

AESOP: Paraphrase Generation with Adaptive Syntactic Control

Jiao Sun^{1,2}, Xuezhe Ma^{1,2} and Nanyun Peng^{1,2,3}

¹Computer Science Department, University of Southern California

²Information Sciences Institute, University of Southern California

³Computer Science Department, University of California, Los Angeles

jiaosun@usc.edu, xuezhema@usc.edu, violetpeng@cs.ucla.edu

Abstract

We propose to control paraphrase generation with carefully chosen target syntactic structures to generate more proper and higher quality paraphrases. Our model, AESOP, leverages a pretrained language model and purposefully selected syntactical control via a retrieval-based selection module to generate fluent paraphrases. Experiments show that AESOP achieves state-of-the-art performances on semantic preservation and syntactic conformation on two benchmark datasets with ground-truth syntactic control from human-annotated exemplars. Moreover, with the retrieval-based target syntax selection module, AESOP generates paraphrases with even better qualities than the current best model using human-annotated target syntactic parses according to human evaluation. We further demonstrate the effectiveness of AESOP to improve classification models' robustness to syntactic perturbation by data augmentation on two GLUE tasks.

1 Introduction

Syntactically-controlled paraphrase generation, which aims to generate paraphrases that conform with given syntactic structures, has drawn increasing attention in the community. On the one hand, paraphrase generation has benefited a wide range of NLP applications, such as neural machine translation (Yang et al., 2019), dialogue generation (Gao et al., 2020), as well as improving model robustness (Huang et al., 2021) and interpretability (Jiang et al., 2019). On the other hand, syntactically-controlled paraphrasing has been used for diverse question generation (Yu and Jiang, 2021), diversifying creative generation (Tian et al., 2021) and improving model robustness (Iyyer et al., 2018; Huang and Chang, 2021).

However, selecting suitable target syntactic structures to control paraphrase generation for diverse and high-quality results is a lesser explored direction. Prior works usually use a fixed set of syntactic

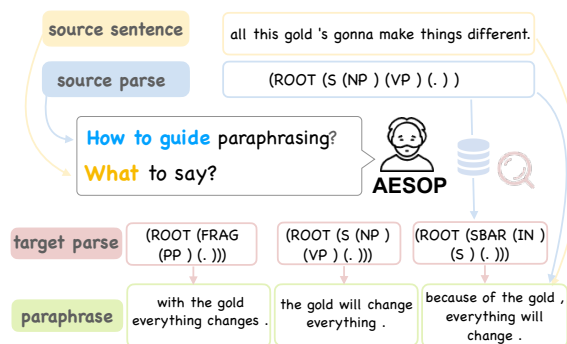


Figure 1: Given a source sentence, AESOP selects target syntactic parses adaptively to guide paraphrase generation. Paraphrases here are all generated by AESOP, which preserve the semantics from source sentences and conform with the selected syntactic parses.

structures for all input sentences (Iyyer et al., 2018; Huang and Chang, 2021). A challenge with this method is that not all sentences can be paraphrased into the same set of syntactic structures. For example, it is impossible to turn a long sentence with multiple clauses into a noun phrase. Thus, Chen et al. (2019b) proposed to use crowd-sourcing to collect exemplars that can provide compatible syntax with the source sentence to guide generation. Disadvantages with this method are that the crowd-sourcing process is costly, and one exemplar sentence can only provide a specific syntactic guidance, while there are many syntactic parses that can properly guide the paraphrase generation (as shown in Figure 1).

In contrast, we propose to automatically select multiple syntactic parse structures to control paraphrase generation for more diverse and higher quality generation. Our **first contribution** is the proposal of AESOP (Adaptive Syntactically-Controlled Paraphrasing), a model that integrates pretrained Language Models (LMs) with a novel retrieval-based target syntactic parse selection module to control paraphrase generation. By leveraging the expressiveness of pretrained LMs and

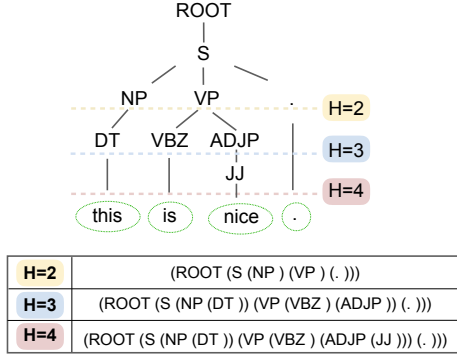


Figure 2: Prune a constituency parse tree at heights H .

the adaptive selection module, AESOP is capable of generating fluent and syntactically-diverse paraphrases. With ground-truth target syntactic parses from human-annotated exemplars, AESOP achieves the state-of-the-art performance on both semantic preservation and syntactic conformation metrics. By human evaluation, we show that AESOP can generate paraphrases with even better quality than the current best model using human annotated exemplars, which points out the importance of studying the adaptive target parse selection for future works on controlled paraphrase generation.

Our **second contribution** is the construction of two datasets containing adversarial examples with syntactic perturbation generated by AESOP that are further validated and labeled by crowd workers. Experiments show that the two datasets are challenging to current classification models, and using AESOP to augment the training data can effectively improve classification models’ robustness to syntactic attacks.¹

2 Task Formulation

We formulate the task of adaptive syntactically-controlled paraphrase generation as: given an input sentence X , find a set of proper syntactic controls Y to generate paraphrases Z , such that Z ’s syntax conforms to Y while retaining the semantics of X .

We use the term *target syntactic parses* to refer to the syntactic structure that guides the generation, which could be from exemplar sentences, a set of fixed templates, or our adaptive selection module.

¹Data and code can be found at <https://github.com/PlusLabNLP/AESOP>

Algorithm 1 Adaptive Target Parse Selection

Input: source parse at level H : T_s^H ; all (source parse, target parse) combinations in the training data $\{(T_{s1}^H, T_{t1}^H), \dots, (T_{sn}^H, T_{tn}^H)\}$; frequencies for each combination $\{F_1, \dots, F_n\}$.

Output: k target parse T_t^H

- 1: **for** $i \in \{1, 2, \dots, N\}$ **do**
- 2: calculate the similarity score S of (T_s^H, T_i^H)
- 3: **end for**
- 4: m parses with highest S with T_s^H : $\{T'_{s1}, \dots, T'_{sm}\}$
- 5: **for** $T'_{si} \in \{T'_{s1}, \dots, T'_{sm}\}$ **do**
- 6: // freq. distribution of possible target parses for T'_{si}
- 7: sample k/m target parses for T'_{si} by distribution
- 8: **end for**

3 AESOP: Adaptive Syntactically-Controlled Paraphrasing

AESOP has two components: i) a retrieval-based module that adaptively selects a set of target syntactic parses to guide the paraphrase generation; ii) an encoder-decoder architecture that leverages BART (Lewis et al., 2020) to generate paraphrases.

3.1 Adaptive Target Syntactic Parse Selection

In AESOP, we propose a retrieval-based strategy to select target syntactic parse adaptively (i.e., Algorithm 1). For a given syntactic parse of source sentence pruned at height H (as shown in Figure 2), denoted as T_s^H , we aim to find k suitable target syntactic parses to guide the generation. First, we collect (source sentence X , paraphrase Z) pairs from the training data. Then, we prune X and Z ’s constituency parse trees at height H simultaneously and get corresponding (T_s^H, T_t^H) pairs. By counting, we have the frequencies of all unique paired combination of pruned source parses with target syntactic parses from their paraphrases, as $\{(T_{s1}^H, T_{t1}^H), \dots, (T_{sn}^H, T_{tn}^H)\}$.

Ranker. For a pruned source parse T_s^H , we calculate the similarity between T_s^H and all other unique parses at height H in the training data $\{T_1^H, \dots, T_i^H, \dots, T_N^H\}$, where N is the number of unique parses pruned at level H . We linearize both T_s^H and T_i^H as constituency parse strings and calculate their similarity score S by calculating weighted ROUGE scores (Lin, 2004) between parse strings:

$$S(T_s^H, T_i^H) = a * \text{ROUGE1} + b * \text{ROUGE2} + c * \text{ROUGEL} \quad (1)$$

Retriever. We rank and get m parses that have the highest similarity scores with T_s^H , denoted as

²Empirically, we use 0.2, 0.3, 0.5 for a, b, c .

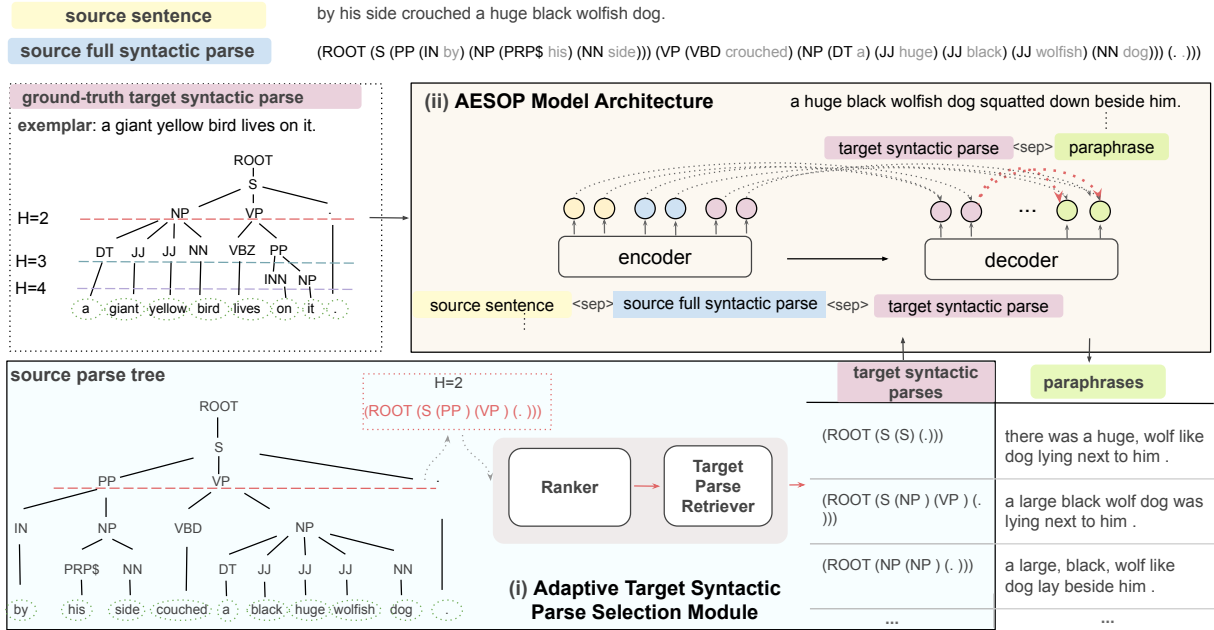


Figure 3: AESOP Framework. With a source sentence as input, AESOP has **i)** a retrieval-based selection module that adaptively chooses a set of target syntactic parses as control signals, together with **ii)** an encoder-decoder architecture to generate fluent paraphrases. With ground-truth target syntactic parses from exemplars, AESOP leverages the syntactic information at different heights from exemplars to guide the generation.

$\{T_{s1}^H, \dots, T_{sm}^H\}$. Then, for each parse T_{si}^H , we retrieve all possible target syntactic parses from pairwise parse combinations from the training data. For each combination, we count how many time it occurs in the training data. For one certain combination with its occurrence frequency $\#(T_{si}^H, T_t^H)$, we divide its frequency over the sum of frequencies for all possible target syntactic parses for T_{si}^H and get a list of frequency ratios. We use the ratio distribution as probabilities to select k/m target syntactic parses T_t^H for each of m parse T_{si}^H as shown in Equation 2, which results in k ($= m * k/m$) target syntactic parses in total.

$$T_t^H \sim P(T_t^H | T_{si}^H) = \frac{\#(T_{si}^H, T_t^H)}{\sum_{j=1}^N \#(T_{si}^H, T_{tj}^H)}. \quad (2)$$

In our later experiments, we use the ranker in Equation 1 to retrieve top-ranked target syntactic parses and their corresponding paraphrases. Using the two-step strategy instead of ranking all syntactic parses based on similarity, we aim to find diverse target syntactic parses suitable for the source sentence. We use the weighted sampling strategy rather than directly choose the most frequently occurred combinations to take care of compatible combinations that occur less in a specific dataset.

3.2 Architecture of AESOP

AESOP takes as inputs the source sentence X , its full syntactic parse T_S and target syntactic parse(s) Y , and generates as outputs a paraphrase Z of X together with a duplication of the target parse Y . Specifically, given source sentences X , we tokenize and get their constituency-based parse trees³, denoted as T_s (shown as *source parse tree* in Figure 3). Similar to previous works (Iyyer et al., 2018; Chen et al., 2019a; Kumar et al., 2020), we linearize the constituency parse tree to a sequence (shown as *source full syntactic parse* in Figure 3).

To utilize the encoder-decoder BART (Lewis et al., 2020) model for syntactic-controlled paraphrase generation, we propose an effective design of having *source sentence*<sep>*source full syntactic parse*<sep>*target syntactic parse* as the input sequence for the encoder. The output sequence from the decoder is the sequence of *target syntactic parse*<sep>*paraphrase*. We will showcase the efficiency of our model design in Section 4 and provide a visual interpretation that AESOP successfully disentangles the semantic and syntactic information in Section 5. During training, we get gold target syntactic parses directly from parallel-annotated paraphrases.

³We used Stanford CoreNLP toolkit (Manning et al., 2014).

| | Model | BLEU \uparrow | ROUGE-1 \uparrow | ROUGE-2 \uparrow | ROUGE-L \uparrow | METEOR \uparrow | TED-R \downarrow | TED-E \downarrow |
|--------------------------------------|--------------------------------------|------------------|--------------------|--------------------|--------------------|-------------------|--------------------|--------------------|
| QQP -Pos | source-as-output | 17.2 | 51.9 | 26.3 | 52.9 | 31.1 | 16.2 | 16.7 |
| | exemplar-as-output | 16.8 | 38.2 | 20.5 | 43.2 | 17.6 | 4.8 | 0.0 |
| | CGEN (Chen et al., 2019a) | 34.9 | 62.6 | 42.7 | 65.4 | 37.4 | 6.7 | 6.0 |
| | SGCP-F (Kumar et al., 2020) | 36.7 | 66.9 | 45.0 | 69.6 | 39.8 | 4.8 | 1.8 |
| | \oplus SGCP-R (Kumar et al., 2020) | 38.0 | 67.6 | 45.3 | 70.0 | 24.8 | 6.6 | 5.7 |
| | AESOP-H2 | 36.8 | 67.1 | 43.8 | 69.0 | 42.2 | 8.0 | 8.6 |
| | \blacklozenge AESOP-H3 | 43.4 | 71.3 | 50.9 | 73.1 | 46.5 | 6.7 | 7.0 |
| | \clubsuit AESOP-H4 | 47.3 | 73.3 | 54.1 | 75.1 | 49.7 | 5.6 | 5.6 |
| | AESOP-F | 40.5 | 69.6 | 49.3 | 72.0 | 43.8 | 4.8 | 1.9 |
| | Para NMT -small | source-as-output | 18.8 | 50.6 | 23.2 | 47.7 | 28.8 | 12.0 |
| exemplar-as-output | | 3.3 | 24.4 | 7.5 | 29.1 | 5.9 | 6.0 | 0.0 |
| CGEN (Chen et al., 2019a) | | 13.6 | 44.8 | 21.0 | 48.3 | 24.8 | 6.7 | 3.3 |
| SGCP-F (Kumar et al., 2020) | | 15.3 | 46.6 | 21.8 | 49.7 | 25.9 | 6.1 | 1.4 |
| \oplus SGCP-R (Kumar et al., 2020) | | 16.4 | 49.4 | 22.9 | 50.3 | 28.8 | 8.7 | 7.0 |
| \blacklozenge AESOP-H2 | | 20.7 | 51.4 | 27.1 | 53.1 | 30.6 | 8.7 | 9.5 |
| AESOP-H3 | | 21.3 | 53.0 | 28.3 | 55.2 | 31.9 | 7.5 | 7.2 |
| \clubsuit AESOP-H4 | | 22.9 | 54.4 | 29.8 | 56.4 | 32.7 | 6.9 | 5.7 |
| AESOP-F | | 20.4 | 52.0 | 27.8 | 55.3 | 30.0 | 6.1 | 1.9 |

Table 1: Performance comparison with ground-truth syntactic control. With coarse syntactic control from shallow height of pruning, \blacklozenge AESOP started to outperform the current state-of-the-art model \oplus SGCP. \clubsuit AESOP-H4 outperforms \oplus SGCP across **all** semantic preservation (BLUE, ROUGE Scores and METEOR) and syntactic conformation metrics (TED-R and TED-E). \uparrow means higher is better, while \downarrow means lower is better. With the full syntactic parse (-F), AESOP achieves its best controllability, which is comparable to previous best performance. *source-as-input* and *exemplar-as-output* are for quality check purpose and not for comparison.

In our setting, we train separate models using pruned trees of target parses at different heights H . During inference, the target syntactic parses are either from exemplar sentences, fixed templates or our adaptive selection module.

4 Paraphrase Generation with Syntactic Control

We train and evaluate AESOP on ParaNMT-small (Chen et al., 2019b) and QQP-Pos (Kumar et al., 2020). Our train/dev/test split follows previous work (Kumar et al., 2020). During our experiments, we aim to answer three research questions:

- **Q1:** Will AESOP conform with the syntactic control while preserving the semantics, given ground-truth target parses? (Section 4.1, Table 1)
- **Q2:** Can AESOP generate fluent paraphrases with the adaptive target parse selection module when ground-truth target parses are unavailable? (Section 4.2, Table 2)
- **Q3:** Does the adaptive selection module produce high-quality target parses? (Section 4.3, Table 3)

Baselines. For supervised models that utilize exemplar sentences to get target parses, we compare with CGEN (Chen et al., 2019a) and two versions

of SGCP (Kumar et al., 2020): SGCP-R and SGCP-F. SGCP prunes constituency parse trees of exemplar sentences from height 3 up to 10. During the evaluation, SGCP-R chooses the best paraphrase out of many, and SGCP-F uses the full parse tree. To the best of our knowledge, SGCP-R is the current state-of-the-art model under this setting. For models that utilize a fixed set of target syntactic parses, we compare with SCPN (Iyyer et al., 2018) that proposes 10 syntactic parses at height 2 to guide the generation.

4.1 Ground-truth Syntactic Control

To answer **Q1**, we evaluate AESOP on both datasets with ground-truth target syntactic parses from exemplar sentences.

Experiment Setup. First, we get the constituency parse trees of exemplar sentences. Then, we remove all leaf nodes (i.e., tokens in the sentences) from the constituency parse trees to prevent any semantics propagating from exemplar sentences into generation. We further prune the parse trees of exemplars at different heights to get different levels of syntactic specifications. Technically, the deeper we prune the parse tree, the more fine-grained syntactic information the model can use. Practically, it is less likely to provide fine-

| | Model | BLEU \uparrow | ROUGE-1 \uparrow | ROUGE-2 \uparrow | ROUGE-L \uparrow | METEOR \uparrow | TED-E@2 \downarrow | Valid@100 \uparrow | Votes \uparrow |
|-----------------------|--------------------|-----------------|--------------------|--------------------|--------------------|-------------------|----------------------|----------------------|------------------|
| QQP -Pos | \oplus SGCP-R | 38.0 | 67.6 | 45.3 | 70.0 | 24.8 | 0.8 | 41.0 | 19.3 |
| | SCPN | 14.9 | 45.9 | 20.9 | 48.1 | 25.4 | 0.7 | 32.0 | 15.3 |
| | AESOP-static | 18.5 | 52.5 | 27.6 | 52.0 | 30.6 | 2.5 | 57.0 | 28.3 |
| | \heartsuit AESOP | 24.6 | 56.2 | 31.5 | 57.6 | 32.8 | 1.1 | 61.0 | 37.0 |
| Para NMT -small | \oplus SGCP-R | 16.4 | 49.4 | 22.9 | 50.3 | 28.8 | 0.7 | 30.0 | 12.0 |
| | SCPN | 12.1 | 35.7 | 15.1 | 32.9 | 23.3 | 0.5 | 54.0 | 30.0 |
| | AESOP-static | 14.4 | 46.0 | 20.5 | 46.5 | 25.5 | 2.9 | 62.0 | 22.0 |
| | \heartsuit AESOP | 15.0 | 47.0 | 21.3 | 47.3 | 26.1 | 2.6 | 68.0 | 36.0 |

Table 2: Performance of AESOP without ground-truth target parse. *Valid@100* is the validity check for the best paraphrases of first 100 test instances, and *Votes* is the percent of received votes for a paraphrase from one model to be the best among 4 models. *Human evaluation* indicates AESOP generate even better-quality paraphrases than the current best model \oplus SGCP that uses the human-annotated target syntactic parse from exemplars.

grained target syntactic parses. For example, it is easy to provide a target syntactic parse at height 2 containing a verb phrase and a noun phrase as (ROOT (S (NP) (VP) (.))), but it is hard to provide more fine-grained syntactic information even for experts. In AESOP, we try to use the syntactic information from exemplar sentences as shallow as possible. We train separate models by using target syntactic parses from pruning the constituency parse tree of paraphrases at heights 2, 3 and 4.⁴ Correspondingly, we denote them as AESOP(-H2/H3/H4). During evaluation, we only use the target syntactic parse from the exemplar sentences at that corresponding height.

Evaluation Metrics. We evaluate the quality of paraphrases with: 1) alignment-based metrics to examine the semantics preservation: including BLEU (Papineni et al., 2002), ROUGE scores (Lin, 2004) and METEOR (Iyer et al., 2016) between the generated paraphrase and gold paraphrase. 2) syntactic conformation metrics: Tree-Edit Distances (TED) scores (Zhang and Shasha, 1989) between the constituency parse trees of generated paraphrases versus exemplar sentences (TED-E) and parallel-annotated paraphrases (TED-R).

Quality Check. We use source sentences and exemplar sentences to check the quality of the datasets in Table 1. Using the source sentences as paraphrases will lead to high semantic preservation scores, but they have distinct syntactic structure with paraphrases, so TED-R scores are poor. On the other hand, exemplar sentences have distinct semantics with both the source sentences and paraphrases, which lead to poor semantic-preservation

metrics. From TED-R scores, we can see that the tree-edit-distance between parse trees of exemplar sentences and paraphrases is low but not 0. It indicates that the quality of such human-annotated exemplar sentences are good yet imperfect.

Experiment Results. Table 1 shows the performance comparison. Unsurprisingly, the deeper we prune target syntactic parse from exemplars, AESOP gets more syntactic information to achieve better controllability. With full target syntactic parse tree, AESOP achieves its best syntactic controllability, which is comparable to previous best performance. On the other hand, AESOP outperforms SGCP-R in semantic-preservation metrics by only using coarse syntactic information from height 2 (AESOP-H2) for ParaNMT-small and height 3 (AESOP-H3) for QQP-Pos. With more syntactic information, AESOP-H4 outperforms the current state-of-the-art SGCP-R in both semantics preservation and syntactic conformation metrics. It showcases AESOP’s great ability of syntactically-controlled paraphrase generation.

4.2 Adaptive Target Parse Selection

To answer Q2, we evaluate AESOP without annotated exemplars. By having SGCP-R in our experiments, we aim to evaluate if AESOP can generate even better paraphrases compared to the current best model with human-annotated exemplars.

Experiment Setup. How to select suitable target syntactic parses to guide the generation is still an open problem in the paraphrase generation community. To fairly compare with SCPN which proposes 10 syntactic templates at height 2, we also adopt AESOP trained at height 2 (shown as AESOP-H2

⁴Implementation details are in Appendix A.1.



| | | <i>top-1</i> | <i>top-3</i> | <i>top-5</i> | <i>top-7</i> | <i>top-10</i> |
|-----------------------|---|---------------------|--------------------|--------------------|--------------------|--------------------|
| QQP -Pos | SCPN | 32.2 (± 7.8) | 32.2 (± 3.3) | 33.4 (± 1.3) | 32.6 (± 0.0) | 33.0 (± 0.0) |
| | AESOP-static | 58.6 (± 4.5) | 58.7 (± 1.8) | 57.5 (± 2.1) | 57.9 (± 0.9) | 58.0 (± 0.0) |
| |  AESOP | 100.0 (± 0.0) | 94.7 (± 0.0) | 90.8 (± 0.0) | 84.3 (± 0.0) | 65.0 (± 0.0) |
| Para NMT -small | SCPN | 16.2 (± 4.1) | 16.9 (± 1.5) | 18.0 (± 1.1) | 17.2 (± 0.9) | 17.4 (± 0.0) |
| | AESOP-static | 47.0 (± 6.2) | 48.9 (± 2.0) | 48.6 (± 1.3) | 48.6 (± 1.3) | 49.0 (± 0.0) |
| |  AESOP | 90.0 (± 0.0) | 86.7 (± 0.0) | 84.4 (± 0.0) | 80.0 (± 0.0) | 70.6 (± 0.0) |

Table 3: Human validity check of *top-k* selected target syntactic parses. All numbers are 10-round mean with standard deviation. In AESOP, we use the ranker in Equation 1 to sort and get *top-k* target parses, while others use random selection. High validity rate of paraphrases indicate the high quality of our retrieved target syntactic parses. The trend that higher-ranked syntactic parses have higher validity rates verifies the efficiency of our ranker.

in Table 1).⁵ Unlike previous work, AESOP uses the adaptive selection module to decide a set of target syntactic parses automatically. For a fair comparison, we also feed the same 10 syntactic target parses from SCPN to AESOP, denoted as AESOP-static. It is hard to evaluate retrieved target syntactic parses because paraphrases are intrinsically diverse, so that many target syntactic parses could be reasonable. Therefore, we use the quality of generated paraphrases, which is our end goal, to reflect the quality of retrieved target syntactic parses. For evaluation, we use automatic metrics together with extensive human evaluations.⁶

Automatic Metrics. First, we generate 10 paraphrases from each model. To establish a strong baseline, we chose the best paraphrase with the highest BLEU scores with source sentences across all models. As shown in Table 2, the improvement from AESOP-static to AESOP indicates the effectiveness of our adaptive selection strategy. SCPN performs better at TED-E@2 metrics on both datasets. After qualitative checks, we share the same finding with previous works (Kumar et al., 2020; Chen et al., 2019a) that SCPN tends to strictly adhere to syntactic parses at the cost of semantics.⁷ On the other hand, AESOP leans towards generating fluent paraphrases and can make up for the case when the target syntactic parse is less reasonable – AESOP achieves a better syntactic conformation when the syntactic control signal is more accurate, indicated by the decreases of TED-E@2 scores in Table 2.

Human Evaluation. We validate the chosen paraphrases for the first 100 instances in the test sets on Amazon Mturk, and report as *Valid@100* in

⁵See experiments with AESOP-H3/H4 in Appendix A.2.

⁶Details of human annotations are in Appendix A.3.

⁷See an example in Appendix A.4.

Table 2. Besides, we show workers 4 paraphrases from all models and ask them to vote for which one is the best. Then we report the percentage of votes that each model got as *votes*. In result, AESOP generates more valid paraphrases than all baselines and gets the most votes, even than SGCP-R that utilizes human-annotated exemplars. Such finding demonstrates the effectiveness of AESOP and points out the importance of studying automatic target parse selection in paraphrase generation.⁸

4.3 Quality of Retrieved Syntactic Parses

To answer Q3, we evaluate the quality of retrieved *top-k* target syntactic parses by checking the validity of their corresponding paraphrases. We generate 10 paraphrases for each of the first 50 test instances (500 in total) using SCPN, AESOP-static, and AESOP and ask workers to validate. After annotation, we use the similarity ranker in Equation 1 to rank and get the *top-k* target syntactic parses and their corresponding paraphrases for AESOP. For other baselines, as they use a fixed set of target syntactic parses and do not have any ranking mechanism, we do random permutation to rank target parses to get *top-k* paraphrases. We run the experiments for 10 rounds and report the validity rate of paraphrases for *top-k* target syntactic parses in Table 3. Comparing to pre-designed syntactic parses, the higher validity rates of paraphrases from AESOP indicate the better quality of our retrieved target syntactic parses. The trend that higher-ranked syntactic parses have higher validity rates also verifies the efficiency of our ranker.

5 Model Analysis and Interpretation

Ablation Studies. We take out each part of sequence in both encoder and decoder and conduct

⁸See a qualitative comparison in Appendix A.5.

| | Model | BLEU \uparrow | ROUGE-1 \uparrow | ROUGE-2 \uparrow | ROUGE-L \uparrow | METEOR \uparrow | TED-R \downarrow | TED-E \downarrow |
|-----------------------|--|-----------------|--------------------|--------------------|--------------------|-------------------|--------------------|--------------------|
| QQP -Pos | 1 AESOP | 47.3 | 73.3 | 54.1 | 75.1 | 49.7 | 5.6 | 5.6 |
| | 2 w/o <i>tp</i> in dec | 39.9 (+7.4) | 68.4 (+4.9) | 49.0 (+5.1) | 70.5 (+4.6) | 44.5 (+5.2) | 8.1 (+2.5) | 8.1 (+2.5) |
| | 3 w/o <i>fp</i> in enc | 42.3 (+5.0) | 71.6 (+1.7) | 50.9 (+3.2) | 73.4 (+1.7) | 45.3 (+4.4) | 6.4 (+0.9) | 6.2 (+0.6) |
| | 4 w/o <i>fp</i>, <i>tp</i> in enc, <i>tp</i> in dec | 23.9 (+23.4) | 56.2 (+17.1) | 32.2 (+21.9) | 57.6 (+17.5) | 34.0 (+15.7) | 12.9 (+7.3) | 13.4 (+7.8) |
| | 5 w/o <i>fp</i> in enc, <i>tp</i> in dec | 38.2 (+9.1) | 67.7 (+5.6) | 47.5 (+6.6) | 70.0 (5.1) | 42.4 (+7.3) | 8.0 (+2.4) | 7.9 (+2.3) |
| Para NMT -small | 1 AESOP | 22.9 | 54.4 | 29.8 | 56.4 | 32.7 | 6.9 | 5.7 |
| | 2 w/o <i>tp</i> in dec | 19.2 (+3.7) | 51.3 (+3.1) | 27.3 (+2.5) | 53.5 (+2.9) | 30.8 (+1.9) | 9.7 (+1.8) | 8.8 (+2.9) |
| | 3 w/o <i>fp</i> in enc | 24.0 (-1.1) | 54.8 (-0.4) | 30.5 (-0.7) | 57.1 (-0.7) | 33.4 (-0.7) | 6.8 (-0.1) | 5.7 (0.0) |
| | 4 w/o <i>fp</i>, <i>tp</i> in enc, <i>tp</i> in dec | 16.7 (+6.2) | 49.8 (+0.6) | 25.2 (+4.6) | 50.4 (+6.0) | 29.1 (+3.6) | 11.7 (+4.8) | 12.8 (7.1) |
| | 5 w/o <i>fp</i> in enc, <i>tp</i> in dec | 20.0 (+2.9) | 53.7 (+0.7) | 29.3 (+0.5) | 55.7 (+0.7) | 31.6 (+1.1) | 8.7 (+1.8) | 7.7 (+2.0) |

Table 4: Ablation studies that justify our model design. + shows how much better AESOP is compared to the that design, while - shows how much worse (*dec*, *enc*: decoder, encoder. *tp*: target parse, *fp*: source full parse).

several ablation studies on AESOP-H4 with exemplars. We show how each part of sequences would influence AESOP’s performance in Table 4. Takeaways from our ablation studies are: 1) AESOP’s performance plummets without any syntactic specifications (row1&row4). 2) Taking out target parse (*tp*) in the output sequence will lead to worse performance in both semantic preservation and syntactic controllability (row1&row2, row3&row4). We will visually interpret the benefit of such design later in this section. 3) Taking out each part in the input sequence for the encoder will leads to a significant performance drop of AESOP on QQP-Pos dataset for both criteria (i.e., semantic preservation and syntactic controllability). The trend is the same for ParaNMT-small dataset, except only taking out the full parse (*fp*) will leads to around 1% improvement on semantic preservation metrics, while the syntactic controllability stays almost the same. Considering the much larger performance drop on criteria, we decided the current design of AESOP.

Interpretation. In Figure 4, we visualize cross attentions between encoder and decoder for two designs, i.e., AESOP with (right) and without (left) target syntactic parse in the decoder on the test set of ParaNMT-small. Technically, we search for the final output with *beam* = 4 and take the average of cross attention scores of 12 attention heads from the last layer of the decoder. Finally, we add the attention of all tokens within each component (*ss*, *fp* and *tp*). To manifest the difference, we denote the highest attention scores as 100, and calculate the relative cross attention to the highest.

Compared to the design without target syntactic parse in the decoder, cross attention between paraphrases and source sentences stays the highest in AESOP. However, the ratio of cross attention

scores of (paraphrases, target parses) and (paraphrases, full source parses) decreases. Such decreases indicate that having target parses in the decoder helps to disentangle semantic and syntactic information from the input sequence. Instead, AESOP learns the syntactic information from target syntactic parses through self-attention in the decoder. As a result, it leads to a performance boost in Table 4. At the same time, target parses influence paraphrase generation directly during decoding through the decoder’s self-attention, which leads to better controllability of AESOP. Take the example in Figure 4, without target parse in the decoder, the model outputs *a large black dog sits in the corner beside him.* as the paraphrase to *by his side crouched a huge black wolfish dog ..* After adding the target parse in the decoder, the model no longer generates prepositional phrase *in the corner* and outputs *a large black dog sits beside him.*, which matches better with the input target parse.

6 Improve Robustness

Recent works show that powerful LMs (e.g., BERT (Devlin et al., 2019)) are capturing the superficial lexical features McCoy et al. (2019) and are vulnerable to simple perturbations (Jin et al., 2020). Motivated by this, we first test if BERT is robust to syntactic perturbations by paraphrasing.

We fine-tune BERT models on two GLUE (Wang et al., 2018) tasks (SST-2 and RTE). Then, we generate 10 paraphrases using AESOP-H2 for each test instance in the dev set and choose *top-5* to get 2 larger dev sets.⁹ We run trained BERT models on new dev sets again.

Human Annotation. We collect the paraphrases where models fail but succeeded at their original

⁹As we do not have the label for test sets on GLUE.

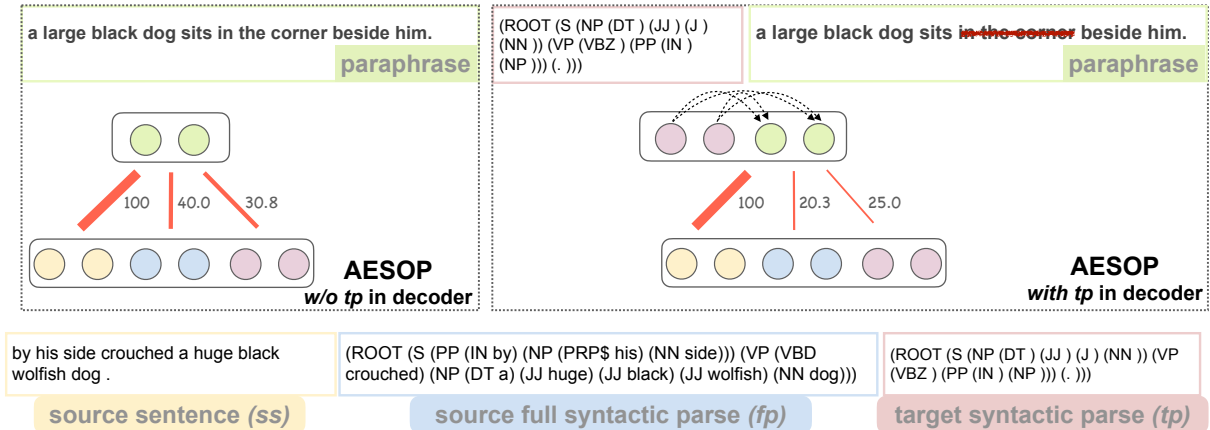


Figure 4: Cross Attention without and with tp (target parse) in the decoder. Line thickness is proportional to relative cross attention scores. By duplicating tp in the decoder, relative cross attention scores for both (paraphrases, full source parse) and (paraphrases, target parse) decrease. It indicates that duplicating target syntactic parses in the decoder lets AESOP disentangle the semantics and syntactic information from the input sequence.

| Dataset | Model | Original Dev | | | Collected | | | Combined | | |
|---------|-------------------------------|--------------|-------------|-------------|-----------|-------------|--------------|----------|-------------|-------------|
| | | Before | After | ParaGAP | Before | After | ParaGAP | Before | After | ParaGAP |
| SST-2 | SCPN (Iyyer et al., 2018) | | 89.7 | -2.2 | | 46.5 | +27.9 | | 76.1 | +8.1 |
| | SynPG (Huang and Chang, 2021) | 91.9 | 85.3 | -6.6 | 18.6 | 47.0 | +28.7 | 68.0 | 73.3 | +5.3 |
| | AESOP-tp | | 88.9 | -3.0 | | 49.5 | +30.9 | | 76.4 | +8.4 |
| | 🏆AESOP | | 91.1 | -0.8 | | 48.5 | +29.9 | | 77.6 | +9.6 |
| RTE | SCPN (Iyyer et al., 2018) | | | 68.6 | | +5.8 | | | 49.6 | +2.7 |
| RTE | SynPG (Huang and Chang, 2021) | 62.8 | 61.7 | -1.1 | 46.9 | 49.0 | +2.1 | 56.0 | 56.6 | +0.6 |
| | AESOP-tp | | 60.3 | -2.5 | | 55.7 | +8.8 | | 57.8 | +1.8 |
| | 🏆AESOP | | 62.5 | -0.3 | | 58.4 | +11.5 | | 61.0 | +5.0 |

Table 5: ParaGAP is the accuracy difference between BERT models after and before using paraphrases augment the training data. Among 4 models, AESOP improves BERT’s robustness to syntactic perturbations the most.

sentence as adversarial examples. We then put all these examples on MTurk and ask workers to re-annotate.¹⁰ For SST-2, we ask workers to assign sentiment labels as positive, negative or undecided (mixed sentiments). For RTE, one test instance has sentence1 and sentence2 with a label if sentence1 entails sentence2. We generate paraphrases for sentence2 and ask workers to binary-decide if sentence1 entails generated paraphrases. We show the statistics of collected adversarial set and original dev set in Table 6. Researchers can test their models’ robustness to syntactic perturbations on our collected datasets.

Augmentation. We augment each training instance with 5 best paraphrases from AESOP-H2. For SynPG and SCPN, as the pre-designed templates for SynPG is a subset of SCPN’s. We generate 5 paraphrases using selected templates in SynPG. Then, we retrain BERT models with augmented training data from each model. Then, we re-

¹⁰See more annotation details in Appendix A.6.

| | Original | Collected | Combined |
|-------|----------|-----------|----------|
| SST-2 | 872 | 404 | 1276 |
| RTE | 277 | 341 | 618 |

Table 6: Dataset statistics. *Combined* is the combination of the original dev set and collected data.

train BERT models after augmentation and get their test accuracies. We define *ParaGAP* as the accuracy difference for after- and before-augmentation using paraphrase generation models. ParaGAP indicates how efficient the augmentation is to improve the model robustness to syntactic perturbations.

Experiment Result. As shown in Table 5, BERT models perform poorly in our collected datasets before augmentation, which indicate that our collected adversarial datasets are challenging, and BERT is vulnerable to syntactic perturbations. After using 4 different paraphrasing models to augment the training data, models’ robustness to such perturbations all get improved. Among all models,

AESOP yields the best ParaGAP on the combined dataset of original dev sets and collected datasets, which shows that using AESOP improves the classification model’s robustness to syntactic perturbations more effectively.¹¹

7 Related Work

Recent advances have been using neural models for syntactically controlled paraphrase generation. From the **modeling** perspective, there are roughly two categories: unsupervised and supervised methods. Unsupervised models do not use parallel paraphrases during training. [Wieting and Gimpel \(2018\)](#); [Wieting et al. \(2017\)](#) use back-translation to generate paraphrases. [Huang and Chang \(2021\)](#) propose a transformer-based model SynPG for paraphrase generation. AESOP is a supervised paraphrase generation model, which means that we require parallel paraphrases during training. Previous supervised paraphrase models are mostly RNN-based models, including SCPN ([Iyyer et al., 2018](#)), CGEN ([Chen et al., 2019a](#)) and SGCP ([Kumar et al., 2020](#)). Such models suffer from generating long sentences and do not utilize the power of recent pretrained language models. [Goyal and Durrett \(2020a\)](#) is a concurrent work with ours that also builds on BART to generate paraphrases but has a different model design. For **syntactic control**, [Goyal and Durrett \(2020b\)](#) use target syntactic parses to reorder source sentences to guide the generation, while other works, including AESOP, directly use target syntactic parses to guide the generation. CGEN ([Chen et al., 2019a](#)) and SGCP ([Kumar et al., 2020](#)) use target syntactic parses from crowd-sourced exemplars, SCPN ([Iyyer et al., 2018](#)) and SynPG ([Huang and Chang, 2021](#)) use pre-designed templates, while AESOP retrieves target syntactic parses automatically.

8 Conclusion and Future Works

In this work, we propose AESOP for paraphrase generation with adaptive syntactic control. One interesting and surprising finding of this paper is that using automatically retrieved parses to control paraphrase generation can result in better qualities than the current best model using human-annotated exemplars. Such finding manifests the benefits of adaptive target parse selection for controlled paraphrase generation – it does not only generate

diverse paraphrases, but also higher quality paraphrases. This suggests future works on syntactically controlled paraphrase generation to pay more attention to target parse selection, and we hope AESOP can serve as a strong baseline for this direction. In our work, we use generated paraphrases to reflect the quality of automatically-selected target parses; future works can design specific metrics to evaluate the quality of retrieved syntactic parses. In addition, we find that having the control signal in the decoder can lead to better controllability of AESOP. Future works can test the generalizability of this modeling strategy in other controlled generation tasks. In addition, we show that AESOP can effectively attack classification models and contribute two datasets to test models’ robustness to syntactic perturbation. We find that using AESOP to augment training data can effectively improve classification models’ robustness to syntactic perturbations.

Acknowledgments

Many thanks to I-Hung Hsu for his constructive suggestion and fruitful discussion for AESOP. We thank Kuan-Hao Huang, Sarik Ghazarian, Yu Hou and anonymous reviewers for their great feedback to improve our work.

Ethical Consideration

Our proposed model AESOP utilizes a pretrained language model to generate paraphrases. Trained on massive online texts, it is well-known that such pretrained language models could capture the bias reflecting the training data. Therefore, AESOP could potentially generate offensive or biased content. We suggest interested parties carefully check the generated content before deploying AESOP in any real-world applications. Note that AESOP might be used for malicious purposes because it does not have a filtering mechanism that checks the toxicity, bias, or offensiveness of source sentences from the input. Therefore, AESOP can generate paraphrases for harmful content that may offend certain groups or individuals.

Our collected datasets are based on the development sets of two public classification tasks on GLUE, including SST-2 for sentiment analysis and RTE for textual entailment. These do not contain any explicit detail that leaks information about a user’s name, health, racial or ethnic origin, religious or philosophical affiliation or beliefs.

¹¹We show that AESOP helps to improve models’ decision boundaries in Appendix A.7.

References

- Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019a. Controllable paraphrase generation with a syntactic exemplar. *Association for Computational Linguistics*.
- Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019b. A multi-task approach for disentangling syntax and semantics in sentence representations. *Association for Computational Linguistics*.
- 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)*. Association for Computational Linguistics.
- Silin Gao, Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020. Paraphrase augmented task-oriented dialog generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Tanya Goyal and Greg Durrett. 2020a. Neural syntactic reordering for controlled paraphrase generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Tanya Goyal and Greg Durrett. 2020b. Neural syntactic reordering for controlled paraphrase generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.
- James Y. Huang, Kuan-Hao Huang, and Kai-Wei Chang. 2021. Disentangling semantics and syntax in sentence embeddings with pre-trained language models. In *NAACL (short)*.
- Kuan-Hao Huang and Kai-Wei Chang. 2021. Generating syntactically controlled paraphrases without using annotated parallel pairs. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics.
- Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, and Luke Zettlemoyer. 2016. Summarizing source code using a neural attention model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.
- Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics.
- Zhengbao Jiang, F. F. Xu, J. Araki, and Graham Neubig. 2019. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.
- Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *AAAI*.
- A. Kumar, Kabir Ahuja, Raghuram Vadapalli, and P. Talukdar. 2020. Syntax-guided controlled generation of paraphrases. *Transactions of the Association for Computational Linguistics*, 8:330–345.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics.
- Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics.
- Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.
- M. McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia Medica*, 22:276 – 282.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Yufei Tian, Arvind krishna Sridhar, and Nanyun Peng. 2021. Hypogen: Hyperbole generation with commonsense and counterfactual knowledge. In *Findings of the Association for Computational Linguistics: EMNLP*.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018.

- GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics.
- John Wieting and Kevin Gimpel. 2018. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.
- John Wieting, Jonathan Mallinson, and Kevin Gimpel. 2017. Learning paraphrastic sentence embeddings from back-translated bitext. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics.
- Xuwen Yang, Yingru Liu, Dongliang Xie, Xin Wang, and Niranjan Balasubramanian. 2019. Latent part-of-speech sequences for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics.
- Xiaojing Yu and Anxiao Jiang. 2021. Expanding, retrieving and infilling: Diversifying cross-domain question generation with flexible templates. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics.
- K. Zhang and D. Shasha. 1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM J. Comput.*, 18:1245–1262.

A Appendix

A.1 Implementation Details

Parameters. We use a learning rate $3 \times e^{-5}$ to train AESOP. We use 6 layers of encoder and 6 layers of decoder with model dimension of 768 and 12 heads. For the input sequence, we set the max length to 128 and max output sequence length as 62. We train 25 epochs for each model. It takes about one days to finish training for ParaNMT-small and about half a day for QQP-Pos on one NVIDIA GeForce RTX 2080.

Optimization. We use Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$) with a linear learning rate decay schedule for optimization. All experiments are done using Huggingface library (Wolf et al., 2020).¹²

A.2 Diverse Syntax with Deeper Pruning

Table 7 is a supplementary to Table 2. Using AESOP-H2 yields a better performance in terms of the semantic preservation metrics. We share the same finding from Section 4.1 that the syntactic controllability will get better when we use the deeper heights of syntactic parse trees. However, the semantic preservation metrics get worse with more fine-grained syntactic control, we hypothesize this is because deeper-level of control signals can be misleading, but such signals restrict models to generate paraphrases that conform to the provided misleading syntactic signals, which impairs the ability of pretrained language models to generate fluent texts.

A.3 Validity Check on Paraphrases

In Section A.3.1, we will give more details of the human validity check in Table 2 and more details of human evaluation of Table 3 in Section A.3.2.

A.3.1 Validity@100 and Votes

We choose the best paraphrases among 10 generated paraphrases from SCPN, AESOP-static and AESOP for the first 100 test instances in the both datasets. For SGCP, we take its output paraphrases that uses the exemplar sentences. Then, we perform the human validity check of these 400 paraphrases on Amazon Mturk platform. For each source sentence, we provide all 4 paraphrases from these four models to three workers. In our instruction, we ask them to annotate three-level of validity: invalid paraphrase, imperfect paraphrase that does

not lose key information, and perfect paraphrases. We binarize worker’s labels with both imperfect and perfect paraphrases as a valid instance, otherwise invalid. Then, we the majority vote of labels among three workers as the final label. We calculate the ratio of valid instances over 100 and report the ratio as *Validity@100* in Table 2. As a supplementary, Table 8 shows the breakdown annotation of three-level validity check. In addition, we ask workers to vote for the best paraphrase among the four paraphrases, and report the ratio of total votes the model gets over all 300 votes as *Votes* in Table 2 to reduce the influence of personal preference. We use fleiss’s kappa scores (McHugh, 2012) to measure the Inter Annotator Agreement (IAA). The IAA for *validity@100* is 0.63, which indicates a substantial agreement among workers.

Mturk Setup Details. We set the qualification as the location needs to be in the US, completed HITs no less than 5000, and approval rate no lower than 98%. Our one HIT contains 10 instances. For one HIT, we have three respondents (workers) to work on it. For payment, we pay workers \$0.8 per HIT with a potential bonus of \$1 if they participate over 5 HITs published by us.

A.3.2 Validity@500

The annotators of the human evaluation in Section 4.3 are three graduate students from our institute. None of them are involved in this project. We have two of them work on validity checks for ParaNMT-small and QQP-Pos, and there was one student who worked on both. We check their understandings about paraphrases before the study and instruct them to only label a paraphrase as valid when the paraphrase is natural, fluent, and preserves the semantics of the source sentence. To understand the Inter Annotator Agreement (IAA), we randomly selected 50 samples of (source sentence, paraphrase) pairs and asked them to annotate if they are valid paraphrases independently. After the annotation, we count it as an agreement if they agree on the same label (either valid paraphrase or invalid). The average IAA is 0.9 between the three of them, which indicates a good agreement. Then, we have these three works to annotate all instances sampled on Table 2. After annotation, we count a paraphrase as a valid paraphrase only if both of 2 annotators think it is valid.

¹²<https://huggingface.co/>.

| | Model | BLUE | ROUGE-1 | ROUGE-2 | ROUGE-L | METEOR | TED-E |
|-------------------|-----------------|------|---------|---------|---------|--------|-------|
| ParaNMT -small | AESOP-H2 | 15.0 | 47.0 | 21.3 | 47.3 | 26.1 | 2.6 |
| | AESOP-H3 | 12.1 | 41.5 | 16.7 | 42.6 | 22.8 | 2.3 |
| | AESOP-H4 | 8.4 | 33.2 | 10.9 | 35.3 | 18.3 | 1.5 |
| QQP-Pos | AESOP-H2 | 24.6 | 56.2 | 31.5 | 57.6 | 32.8 | 0.7 |
| | AESOP-H3 | 22.5 | 54.8 | 29.7 | 56.1 | 31.5 | 0.8 |
| | AESOP-H4 | 19.9 | 51.4 | 25.9 | 52.0 | 30.6 | 1.1 |

Table 7: A supplementary to Table 2. When we use the deeper levels of syntactic parse trees, the syntactic controllability of AESOP will get better. However, the semantic preservation metrics get worse, because such signals can be misleading and it restricts models to generate paraphrases that conform to the control signal.

| Dataset | Model | Invalid | Imperfect | Perfect |
|------------------------|--------------|---------|-----------|---------|
| QQP -Pos | SGCP | 59 | 19 | 22 |
| | SCPN | 68 | 12 | 20 |
| | AESOP-static | 43 | 23 | 34 |
| | AESOP | 39 | 24 | 37 |
| Para -NMT -small | SGCP | 70 | 21 | 9 |
| | SCPN | 46 | 24 | 30 |
| | AESOP-static | 38 | 20 | 42 |
| | AESOP | 32 | 27 | 41 |

Table 8: Three-level validity annotation breakdowns for *Validity@100*.

A.4 Case Study with Invalid Target Syntax

Strict conformation to inappropriate target syntactic parse sometimes leads to semantics lost and abrupt termination of sentences, which hurts the goal of generating fluent and natural paraphrases as indicated in Section 4.2. For example, given the input sentence *i had a dream yesterday and it was about you* and a target syntactic parse with height 2 ($ROOT(S(ADVP)(NP)(VP)(.))$), SCPN generates *maybe it was about you*. that has the same syntactic parse with the target parse, while AESOP generates *you were in my dream last night*. whose syntax parse at height 2 is ($ROOT(S(NP)(VP)(.))$).

A.5 Qualitative Comparison

We provide a qualitative comparison between AESOP and other competitive paraphrase generation models under both settings with or without exemplar sentences in Table 9. We show that with ground-truth syntactic control (Setting I), AESOP can generate paraphrases that are closer to ground-truth paraphrases. Without ground truth, AESOP can generate diverse paraphrases that are more natural and better preserve the semantics than SCPN.

A.6 Adversarial Set Collection

We contribute two datasets constructed from AESOP in Section 6 by crowd-sourcing. We collect all adversarial examples that successfully attacked

the models, as shown in the *all* column of Table 5 and put them on Amazon MTurk to annotate if the paraphrases are valid. We set the qualification as the location needs to be in the US, completed HITs no less than 5000, and approval rate no lower than 98%. One HIT contains 12 instances and have 3 respondents (workers) work on it. For payment, we pay workers \$0.4 per HIT as qualification test. After selecting qualified workers, we pay them \$1 per HIT with another potential bonus of \$1 if they participate over 5 HITs published by us. On average, experienced workers spent around 10 minutes to complete one HIT, which means our payment is above the federal minimum wage in the US.

Instruction and Annotation. As sentiment analysis on SST-2 is intuitive, we list examples as an instruction to guide the annotation. We count it as an agreement if all of three workers given the same label to one instance (i.e., positive, negative or undecided), and we calculate IAA as the ratio of agreements over all instances for qualified workers. The average IAA of three workers among all instances are 0.8, which indicates a good agreement. During the dataset collection, we use the majority vote to decide the final label of one instance. For textual entailment on RTE dataset, we refer to the guideline from the original guide line of RTE-4¹³ to explain the textual entailment task itself with examples. The IAA for RTE annotation is 0.71.

A.7 AESOP Helps to Improve the Decision Boundary

We conduct a study on how augmenting the training data would influence models’ decision boundaries. More specifically, we test BERT models before and after augmentation with AESOP, on the combination of the original gold test set and our collected adversarial datasets on two downstream tasks. For

¹³<https://tac.nist.gov/2008/rte/rte.08.guidelines.html>

| | Model | Exemplar/Selected Target Parses | Generated Paraphrases |
|-----------------------|-------|---|--|
| | | source sentence: what is the best way to get manchester united tickets ? | |
| QQP -Pos | SGCP | (ROOT (FRAG (NP (NP (NNS)) (NP (DT) (NN))) (.))) | whats the way ? how can i get free manchester united tickets ? |
| | SCPN | (ROOT (S (NP) (VP) (.)) (ROOT (FRAG (SBAR) (.))) | that 's the best way to get manchester united tickets ? what 's the best way to get manchester ? |
| | AESOP | (ROOT (SBARQ (WHADVP) (SQ) (.))) (ROOT (SQ (VBZ) (NP) (VP) (.))) | how can i get free manchester united tickets ? is there any way to get free manchester united tickets ? |
| | | source sentence: by his side crouched a huge black wolfish dog . | |
| Para NMT -small | SGCP | (ROOT (S (NP (DT) (JJ) (JJ) (NN)) (VP (VBZ) (PP (IN) (NP))) (.))) | his side waving a huge black dog . a large black dog sits beside him . |
| | SCPN | (ROOT (S (NP) (VP) (.)) (ROOT (NP (NP) (.))) | his side was a huge black dog . a huge black dog on his side . |
| | AESOP | (ROOT (S (S) (NP) (VP) (.))) (ROOT (NP (NP) (.))) | there was a big black wolf lying next to him . a large , black , wolf like dog lay beside him . |

Table 9: A qualitative comparison of generated paraphrases with or without exemplar sentences from AESOP. SGCP and AESOP-H4 use target syntactic parses from exemplar sentences to guide the generation. SCPN use fixed target syntactic templates, while AESOP retrieves target syntactic parses automatically.

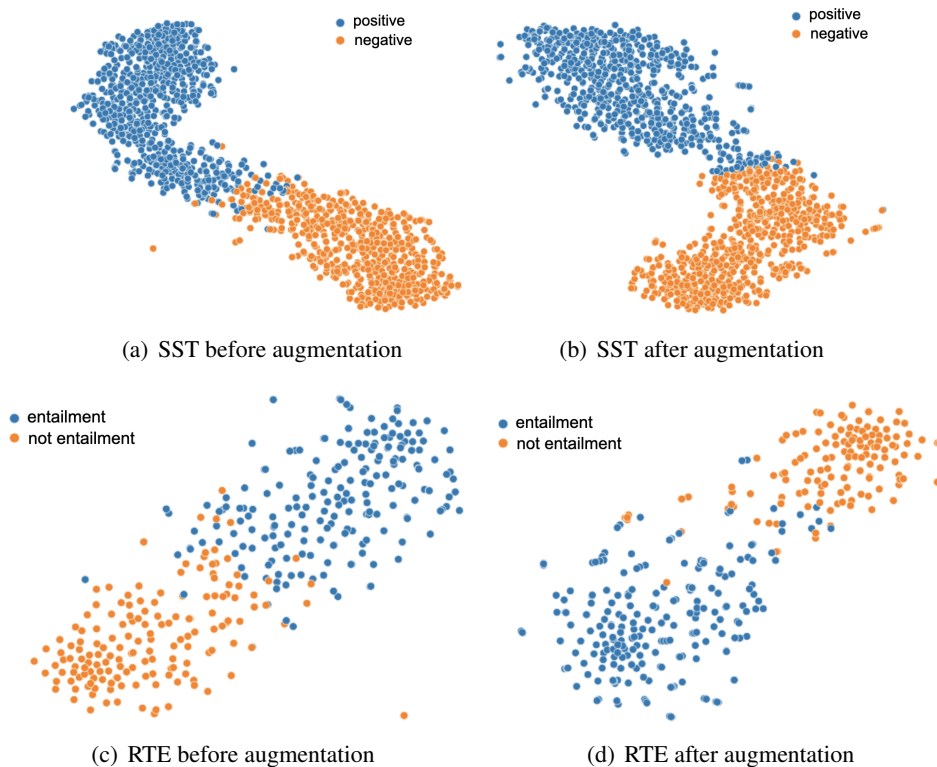


Figure 5: AESOP helps to improve the model decision boundary. For visualization, we use TSNE to reduce the dimension of **[CLS]** token from the last layer of BERT model combining the collected data and dev set for SST-2 and RTE.

visualization, we use TSNE to reduce the dimension of **[CLS]** token from the last layer of BERT model. Figure 5 show that AESOP helps BERT models to improve the decision boundary to be more clear, which is also indicated by Table 5 in the main content.