layer of Enc-Dec-Attention for example. The x-axis represents mt and the y-axis represents pe. The darker the color, the higher the copying scores.

We find that words "mit" and "dem" are identified by the Predictor. Accordingly, the attention weights corresponding to these words are decreased since the columns corresponding to these words have lighter color. As a result, all words in mt other than "mit" and "dem" are copied to pe.

[8] As POSTECH only used small data for training, the eSCAPE datasets were not used in this experiment for a fair comparison.

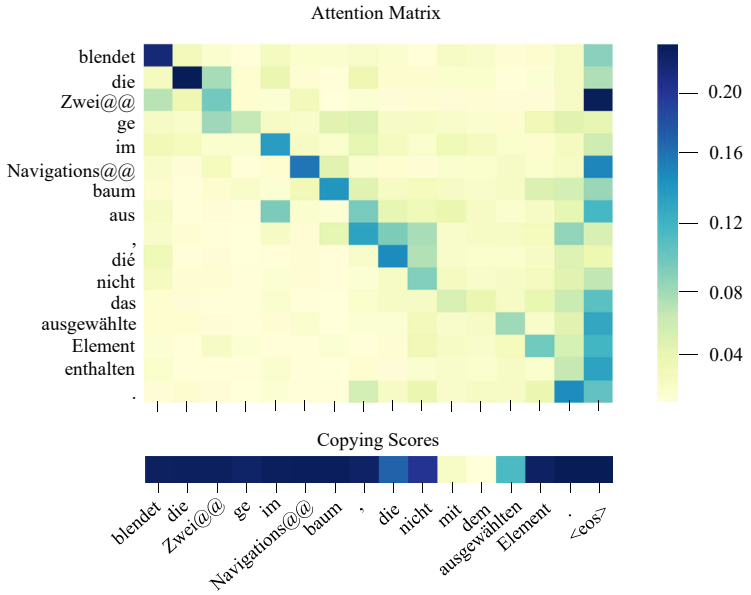

Figure 5: Example of the heatmap of attention and copying scores. The x-axis is mt and the y-axis is pe. The Predictor successfully detects the incorrect word "mit" and "dem" and gives low copying scores to these words and then decrease the importance of them in attention. Other words in mt are correctly copied to pe.

5 Related Work

5.1 Multi-source Sequence-to-Sequence Learning

Recently, multi-source Transformer-based APE systems (Junczys-Dowmunt and Grundkiewicz, 2018; Pal et al., 2018; Tebbifakhr et al., 2018; Shin and Lee, 2018) have achieved the state-of-the-art results on the datasets of the WMT APE shared task. Our work differs from prior studies by enabling interactions between src and mt and explicitly detecting words to be copied.

5.2 The Copying Mechanism

Zhao et al. (2019) apply CopyNet (Gu et al., 2016) to grammar error correction. Their approach generates labels similar to ours, but only uses them to perform mutli-task learning. Libovický et al. (2016) first introduce CopyNet to APE but do not provide a detailed description of their method and experimental results. We show that interactive representation learning and explicit indication of words are important for modeling copying in APE.

5.3 Interactive Representation Learning

Niehues et al. (2016) simply concatenate the output of the PBSMT system and the source sentence to serve as the input the NMT system without enabling multi-layer interactive learning. Lample and Conneau (2019) used cross-lingual setting to enable cross-lingual language model pre-training. We propose to let src and mt fully interact with each other to make it easier to decide which words in mt should be copied.

6 Conclusion

We have presented a new method for modeling the copying mechanism for automatic post-editing. By making the source sentence and machine translation attend to each other, representations learned in such an interactive way help to identify whether a target word should be copied or be re-generated. We also find that explicitly predicting words to be copied is beneficial for improving the performance of post-editing. Experiments show that our approach achieves new state-of-the-art results on the WMT 2016 & 2017 APE PBSMT sub-tasks.

Acknowledgments

We thank all anonymous reviewers for their valuable comments. This work is supported by the National Key R&D Program of China (No. 2017YFB0202204), National Natural Science Foundation of China (No. 61761166008, No. 61432013), Beijing Advanced Innovation Center for Language Resources (No. TYR17002), and the NExT++ project supported by the National Research Foundation, Prime Ministers Office, Singapore under its IRC@Singapore Funding Initiative. This research is also supported by Sogou Inc.

References

Hanna Béchara, Yanjun Ma, and Josef van Genabith. 2011. Statistical post-editing for a statistical mt system. In *MT Summit*.

Rajen Chatterjee, M. Amin Farajian, Matteo Negri, Marco Turchi, Ankit Srivastava, and Santanu Pal. 2017. Multi-source neural automatic post-editing: Fbk's participation in the wmt 2017 ape shared task. In *Proceedings of WMT*.

Rajen Chatterjee, Matteo Negri, Raphaël Rubino, and Marco Turchi. 2018. Findings of the wmt 2018 shared task on automatic post-editing. In *Proceedings of WMT*.

Rajen Chatterjee, Marion Weller, Matteo Negri, and Marco Turchi. 2015. Exploring the planet of the apes: a comparative study of state-of-the-art methods for mt automatic post-editing. In *Proceedings of ACL*.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O. K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of ACL*.

Tianyu He, Xu Tan, Yingce Xia, Di He, Tao Qin, Zhibo Chen, and T. M. Liu. 2018. Layer-wise coordination between encoder and decoder for neural machine translation. In *Proceedings of NeurIPS*.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Log-linear combinations of monolingual and bilingual neural machine translation models for automatic post-editing. In *Proceedings of WMT*.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2017. An exploration of neural sequence-to-sequence architectures for automatic post-editing. In *Proceedings of IJCNLP*.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2018. Ms-uedin submission to the wmt2018 ape shared task: Dual-source transformer for automatic post-editing. In *Proceedings of WMT*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proceedings of ICLR*.

Kevin Knight and Ishwar Chander. 1994. Automated postediting of documents. In *Proceedings of AAAI*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Jindrich Libovický, Jindrich Helcl, Marek Tlustý, Pavel Pecina, and Ondrej Bojar. 2016. Cuni system for wmt16 automatic post-editing and multimodal translation tasks. In *Proceedings of WMT*.

Kathleen McKeown, Kristen Parton, Nizar Habash, Gonzalo Iglesias, and Adrià de Gispert. 2012. Can automatic post-editing make mt more meaningful? In *Proceedings of EAMT*.

Matteo Negri, Marco Turchi, Rajen Chatterjee, and Nicola Bertoldi. 2014. escape: a large-scale synthetic corpus for automatic post-editing. In *Proceedings of LREC*.

Jan Niehues, Eunah Cho, Thanh-Le Ha, and Alexander H. Waibel. 2016. Pre-translation for neural machine translation. In *Proceedings of COLING*.

Santanu Pal, Nico Herbig, Antonio Krüger, and Josef van Genabith. 2018. A transformer-based multi-source automatic post-editing system. In *Proceedings of WMT*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of ACL*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. In *Proceedings of ACL*.

Jaehun Shin and Jong-Hyeok Lee. 2018. Multi-encoder transformer network for automatic post-editing. In *Proceedings of WMT*.

Michel Simard, Cyril Goutte, and Pierre Isabelle. 2007. Statistical phrase-based post-editing. In *Proceedings of HLT-NAACL*.

Amirhossein Tebbifakhr, Ruchit Agrawal, Matteo Negri, and Marco Turchi. 2018. Multi-source transformer for automatic post-editing. In *Proceedings of CLiC-it*.

Inigo Jauregi Unanue, Ehsan Zare Borzeshi, and Massimo Piccardi. 2018. A shared attention mechanism for interpretation of neural automatic post-editing systems. In *Proceedings of NMT@ACL*.

Dusan Varis and Ondrej Bojar. 2017. Cuni system for wmt17 automatic post-editing task. In *Proceedings of WMT*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NeurIPS*.

Thuy-Trang Vu and Gholamreza Haffari. 2018. Automatic post-editing of machine translation: A neural programmer-interpreter approach. In *Proceedings of EMNLP*.

Robert A. Wagner and Michael J. Fischer. 1974. The string-to-string correction problem. *Journal of the ACM*, 21:168–173.

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 preprint arXiv:1609.08144*.

Baosong Yang, Zhaopeng Tu, Derek F. Wong, Fandong Meng, Lidia S. Chao, and Tong Zhang. 2018. Modeling localness for self-attention networks. In *Proceedings of EMNLP*.

Jiacheng Zhang, Yanzhuo Ding, Shiqi Shen, Yong Cheng, Maosong Sun, Huan-Bo Luan, and Yang Liu. 2017. Thumt: An open source toolkit for neural machine translation. *arXiv preprint arXiv:1706.06415*.

Wei Ke Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data. In *Proceedings of NAACL-HLT*.

Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. In *Proceedings of HLT-NAACL*.