IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 10, 2022, accepted February 13, 2022, date of publication February 15, 2022, date of current version March 1, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3152001

An Automatic Post Editing With Efficient and Simple Data Generation Method

HYEONSEOK MOON, CHANJUN PARK, JAEHYUNG SEO, SUGYEONG EO, AND HEUISEOK LIM

Department of Computer Science and Engineering, Korea University, Seoul 02841, Republic of Korea

Corresponding author: Heuiseok Lim (limhseok@korea.ac.kr)

This work was supported in part by the Ministry of Science and ICT (MSIT), South Korea, under the Information Technology Research Center (ITRC) Support Program IITP-2018-0-01405 supervised by the Institute of Information and Communications Technology Planning and Evaluation (IITP); in part by the Korean Government (MSIT) under IITP Grant 2020-0-00368 (A Neural-Symbolic Model for Knowledge Acquisition and Inference Techniques); and in part by the MSIT under the ICT Creative Consilience Program IITP-2021-2020-0-01819 supervised by the IITP.

ABSTRACT Automatic post-editing (APE) research considers methods for correcting translation results inferred by machine translation systems. The training of APE models, generally require triplets including a source sentence (*src*), machine translation sentence (*mt*), and post-edited sentence (*pe*). As considerable expert-level human labor is required in creating *pe*, APE researches have encountered difficulty in constructing suitable dataset for most of language pairs. This has led to the absence of APE data for most of language pairs, such as Korean-English, and imposed limitation to the sustainable researches of APE. Motivated by this problem, we propose a method that can generate APE triplets using only a parallel corpus without human labor. Our proposal comprises three noise generation techniques, including random, part of speech tagging (POS) based, and semantic level noises, and the effectiveness of these methods are verified by the results of quantitative and qualitative experiments on Korean-English APE tasks. As a result of our experiments, we find that POS based noise encourages the best APE performance. The proposed method is influential in that it can obviate expert human labor which was generally required in APE data construction, and enable the sustainable APE researches for the most language pairs where human-edited APE triplets are unavailable.

INDEX TERMS Automatic post editing, neural machine translation, data generation, machine translation, post editing.

I. INTRODUCTION

Automatic post-editing (APE) is a sub-field of machine translation focusing on the automated correction of errors produced by machine translation systems. APE has attracted considerable attention in that it alleviates the need for human efforts to correct machine-generated translations to human levels [1] and can contribute to domain specialized translations [2], [3]. APE is currently being actively studied as a shared task in the Conference on Machine Translation (WMT) [4].

However, a chronic problem remains in the APE researches with respect to the data generation. APE models require triplet data including a source sentence (*src*), a corresponding machine translation sentence (*mt*), and an associated post-edited sentence (*pe*), which is directly post-processed by human experts. As substantial human revisions are essential in correcting errors in *mt*, considerable expert-level human labor is required in APE data generation.

The obligation to associating expert human labor in data generation yields significant difficulties for most of the language pairs. Currently, open APE triplets have been provided only for very few language pairs, such as English-German [4], [5], while open data suitable for the implementation of APE data has not been released in most language pairs, such as Korean-English. Accordingly, it can be observed that APE research may become more concentrated toward some specific language pairs where the appropriate data has already been released.

In this study, we relieve the high dependency of APE research on the human-generated data, and propose a method to conduct APE studies on language pairs without

The associate editor coordinating the review of this manuscript and approving it for publication was Zijian Zhang.

human-edited APE triplets. In particular, we introduce several methods for automatically generating APE triplets from parallel corpora without human labor, and evaluate their performance by training APE models leveraging each approach. We propose three different APE triplet generation methods which are based on various noising schemes; Random noise, POS based noise, and semantic level noised. The effectiveness of our proposed method is validated by applying it to Korean-English pairs.

These methods commonly regard source and target sentences of parallel corpora as *src* and *pe* of APE triplets, respectively. The proposed noising schemes serve to generate a before-editing sentence, which is regarded as *mt* for APE triplets. The methods proposed in the present work were inspired by [6], which suggested the application of noising schemes to parallel corpora to generate APE triplets in English-German (En-De) language pairs. We define Random noise based APE triplet generation as a method to generate noise based on the retrieval of a training corpus. APE triplet generation utilizing POS based noise and semantic level noise indicate the methods to create *mt* by imposing noise by referring its corresponding POS tagging and semantic information retrieved by WordNet [7], respectively.

Through experiments, we quantitatively and qualitatively verified the effectiveness of these methodologies and evaluated their performance on APE tasks. Furthermore, we additionally leveraged translation system based APE triplet and confirmed that we can achieve additional improvement via utilizing translation system based APE triplet. Overall, the main contributions of this paper are as follows:

- We propose a method to generate APE models with only parallel corpora, and substantially alleviated the needs for the expert human labor in APE data generation.
- Through our proposal, we enable the sustainable APE researches for most of language pairs where APE data has not been released.
- Through our comparative analyses between several noising schemes and training strategies, we have derived the optimal strategy that trains APE model only with parallel corpus.

II. RELATED WORK AND BACKGROUND

Recent studies on APE primarily focus on techniques to alleviate data sparsity problems in APE. Representatively, data augmentation method utilizing parallel corpora have been widely adopted. In these approach, source and target sentences in parallel corpora are generally regarded as *src* and *pe* for APE triplet, respectively, and *mt* is generated by utilizing parallel corpus. In these work, *mt* indicates before-editing sentence which should be revised through APE models.

One major approach in generating *mt* through parallel corpus is to leveraging machine translation system. Reference [8] proposed a method to generate *mt* by translating *src* through a translation system. Recent studies have demonstrated significant improvements in APE models by utilizing this method [9], [10]. However, the corresponding method involves a translation system for the generation of *mt*. Therefore, we observe that the implementation of such methods may not suitable for low resource languages (LRLs) for which it is difficult to construct high-performance translation systems owing to insufficient parallel data [11]. That is, when relatively few parallel sentences are available, it is difficult to assure that an equivalent performance improvement can be achieved by the corresponding method. Moreover, because the *mt* generated through the translation system was created independently of *pe*, it is hard to say that corresponding *mt* contains information on errors that need to be corrected through humans [12]. This may mislead the APE model to the different objectives from the original purpose of APE: generating *pe* through correcting errors reside in *mt*.

Considering the limitations of such translation system based APE triplet generation, a noising scheme based APE triplet generation method was proposed [6]. In corresponding methods, *mt* is generated by intentionally imposing defect to *pe*, which was originally target sentence in parallel corpus. Four noising schemes are proposed, including adding new tokens (insertion), deleting tokens (deletion), replacing tokens with other token (substitution), and reordering tokens (shifting). An advantage of such methods is that they can also be applied in LRLs because they generate *mt* by adding noise to target sentences without the necessity of constructing a translation system.

III. PROPOSED METHOD

A. AUTOMATIC NOISE GENERATION FOR LRL APE TRIPLET

We propose a method to generate pseudo-triplets $T = \{(X^{(i)}, \hat{Y}^{(i)}, Y^{(i)})\}_{i=1}^{d}$ for APE by applying a noising scheme from parallel corpus $P = \{(X^{(i)}, Y^{(i)})\}_{i=1}^{d}$. In this notation, $X^{(i)}$, $\hat{Y}^{(i)}$, $Y^{(i)}$ indicate *src*, *mt* and *pe*, respectively. Similar to [6], the present work utilizes noising schemes in generating pseudo-triplets of APE. T and P share the same $X^{(i)}$ and $Y^{(i)}$, and each $\hat{Y}^{(i)}$ in T is generated by imposing noise to $Y^{(i)}$, As part of this study, we conducted a comparative analysis on three noising schemes, including random noise, POS based noise, and semantic level noise.

1) RANDOM NOISE

Random noise refers to a noising scheme that generates $\hat{Y}^{(i)}$ from $Y^{(i)}$ by replacing some words in the $Y^{(i)}$ with random words. Prior to the noising process, we construct a word list L by referring $Y^{(i)}$ in P for the latter use. L is defined by equation (1).

$$L = \{y_j^{(i)} | \forall y_j^{(i)} \in Y^{(i)}, \forall Y^{(i)} \in P\} \quad (1)$$

In this equation, $Y^{(i)} = \{y_j^{(i)}\}_{j=1}^{n_i}$ where n_i is the token length of $Y^{(i)}$ which is segmented by NLTK tokenizer [13]. L combines all the segmented tokens in every $Y^{(i)}$ in P, without overlapping.

In this noising process, $\hat{Y}^{(i)}$ is generated by noising some words in a $Y^{(i)}$ with random words selected from L, without

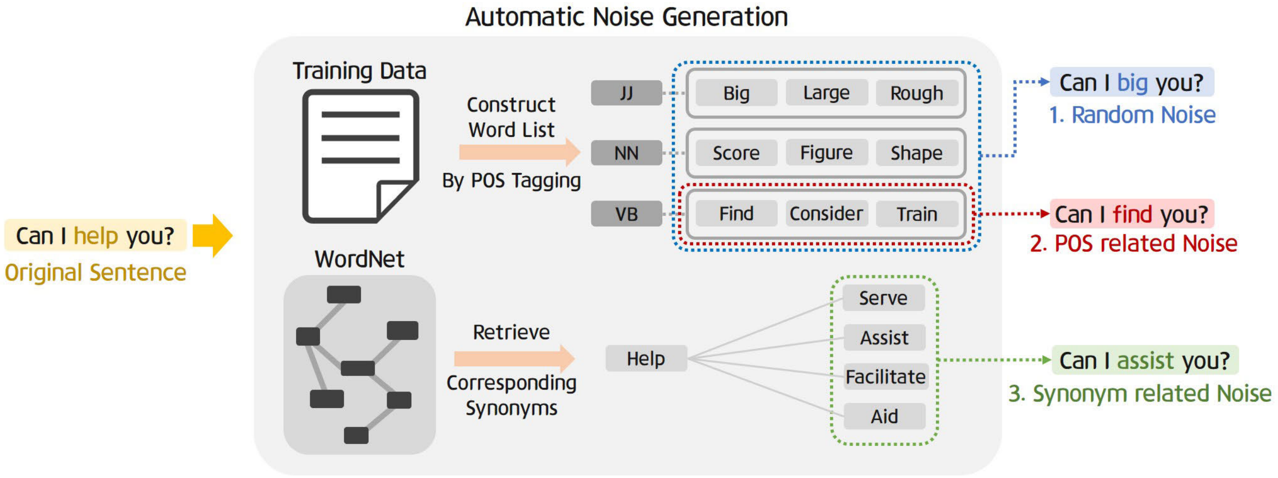

FIGURE 1. Overall process of automatic noise generation for LRL APE triplets.

any consideration of contextual information. In addition to replacing words in a $Y^{(i)}$ with random words, we utilize insertion, deletion, replacing and shifting noise, which denote adding new words, deleting original words, changing original word into different one, and changing the positions of words in the $Y^{(i)}$, respectively. These noising schemes were proposed by [6], and we generate $\hat{Y}^{(i)}$ by combining these noising schemes together.

The ratio of noise to be imposed to $Y^{(i)}$ is determined by the probability. In noising process, each word in $Y^{(i)}$ is judged to be noised or not, based on the probability p, selected from the uniform distribution [0, 1). The noising probability applied to each $Y^{(i)}$ varies throughout the whole training process. This can enable the model to obtain the robust error-revising capacity. Detailed procedure of generating $\hat{Y}^{(i)}$ can be formularized as Equation (2)

$$\hat{y}_j^{(i)} = \begin{cases} t : y_j^{(i)}(t \in L) & \text{if } r \in [0, \frac{p}{4}) \\ \text{No Token} & \text{if } r \in [\frac{p}{4}, \frac{p}{2}) \\ t \ (t \in L) & \text{if } r \in [\frac{p}{2}, \frac{3p}{4}) \\ y_k^{(i)}(k \in [1, n_i]) & \text{if } r \in [\frac{3p}{4}, p) \\ y_j^{(i)} & \text{if } r \in [p, 1) \end{cases} \tag{2}$$

Each noising scheme in Equation (2) refers to the insertion, deletion, replacing, shifting noise, and skipping noise, respectively in order from the top. With total probability p, each noise scheme is selected to be equally-distributed.

In imposing noise schemes to $y_j^{(i)}$ for generating $\hat{Y}^{(i)}$, $\{y_j^{(i)}\}_{j=1}^{n_i}$ is generated by segmenting each $Y^{(i)}$ with NLTK tokenizer. Then $\hat{Y}^{(i)}$ can be obtained by imposing noise schemes with probability p by the random variable r selected from the uniform distribution [0, 1). For the insertion and replacing noise, random token t is extracted from the word list L, and for the shifting noise, $\hat{y}_k^{(i)}$ is shifted to $\hat{y}_j^{(i)}$ simultaneously.

2) POS BASED NOISE

In applying POS based noise, a similar noising process with the random noise is implemented, but different from random noise, part of speech (POS) tagging-based word list L_{pos} is used to determine tokens for the replacement. For the construction of L_{pos}, all the $y_j^{(i)}$ which POS tag is pos are accumulated as shown in equation (3).

$$L_{pos} = \{y_j^{(i)} | \text{POS}(y_j^{(i)}) = pos, \forall y_j^{(i)} \in P\} \tag{3}$$

In this equation, pos refers to the POS tag of $y_j^{(i)}$, which is a replacing token in Y_i. During the noising process, each word used to replace another is selected randomly from L_{pos}. Specifically, the similar noising process with equation (2) is proceeded, but t is selected from L_{pos}, not from L. Through this process, a verb in the $Y^{(i)}$, such as "help" is replaced with another verb such as "find", not with a word in another POS tag such as noun or adjective.

These noising processes follow the traditional NLP pipeline [14] in imposing noise to each sentence. Thus, we can expect higher performance compared with the application of random noise, which does not have any standard. We utilized the NLTK toolkit [13] to perform POS tagging.

3) SEMANTIC LEVEL NOISE

In semantic level noise, wordnet [7] is utilized in imposing noise. Similar with previous noising schemes, $\hat{Y}^{(i)}$ is generated from $Y^{(i)}$ by replacing some words in the $Y^{(i)}$ with others. Once a word $y_j^{(i)}$ is selected to be noised, its corresponding wordnet information is retrieved, especially a list of its synonyms. These synonyms are regarded as candidates for the replacements, and during the noising process, a corresponding synonym is randomly selected and replaced. Specifically, in applying equation (2), newly replaced token t is selected from the synonym list of $y_j^{(i)}$, not L.

This can be viewed as a noising scheme used to train a type of human-level editing. Representative errors in the translation results, which should be edited by human experts,

TABLE 1. Statistics of training, validation, and test sets. SRC, MT, and PE indicate source, machine translation, and post-edit, respectively. "Avg T_len" and "Avg C_len" indicate average token length and average character length of sentences in each dataset, respectively.

Dataset		TER (MT-PE)	BLEU (MT-PE)	SRC Avg T_len	MT Avg T_len	PE Avg T_len	SRC Avg C_len	MT Avg C_len	PE Avg C_len	# Triplets
Training		-	-	23.88	-	30.39	59.92	-	150.50	96,000
Validation		52.841	34.69	24.00	30.44	30.54	60.09	150.06	151.23	12,000
Test	Google	52.983	34.49	23.95	30.44	30.59	60.02	149.74	151.22	12,000
	Microsoft[1]	59.478	26.49	23.95	29.67	30.59	60.02	141.11	151.22	12,000
	Amazon[2]	60.421	23.32	23.95	27.42	30.59	60.02	133.19	151.22	12,000

include the miss-choice of synonyms, which have the same meaning but play a different role in a given domain or context. For instance, "help" and "assist" have the same meaning as "give help or assistance, or be of service." However, considering formality and context, these two phrases should be considered differently. By imposing this type of error in the noising process, a model could be trained to mimic human-level editing.

A brief structure for the application of each noising scheme is shown in Figure 1. Representatively, replacing noise is depicted in this figure. During the generation of $\hat{Y}^{(i)}$ in each noising scheme, some words in the $Y^{(i)}$ were selected randomly and replaced with other words according to the corresponding noising scheme. In random noise, words are replaced randomly from the word list constructed by combining words of $Y^{(i)}$ in the whole corpus P, without any consideration of contextual or semantic information. In POS based noise, replacing words are selected from the word list, according to its corresponding POS tag, and in semantic level noise, synonym lists of replacing words in $Y^{(i)}$ are retrieved, and then random synonyms are selected from their corresponding synonym lists.

B. EFFICIENT TRAINING OF APE MODEL

In this study, we construct an APE model by fine tuning APE tasks to a pretrained vanilla transformer [15] based NMT model. Transfer learning strategy [16] that fine tuning APE task to the pretrained NMT model is shown to be effective in improving APE performance [17], [18]. As a pretrained language model which has a large amount of parameters and is trained with a large amount of training data, such as XLM [19], is not required, this approach is also effective in aspect of training efficiency.

In the APE fine tuning process, we utilized a bottleneck adapter layer (BAL) [20] structure for more efficient training [17], [21]. The BAL comprises two dense layers with one activation function. We adopted ReLU as the activation function. The forward processing through a single BAL structure can be described as equation (4).

$$BAL(x) = W_2 \cdot ReLU(W_1 \cdot x + b_1) + b_2 + x \qquad (4)$$

In equation (4), x is an input embedding vector that is $x \in R^{d_{model} \times len}$, where len indicates max token length. W_1, W_2, b_1, b_2 indicate trainable parameters where $W_1 \in \mathrm{R}^{d_{bal} \times d_{model}}$, $W_2 \in \mathrm{R}^{d_{model} \times d_{bal}}$, $b_1 \in \mathrm{R}^{d_{bal} \times 1}$, $b_2 \in \mathrm{R}^{d_{model} \times 1}$. By setting the output dimension of W_2 to be equal to d_{model}, the final dimensionality of $BAL(x)$ is made to be the same as the dimension of x. The BAL structure is applied to the pre-trained transformer based NMT model. For each transformer layer, two BAL structure is added to the posterior position of self-attention structure, and the feed forward network structure. During the fine tuning process of the BAL-added transformer model, we froze the parameters of the pre-existing transformer model, and train only the BAL structure.

Adopting BAL structure in fine tuning process noticeably improves training efficiency. In particular, the amount of training parameters involved in a fine tuning process can be significantly reduced, compared with naive fine tuning strategy [20]. In this work, the amount of trainable parameters in APE fine tuning is 1.6M, which is about 1.6% of the total amount of parameters in the whole model structure. This enable the model to have considerable performance only by training relatively small amount of parameters, and accordingly can lead to the reduction of the computing power and training time required in model training.

APE fine tuning process by utilizing our proposed noising scheme can be described as follows. First, we generate pseudo-triplet APE data $T = \{(X^{(i)}, \hat{Y}^{(i)}, Y^{(i)})\}_{i=1}^{d}$ from parallel corpus $P = \{(X^{(i)}, Y^{(i)})\}_{i=1}^{d}$, by following our proposal. Then by utilizing T, we train the APE model θ with the training objective to maximize a sequence to sequence based probability as equation (5).

$$\sum_{i=1}^{d} \log \left[\prod_{k=1}^{m_i} P(z_k^{(i)} \mid X^{(i)}, \hat{Y}^{(i)}, z_{t<k}^{(i)}, \theta) \right] \qquad (5)$$

In this equation, $z_k^{(i)}$ refers to the k^{th} token in $Y^{(i)}$, which is segmented by our sentencepiece tokenizer. We can denote it as $Y^{(i)} = \{z_k^{(i)}\}_{k=1}^{m_i}$ where m_i is the max token length of $Y^{(i)}$. Equation (5) indicates that in fine tuning process, a concatenated sentence of $X^{(i)}$ and $\hat{Y}^{(i)}$ are utilized as an input structure. Then, by a sequence-to-sequence [22] based training process, a model is trained to generate $Y^{(i)}$. These fine tuning processes are suggested by [17], and we modified the corresponding processes appropriately in our experiments.

IV. EXPERIMENTS AND RESULTS

A. DATASET DETAILS

To verify the effectiveness of our proposed approach, we adopted a Korean-English parallel corpus, released by

AIhub[3] [23]. This released data, comprising 1.6 million sentence pairs, was generated by the NMT model and then inspected by human experts. This corpus is being adopted in many Korean-English translation studies [24], [25]. As a precise human inspection was engaged in data generation, we can ensure the data quality to be sufficiently high. We filtered out several sentences with less than two words or more than 200 words for consistent training [24], [26]. We extracted 120,000 sentence pairs from these data and utilized them to generate APE triplets, while others were utilized to train the NMT models. From the 120,000 extracted sentence pairs, we arbitrarily selected 12,000 sentence pairs each as validation and test datasets.

We used translation edit rate (TER) [27] and BLEU [28] score as our evaluation metrics. To measure each metric, we employed TER measurement software[4] known as tercom for TER score, and mteval13.pl[5] presented by Moses for BLEU score. BLEU is the most representative metrics that measures the performance of machine translation system based on the n-gram similarity with length penalty [29], and is being adopted for the auxiliary evaluation metric of APE research [4]. TER is now being adopted for the major evaluation metric for the APE system that measures the minimum number of editing required in revising translated sentence into reference one [30]. The performance evaluation of our proposed model was conducted based on three different test sets, which were generated by Google, Amazon, and Microsoft translation systems. The statistics for training, validation, and test datasets used in this paper are as shown in Table 1.

As shown in Table 1, Google Translate had the best MT quality, while Microsoft and Amazon showed relatively lower MT quality. By conducting evaluations on these test sets that show different qualities, we verified the effectiveness of our proposal.

B. MODEL DETAILS

In our experiment, we constructed an NMT model which has an vanilla transformer model structure, and then fine-tuned the APE task to that model. The adopted transformer-based model consisted of six encoder-decoder layers with a hidden size of 512. The vocab size of the model was set to 50,000, and we utilized SentencePiece [31] uni-gram model as a tokenizer. Translation training was conducted using `fairseq` [32]. Specifically, 200K training steps of 16,384 max tokens was proceeded with early stopping based on the validation BLEU score. We followed training instruction given by `fairseq` [6] where learning rate is empirically selected to 5e-4 with inverse square root learning rate scheduler.

Prior to the fine tuning of the APE task, we added the BAL structure to the pretrained NMT model structure. We set BAL size d_{bal} to be 64, which is 1/8 of the pretrained model's hidden size d_{model}, to prevent overfitting and improve efficient learning. Model parameter settings are based on [21], where d_{bal} is set to be smaller than d_{model}. Total amount of parameters in our NMT model is 96.4M and the amount of parameters added by utilizing BAL structure is 1.6M. This indicates that during APE fine tuning process, only 1.6M parameters are trained among 98.0M parameters of the whole model. We utilized Huggingface [33] for training the BAL-added model structure. For the training process, we adopted Adam optimizer [34] and cosine annealing scheduler [35] with learning rate 3e-5, selected empirically. One RTX A6000 was used for the training, and every training process takes within a day. Specifically, early stopping was applied based on the validation BLEU score.

C. EXPERIMENTAL RESULTS

1) MAIN RESULTS

In this section, we present the experimental results of verifying the effectiveness of our proposed method. To perform this verification, we generated APE training data by applying the proposed noising schemes, including random, POS based, semantic level noise, and trained three different APE models by utilizing each data. The effectiveness of each noising scheme is measured by the performance of its corresponding model. Our results are shown in Table 2.

As shown in Table 2, by utilizing the APE triplets generated by the three proposed noising schemes, it was possible to create an APE model that effectively correct errors included in the *mt*. Among our three noising schemes, APE triplets utilizing POS based noise demonstrated the best performance. This shows that POS information is influential if utilized in imposing noise, and elaborate consideration in noising schemes may further improve the APE performance. By comparing such results with random noise, we can also find that maintaining structural consistency in injecting noise may lead to better performance, as in generating *mt*, tokens in *pe* is substituted with the tokens with the same POS.

Through these experiments, we can observe that APE triplets generated by semantic level noise achieved relatively poor performance. We can infer that as the entire surrounding context is not fully considered in the noising process, the proper synonym may not have been selected as its original meaning in the *pe*, and thereby adequate training has not been performed.

Additionally, the APE model trained by the APE triplets generated by semantic level noise demonstrate even worse performance than the model utilizing random noise. This can be interpreted that the word list used in the noising process, which was extracted from the synset of WordNet, may have led to a slight degradation in APE performance as the replaced words were derived independently from the *pe* corpus. This can be interpreted that the model is more likely sensitive to the domain specific vocabulary. Note that in semantic level noise, replacing words are selected from the vocabulary

[3] https://aihub.or.kr/

[4] http://www.cs.umd.edu/ snover/tercom/

[5] https://github.com/moses-smt/mosesdecoder

[6] https://github.com/pytorch/fairseq/tree/main/examples/translation

TABLE 2. Performances of each APE model trained by APE triplets generated by our proposed noising scheme. Baseline refers to the machine translation result before the editing process by the APE model.

	Amazon		Microsoft		Google	
	TER ↓	BLEU ↑	TER ↓	BLEU ↑	TER ↓	BLEU ↑
Baseline	60.421	23.32	59.478	26.49	52.983	34.49
Random Noise	45.870 (-14.551)	43.35 (+20.03)	45.921 (-13.557)	43.26 (+16.77)	**45.743** **(-7.240)**	43.44 (+8.95)
POS Based Noise	**45.801** **(-14.620)**	**43.42** **(+20.10)**	**45.832** **(-13.646)**	**43.41** **(+16.92)**	45.758 (-7.225)	**43.47** **(+8.98)**
Semantic Level Noise	46.596 (-13.825)	42.70 (+19.38)	46.685 (-12.793)	42.59 (+16.10)	46.114 (-6.869)	43.09 (+8.60)

TABLE 3. Qualitative analysis for each APE model trained by the APE triplets generated by our proposed noising schemes. *mt* was obtained by translating the source sentence in the APE test set with the Amazon translation system. Colored words in each example indicate semantically related words.

APE Triplet	Example-1	Example-2
src	병사의 얼굴, 옷 주름 등에 명암을 표현하고 있어 중국의 서양화법을 수용하였음을 알 수 있다. (Byeongsaui eolgul, ot juleum deung-e myeong-ameul pyohyeonhago iss-eo jung-gug-ui seoyanghwabeob-eul suyonghayeoss-eum-eul al su issda)	KPOP 아이돌 그룹 내에 글로벌 바람이 분 것은 이미 오래된 일이다. (KPOP aidol geulub naee geullobeol balam-i bun geos-eun imi olaedoen il-ida.)
mt	The contrast is expressed on the soldier's face and the wrinkles of clothes, indicating that it has adopted the Western painting method of China.	The global wind within the KPOP Idol Group is already old.
pe	The light and dark effect **reflected in the faces of the soldiers and the folds in their clothes suggest that this painter had adopted the** Western **painting style of China.**	**It** has been long **since a global wind is spreading** within **the KPOP idol groups.**

APE model	Post Edited Example-1	Post Edited Example-2
Random Noise	The use of the Chinese traditional painting technique can be seen as it expresses light and shade on the face and wrinkles of the soldiers.	The global wind has already been blowing in the KPOP idol group.
POS Based Noise	It can be seen that the Chinese style of Western painting was accepted as it expresses light and shade on the face and clothes of soldiers.	It has already been a long time since the global wind blew within the KPOP idol group.
Semantic Level Noise	The face and wrinkles of the soldiers show that the Chinese style of Western painting was adopted.	The global wind in the KPOP idol group is already old.

extracted from the synonym list in wordnet, which is retrieved independently of the domain of training data.

Performance difference between the POS based noise utilizing model, and the semantic level noise utilizing model also shows the importance of domain-consistency in APE data generation. POS based noise may encourage the optimal performance by considering contextual information as well as reflecting the domain specificity of training data, in generating APE data.

Though our three methods yielded slightly different results, the overall performances of the APE models utilizing the corresponding APE triplets are quite prominent. For instance, our APE model trained with POS based noise improves the quality of translation results obtained from Amazon translation by 14.620 TER and 20.10 BLEU score. As our proposal is a data generation method that automatically generates APE triplets from a parallel corpus without the need for human experts to perform editing, it is expected to be of benefit especially in application to LRL, for which human-edited APE triplets have not been released. Furthermore, as it exclude the needs for the expert level human labor in APE data generation, which has been the major obstacles in vigorous APE studies throughout the universal language pairs, this work contribute to the sustainable APE research.

2) QUALITATIVE ANALYSIS

To perform a more reliable verification of our proposal, we additionally conducted a qualitative analysis. We analyzed the actual editing results of the APE models, which were trained through APE triplets generated by our proposed three noising schemes. The results are shown in Table 3.

Based on the results, we can observe that there exist considerable differences between the edited results for each model, especially with respect to the order of words within each sentence, or in the overall meaning of each sentence. We can observe that the best qualitative results are obtained through the model that leveraged the POS based noise. The model, which is trained by semantic noise based APE triplets,

TABLE 4. Performances of the APE model for the various mixing ratios. m refers to mixing ratio, indicating the occupying ratio of d_{noise} among d_{MT} and d_{noise} in batch configuration for each training step. $m = 0.0$ indicates the model trained only using d_{noise}, and $m = 1.0$ indicates the model trained only using d_{MT}.

		Amazon		Microsoft		Google	
		TER ↓	BLEU ↑	TER ↓	BLEU ↑	TER ↓	BLEU ↑
Baseline		60.421	23.32	59.478	26.49	52.983	34.49
Random Noise	$m = 1.00$	45.870	43.35	45.921	43.26	45.743	43.44
	$m = 0.90$	45.642	43.53	45.568	43.59	45.509	43.63
	$m = 0.75$	45.531	43.65	45.524	43.61	45.345	43.76
	$m = 0.50$	45.198	43.50	45.175	43.67	44.822	44.16
	$m = 0.25$	44.859	44.15	**44.811**	**44.22**	**44.812**	**44.23**
	$m = 0.10$	**44.789**	**44.20**	44.837	44.20	44.852	44.21
	$m = 0.00$	44.913	44.10	44.909	44.12	44.897	44.08
POS Based Noise	$m = 1.00$	45.801	43.42	45.832	43.41	45.758	43.47
	$m = 0.90$	45.773	43.35	45.921	43.26	45.743	43.44
	$m = 0.75$	45.643	43.50	45.685	43.47	45.392	43.77
	$m = 0.50$	45.232	43.82	45.087	43.98	44.885	44.15
	$m = 0.25$	**44.861**	**44.16**	**44.747**	**44.22**	**44.770**	**44.24**
	$m = 0.10$	44.879	44.10	44.853	44.14	44.936	44.08
	$m = 0.00$	44.913	44.10	44.909	44.12	44.897	44.08
Semantic Level Noise	$m = 1.00$	46.596	42.70	46.685	42.59	46.114	43.09
	$m = 0.90$	46.142	43.25	46.224	43.17	45.988	43.29
	$m = 0.75$	45.859	43.49	45.877	43.42	45.760	43.50
	$m = 0.50$	45.434	43.64	45.396	43.68	45.166	43.91
	$m = 0.25$	44.943	44.04	44.909	44.09	**44.806**	44.16
	$m = 0.10$	**44.819**	**44.20**	**44.852**	**44.19**	44.827	**44.17**
	$m = 0.00$	44.913	44.10	44.909	44.12	44.897	44.08

ignored the information on "the light and dark effect," and lost is tense such as "has been", and in leveraging random noise, a word from the source sentence, "서양" that should be translated to "Western" is omitted in the edited sentence.

However, for the APE model that leverages POS based noise, these errors were properly corrected, and the edited sentences expressed the overall meaning of the post-edit sentences properly. These results indicate that POS based noise is of particular benefit in APE training.

3) MUTUAL SUPPLEMENTATION EFFECT WITH TRANSLATION SYSTEM

Referring to recent studies on APE, we can identify two different methods of generating pseudo-APE triplets from parallel corpora. The first method is generating *mt* by translating *src* with machine translation systems, similar to the eSCAPE [8], and the second method is to utilize noising schemes as used in the present work. For the simplicity, we denote APE data which is augmented by machine translation system as d_{MT}, and APE data generated by applying noising scheme as d_{noise}.

In previous study, [6] showed that utilizing both d_{MT} and d_{noise} can substantially improve the APE performance. Inspired by this, we investigate the optimal approach to use both APE triplet generation methods for improving APE performance. If the performance of the APE model can be improved by being trained with both triplets, compared with the model trained by the respective APE triplet, we denote it as a mutual supplementation effect.

Through this experiments, we investigate how this mutual supplementation effect can be obtained, and inspect the optimal strategy of utilizing both d_{MT} and d_{noise} in the training process, with respect to the batch configuration. Specifically, we extend the previous study [6] where the occupying ratio of d_{noise} in the whole batch configuration is fixed, and experimented for the various batch configurations.

We denote the occupying ratio of d_{noise} in a batch configuration of the training process where d_{MT} and d_{noise} are utilized together, as the mixing ratio m. We experimented with various mixing ratios and figure out the optimal m for achieving mutual supplementation effect. Our results are shown in Table 4.

The results demonstrate that the proposed model trained with d_{MT} and d_{noise} together showed improved performance compared with the model trained by d_{MT} or d_{noise} alone, for all of our three noising schemes. We also found that generally low m lead to higher performance; if we set the mixing ratio to be lower than 0.25, the models were largely able to obtain higher performance than the model trained by d_{MT} alone. These results indicate that the mutual supplementation effect can be achieved by strategically combining d_{MT} and d_{noise} together in batch configuration. That is, by adopting d_{MT} along with our proposed method, we can obtain the further improvement of APE model performance. As d_{MT} can also be generated without expert human level labor, this can substantially assist our proposed noising schemes, without deteriorating the original intention of our proposal: sustainability of APE data generation and APE research.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a method to automatically generate APE triplets from parallel corpora without human labor. Three noising schemes were proposed, including random noise, POS based noise, and semantic level noise, and the performance of each method was evaluated using an APE model trained with APE triplets generated by the corresponding noising schemes. We confirmed that decent APE model can be constructed by utilizing our proposal, without leveraging the APE data which should be generated by expert level human labor. We figured out that by additionally utilizing translation system, even higher APE performance can be obtained. Our proposal enable the sustainable APE researches even for the language pairs where appropriate APE data has not been released. In the future, we plan to study APE considering industrial services based on a data-centric methodology.

ACKNOWLEDGMENT

(Hyeonseok Moon and Chanjun Park contributed equally to this work.)

REFERENCES

[1] S. Pal, N. Herbig, A. Krüger, and J. van Genabith, "A transformer-based multi-source automatic post-editing system," in *Proc. 3rd Conf. Mach. Transl., Shared Task Papers*, 2018, pp. 827–835.

[2] P. Isabelle, C. Goutte, and M. Simard, "Domain adaptation of MT systems through automatic post-editing," MT Summit XI, Interact. Lang. Technol. Group, NRC Inst. Inf. Technol., Gatineau, QC, Canada, Tech. Rep., 2007, vol. 102.

[3] S. Chollampatt, R. H. Susanto, L. Tan, and E. Szymanska, "Can automatic post-editing improve NMT?" 2020, *arXiv:2009.14395*.

[4] R. Chatterjee, C. Federmann, M. Negri, and M. Turchi, "Findings of the WMT 2019 shared task on automatic post-editing," in *Proc. 4th Conf. Mach. Transl.*, 2019, pp. 11–28.

[5] R. Chatterjee, M. Negri, R. Rubino, and M. Turchi, "Findings of the WMT 2018 shared task on automatic post-editing," in *Proc. 3rd Conf. Mach. Transl., Shared Task Papers*, 2018, pp. 646–659. [Online]. Available: https://aclanthology.org/2020.wmt-1.75

[6] W. Lee, J. Shin, B. Jung, J. Lee, and J.-H. Lee, "Noising scheme for data augmentation in automatic post-editing," in *Proc. 5th Conf. Mach. Transl.*, 2020, pp. 783–788.

[7] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[8] M. Negri, M. Turchi, R. Chatterjee, and N. Bertoldi, "ESCAPE: A large-scale synthetic corpus for automatic post-editing," 2018, *arXiv:1803.07274*.

[9] A. V. Lopes, M. A. Farajian, G. M. Correia, J. Trenous, and A. F. T. Martins, "Unbabel's submission to the WMT2019 APE shared task: BERT-based encoder-decoder for automatic post-editing," 2019, *arXiv:1905.13068*.

[10] M. Junczys-Dowmunt and R. Grundkiewicz, "MS-UEdin submission to the WMT2018 APE shared task: Dual-source transformer for automatic post-editing," 2018, *arXiv:1809.00188*.

[11] B. Zoph, D. Yuret, J. May, and K. Knight, "Transfer learning for low-resource neural machine translation," 2016, *arXiv:1604.02201*.

[12] W. Lee, B. Jung, J. Shin, and J.-H. Lee, "Adaptation of back-translation to automatic post-editing for synthetic data generation," in *Proc. 16th Conf. Eur. Chapter Assoc. Comput. Linguistics, Main Volume*, 2021, pp. 3685–3691.

[13] S. Bird and E. Loper, "NLTK: The natural language toolkit," in *Proc. ACL Interact. Poster Demonstration Sessions*, Barcelona, Spain, Jul. 2004, pp. 214–217. [Online]. Available: https://www.aclweb.org/anthology/P04-3031

[14] I. Tenney, D. Das, and E. Pavlick, "BERT rediscovers the classical NLP pipeline," 2019, *arXiv:1905.05950*.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, *arXiv:1706.03762*.

[16] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[17] R. Chatterjee, J. G. C. de Souza, M. Negri, and M. Turchi, "HW-TSC's participation at WMT 2020 automatic post editing shared task," in *Proc. 5th Conf. Mach. Transl.*, 2020, pp. 797–802.

[18] H. Moon, C. Park, S. Eo, J. Seo, and H. Lim, "The verification of the transfer learning-based automatic post editing model," *J. Korea Converg. Soc.*, vol. 12, no. 10, pp. 27–35, 2021.

[19] G. Lample and A. Conneau, "Cross-lingual language model pretraining," 2019, *arXiv:1901.07291*.

[20] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for NLP," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 2790–2799.

[21] H. Moon, C. Park, S. Eo, J. Seo, and H. Lim, "An empirical study on automatic post editing for neural machine translation," *IEEE Access*, vol. 9, pp. 123754–123763, 2021.

[22] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," 2014, *arXiv:1409.3215*.

[23] C. Park and H. Lim, "A study on the performance improvement of machine translation using public Korean-English parallel corpus," *J. Digit. Converg.*, vol. 18, no. 6, pp. 271–277, 2020.

[24] C. Park, S. Eo, H. Moon, and H. Lim, "Should we find another model: Improving neural machine translation performance with ONE-piece tokenization method without model modification," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol., Ind. Papers*, 2021, pp. 97–104.

[25] S. Eo, C. Park, H. Moon, J. Seo, and H.-S. Lim, "Dealing with the paradox of quality estimation," in *Proc. 4th Workshop Technol. MT Low Resource Lang. (LoResMT)*, 2021, pp. 1–10.

[26] H. Moon, C. Park, S. Eo, J. Park, and H. Lim, "Filter-mBART based neural machine translation using parallel corpus filtering," *J. Korea Converg. Soc.*, vol. 12, no. 5, pp. 1–7, 2021.

[27] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A study of translation edit rate with targeted human annotation," in *Proc. 7th Conf. Assoc. Mach. Transl. Americas, Tech. Papers*, 2006, pp. 223–231.

[28] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2001, pp. 311–318.

[29] L. Barrault, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, S. Huang, M. Huck, P. Koehn, Q. Liu, V. Logacheva, C. Monz, M. Negri, M. Post, R. Rubino, L. Specia, and M. Turchi, "Findings of the 2020 conference on machine translation (WMT20)," in *Proc. 5th Conf. Mach. Transl.*, Nov. 2020, pp. 1–55. [Online]. Available: https://www.aclweb.org/anthology/2020.wmt-1.1

[30] R. Chatterjee, M. Freitag, M. Negri, and M. Turchi, "Findings of the WMT 2018 shared task on automatic post-editing," in *Proc. 5th Conf. Mach. Transl.*, Nov. 2020, pp. 646–659. [Online]. Available: https://www.aclweb.org/anthology/2020.wmt-1.75

[31] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," 2018, *arXiv:1808.06226*.

[32] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "Fairseq: A fast, extensible toolkit for sequence modeling," 2019, *arXiv:1904.01038*.

[33] T. Wolf *et al.*, "HuggingFace's transformers: State-of-the-art natural language processing," 2019, *arXiv:1910.03771*.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[35] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," 2016, *arXiv:1608.03983*.

HYEONSEOK MOON received the B.S. degree from the Department of Science in Mathematics and Engineering, Korea University, Seoul, South Korea, in 2021, where he is currently pursuing the Ph.D. degree in computer science and engineering with the Natural Language Processing and Artificial Intelligence Laboratory, under an integrated master's and Ph.D. course. His research interests include natural language processing, neural machine translation, automatic post editing, and parallel corpus filtering.

CHANJUN PARK received the B.S. degree in natural language processing and creative convergence from the Busan University of Foreign Studies, Busan, South Korea, in 2019. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Korea University, Seoul, South Korea. From June 2018 to July 2019, he worked at SYSTRAN as a Research Engineer. His research interests include machine translation, grammar error correction, simultaneous speech translation, and deep learning.

JAEHYUNG SEO received the B.S. degree from the Department of English Language and Literature, Korea University, Seoul, South Korea, in 2020, where he is currently pursuing the Ph.D. degree in computer science and engineering with the Natural Language Processing and Artificial Intelligence Laboratory, under an integrated master's and Ph.D. course. His research interests include language generation and decoding strategy, where he tries to find inspiration from how humans do it and build generative model based on commonsense reasoning.

SUGYEONG EO received the B.S. degree in linguistics and cognitive science from the Hankuk University of Foreign Studies, Yongin-si, South Korea, in 2020. She is currently pursuing the Ph.D. degree with the Natural Language Processing and Artificial Intelligence Laboratory, Korea University, Seoul, South Korea, under an integrated master's and Ph.D. course. Her research interests include neural machine translation and quality estimation, where she tries to predict machine translation quality that minimizes human labor.

HEUISEOK LIM received the B.S., M.S., and Ph.D. degrees in computer science and engineering from Korea University, Seoul, South Korea, in 1992, 1994, and 1997, respectively. He is currently a Professor with the Department of Computer Science and Engineering, Korea University. His research interests include natural language processing, machine learning, and artificial intelligence.