Towards Semi-Supervised Learning of Automatic Post-Editing: Data-Synthesis by Infilling Mask with Erroneous Tokens

Anonymous ACL submission

Abstract

Semi-supervised learning that leverages synthetic training data has been widely adopted in the field of Automatic post-editing (APE) to overcome the lack of human-annotated training data. In that context, data-synthesis methods to create high-quality synthetic data have also received much attention. Considering that APE takes machine-translation outputs containing translation errors as input, we propose a noising-based data-synthesis method that uses a mask language model to create noisy texts through substituting masked tokens with erroneous tokens, yet following the error-quantity statistics appearing in genuine APE data. In addition, we propose corpus interleaving, which is to combine two separate synthetic data by taking only advantageous samples, to further enhance the quality of the synthetic data created with our noising method. Experimental results reveal that using the synthetic data created with our approach results in significant improvements of APE performance upon using other synthetic data created with different existing data-synthesis methods. Our research findings are available at (*the link will be located here after anonymous period*).

1 Introduction

Automatic post-editing (APE) (Chatterjee et al., 2015, 2020) is a field of study that seeks to correct errors in machine translation (MT) outputs to provide high-quality translations. Generally, APE is regarded as a multi-source sequence-to-sequence problem (Figure 1) as it simultaneously takes a source text (src) and its MT output (mt) to produce a post-edited text (pe). Due to this nature of APE, supervised learning of APE models requires training data that have the form $\langle src, mt, pe \rangle$, where pe is assumed to be a product of **minimal correction** on mt, which is an underlying assumption of APE (Bojar et al., 2015).

However, the quantity of currently available human-made (gold-standard) APE triplets is heavily insufficient to train a sequence-to-sequence APE model robustly. Thus, 'semi-supervised learning' leveraging synthetic data in addition to gold-standard data for model training has been adopted widely; accordingly, a fair number of studies (Junczys-Dowmunt and Grundkiewicz, 2016; Negri et al., 2018; Lee et al., 2020, 2021) have attempted to explore data-synthesis methods to obtain high-quality synthetic training data.

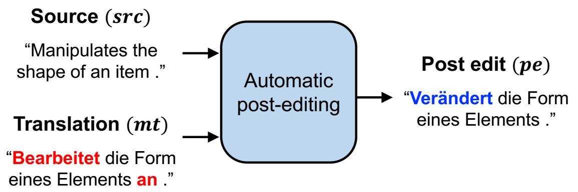

Figure 1: An overview the APE process. Erroneous words and post-edited words are highlighted in **bold**.

In particular, there have been several studies (Negri et al., 2018; Lee et al., 2020, 2021) that utilize parallel corpora, which comprise pairs of a source text (src) and its reference text (ref): $\langle src, ref \rangle$, which is also called a 'bitext'. Such methods have a common feature that bitexts' ref are used to server as pe in synthetic APE triplets, yet each method differs in its method to create mt data to construct synthetic APE triplets.

One such method (Negri et al., 2018) is to create mt by simply translating src with an MT system. Although being plausible in that this method creates synthetic triplets in the same manner as gold-standard triplets are created, this method has the limitation that ref is not guaranteed to be a product of minimal correction on mt. Thus, mt created with this method is likely to contain a much larger amount of error than gold-standard mt.

Another approach (Lee et al., 2020) is to create mt by randomly injecting noise into ref with regards to the actual quantity of errors appearing in gold-standard mt. Even though this method may successfully reflect the distribution of the error quantities in gold-standard mt, the resulting synthetic mt still could be significantly distant from an MT output qualitatively because the injected noise is not originated by an MT system.

To amalgamate the advantages of those two approaches, we propose a data-synthesis method using parallel corpora to obtain synthetic mt that contain errors that are likely to appear in an MT output while controlling the quantity of errors in the synthetic mt by following the error-quantity distribution of gold-standard mt.

Our approach, **MLM noising**, is inspired by the “masked language model” (MLM; Devlin et al. 2019), which predicts a proper substitute for each masked token `<MASK>` in an input sequence. The basic idea of this approach is to let an MLM substitute each `<MASK>` with an ‘erroneous’ token that is likely to appear in an MT output while restricting the number of `<MASK>` to the number of errors contained in gold-standard mt.

Also, to further improve the quality of synthetic data created with our approach, we propose **corpus interleaving**, which is to incorporate our synthetic data into an existing synthetic data (both are made of the same bitexts) by adopting our new triplet if it is considered better than the original triplet.

2 Background

2.1 Problem Statement

APE has been addressed in the frame of the multi-source sequence-to-sequence problem $(src, mt) \to pe$, which has two inputs: src, which provides contextual information that helps identifying translation errors, and mt, which is the object of correction. Formally, let $D = \{\langle \mathbf{x}, \widetilde{\mathbf{y}}, \mathbf{y}\rangle_{i=1}^{n}\}$ denotes a set of n APE triplets (whether they are gold-standard or synthetic triplets), where $\mathbf{x} = (x_1 \ldots x_{T_x})$, $\widetilde{\mathbf{y}} = (\widetilde{y}_1 \ldots \widetilde{y}_{T_{\widetilde{y}}})$, and $\mathbf{y} = (y_1 \ldots x_{T_y})$ indicate src, mt, and pe, respectively; An APE model learns to predict pe by following the conditional probability,

$$P(\mathbf{y}) = \prod_{i=1}^{T_y} P(y_i|\mathbf{x}, \widetilde{\mathbf{y}}, y_{<i}; \theta), \tag{1}$$

where θ is a set of model parameters.

2.2 Existing Data-Synthesis Methods

In this section, we outline several existing methods to create synthetic triplets by using parallel corpora.

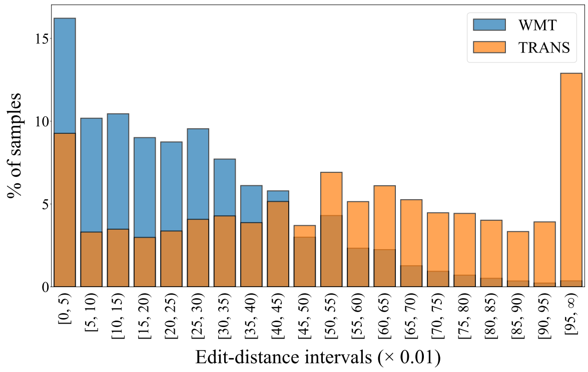

Figure 2: The mt–pe edit-distance distribution of gold-standard data (released by WMT (Chatterjee et al., 2018)) vs. the mt–ref edit-distance distribution of TRANS synthetic data.

Translation Approach (TRANS) An early study (Negri et al., 2018) of creating synthetic APE triplets by using parallel corpora, this method uses an MT system to translate src of bitexts to the target language and obtain mt. The result is a set of synthetic triplets in the form of $\langle src, mt, ref\rangle$.

The strength of this method is that its mt is created by an MT system as in creating gold-standard mt. However, this method has a key limitation that ref is not guaranteed to be a product of minimal correction on the mt. We observe a significant discrepancy in the edit-distance distribution between mt–ref and mt–pe (Figure 2). Thus, this method leaves much room for improvement in the quality and effect of the resulting synthetic triplets in that they are likely to guide an APE models to give unnecessarily excessive correction on mt in the inference stage.

Random Noising Approach (RAND) This method (Lee et al., 2020) creates synthetic mt by applying editing operations (each editing operation corresponds to a certain type of translation errors) to ref. Specifically, this method randomly inserts, deletes, or substitutes tokens in ref with regards to the error-quantity distribution (i.e., the probability of the occurrence of each operation) of gold-standard mt. The result is a set of synthetic triplets in the form of $\langle src, ref_{\text{noise}}, ref\rangle$, where ref_{noise} serves as mt. In contrast to TRANS, the synthetic mt created with this method have the advantage of reflecting the error distribution of gold-standard mt. However, the synthetic triplets could still significantly differ from gold-standard triplets because the noising procedure is unsystematic due to its randomness.

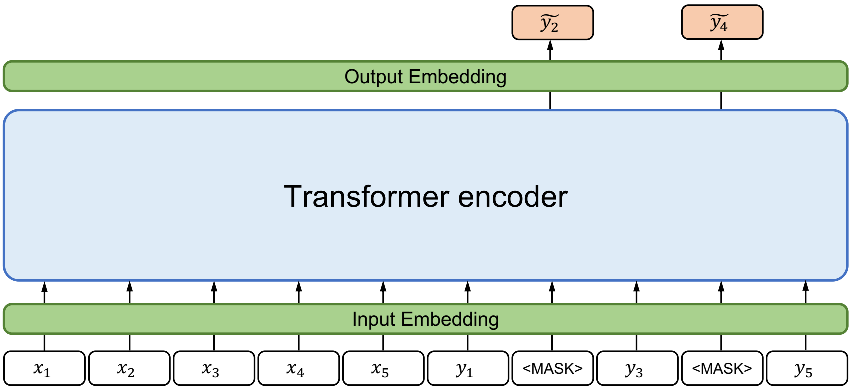

Figure 3: The overall architecture of the MLM noising model. $x_i, \widetilde{y}_j$ and y_k denote a token in src, mt, and ref, respectively.

Back-Translation Approach (BT)

A recently proposed method (Lee et al., 2021) is to adapt back-translation (Sennrich et al., 2016) to APE to improve TRANS so that the resulting synthetic mt ("$\widetilde{mt}$") can reduce the excessive mt–ref edit distance of the existing TRANS triplets. They suggest two such methods to produce $\widetilde{mt}$:

- forward generation (BT-FG): creating $\widetilde{mt}$ by partially correcting mt by using APE.
- backward generation (BT-BG): reversing APE in the form of $(src, pe) \rightarrow mt$ and creating $\widetilde{mt}$ by partially noising ref with probable translation errors.

Compared to TRANS, although this method succeeds in reducing the edit distance in the overall distribution, the resulting BT synthetic data fails to improve the APE performance of trained models when it is used solely.

3 Approach

Considering that each one of TRANS and RAND has a limitation that the other overcomes, we aim to construct a new set of synthetic triplets that incorporates the advantages of both the methods and solves their problems. Our **MLM noising** approach is to create new synthetic mt, the errors of which are likely to appear in gold-standard mt not only qualitatively but also quantitatively: the error quantity and types are determined with regards to the distribution of gold-standard data.

In addition, we also explain the **corpus interleaving** approach, which is to combine the products of TRANS and our approach, to further improve the quality of our new synthetic data.

3.1 MLM Noising

MLM Architecture

MLM is a denoising method that substitutes <MASK> in the input sequence with appropriate tokens that are likely to be the original tokens, which is equivalent to 'text-infilling', by using a Transformer (Vaswani et al., 2017) encoder.

The difference between the original work and our approach is that our MLM noising model is fed with ref_{mask}, which derives from ref through replacement of certain tokens by <MASK>, and performs text-infilling to predict probable mt tokens to be in the positions of <MASK>; for this purpose, we substitute <MASK> in ref_{mask} with the desired mt tokens by using MT outputs and obtain ref_{noise}. We expect that this text-infilling will allow the MLM model to learn injecting mt errors into ref.

In sum, the training data of our MLM model have the form of $\langle src, ref_{\text{mask}}, ref_{\text{noise}} \rangle$, of which src (yet without masking) is also fed into the model to provide contextual information (Figure 3), following the basic nature of APE (§1).

Training: MLM Training-Data Construction

To arrange $\langle src, ref_{\text{mask}}, ref_{\text{noise}} \rangle$ for the model training, we need to determine (1) what output tokens correspond to <MASK>, (2) where to place <MASK>, and (3) how many <MASK> should be.

To this end, we utilize word-to-word alignments[1] between mt and ref, which are the byproducts of the mt–ref edit-distance calculation process. By examining the alignment of each ref token, we can identify whether it is (1) a correct translation, (2) a substitution error, (3) an insertion error, or

[1] We obtain these alignments by running tercom software: https://github.com/jhclark/tercom.git

Algorithm 1: Construction of MLM Training Data

Input: $D := \{\langle \mathbf{x}, \widetilde{\mathbf{y}}, \mathbf{y} \rangle_{i=1}^{n}\}$
$Dist :=$ edit-distance distribution of gold data
$A := \{a_{i=1}^{n}\}$ `// mt-ref alignments`
Output: $D_{\text{MLM}} = \{\langle \mathbf{x}, \mathbf{y}^{\text{mask}}, \mathbf{y}^{\text{noise}} \rangle_{i=1}^{n}\}$
$D_{\text{MLM}} \leftarrow \{\}$
for $i \in [1, n]$ **do**
  $\mathbf{y}_i^{\text{mask}} \leftarrow \varnothing$
  $\mathbf{y}_i^{\text{noise}} \leftarrow \varnothing$
  $e_i \sim Dist$
  $\widetilde{e}_i \leftarrow$ edit_distance$(\widetilde{\mathbf{y}}_i, \mathbf{y}_i)$
  $a_i^{err} \leftarrow \{(y, \widetilde{y}) \mid \forall a_i, y \neq \widetilde{y}\}$
  if $\widetilde{e}_i > e_i$ **then**
    num_mask $\leftarrow \lceil (\text{Len}(\mathbf{y}_i) * e_i) \rceil$
    $\widetilde{a}_i \leftarrow$ random_choice$(a_i^{err}$, num_mask$)$
  else
    $\widetilde{a}_i \leftarrow a_i^{err}$
  for *each* $\langle y, \widetilde{y} \rangle \in a_i$ **do**
    if $\langle y, \widetilde{y} \rangle \in \widetilde{a}_i$ **then**
      $\mathbf{y}_i^{\text{mask}} \leftarrow \text{Append}(\mathbf{y}_i^{\text{mask}}, \langle \texttt{MASK} \rangle)$
      $\mathbf{y}_i^{\text{noise}} \leftarrow \text{Append}(\mathbf{y}_i^{\text{noise}}, \widetilde{y})$
    else
      $\mathbf{y}_i^{\text{mask}} \leftarrow \text{Append}(\mathbf{y}_i^{\text{mask}}, y)$
      $\mathbf{y}_i^{\text{noise}} \leftarrow \text{Append}(\mathbf{y}_i^{\text{noise}}, y)$
  $D_{\text{MLM}} \leftarrow D_{\text{MLM}} \cup \{\langle \mathbf{x}_i, \mathbf{y}_i^{\text{mask}}, \mathbf{y}_i^{\text{noise}} \rangle\}$

Algorithm 2: Construction of MLM Inference Data

Input: $D := \{\langle \mathbf{x}, \mathbf{y} \rangle_{i=1}^{n}\}$
$\mu := \{\mu_{\text{keep}}, \mu_{\text{sub}}, \mu_{\text{ins}}, \mu_{\text{del}}\}$ s.t. $\sum \mu = 1$
Output: $D_{\text{MLM}} = \{\langle \mathbf{x}, \mathbf{y}^{\text{mask}} \rangle_{i=1}^{n}\}$
$D_{\text{MLM}} \leftarrow \{\}$
for $i \in [1, n]$ **do**
  $\mathbf{y}_i^{\text{mask}} \leftarrow \varnothing$
  for *each* $y_j \in \mathbf{y}_i$ **do**
    op $\sim Categorical(\text{op} \mid \mu)$
    if *op is keep* **then**
      $\mathbf{y}_i^{\text{mask}} \leftarrow \text{Append}(\mathbf{y}_i^{\text{mask}}, y_j)$
    else if *op is substitution* **then**
      $\mathbf{y}_i^{\text{mask}} \leftarrow \text{Append}(\mathbf{y}_i^{\text{mask}}, \langle \texttt{MASK} \rangle)$
    else if *op is insertion* **then**
      $\mathbf{y}_i^{\text{mask}} \leftarrow \text{Append}(\mathbf{y}_i^{\text{mask}}, y_j)$
      $\mathbf{y}_i^{\text{mask}} \leftarrow \text{Append}(\mathbf{y}_i^{\text{mask}}, \langle \texttt{MASK} \rangle)$
    else if *op is deletion* **then**
      continue
  $D_{\text{MLM}} \leftarrow D_{\text{MLM}} \cup \{\langle \mathbf{x}_i, \mathbf{y}_i^{\text{mask}} \rangle\}$

(4) a deletion error. For a given ref sequence, we place `<MASK>` in the positions of the tokens the alignments of which are identified as errors[2] and take their aligned mt tokens as the desired output tokens.

Next, we control the quantity of errors to be in ref_{mask} by following the error-quantity distribution of gold-standard mt. We compare the edit distance of every $\langle mt, ref \rangle$ pair with that of sampled edit distance from the mt–pe edit-distance distribution of gold-standard data (Figure 2); if the edit distance of the given $\langle mt, ref \rangle$ sample is larger than the sampled value, we restrict the number of `<MASK>` to the sampled value. We describe this MLM training-data construction process with Algorithm 1.

Inference: APE Training-Data Construction

Once the training has finished, we then make new synthetic APE triplets by using our trained MLM. We first mask tokens in a given ref sequence with regards to the error-type statistics of gold-standard mt so that the resulting ref_{mask} simulates error patterns appearing in gold-standard mt. We refer to a categorical distribution $\mu = \{\mu_{\text{keep}}, \mu_{\text{sub}}, \mu_{\text{ins}}, \mu_{\text{del}}\}$[3], where each term indicates the probability of each mt token's being a correct translation, a substitution error, an insertion error, and a deletion error, respectively.

After making ref_{mask} in this stochastic manner according to the procedure described in Algorithm 2, we provide our trained MLM model with $\langle src, ref_{\text{mask}} \rangle$ and let it perform text-infilling to predict ref_{noise}. This ref_{noise} is our new synthetic mt; our new synthetic APE triplets are thus $\langle src, ref_{\text{noise}}, ref \rangle$.

3.2 Corpus Interleaving

Although many triplets in TRANS have excessively large mt–ref edit distances as mentioned above (§2.2), the triplets whose mt–ref edit distance are similar to the average mt–pe edit distance can be regarded as a suitable training sample for APE (e.g., the overlapping region in Figure 2); utilizing them together with our new synthetic APE triplets would be helpful for APE models eventually.

Thus, we suggest corpus interleaving, which is to adopt the better triplet between TRANS' and ours, both of which share the same src and ref. In this regard, we apply the 3-sigma rule (Pukelsheim, 1994) to make a choice between mt and ref_{noise}

[2] We ignore alignments indicating deletion errors because all deletion `<MASK>` are mapped onto a single output token. We presume that this mapping will cause deletion-biased prediction. Instead, we simulate deletion errors at the inference time.

[3] We obtain the statistics through mt–pe edit-distance calculation. For the calculation, we used tercom software: `https://github.com/jhclark/tercom.git`

for every bitext $\langle src, ref \rangle$, i.e.,

$$mt = \begin{cases} mt & \text{if } |\text{edit}(mt, ref) - \mu| \leq \lambda\sigma \\ ref_{\text{noise}} & \text{otherwise,} \end{cases} \tag{2}$$

where $\text{edit}(\cdot)$ denotes the edit distance; μ and σ are the mean and standard deviation of the mt–pe edit distances, respectively; $\lambda \in [1, 3]$ is a hyperparameter.

4 Experiments

4.1 Setup

Evaluation Metric

Following the WMT APE shared task (Chatterjee et al., 2018), we adopted TER (Snover et al., 2006) as our primary metric, and BLEU (Koehn et al., 2007) as the secondary metric. We conducted all evaluations case-sensitively.

Datasets

Our benchmark dataset is the dataset released for the WMT'18 APE shared task, which is a set of human-annotated English-to-German (EN–DE) APE triplets (i.e., src is in English while mt and pe are in German), consisting of 23K training data, 1K development data, and three kinds of test data: Test16, Test17, and Test18, each of which contains 2K data.

In our experiments, TRANS refers to the dataset[4] released by Negri et al. (2018), which contains about 7M triplets. We used the TRANS triplets in both the training of our MLM noising model (we randomly extracted 2K held-out data from the whole data to use as development data) and the construction of our new synthetic APE triplets (§3.1). For a fair comparison with the existing methods (§2.2), we used TRANS' bitexts also for RAND (note that BT[5] has been already made from TRANS). All words in the datasets that we used were tokenized into subword by SentencePiece[6].

Model Configuration

We implemented our MLM noising model by modifying the RoBERTa (Liu et al., 2019) implementation released by Huggingface[7] (note that we trained our MLM model from scratch, not using a 'pre-trained' RoBERTa model). We followed most of the default hyperparameter configuration of Huggingface's RoBERTa implementation.

To evaluate the effect of synthetic APE training data, we used OpenNMT-py[8] to implement the "concat-based" APE model (Figure 4) proposed by Shin et al. (2021), a basic Transformer-based APE model containing relatively few parameters yet showing a satisfactory performance. This model follows the 'Transformer-base' (Vaswani et al., 2017) hyperparameter settings, and we adopted the same settings. We report both model's configurations including their hyperparameters in Table 1.

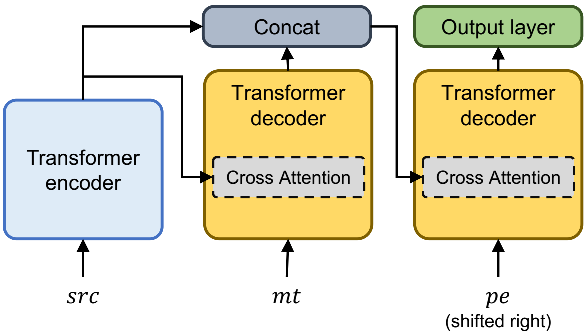

Figure 4: A schema describing the architecture of the concat-based APE model proposed by Shin et al. (2021).

Settings	MLM	APE
Optimizer	Adam ($\epsilon = 10^{-9}, \beta = (0.9, 0.998)$)	
Batch size (# samples)	384	
# layers	12	6
# heads	12	6
Hidden size	768	512
Feed-forward	3,072	2,048
Activation	GeLU	ReLU
Learning rates	2e-4	5e-4
Warmup steps	7,000	6,000
Decay function	linear	inverse sqrt.
Train steps	40K	10K
Train times	2.5 days	12 hours
GPUs	A100 $\times$ 8ea	A5000 $\times$ 4ea
# params	110M	85M

Table 1: The configurations of the MLM noising model and the APE model used in our experiments.

[4] https://ict.fbk.eu/escape/
[5] https://github.com/wonkeelee/APE-backtranslation.git
[6] https://github.com/google/sentencepiece
[7] https://github.com/huggingface/transformers
[8] https://github.com/OpenNMT/OpenNMT-py

Approaches	Test16 TER(↓)	Test16 BLEU(↑)	Test17 TER(↓)	Test17 BLEU(↑)	Test18 TER(↓)	Test18 BLEU(↑)	Test Avg. TER(↓)	Test Avg. BLEU(↑)
TRANS	**16.87**	73.95	**17.30**	**73.08**	17.80	72.41	17.32	**73.15**
RAND	17.23	73.59	17.61	72.69	17.81	72.38	17.55	72.88
BT-FG	17.26	73.56	17.56	72.78	17.89	72.14	17.57	72.82
BT-BG	17.61	73.04	17.60	72.49	18.01	71.89	17.74	72.47
MLM Noising	$16.90^{\blacktriangle\dagger\ddagger}$	$\mathbf{74.03}^{\blacktriangle\dagger\ddagger}$	$17.31^{\blacktriangle\ddagger}$	$72.90^{\ddagger}$	$\mathbf{17.62}^{\ddagger}$	$\mathbf{72.43}^{\ddagger}$	$\mathbf{17.28}^{\blacktriangle\dagger\ddagger}$	$73.12^{\blacktriangle\dagger\ddagger}$

Table 2: A fair comparison of the evaluation results of four APE models, each of which is trained on different synthetic data. $*$, ▲, †, ‡ indicate that our MLM noising approach's result outperforms TRANS, RAND, BT-FG, and BT-BG, respectively, with a statistical significance of $p < 0.05$. The best result in each column is highlighted in **bold**.

	Test Avg. TER(↓)	Test Avg. BLEU(↑)	Sample Ratio MLM	Sample Ratio TRANS
$\lambda = 0$ (MLM Noising)	17.32	73.15	100.0%	0.0%
$\lambda = 1$	17.25	73.26	71.5%	28.5%
$\lambda = 2$	$\mathbf{16.96}^{**}$	$\mathbf{73.75}^{**}$	41.5%	58.5%
$\lambda = 3$	17.03^{**}	73.48^{**}	20.8%	79.2%
$\lambda = \infty$ (TRANS)	17.28	73.12	0.0%	100.0%

Table 3: The effect of corpus interleaving with varying λ. $**$ indicates the improvement is statistically significant compared to both $\lambda = 0$ and $\lambda = \infty$ with $p < 0.01$. The best result in each column is highlighted in **bold**.

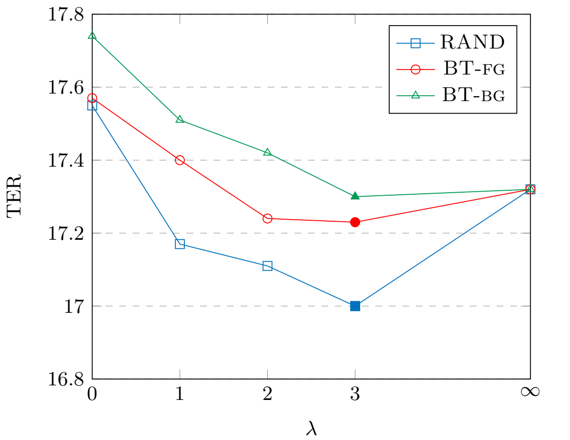

Figure 5: The effects of the corpus interleaving when applied to other existing synthetic APE data. The λ that records the best performance for each data is marked with a color.

4.2 Results

We trained the APE model in the same experimental environment (including the hyperparameter, codebase, and training seed) but with different synthetic data: TRANS, RAND, BT-FG, BT-BG, and MLM noising. We evaluate these five trained models on the test data to compare the effectiveness of each synthetic data (Table 2).

We observe that using the synthetic data created with MLM noising results in significant improvements in APE performance compared to most of the other data-synthesis methods. However, the improvement over TRANS was not statistically significant in our experiments. As aforementioned (§3.2), we surmise that TRANS contained numbers of triplets that are as helpful for APE models as our synthetic triplets.

To verify whether the effectiveness of MLM noising is further enhanced when corpus interleaving is applied, we also trained the APE model on the integration of the TRANS data and our MLM noising triplets by using corpus interleaving. Through experiments, we ascertain that corpus interleaving is helpful in enhancing APE performance and that this enhancing effect is statistically significant (Table 3). Also, we found that taking almost equal parts ($\lambda = 2$) from TRANS and our MLM noising data leads to the best APE performance among other ratios; this finding supports our speculation (§3.2) that each synthetic data has its own advantages.

5 Analysis and Discussion

5.1 Effect of Corpus Interleaving

To examine the effect of corpus interleaving independently, we conducted additional experiments where we interleave the TRANS data with different synthetic training data other than ours: RAND, BT-FG, and BT-BG.

Considering the experimental results (Figure 5), corpus interleaving appears to be effective in creating better training data, regardless of what data-generating method is used as the counterpart of the TRANS data. We found that all the instances of corpus interleaving outperform using the TRANS data solely ($\lambda = \infty$) and using the corresponding synthetic data solely ($\lambda = 0$). We speculate that

src	How to choose the right trainee ?
ref	Wie **wählen Sie den richtigen Praktikanten ?**
mt (TRANS)	How bis choose the right Auszubildender ?
ref_{noise} (MLM noising)	**wählen Sie den richtigen Praktikanten ?**

Table 4: An example of ref_{noise}, a new mt created with our proposed method, showing a better quality than the corresponding mt (TRANS). The corresponding src, mt, and ref compose a triplet in the original TRANS dataset. **Boldface** words in ref are overlapped with the other boldface words either in mt (TRANS) or in ref_{noise}.

src	What happens if I want to leave ?
ref	**Was geschieht , wenn ich** wieder **gehen** will **?**
mt (TRANS)	**Was** passiert **, wenn ich** verlassen wollen ?
ref_{noise} (MLM noising)	**Was** passiert **geschieht , wenn ich** ich zu **gehen** wollen .

Table 5: An example of ref_{noise}, a new mt created with our proposed method, showing a poorer quality than the corresponding mt (TRANS). The corresponding src, mt, and ref compose a triplet in the original TRANS dataset. **Boldface** words in ref are overlapped with the other boldface words either in mt (TRANS) or in ref_{noise}.

this effect proceeds from the common feature of all the experimented synthetic data that they are built upon the TRANS data to further reduce the edit distance between mt and ref.

We also found that corpus interleaving with RAND leads to a bigger improvement in APE performance than the others (except ours). The reason could be that ref_{noise} of the RAND data reflects the error-quantity distribution of gold-standard mt as ours does although their noising procedure is random unlike our MLM noising approach. This finding can also be evidence that reflecting the error-quantity distribution of gold-standard mt is the crux of constructing good synthetic APE triplets.

5.2 Case Analysis on Synthetic Data

We provide two examples of how the relation between mt and ref changes when our proposed method is applied. Whereas the mt sentence in the first example (Table 4) contains only two German words 'bis' (translated as 'until' or 'to') and 'Auszubildender' ('trainee') and all the other words are still English words, the new mt sentence (translated as 'do you choose the right trainee?') only omits one German word 'Wie' ('How').

This first example implies that our proposed method can successfully supply improved APE triplets, mt of which has a similar amount of error as that in gold-standard mt sentences. Furthermore, while the ref sentence cannot be a minimally post-edited sentence for the mt sentence due to the synonymity between 'Auszubildender' and 'Praktikanten' (also 'trainee'), it obviously is for the new mt sentence.

The second example (Table 5) is the opposite case, where our method fails to supply an improved APE triplet. In this example, inserted words such as 'passiert' ('happens') and substituted words such as 'wollen' ('want to') make the new mt sentence as corrupted as the mt sentence.

This example implies that our MLM can perform implausible substitutions of masked tokens when the number of errors in the mt sentence is already moderate with regards to the gold-standard statistics and thus the substitution of the remaining masked tokens requires the learning of relatively 'sophisticated' substitution; for instance, our MLM appears to have substituted a `<MASK>` for 'wieder' ('again') by 'zu' ('to') to reflect the existence of 'to' in src, but it is a wrong substitution.

Nevertheless, because our new mt basically takes a large part of ref, we can still expect that the new mt may have the advantage of choosing the given ref as its minimally post-edited result while mt does not; in this example, the ref sentence cannot be a minimally post-edited sentence for the mt sentence due to the synonymity between 'passiert' and 'geschiet' (also 'happens').

6 Related Work

Outside the APE field, the field of quality estimation (QE) shares the same training data as the APE field, and thus training data shortage is a problem for both fields. To mitigate the problem, Tuan et al. (2021) propose a method to create synthetic training data for QE models by using an MLM. First, they randomly select spans of words in a given ref to delete, insert `<MASK>` tokens, or mask with

<MASK> tokens. Then, their MLM performs text-infilling to produce a synthetic mt as a result.

Although their approach appears to be based on the same intuition as ours, the two approaches differ in the following respects. First, they used an off-the-shelf multilingual BERT (Devlin et al., 2019) model in their experiments and do not address the MLM's training. Second, their multilingual BERT does not learn cross-lingual representations in the process of self-attention on the contrary to our MLM, which takes src and mt at once as its input. Lastly, because their MLM is trained on clean training data, it learns to predict a correct substitute for each masked token, whereas our MLM learns to predict substitutes that are likely to reproduce MT errors.

7 Conclusion

In this paper, we introduce a new method to construct a synthetic APE dataset with parallel corpora. To this end, inspired by the text-infilling process performed by an MLM, we propose the MLM noising approach, which is to let an MLM inject translation errors into ref to obtain new synthetic mt. Our MLM applies text-infilling to learning the prediction of erroneous tokens that are likely to be outputs of an MT system while the error quantities to be injected are controlled using the statistics of gold-standard data.

Because we find that the TRANS approach, an existing data-synthesis method that simply translates src to the target language to obtain mt, still has distinctive advantages, we also propose corpus interleaving, which is to combine TRANS and ours for a further enhancement of the APE performance that our data produces.

Through experiments, we found that our MLM noising method significantly outperforms other existing data-synthesis methods in terms of the resulting APE performance. However, we also find that our approach may not have a significant effect when the number of translation errors already included in mt is not excessively big with regards to the gold-standard statistics because our new synthetic mt has a similar number of implausible mask substitutions in that case. We therefore expect that applying adversarial learning to our MLM (as ELECTRA (Clark et al., 2020) does) to discriminate whether the text-infilling result is plausible will be an way to further improve our method in the future.

References

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 Workshop on Statistical Machine Translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal. Association for Computational Linguistics.

Rajen Chatterjee, Markus Freitag, Matteo Negri, and Marco Turchi. 2020. Findings of the WMT 2020 Shared Task on Automatic Post-Editing. In *Proceedings of the Fifth Conference on Machine Translation*, pages 646–659, Online. Association for Computational Linguistics.

Rajen Chatterjee, Matteo Negri, Raphael Rubino, and Marco Turchi. 2018. Findings of the WMT 2018 Shared Task on Automatic Post-Editing. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 710–725, Belgium, Brussels. Association for Computational Linguistics.

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 the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 156–161, Beijing, China. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. arXiv:2003.10555 [cs.CL].

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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 the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 751–758, Berlin, Germany. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open

Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

WonKee Lee, Baikjin Jung, Jaehun Shin, and Jong-Hyeok Lee. 2021. Adaptation of Back-translation to Automatic Post-Editing for Synthetic Data Generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3685–3691, Online. Association for Computational Linguistics.

WonKee Lee, Jaehun Shin, Baikjin Jung, Jihyung Lee, and Jong-Hyeok Lee. 2020. Noising Scheme for Data Augmentation in Automatic Post-Editing. In *Proceedings of the Fifth Conference on Machine Translation*, pages 783–788, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 [cs.CL].

Matteo Negri, Marco Turchi, Rajen Chatterjee, and Nicola Bertoldi. 2018. ESCAPE: a Large-scale Synthetic Corpus for Automatic Post-Editing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Friedrich Pukelsheim. 1994. The Three Sigma Rule. *The American Statistician*, 48(2):88–91.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Jaehun Shin, Wonkee Lee, Byung-Hyun Go, Baikjin Jung, Youngkil Kim, and Jong-Hyeok Lee. 2021. Exploration of effective attention strategies for neural automatic post-editing with transformer. 20(6).

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, pages 223–231, Cambridge, Massachusetts, USA.

Yi-Lin Tuan, Ahmed El-Kishky, Adithya Renduchintala, Vishrav Chaudhary, Francisco Guzmán, and Lucia Specia. 2021. Quality Estimation without Human-labeled Data. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 619–625, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 6000–6010, Long Beach, California, USA. Curran Associates Inc.