Bi-SimCut: A Simple Strategy for Boosting Neural Machine Translation

Anonymous ACL submission

Abstract

We introduce Bi-SimCut: a simple but effective strategy to boost neural machine translation (NMT) performance. It consists of two training procedures: bidirectional pretraining and unidirectional finetuning. Both procedures utilize SimCut, a simple regularization method that forces the consistency between the output distributions of the original and the cutoff samples. Without utilizing extra dataset via back-translation or integrating large-scale pretrained model, Bi-SimCut achieves strong translation performance across five translation benchmarks (data sizes range from 160K to 20.1M): BLEU scores of 31.16 for `en` → `de` and 38.37 for `de` → `en` on the IWSLT14 dataset, 30.78 for `en` → `de` and 35.15 for `de` → `en` on the WMT14 dataset, and 27.17 for `zh` → `en` on the WMT17 dataset. SimCut is not a new method, but a version of Cutoff (Shen et al., 2020) simplified and adapted for NMT, and it could be considered as a perturbation-based method. Given the universality and simplicity of Bi-SimCut and SimCut, we believe they can serve as strong baselines for future NMT research.

1 Introduction

The state of the art in machine translation has been dramatically improved over the past decade thanks to the neural machine translation (NMT) (Wu et al., 2016), and transformer-based models (Vaswani et al., 2017) often deliver state-of-the-art performance with large-scale corpora (Ott et al., 2018). Along with the development in the NMT field, consistency training has been widely adopted and shown great promise to improve NMT performance. It simply regularizes NMT model predictions to be invariant to either small perturbations applied to the inputs (Sano et al., 2019; Shen et al., 2020) and hidden states (Chen et al., 2021) or the model randomness and variance existed in the training procedure (Liang et al., 2021).

Specifically, Shen et al. (2020) introduced a set of cutoff data augmentation methods and utilized Jensen-Shannon (JS) divergence loss to force the consistency between the output distributions of the original and the cutoff augmented samples in the training procedure. Despite its impressive performance, finding the proper values for the four additional hyper-parameters introduced in cutoff augmentation seems to be time-consuming if there are limited resources available, which hinders its practical value in the NMT field.

In this paper, our main goal is to provide a simple, easy-to-reproduce, but tough-to-beat strategy for training NMT models. Inspired by cutoff augmentation (Shen et al., 2020) and virtual adversarial regularization (Sano et al., 2019) for NMT, we firstly introduce a simple yet effective regularization method named SimCut. Technically, SimCut is not a new method and can be viewed as a simplified version of Token Cutoff proposed in Shen et al. (2020). We show that bidirectional backpropagation in Kullback-Leibler (KL) regularization plays a key role in improving NMT performance. We also regard SimCut as a perturbation-based method and discuss its robustness to the noisy inputs. At last, we present Bi-SimCut, a two-stage training strategy consisting of bidirectional pretraining and unidirectional finetuning equipped with SimCut regularization.

The contributions of this paper can be summarized as follows:

- We propose a simple but effective regularization method, SimCut, for improving the generalization of NMT models. SimCut could be regarded as a perturbation-based method and serves as a strong baseline for the methods of perturbations.

- We propose Bi-SimCut, a training strategy for NMT that consists of bidirectional pretraining and unidirectional finetuning with SimCut regularization.

- Our experimental results show that NMT training with Bi-SimCut achieves significant improvements over the transformer model on five translation benchmarks (data sizes range from 160K to 20.1M), and outperforms the current state-of-the-art method BiBERT (Xu et al., 2021) on several benchmarks.

2 Background

2.1 Neural Machine Translation

The NMT model refers to a neural network with an encoder-decoder architecture, which receives a sentence as input and returns a corresponding translated sentence as output. Assume $\mathbf{x} = x_1, ..., x_I$ and $\mathbf{y} = y_1, ..., y_J$ that correspond to the source and target sentences with lengths I and J respectively. Note that y_J denotes the special end-of-sentence symbol $\langle eos \rangle$. The encoder first maps a source sentence $\mathbf{x}$ into a sequence of word embeddings $e(\mathbf{x}) = e(x_1), ..., e(x_I)$, where $e(\mathbf{x}) \in \mathbb{R}^{d \times I}$, and d is the embedding dimension. The word embeddings are then encoded to the corresponding hidden representations $\mathbf{h}$. Similarly, the decoder maps a shifted copy of the target sentence $\mathbf{y}$, i.e., $\langle bos \rangle, y_1, ..., y_{J-1}$, into a sequence of word embeddings $e(\mathbf{y}) = e(\langle bos \rangle), e(y_1), ..., e(y_{J-1})$, where $\langle bos \rangle$ denotes a special beginning-of-sentence symbol, and $e(\mathbf{y}) \in \mathbb{R}^{d \times J}$. The decoder then acts as a conditional language model that operates on the word embeddings $e(\mathbf{y})$ and the hidden representations $\mathbf{h}$ learned by the encoder.

Given a parallel corpus $\mathcal{S} = \{\mathbf{x}^i, \mathbf{y}^i\}_{i=1}^{|\mathcal{S}|}$, the standard training objective is to minimize the empirical risk:

$$\mathcal{L}_{ce}(\theta) = \mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{y}) \in \mathcal{S}} [\ell(f(\mathbf{x}, \mathbf{y}; \theta), \ddot{\mathbf{y}})], \quad (1)$$

where ℓ denotes the cross-entropy loss, θ is a set of model parameters, $f(\mathbf{x}, \mathbf{y}; \theta)$ is a sequence of probability predictions, i.e.,

$$f_j(\mathbf{x}, \mathbf{y}; \theta) = P(y|\mathbf{x}, \mathbf{y}_{<j}; \theta), \quad (2)$$

and $\ddot{\mathbf{y}}$ is a sequence of one-hot label vectors for $\mathbf{y}$.

2.2 Cutoff Augmentation

Shen et al. (2020) introduced a set of cutoff methods which augments the training by creating the partial views of the original sentence pairs and proposed Token Cutoff for the machine translation task. Given a sentence pair $(\mathbf{x}, \mathbf{y})$, N cutoff samples $\{\mathbf{x}^i_{\text{cut}}, \mathbf{y}^i_{\text{cut}}\}_{i=1}^N$ are constructed by randomly setting the word embeddings of $x_1, ..., x_I$ and $y_1, ..., y_J$ to be zero with a cutoff probability p_{cut}. For each sentence pair, the training objective of Token Cutoff is then defined as:

$$\mathcal{L}_{tokcut}(\theta) = \mathcal{L}_{ce}(\theta) + \alpha \mathcal{L}_{cut}(\theta) + \beta \mathcal{L}_{kl}(\theta), \quad (3)$$

where

$$\mathcal{L}_{ce}(\theta) = \ell(f(\mathbf{x}, \mathbf{y}; \theta), \ddot{\mathbf{y}}),$$

$$\mathcal{L}_{cut}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \ell(f(\mathbf{x}^i_{\text{cut}}, \mathbf{y}^i_{\text{cut}}; \theta), \ddot{\mathbf{y}}),$$

$$\mathcal{L}_{kl}(\theta) = \frac{1}{N+1} \{\sum_{i=1}^{N} \text{KL}(f(\mathbf{x}^i_{\text{cut}}, \mathbf{y}^i_{\text{cut}}; \theta) \| p_{\text{avg}}) + \text{KL}(f(\mathbf{x}, \mathbf{y}; \theta) \| p_{\text{avg}})\},$$

$$p_{\text{avg}} = \frac{1}{N+1} \{\sum_{i=1}^{N} f(\mathbf{x}^i_{\text{cut}}, \mathbf{y}^i_{\text{cut}}; \theta) + f(\mathbf{x}, \mathbf{y}; \theta)\},$$

in which $\text{KL}(\cdot \| \cdot)$ denotes the Kullback-Leibler (KL) divergence of two distributions, and α and β are the scalar hyper-parameters that balance $\mathcal{L}_{ce}(\theta)$, $\mathcal{L}_{cut}(\theta)$ and $\mathcal{L}_{kl}(\theta)$.

3 Datasets and Baseline Settings

In this section, we describe the datasets used in experiments as well as the model configurations. For fair comparisons, we keep our experimental settings consistent with previous works.

	IWSLT	WMT	
	en↔de	en↔de	zh→en
train	160239	4468840	20184941
valid	7283	6003	2002
test	6750	3003	2001

Table 1: Number of sentence pairs used in our machine translation experiments.

Datasets We initially consider a low-resource (IWSLT14 en↔de) scenario and then show further experiments in standard (WMT14 en↔de) and high (WMT17 zh→en) resource scenarios in Sections 5 and 6. The detailed information of the datasets are summarized in Table 1. We here conduct experiments on the IWSLT14 English-German dataset, which has 160K parallel bilingual sentence pairs. Following the common practice, we lowercase all words in the dataset. We build a shared dictionary with 10K byte-pair-encoding (BPE) (Sennrich et al., 2016) types.

Settings We implement our approach on top of the Transformer (Vaswani et al., 2017). We apply a 6-layer Transformer with 4 attention heads, embedding size 512, and FFN layer dimension 1024. We apply cross-entropy loss and set max tokens per batch to be 4096. We use Adam optimizer with Beta $(0.9, 0.98)$, 4000 warmup updates, and inverse square root learning rate scheduler with initial learning rates $5e^{-4}$. We use dropout 0.3 and beam search decoding with beam size 5 and length penalty 1.0. We apply the same training configurations in both pretraining and finetuning stages which will be discussed in the following sections. We use `multi-bleu.pl`[1] for BLEU evaluation. We train all models until convergence on a single NVIDIA Tesla V100 GPU. All reported BLEU scores are from a single model. For all the experiments below, we select the saved model state with the best validation perplexity.

4 Bi-SimCut

In this section, we formally propose Bidirectional Pretrain and Unidirectional Finetune with Simple Cutoff Regularization (Bi-SimCut), a simple but effective training strategy that can greatly enhance the generalization of the NMT model. Bi-SimCut consists of a simple cutoff regularization and a two-phase pretrain and finetune strategy. We introduce the details of each part below.

4.1 SimCut: A Simple Cutoff Regularization for NMT

Despite the impressive performance reported in Shen et al. (2020), finding the proper hyper-parameters $(p_{\text{cut}}, \alpha, \beta, N)$ in Token Cutoff seems to be time-consuming if there are limited resources available, which hinders its practical value in the NMT community. To reduce the burden in hyper-parameter searching, we propose SimCut, a simple regularization method that forces the consistency between the output distributions of the original sentence pairs and the cutoff samples.

Our problem formulation is motivated by Virtual Adversarial Training (VAT), where Sano et al. (2019) introduces adversarial regularization that forces the output distribution of the samples with adversarial perturbations $\boldsymbol{\delta}_{\mathbf{x}}$ and $\boldsymbol{\delta}_{\mathbf{y}}$ to be consistent with that of the original samples:

$$\text{KL}(f(e(\mathbf{x}), e(\mathbf{y}); \theta) \| f(e(\mathbf{x}) + \boldsymbol{\delta}_{\mathbf{x}}, e(\mathbf{y}) + \boldsymbol{\delta}_{\mathbf{y}}; \theta)).$$

Instead of generating perturbed samples by gradient-based adversarial methods, for each sentence pair $(\mathbf{x}, \mathbf{y})$, we only generate one cutoff sample $(\mathbf{x}_{\text{cut}}, \mathbf{y}_{\text{cut}})$ by following the same cutoff strategy used in Token Cutoff. For each sentence pair, the training objective of SimCut is defined as:

$$\mathcal{L}_{simcut}(\theta) = \mathcal{L}_{ce}(\theta) + \alpha \mathcal{L}_{simkl}(\theta), \quad (4)$$

where

$$\mathcal{L}_{simkl}(\theta) = \text{KL}(f(\mathbf{x}, \mathbf{y}; \theta) \| f(\mathbf{x}_{\text{cut}}, \mathbf{y}_{\text{cut}}; \theta)).$$

There are only two hyper-parameters α and p_{cut} in SimCut, which greatly simplify the hyper-parameter searching step in Token Cutoff. Note that VAT only allows the gradient to be backpropagated through the right-hand side of the KL divergence term, while the gradient is designed to be backpropagated through both sides of the KL regularization in SimCut. We can see that the constraint introduced by $\mathcal{L}_{tokcut}(\theta)$ and $\mathcal{L}_{kl}(\theta)$ in (3) still implicitly hold in (4):

- $\mathcal{L}_{tokcut}(\theta)$ in Token Cutoff is designed to guarantee that the output of the cutoff sample should close to the ground-truth to some extent. In SimCut, $\mathcal{L}_{ce}(\theta)$ requires the outputs of the original sample close to the ground-truth, and $\mathcal{L}_{simkl}(\theta)$ requires the output distributions of the cutoff sample close to that of the original sample. The constraint introduced by $\mathcal{L}_{tokcut}(\theta)$ then implicitly holds.

- $\mathcal{L}_{kl}(\theta)$ in Token Cutoff is designed to guarantee that the output distributions of the original sample and N different cutoff samples should be consistent with each other. In SimCut, $\mathcal{L}_{simkl}(\theta)$ guarantees the consistency between the output distributions of the original and cutoff samples. Even though SimCut only generates one cutoff sample at each time, different cutoff samples of the same sentence pair will be considered in different training epochs. Such constraint raised by $\mathcal{L}_{kl}(\theta)$ still implicitly holds.

[1] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

Method	en→de	de→en
Transformer	28.70	34.99
VAT	29.45	35.52
R-Drop	30.73	37.30
Token Cutoff†	-	37.60
SimCut	30.98	37.81

Table 2: SimCut achieves the superior or comparable performance on IWSLT14 en ↔ de translation tasks over the strong baselines such as VAT, R-Drop, and Token Cutoff. † denotes the number is reported from Shen et al. (2020), others are based on our runs.

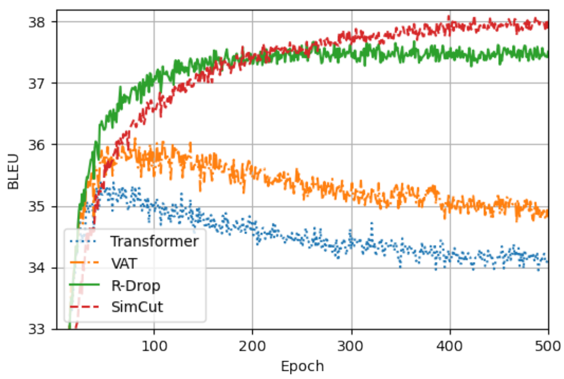

Figure 1: On the IWSLT14 de→en validation set, the BLEU score increases monotonously over epoch number in model training using SimCut. In contrast, the BLEU scores of the other three baselines all stop increasing before 250 epochs. The results suggest that the use of SimCut can effectively alleviate the model training from overfitting.

4.2 Analysis on SimCut

4.2.1 How Does the Simplification Affect Performance?

We here investigate whether our simplification on Token Cutoff hurts its performance on machine translation tasks. We compare SimCut with VAT, Token Cutoff, and R-Drop (Liang et al., 2021), a strong regularization baseline that forces the output distributions of different sub-models generated by dropout to be consistent with each other. Table 2 shows that SimCut achieves superior or comparable performance over VAT, R-Drop, and Token Cutoff, which clearly shows the effectiveness of our method. Due to the tedious and time-consuming hyper-parameter searching in Token Cutoff, we will not include its results in the following sections and show the results of SimCut directly.

Figure 1 shows the evolution of different training methods' validation BLEU scores. On the IWSLT14 de→en validation set, the performance of all methods stop increasing before 250 epochs except for SimCut. The results on VAT are consistent with the previous studies on adversarial overfitting, i.e., virtual adversarial training easily suffering from overfitting (Rice et al., 2020). Note that the BLEU score of SimCut continuously increases in the first 500 epochs.

4.2.2 How Does the Bidirectional Backpropagation Affect Performance?

Even though the problem formulation of SimCut is similar to that of VAT, one key difference is that the gradients are allowed to be backpropagated bidirectionally in the KL regularization in SimCut. We here investigate the impact of the bidirectional backpropagation in the regularization term on the performance of the NMT model. Table 3 shows the translation results of VAT and SimCut with or without bidirectional backpropagation. We can see that both VAT and SimCut benefit from the bidirectional gradient backpropagation in the KL regularization.

Method	en→de	de→en
VAT	29.45	35.52
+ Bi-backpropagation	29.69	36.26
SimCut	30.98	37.81
- Bi-backpropagation	30.29	36.91

Table 3: Bidirectional backpropagation achieves better performance on IWSLT14 en ↔ de translation tasks compared with unidirectional backpropagation in the KL regularization.

4.2.3 Performance on Perturbed Inputs

Given the similar problem formulations of VAT and SimCut, it is natural to regard cutoff operation as a special perturbation and consider SimCut as a perturbation-based method. We here investigate the robustness of NMT models on the perturbed inputs. As discussed in Takase and Kiyono (2021), simple techniques such as word replacement and word drop can achieve comparable performance to sophisticated perturbations. We hence include them as baselines to show the effectiveness of our method.

- **UniRep**: Word replacement approach constructs a new sequence whose tokens are randomly replaced with sampled tokens. For each token in the source sentence $\mathbf{x}$, we sample $\hat{x}_i$ uniformly from the source vocabulary, and use it for the

Input	wir denken (festgelegten), dass wir in der realität nicht so gut sind wie in spielen.
Reference	we feel that we are not as good in reality as we are in games.
Vaswani et al. (2017) on Input	we think we’re not as good in reality as we are in games.
on Noisy Input	we realized that we weren’t as good as we were in real life.
SimCut on Input	we think in reality, we’re not as good as we do in games.
on Noisy Input	we realized that we’re not as good in reality as we are in games.

Table 4: SimCut is more robust to small perturbations in an authentic context. SimCut captures the translation of “in spielen” under the noisy input while the vanilla Transformer ignores the translation of “in spielen” due to the replacement of “denken” with “festgelegten”.

new sequence $\mathbf{x}'$ with probability $1 - p'$:

$$x'_i = \begin{cases} x_i, & \text{with probability } p', \\ \hat{x}_i, & \text{with probability } 1 - p'. \end{cases} \quad (5)$$

We construct $\mathbf{y}'$ from the target sentence $\mathbf{y}$ in the same manner. Following the curriculum learning strategy used in Bengio et al. (2015), we adjust p' with the inverse sigmoid decay:

$$p'_t = \max(q, \frac{k}{k + \exp\left(\frac{t}{k}\right)}), \quad (6)$$

where q and k are hyper-parameters. p'_t decreases to q from 1, depending on the training epoch number t. We use p'_t as p' in epoch t. We set q and k to be 0.9 and 25 respectively in the experiments.

- **WordDrop**: Word drop randomly applies the zero vector instead of the word embedding $e(x_i)$ or $e(y_i)$ for the input token x_i or y_i (Gal and Ghahramani, 2016). For each token in both source and target sentences, we keep the original embedding with the probability β and set it to be the zero vector otherwise. We set β to be 0.9 in the experiments.

We construct noisy inputs by randomly replacing words in the source sentences based on a predefined probability. If the probability is 0.0, we use the original source sentence. If the probability is 1.0, we use completely different sentences as source sentences. We set the probability to be 0.00, 0.01, 0.05, and 0.10 in our experiments. We randomly replace each word in the source sentence with a word uniformly sampled from the vocabulary. We apply this procedure to IWSLT14 de→en test set. Table 5 shows the BLEU scores of each method on the perturbed test set. Note that the BLEU scores are calculated against the original reference sentences. We can see that all methods improve the robustness of the NMT model, and SimCut achieves the best performance among all the methods on both the clean and perturbed test sets. The performance results indicate that SimCut could be considered as a strong baseline for the perturbation-based method for the NMT model.

Method	probability			
	0.00	0.01	0.05	0.10
Transformer	34.99	34.01	30.38	25.70
UniRep	35.67	34.91	31.54	27.24
WordDrop	35.65	34.73	31.22	26.46
VAT	35.52	34.65	30.48	25.44
R-Drop	37.30	36.24	32.27	27.19
SimCut	**37.81**	**36.94**	**33.16**	**27.93**

Table 5: The model trained by SimCut achieves high robustness on the perturbed test set and high performance on the clean test set. Entries represent BLEU scores on IWSLT14 de→en test set when we inject perturbations to source sentences with different probability.

As shown in Table 4, the baseline model completely ignores the translation of “in spielen (in games)” due to the replacement of “denken (think)” with “festgelegten (determined)” in the source sentence. In contrast, our model successfully captures the translation of “in spielen” under the noisy input. This result shows that our model is more robust to small perturbations in an authentic context.

4.2.4 Effects of α and p_{cut}

We here investigate the impact of the scalar hyper-parameters α and p_{cut} in SimCut. α is a penalty parameter that controls the regularization strength in our optimization problem. p_{cut} controls the percentage of the cutoff perturbations in SimCut. We here vary α and p_{cut} in $\{1, 2, 3, 4, 5\}$ and $\{0.00, 0.05, 0.10, 0.15, 0.20\}$ respectively and conduct the experiments on the IWSLT14 de→en dataset. Note that SimCut is simplified to R-Drop approximately when $p_{\text{cut}} = 0.00$. The test BLEU scores are reported in Figure 2. By checking model performance under different combinations of α and p_{cut}, we have the following observations: 1) A too small α (e.g., 1) cannot achieve as good performance as larger α (e.g., 3), indicating a certain degree of regularization strength during NMT model training is conducive to generalization. Meanwhile, an overwhelming regularization ($\alpha = 5$) is not plausible for learning NMT models. 2) When $\alpha = 3$, the best performance is achieved when $p_{\text{cut}} = 0.05$, and $p_{\text{cut}} = 0.00$ performs sub-optimal among all selected probabilities. Such an observation demonstrates that the cutoff perturbation in SimCut can effectively promote the generalization compared with R-Drop.

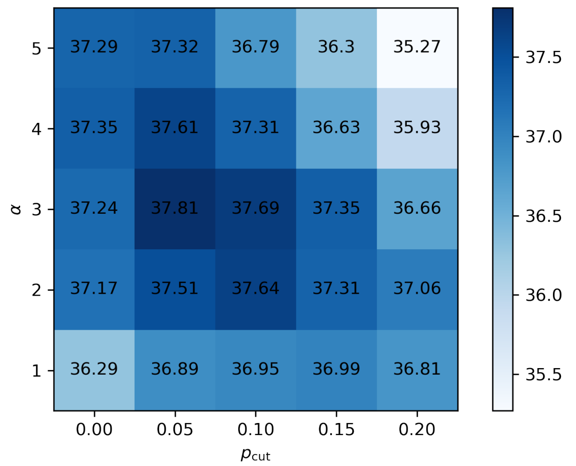

Figure 2: BLEU scores with different α and p_{cut} on IWSLT14 de→en dataset.

4.3 Training Strategy: Bidirectional Pretrain and Unidirectional Finetune

Bidirectional Pretrain is shown to be very effective to improve the translation performance of the unidirectional NMT system (Ding et al., 2021; Xu et al., 2021). The main idea is to pretrain a bidirectional NMT model at first and use it as the initialization to finetune a unidirectional NMT model. Assume we want to train an NMT model for "English→German", we first reconstruct the training sentence pairs to "English+German→German+English", where the training dataset is doubled. We then firstly train a bidirectional NMT model with the new training

Method	en→de	de→en
Transformer	28.70	34.99
Bi-Pretrain	28.94	35.64
+ Finetune	28.82	35.66
Bi-R-Drop Pretrain	30.30	37.01
+ R-Drop Finetune	30.85	37.55
Bi-SimCut Pretrain	30.57	37.70
+ SimCut Finetune	31.16	38.37

Table 6: Bidirectional pretrain and unidirectional finetune results on IWSLT14 en ↔ de datasets. Note that the results of bidirectional pretrain are from one model for dual-directional translations.

Method	en→de	de→en	Average
Transformer	28.70	34.99	31.85
VAT	29.45	35.52	32.49
Mixed Rep†	29.93	36.41	33.17
UniDrop†	29.99	36.88	33.44
R-Drop	30.73	37.30	34.02
BiBERT†	30.45	**38.61**	34.53
Bi-SimCut	**31.16**	38.37	**34.77**

Table 7: Our method achieves the superior performance over the existing methods on the IWSLT14 en↔de translation benchmark. † denotes the numbers are reported from the papers, others are based on our runs.

sentence pairs:

$$\mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{y})\in\mathcal{S}}[\ell(f(\mathbf{x},\mathbf{y};\theta),\ddot{\mathbf{y}}) + \ell(f(\mathbf{y},\mathbf{x};\theta),\ddot{\mathbf{x}})], \quad (7)$$

and finetune the model with "English→German" direction. We follow the same training strategy and apply SimCut regularization to both pretraining and finetuning procedures. Table 6 shows that our training strategy with SimCut could achieve superior performance compared with strong baseline such as R-Drop.

Comparison with Existing Methods We summarize the recent results of several existing works on IWSLT14 en↔de benchmark in Table 7. The existing methods vary from different aspects, including Virtual Adversarial Training (Sano et al., 2019), Mixed Tokenization for NMT (Wu et al., 2020), Unified Dropout for the transformer model (Wu et al., 2021), Regularized Dropout (Liang et al., 2021), and BiBERT (Xu et al., 2021). We can see that our approach achieves an improvement of 2.92 BLEU score over Vaswani et al. (2017) and surpass the current state-of-the-art (SOTA) method BiBERT that incorporates large-scale pretrained model, stochastic layer selection, and bidirectional pretraining. Given the simplicity of Bi-SimCut, we believe it could be considered as a strong baseline for the NMT task.

5 Standard Resource Scenario

We here investigate the performance of Bi-SimCut on the larger translation benchmark compared with the IWSLT14 benchmark.

5.1 Dataset Description and Model Configuration

For the standard resource scenario, we evaluate NMT models on the WMT14 English-German dataset, which contains 4.5M parallel sentence pairs. We combine newstest2012 and newstest2013 as the validation set and use newstest2014 as the test set. We collect the pre-processed data from Xu et al. (2021)'s release[2], where a shared dictionary with 52K BPE types is built. We apply a standard Transformer Big model with 16 attention heads, embedding size 1024, and FFN layer dimension 4096. We apply cross-entropy loss and set max tokens per batch to be 4096. We use Adam optimizer with Beta $(0.9, 0.98)$, 4000 warmup updates, and inverse square root learning rate scheduler with initial learning rates $1e^{-3}$. We decrease the learning rate to $5e^{-4}$ in the finetuning stage. We select the dropout rate from 0.3, 0.2, and 0.1 based on the validation performance. We use beam search decoding with beam size 4 and length penalty 0.6. We train all models until convergence on 8 NVIDIA Tesla V100 GPUs. All reported BLEU scores are from a single model.

5.2 Results

We report test BLEU scores of all comparison methods and our approach on the WMT14 dataset in Table 8. With Bi-SimCut pretraining and finetuning procedures, our model achieves strong or state-of-the-art BLEU scores on en→de and de→en translation benchmarks. We fix p_{cut} to be 0.05 and tune the hyperparameter α in both R-Drop and SimCut based on the performance on the validation set. Note that the BLEU scores of R-Drop are lower than that reported in Liang et al. (2021). Such gap might be due to the different prepossessing steps used in Liang et al. (2021) and Xu et al. (2021). It is worth mentioning that Bi-SimCut outperforms BiBERT on de→en direction even though BiBERT incorporates bidirectional pretraining, large-scale pretrained contextualized embeddings, and stochastic layer selection mechanism.

[2] https://github.com/fe1ixxu/BiBERT

6 High Resource Scenario

To investigate the performance of Bi-SimCut on the distant language pairs which naturally do not share dictionaries, we here discuss the effectiveness of Bi-SimCut on the Chinese-English translation task.

6.1 Dataset Description and Model Configuration

For the high resource scenario, we evaluate NMT models on the WMT17 Chinese-English dataset, which consists of 20.1M training sentence pairs, and we use devtest-2017 as the validation set and newstest-2017 as the test set. We firstly build the source and target vocabularies with 32K BPE types separately and treat them as separated or joined dictionaries in our experiments. We apply the same Transformer Big model and training configurations used in the WMT14 experiments. We use beam search decoding with beam size 5 and length penalty 1. We train all models until convergence on 8 NVIDIA Tesla V100 GPUs. All reported BLEU scores are from a single model.

6.2 Results

We report test BLEU scores of the baselines and our approach on the WMT17 dataset in Table 9. The NMT models with separated dictionaries perform slightly better than those with the shared dictionary. We can see that our approach significantly improves translation performance. In particular, Bi-SimCut achieves more than 1.5 BLEU improvement over Vaswani et al. (2017), showing the effectiveness and universality of Bi-SimCut on the distant language pair.

7 Related Work

Adversarial Perturbation SimCut could be regarded as a perturbation base method. Adversarial perturbation was firstly introduced in the field of computer vision (Szegedy et al., 2014; Goodfellow et al., 2015). Miyato et al. (2017) considered adversarial perturbations in the embedding space and showed its effectiveness on the text classification tasks. In the NMT field, Sano et al. (2019) and Wang et al. (2019) applied adversarial perturbations in the embedding space during training of the

Method	en→de	de→en	Average
Transformer + Large Batch† (Ott et al., 2018)	29.30	-	-
Evolved Transformer† (So et al., 2019)	29.80	-	-
BERT Initialization (12 layers)† (Rothe et al., 2020)	30.60	33.60	32.10
BERT-Fuse† (Zhu et al., 2020)	30.75	-	-
R-Drop (Liang et al., 2021)	30.13	34.54	32.34
BiBERT† (Xu et al., 2021)	**31.26**	34.94	**33.10**
SimCut	30.56	34.86	32.71
Bi-SimCut Pretrain	30.10	34.42	32.26
+ SimCut Finetune	30.78	**35.15**	32.97

Table 8: Our method achieves the superior or comparable performance over the existing methods on the WMT14 en↔de translation benchmark. † denotes the numbers are reported from Xu et al. (2021), others are based on our runs.

Method	share	zh→en
Transformer	✗	25.53
Transformer	✓	25.31
SimCut	✗	26.86
SimCut	✓	26.74
Bi-SimCut Pretrain	✓	26.13
+ SimCut Finetune	✓	**27.17**

Table 9: Our method achieves strong performance on the WMT17 zh→en translation benchmark. share denotes whether a shared dictionary is applied.

encoder-decoder NMT model. Cheng et al. (2019) leveraged adversarial perturbations and generated adversarial examples by replacing words in both source and target sentences. They introduced two additional language models for both sides and a candidate word selection mechanism for replacing words in the sentence pairs. Takase and Kiyono (2021) compared perturbations for the NMT model in view of computational time and showed that simple perturbations are sufficiently effective compared with complicated adversarial perturbations.

Consistency Training Besides perturbation-based methods, our approach also highly relates to a few works of consistency training in the NMT field on dropout models and data augmentation. Among them, the most representative methods are R-Drop (Liang et al., 2021) and Cutoff (Shen et al., 2020). R-Drop only considers the output consistency between two dropout sub-models with the same inputs. Cutoff considers consistency training from a data perspective by regularizing the inconsistency between the original sample and the augmented samples with part of the information within the input sentence pair being dropped. Note that Cutoff takes the dropout sub-models into account during the training procedure as well. We want to emphasize that SimCut is not a new method, but a version of Cutoff simplified and adapted for NMT tasks.

8 Conclusion

In this paper, we propose Bi-SimCut: a simple but effective two-stage training strategy to improve NMT performance. Bi-SimCut consists of bidirectional pretraining and unidirectional finetuning procedures equipped with SimCut regularization for improving the generality of the NMT model. Experiments on low (IWSLT14 en↔de), standard (WMT14 en↔de), and high (WMT17 zh→en) resource translation benchmarks demonstrate Bi-SimCut and SimCut's capabilities to improve translation performance and robustness. Given the universality and simplicity of Bi-SimCut and SimCut, we believe: a) SimCut could be regarded as a perturbation-based method, and it could be used as a strong baseline for the robustness research. b) Bi-SimCut outperforms many complicated methods which incorporate large-scaled pretrained models or sophisticated mechanisms, and it could be used as a strong baseline for future NMT research. We hope researchers of perturbations and NMT could use SimCut and Bi-SimCut as strong baselines to make the usefulness and effectiveness of their proposed methods clear. For future work, we will explore the effectiveness of SimCut and Bi-SimCut on more sequence learning tasks, such as text classification, natural language understanding, etc.

References

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1, pages 1171–1179.

Guandan Chen, Kai Fan, Kaibo Zhang, Boxing Chen, and Zhongqiang Huang. 2021. Manifold adversarial augmentation for neural machine translation. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 3184–3189.

Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust neural machine translation with doubly adversarial inputs. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4324–4333.

Liang Ding, Di Wu, and Dacheng Tao. 2021. Improving neural machine translation by bidirectional training. arXiv preprint arXiv:2109.07780.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. Advances in neural information processing systems, 29:1019–1027.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In Proceedings of the 3rd International Conference on Learning Representations.

Xiaobo Liang, Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, and Tie-Yan Liu. 2021. R-drop: regularized dropout for neural networks. CoRR abs/2106.14448.

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2017. Adversarial training methods for semi-supervised text classification. In Proceedings of the 5th International Conference on Learning Representations (ICLR).

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. arXiv preprint arXiv:1806.00187.

Leslie Rice, Eric Wong, and Zico Kolter. 2020. Overfitting in adversarially robust deep learning. In International Conference on Machine Learning, pages 8093–8104. PMLR.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. Transactions of the Association for Computational Linguistics, 8:264–280.

Motoki Sano, Jun Suzuki, and Shun Kiyono. 2019. Effective adversarial regularization for neural machine translation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 204–210.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 1715–1725.

Dinghan Shen, Mingzhi Zheng, Yelong Shen, Yanru Qu, and Weizhu Chen. 2020. A simple but tough-to-beat data augmentation approach for natural language understanding and generation. arXiv preprint arXiv:2009.13818.

David So, Quoc Le, and Chen Liang. 2019. The evolved transformer. In International Conference on Machine Learning, pages 5877–5886. PMLR.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In Proceedings of the 2nd International Conference on Learning Representations (ICLR).

Sho Takase and Shun Kiyono. 2021. Rethinking perturbations in encoder-decoders for fast training. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5767–5780.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. Advances in neural information processing systems, 30:5998–6008.

Dilin Wang, Chengyue Gong, and Qiang Liu. 2019. Improving neural language modeling via adversarial training. In Proceedings of the 36th International Conference on Machine Learning, volume 97, pages 6555–6565. PMLR.

Lijun Wu, Shufang Xie, Yingce Xia, Yang Fan, Jian-Huang Lai, Tao Qin, and Tieyan Liu. 2020. Sequence generation with mixed representations. In Proceedings of the 37th International Conference on Machine Learning, volume 119, pages 10388–10398. PMLR.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144.

Zhen Wu, Lijun Wu, Qi Meng, Yingce Xia, Shufang Xie, Tao Qin, Xinyu Dai, and Tie-Yan Liu. 2021. Unidrop: A simple yet effective technique to improve transformer without extra cost. arXiv preprint arXiv:2104.04946.

Haoran Xu, Benjamin Van Durme, and Kenton Murray. 2021. BERT, mBERT, or BiBERT? a study on contextualized embeddings for neural machine translation. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 6663–6675.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. 2020. Incorporating bert into neural machine translation. arXiv preprint arXiv:2002.06823.