

AdMix: A Mixed Sample Data Augmentation Method for Neural Machine Translation

Chang Jin*, Shigui Qiu, Nini Xiao, Hao Jia

Institute of Artificial Intelligence, School of Computer Science and Technology, Soochow University
 {cjin, sgqiu, nnxiaoxiao, hjia}@stu.suda.edu.cn

Abstract

In Neural Machine Translation (NMT), data augmentation methods such as back-translation have proven their effectiveness in improving translation performance. In this paper, we propose a novel data augmentation approach for NMT, which is independent of any additional training data. Our approach, *AdMix*, consists of two parts: 1) introduce faint discrete noise (word replacement, word dropping, word swapping) into the original sentence pairs to form augmented samples; 2) generate new synthetic training data by softly mixing the augmented samples with their original samples in training corpus. Experiments on three translation datasets of different scales show that *AdMix* achieves significant improvements (1.0 to 2.7 BLEU points) over strong Transformer baseline. When combined with other data augmentation techniques (e.g., back-translation), our approach can obtain further improvements.

1 Introduction

Data augmentation are techniques used to increase the amount of data by adding slightly modified copies of already existing data or newly created synthetic data from existing data [Li *et al.*, 2021]. These methods can significantly boost the accuracy of deep learning methods. For image classification tasks, there are various data augmentation methods like random rotation, mirroring, cropping and cut-out [Krizhevsky *et al.*, 2012; DeVries and Taylor, 2017]. However, universal data augmentation techniques in natural language processing (NLP) tasks have not been thoroughly explored due to the discrete nature of natural language [Wei and Zou, 2019; Gao *et al.*, 2019].

According to the diversity of augmented data, Li *et al.* [2021] frame data augmentation methods into three categories, including paraphrasing, noising, and sampling. Specifically, the purpose of the paraphrase-based methods is to generate augmented samples that are semantically similar to the original samples. Back-translation [Sennrich *et al.*, 2016a] is a typical paraphrase-based method to translate

monolingual data by an inverse translation model. While effective, back-translation method becomes inapplicable when target-side monolingual data is limited. The sampling-based methods master the distribution of the original data to sample novel data points as augmented data. For this type of methods, a common approach is to use the pretrained language model to generate labeled augmented sentences [Anaby-Tavor *et al.*, 2020; Kumar *et al.*, 2020]. The last category is based on noising, which adds discrete or continuous noise under the premise of guaranteeing validity. The noising-based methods consist of word dropping, word replacement, word swapping and Mixup, etc. Due to their simplicity and effectiveness, we focus on these methods in this paper. Nevertheless, the previously proposed noise-based methods have their limitations. For example, injecting faint discrete noise (including word replacement, word dropping and word swapping) into sentences produces sentences with limited diversity. Guo *et al.* [2020] propose a sequence-level variant of Mixup [Zhang *et al.*, 2017]. Despite its effectiveness, this method is less interpretable and more difficult than the discrete noising-based methods.

In this paper, we propose a new data augmentation method for neural machine translation task. Our approach, *AdMix*, first introduces an appropriate amount of discrete noise to obtain augmented samples, and then mixes these augmented samples by randomly sampling from the Dirichlet(α, \dots, α) distribution. Once these augmented sentences are mixed, we use a residual connection to combine the mixed samples with their original samples through a second random convex combination sampled from a Beta(β, β) distribution. In this way we obtain the final synthetic samples. To verify the effectiveness of our method, we conduct experiments on three machine translations tasks, including IWSLT14 German to English, LDC Chinese to English, and WMT14 English to German translation tasks. The experimental results show that our method yields significant improvements over strong Transformer baselines on datasets of different scales.

We highlight our contributions in three aspects:

- We propose a simple but effective strategy to boost NMT performance, which linearly interpolates the original sentence with its discrete transformed sentence to generate new synthetic training data.
- Experiments on three translation benchmarks show that our approach achieves significant improvements over

*Contact Author

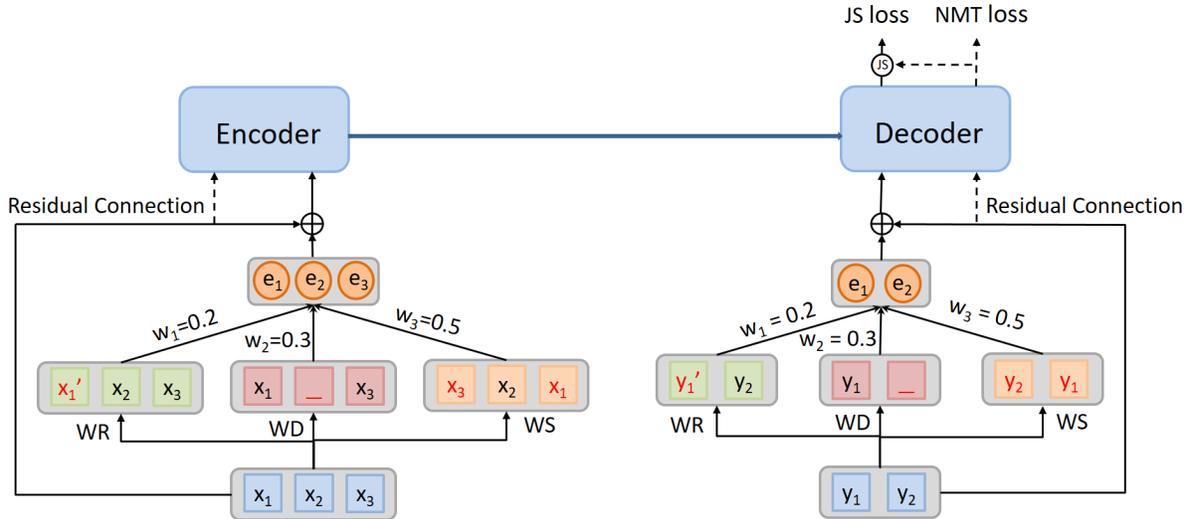


Figure 1: Illustration of the proposed method, *AdMix*, which linearly interpolates the original sentence with its discrete transformed sentence to generate synthetic training sample. The dashed line represents the data flow of the original sentence, the solid line represents the data flow of the synthetic sentence.

various strong baselines.

- Our method can combine with other data augmentation techniques to yield further improvements.

2 Related Work

A large number of data augmentation methods have been proposed recently, and we will present several related works on data augmentation for NMT.

One popular study is back-translation [Sennrich *et al.*, 2016a] which generates synthetic samples based on the conditional probability distribution of the reverse translation model. But back-translation requires training a reverse translation model, which usually consumes giant computing resources. Another similar method is self-training [He *et al.*, 2020], which augments the original labeled dataset with unlabeled data paired with the model’s prediction. Different from back-translation, it uses source-side monolingual data to generate pseudo-parallel sentences. Self-training avoids the need to train a reverse translation model, but it is also not suitable in the case of limited monolingual data.

The noising-based methods are also common data augmentation technologies. Such methods are not only easy to use, but also to improve the robustness of the model [Miyato *et al.*, 2017]. Artetxe *et al.* [2017] randomly choose several words in the sentence and swap their positions. Lample *et al.* [2018] randomly remove each word in the source sentence with probability p to help train the unsupervised NMT model. Xie *et al.* [2017] randomly replace the word with a placeholders token. Similarly to Xie *et al.* [2017], Wang *et al.* [2018] propose a method that randomly replaces words in both the source and target sentences with other words in the vocabulary. Although faint discrete noises are not strong enough to alter the meaning of the input sentence, they may result in the loss of lexical meaning of specific words. Thus, it affects the correspondence between source-side sentences

and target-side sentences. Another problem with the discrete noising-based approach is the limited diversity of its operations. Cheng *et al.* [2018] simply add the Gaussian noise to all of word embeddings in a sentence to simulate possible types of perturbations. The formula is

$$\mathbf{E}[x'_i] = \mathbf{E}[x_i] + \epsilon, \quad \epsilon \sim \mathbf{N}(0, \sigma^2 \mathbf{I}), \quad (1)$$

where the vector ϵ is sampled from a Gaussian distribution with variance σ^2 . σ is a hyperparameter. This method is also not guaranteed to maintain semantics.

In addition to the above methods, there is another data augmentation method that has attracted a lot of attention recently, called Mixup. The Mixup data augmentation technique was first presented in image classification by Zhang *et al.* [2017]. Hendrycks *et al.* [2020] proposed an advanced Mixup method called AugMix, a data processing technique that mixed augmented images and enforced consistent embeddings of the augmented images. Inspired by the work, Guo *et al.* [2020] proposed a sequence-level variant of Mixup. As described by Guo *et al.* [2020], a pair of training examples (x_1, y_1) and (x_2, y_2) are sampled from the training set. And the synthetic sentences are:

$$\begin{aligned} \hat{x} &= \lambda x_1 + (1 - \lambda)x_2, \\ \hat{y} &= \lambda y_1 + (1 - \lambda)y_2. \end{aligned}$$

λ is drawn from a Beta(β, β) distribution, which is controlled by the hyperparameter β . Different from Guo *et al.* [2020], Cheng *et al.* [2020] firstly build their adversarial samples, and then generate a new synthetic sample by interpolating between the adversarial samples. Although Cheng *et al.* [2020] guarantee the invariance of the original sentences semantic information compared to Guo *et al.* [2020], they need an additional language model (LM) to generate the adversarial samples. Compared with these studies, our method does not require additional resources, and the generated synthetic samples are more interpretable than Mixup method.

3 AdMix

In our approach, *AdMix*, the goal is to generate new synthetic training examples as illustrated in Figure 1. It is applied to both the source and target sequences to build new synthetic samples, which are used as augmented translation pairs for training purposes. This means that our approach only influences the training process of NMT without changing its inference process. *AdMix* is divided into two stages: discrete noise and data mixing, and the details are described as follows.

3.1 Discrete Noise

In this stage, we introduce an appropriate amount of discrete noises to obtain several augmented samples. Let $X \in \mathbb{R}^{s \times V}$, $Y \in \mathbb{R}^{t \times V}$ represent a source sequence of length s and a target sequence of length t , respectively. V denotes the size of vocabulary. Given a pair of training example (X, Y) in training set, we perform word replacement (WR), word swapping (WS), word dropping (WD) operations on the source and target sentences to obtain (X_{wr}, Y_{wr}) , (X_{ws}, Y_{ws}) , (X_{wd}, Y_{wd}) respectively. For example, we swap the positions of x_1 and x_3 , replace x_1 with x'_1 , and remove x_2 in the sentence for WS, WR, WD respectively (see Figure 1).

In practice, to ensure that the amount of noise is proportional to the sentence length l , we set a hyperparameter γ . For WS and WR operations, the number of changed words is $n = \gamma l$. For WD operation, we randomly remove each word in the sentence with probability γ .

3.2 Data Mixing

After obtaining the augmented sentences, we first obtain their embedding sequences separately: $(\mathbf{E}[X_{wr}], \mathbf{E}[Y_{wr}])$, $(\mathbf{E}[X_{ws}], \mathbf{E}[Y_{ws}])$, $(\mathbf{E}[X_{wd}], \mathbf{E}[Y_{wd}])$. Inspired by Hendrycks et al. [2020], we choose to use elementwise convex combinations to mix them and the coefficient vector is randomly sampled from a Dirichlet(α, \dots, α) distribution. The word embedding of these augmented data after softly mixing is $(\mathbf{E}[X_{ad}], \mathbf{E}[Y_{ad}])$. Finally, we use a residual connection to combine the mixed samples with their original samples by sampling m from a Beta(β, β) distribution, and the word embedding of the final synthetic sentence is

$$\mathbf{E}[X_{admix}] = m\mathbf{E}[X_{ad}] + (1 - m)\mathbf{E}[X], \quad (2)$$

$$\mathbf{E}[Y_{admix}] = m\mathbf{E}[Y_{ad}] + (1 - m)\mathbf{E}[Y], \quad (3)$$

where $\mathbf{E}[X]$, $\mathbf{E}[Y]$ denote the word embedding at the source and target sides of original sentence pairs respectively.

3.3 Training Objectives

The pseudocode of *AdMix* is shown in Algorithm 1. We couple this augmentation scheme with a loss that enforces a consistent embedding by the model across diverse augmentations of the same input sentences. Since the semantic meanings of the sentences are approximately preserved after *AdMix* operation, we can incorporate Jensen-Shannon divergence consistency loss into the training objective by encouraging the model to make similar predictions between original samples and synthetic samples. For this purpose, we minimize

Algorithm 1 AdMix Pseudocode

```

1: Input: Model  $p$ , CE Loss  $\mathcal{L}$ , sentence  $x_{orig}, y_{orig}$ ,
   Operations = {replace, drop, swap}
2: function AdMix( $x_{orig}, y_{orig}, k = 3, \alpha = 1, \beta = 1$ ):
3:   Fill  $x_{ad}, y_{ad}$  with zeros
4:   Sample mixing weights  $(w_1, w_2, \dots, w_k) \sim$ 
     Dirichlet( $\alpha, \alpha, \dots, \alpha$ )
5:   for each  $i \in [1, k]$ 
6:      $x_{noise}^i = \text{Operations}[i](x_{orig})$ 
7:      $y_{noise}^i = \text{Operations}[i](y_{orig})$ 
8:      $x_{ad} += w_i \cdot \text{Embedding}(x_{noise}^i)$ 
9:      $y_{ad} += w_i \cdot \text{Embedding}(y_{noise}^i)$ 
10:  end for
11:   $x_{orig} = \text{Embedding}(x_{orig})$ 
12:   $y_{orig} = \text{Embedding}(y_{orig})$ 
13:  Sample weights  $m \sim \text{Beta}(\beta, \beta)$ 
14:  Interpolate  $x_{admix} = mx_{ad} + (1 - m)x_{orig}$ 
15:  Interpolate  $y_{admix} = my_{ad} + (1 - m)y_{orig}$ 
16:  return  $x_{admix}, y_{admix}$ 
17: end function
18:  $x_{admix}, y_{admix} = \text{AdMix}(x_{orig}, y_{orig})$ 
19: Loss Output:  $\mathcal{L}(p(y|x_{orig}; y_{orig})) + \lambda JS(p(y|x_{orig};$ 
20:    $y_{orig}); p(y|x_{admix}; y_{admix}))$ 

```

the Jensen-Shannon divergence among the posterior distributions of the original sample (x, y) and its augmented variants (x_{admix}, y_{admix}) . The training objective can be written as:

$$\mathcal{L} = \mathcal{L}_{ce}(x, y) + \lambda JS(p_{orig}; p_{admix}), \quad (4)$$

$$\mathcal{L}_{ce} = - \sum_{i=1}^T \log P(y_i | x; y_{<i}), \quad (5)$$

where \mathcal{L}_{ce} denotes the standard cross-entropy loss applied to the original samples, and T refers to the length of the target sentence. JS denotes Jensen-Shannon divergence. p_{orig} and p_{admix} are the probability distributions of model by respectively feeding the original sentence and the augmented sentence. λ is a hyperparameter, which balances the importance between two loss function.

The Jensen-Shannon divergence can be understood to measure the average information that the sample reveals about the identity of the distribution from which it was sampled [Hendrycks et al., 2020]. This loss can be computed by

$$JS(p_{orig}; p_{admix}) = 1/2(KL[p_{orig}||M] + KL[p_{admix}||M]), \quad (6)$$

where $M = (p_{orig} + p_{admix})/2$. Although KL-divergence has been widely adopted as the divergence metric in previous works [Miyato et al., 2017; Clark et al., 2018; Xie et al., 2020], it has been shown that the JS divergence loss can endow the model with more stability and consistency across a diverse set of inputs [Kannan et al., 2018; Hendrycks et al., 2020]. Therefore, we utilize the Jensen-Shannon (JS) divergence consistency loss as our loss function in this paper.

Method	Zh-En						En-De	De-En
	NIST02	NIST03	NIST04	NIST05	NIST08	AVG		
Transformer	47.48	46.77	47.92	47.02	37.99	45.44	27.30	34.43
Swap	48.46	47.01	48.51	47.86	37.71	45.91	27.48	34.65
WordDrop	48.00	47.05	48.28	47.35	38.06	45.75	27.55	35.03
Switchout	48.02	47.96	48.06	48.73	39.07	46.37	27.60	35.04
SeqMix	48.37	47.23	48.41	48.79	37.63	46.11	28.10	35.49
AdMix	48.98	48.41	49.62	48.38	40.32	47.14	28.26	37.10

Table 1: BLEU scores on LDC Zh-En, WMT En-De, and IWSLT De-En translation.

4 Experiments

We conduct experiments on the following machine translation tasks to evaluate our method: LDC Chinese-English (Zh-En), IWSLT14 German-English (De-En), and WMT14 English-German (En-De). The performance is evaluated with the 4-gram BLEU score [Papineni *et al.*, 2002] calculated by the multi-bleu.perl script. For De-En and En-De, we report case-sensitive tokenized BLEU scores, while for Zh-En, we report case-insensitive tokenized BLEU scores.

4.1 Setup

Datasets. For IWSLT14 German-English, following Edunov *et al.* [2018], we apply the byte-pair encoding (BPE) [Sennrich *et al.*, 2016b] script to preprocess the training corpus with 10K joint operations, which consists of 0.16M sentence pairs. The validation set is split from the training set and the test set is the concatenation of tst2010, tst2011, tst2012, dev2010, and dev2012. For the Chinese-English translation task, the training set is the LDC corpus which contains 1.25M sentence pairs. The validation set is the NIST 06 dataset, and test sets are NIST 02, 03, 04, 05, 08. We apply BPE for Chinese and English respectively, and the merge operations are both 32k. For English-German translation, we use the WMT14 corpus consisting of 4.5M sentence pairs. The validation set is newstest2013 and the test set is newstest2014. We build a shared vocabulary of 32K sub-words using the BPE script.

Training Details. We choose Transformer as our translation model. For IWSLT14 German-English tasks, the dimensions of the embedding, feed-forward network, and the number of layers of the Transformer models are 512, 1024, and 6 respectively. The dropout rate is 0.3, and the batch size is 8192 tokens. For LDC Chinese-English task and WMT14 German-English task, the dimensions of the embedding, feed-forward network, and the number of layers of the Transformer models are 512, 2048, and 6 respectively. The dropout rate is 0.3, 0.1 separately for Zh-En and En-De tasks, and the batch size is both 8192 tokens. We train on two V100 GPUs and accumulate the gradients 2 times before updating. For all models but except the En-De task model, we use Adam with learning rate 5×10^{-4} and the inverse sqrt learning rate scheduler to optimize the models. For En-De task model, we use Adam with learning rate 7×10^{-4} and the inverse sqrt learning rate scheduler. There are two important hyperparameters in our approach, λ in objective loss function and the discrete noise fractions γ . For all datasets, we set the noise

fractions γ as 0.1 and the hyperparameter λ as 10 by default. we also explore the effect of noise fractions γ and hyperparameter λ on the validation set, and then we choose the best hyperparameters. Exact details regarding the result can be found in 4.5.

4.2 Baselines

We compare our approach with the following baselines:

- Transformer: The vanilla Transformer model without any data augmentation [Vaswani *et al.*, 2017];
- Swap: Randomly choose several tokens in the sentence and swap their positions [Artetxe *et al.*, 2017];
- WordDrop: Randomly drop each token in the sentence with probability γ [Lample *et al.*, 2018];
- Switchout: Randomly replace tokens over the vocabulary by position [Wang *et al.*, 2018];
- SeqMix: Randomly sample two sentences from training dataset and softly combine them [Guo *et al.*, 2020].

For Swap and WordDrop, we set the probability $\gamma = 0.15$ of each word token to be swapped or replaced in the training phase. As for Switchout, both the source and target side temperature parameters are set to 1.0 by us. When adopting the SeqMix training strategy, we set β of Beta(β, β) distribution as 1.0, 1.0 and 0.1 for Zh-En, De-En and En-De respectively. For AdMix, we fix $\beta = 1.0$ of the Beta(β, β) distribution and $\alpha = 1.0$ of the Dirichlet(α, \dots, α) distribution in all experiments.

4.3 Main Results

We show the BLEU scores of different methods for Zh-En, En-De, and De-En translation tasks in Table 1. We compare our method with vanilla Transformer and existing methods including Swap, WordDrop, Switchout, SeqMix. AdMix significantly outperforms the Transformer baseline on all tasks. Specifically, compared to the baseline system, AdMix improves the performance of Zh-En, En-De, and De-En by 1.70, 0.96, and 2.67 points respectively. The experimental result also shows that our method is more effective on small datasets.

AdMix shows performance advantages over other data augmentation methods, outperforming the best-in-class SwitchOut by 0.8 points in Zh-En and SeqMix by 1.61 points in De-En. In particular, the superiority of AdMix over SeqMix [Guo *et al.*, 2020] verifies that we propose a more effective method to generate synthetic examples in NMT. Our approach consistently outperforms SeqMix [Guo *et al.*, 2020],

yielding significant 1.03, 0.16, and 1.61 BLEU point gains on the Zh-En, En-De, and De-En translation tasks, respectively. Compared with the method of discrete noise methods (such as WordDrop [Lample *et al.*, 2018]), our method yields significant 1.39, 0.71, 2.07 BLEU point gains on the Zh-En, En-De, and De-En translation tasks respectively.

Method	De-En
Transformer	34.43
AdMix	37.10
w/o Replace	37.05
w/o Swap	36.73
w/o Drop	37.01
w/o Residual Connection	36.95
Only Source	36.97
Only Target	35.62

Table 2: Ablation study on German-English translation. Only Source and Only Target mean only applying AdMix on the source or target sentences respectively.

4.4 Ablation Study

To investigate the importance of each component of AdMix, the BLEU scores from the ablation experiments on the German-English dataset are presented in Table 2. Firstly, we remove one of the three discrete transformation operations which include word replacement, word dropping, and word swapping. And the results appear that no matter which of the three discrete noises is removed, the empirical performance will be damaged. Especially, the BLEU score in the test set drops by 0.37 after removing word swapping. Therefore, it is important to introduce diversity through these three discrete transformation operations. And among these three operations, the word swapping operation can bring more diversity with the same noise levels. We also remove the residual connection to observe the BLEU score, which drops from 37.10 to 36.95 in the test set. The residual connection is effective probably because it preserves the semantic information of clean sentences, allowing a better correspondence between the source and target sentences.

Finally, We wonder how the BLEU scores would change by only applying AdMix on the source or target sentences. As shown in Table 2, the data augmentation strategies performed on either the source or target sentences are not as effective as on both sides simultaneously. Specially, when we employ AdMix only on the target sentences, the BLEU score drops 1.48 points in the test set. Nevertheless, both strategies have different degrees of improvement compared to the baseline system. The former brings 2.54 BLEU improvement and the latter yields 1.19 BLEU improvement in the test set.

4.5 Effect of λ and γ

We select different values of the hyperparameter λ (ranging from 0.0 to 40.0) to investigate the importance of incorporating the Jensen-Shannon (JS) divergence consistency loss. The coefficient of the cross-entropy (CE) loss term is set as 1, and the JS divergence consistency loss term is controlled as the relative weight through the coefficient λ . The results of the

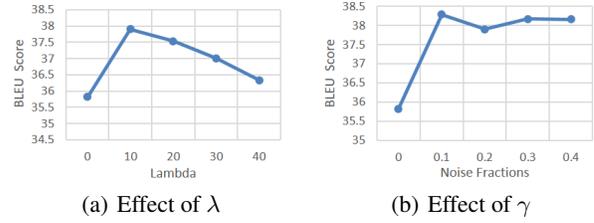


Figure 2: The effect of lambda and noise fractions on the validation set of German-English translation task.

BLEU scores on the German-English dataset are shown in the left panel of Figure 2(a). The performance of the model tends to increase and then decrease as the lambda increases, producing the best empirical results on the validation set when the value λ is 10. Compared with the standard translation loss, the JS divergence loss can lead to performance improvements. Because it can approximate the original sample output distribution $p(\tilde{y}|x; y)$ and the synthetic sample output distribution $p(\tilde{y}|\hat{x}; \hat{y})$, i.e. to ensure consistency at the semantic level.

Another important hyperparameter with the AdMix approach is the noise fractions γ , where the fractions can be regarded as the magnitude of perturbations applied to the input sentence. As shown in the right panel of Figure 2(b), we apply various noise fractions for AdMix, including 0.1, 0.2, 0.3 and 0.4. It can be observed that the noise fractions of 0.1 shows the best translation performance of the obtained model. Using a larger noise fractions within a certain range can also obtain higher levels of performance than the baseline. The results show that the introduction of discrete noise in the training samples has a positive impact on improving the translation quality of the model, and AdMix can maintain a stable bleu score even at higher noise levels.

Method	Op-0	Op-1	Op-2	Op-3
Transformer	35.82	33.68	32.04	29.75
Swap	35.85	33.50	31.65	29.51
WordDrop	36.46	34.38	32.96	31.06
Switchout	36.22	34.23	32.50	30.52
SeqMix	36.50	34.32	32.53	30.61
AdMix	38.28	36.23	34.52	32.58

Table 3: Results on German-English validation set. The column lists results for different noise levels.

4.6 Results to Noisy Inputs

The noising-based data augmentation methods can not only improve the translation quality of the model, but also improve its robustness [Miyato *et al.*, 2017; Li *et al.*, 2021]. To test robustness on noisy inputs, similarly to Cheng et al. [2020], we construct a noisy data set by randomly replacing a word in each sentence of the standard German-English validation set with a relevant alternative. The cosine similarity of word embeddings is used to determine the relevance of words. We repeat the process in an original sentence according to the

number of operations where zero operation yields the original clean dataset.

As shown in Table 3, we can find that our training approach consistently outperforms all baseline methods on all the numbers of operations. This demonstrates that our approach has the ability to resist perturbations. The performance on Transformer baseline decreases rapidly as the number of operations increases. Although the performance of our approach also drops, we can see that our approach consistently exceeds baseline. After conducting three operations on the original validation set, the vanilla Transformer decreases from 35.82 to 29.75, while AdMix decreases from 38.28 to 32.58.

Method	De-En
Transformer	34.43
+BT	35.70
AdMix	37.10
+BT	37.46

Table 4: The results of our method combined with back-translation.

4.7 Results with Back-translation

To explore the effect of combining our method with back-translation (BT), we sample 0.16M English sentences from the WMT13 english monolingual corpus¹. We compare AdMix against BT on De-En translation task. As shown in Table 4, with using extra monolingual data, the back-translation approach yields a boost of 1.27 BLEU compared to the vanilla Transformer. But the gain delivered by BT is less significant than the gain delivered by AdMix. In addition, AdMix and back translation are not mutually exclusive, and we can apply AdMix on the pseudo-parallel sentences obtained by BT to further improve the BLEU score. When we combine BT with AdMix, it will yield 0.36 BLEU improvement than using AdMix alone.

5 Conclusion

For machine translation, we propose a new data augmentation method, *AdMix*, which performs augmentation in both source and target sentences. We generate new synthetic samples by softly mixing the original sentences with their discrete transformed sentences. Our approach guarantees semantic invariance at both the source and the target sentences compared to the noising-base methods, so it enables the model to focus more on learning the correspondence between the source and target sentences. In the experiment, AdMix delivers improvements over translations tasks at different scales. Experimental results on Chinese-English, German-English, and English-German translation tasks demonstrate the capability of our approach to improve both translation performance and robustness. In the future, besides focusing on the machine translations task in this paper, we are interested in exploring the application of natural language understanding tasks, such as the GLUE benchmark.

¹<https://www.statmt.org/wmt13/training-monolingual-news-2007.tgz>

Acknowledgments

We would like to thank the anonymous reviewers for the helpful comments. This work was supported by Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

- [Anaby-Tavor *et al.*, 2020] Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. Do not have enough data? deep learning to the rescue! In *AAAI*, volume 34, pages 7383–7390, 2020.
- [Artetxe *et al.*, 2017] Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. Unsupervised neural machine translation. *arXiv preprint arXiv:1710.11041*, 2017.
- [Cheng *et al.*, 2018] Yong Cheng, Zhaopeng Tu, Fandong Meng, Junjie Zhai, and Yang Liu. Towards robust neural machine translation. In *ACL*, pages 1756–1766, 2018.
- [Cheng *et al.*, 2020] Yong Cheng, Lu Jiang, Wolfgang Macherey, and Jacob Eisenstein. AdvAug: Robust adversarial augmentation for neural machine translation. In *ACL*, pages 5961–5970, 2020.
- [Clark *et al.*, 2018] Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. Semi-supervised sequence modeling with cross-view training. In *EMNLP*, pages 1914–1925, 2018.
- [DeVries and Taylor, 2017] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.
- [Edunov *et al.*, 2018] Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and Marc’Aurelio Ranzato. Classical structured prediction losses for sequence to sequence learning. In *NAACL*, pages 355–364, 2018.
- [Gao *et al.*, 2019] Fei Gao, Jinhua Zhu, Lijun Wu, Yingce Xia, Tao Qin, Xueqi Cheng, Wengang Zhou, and Tie-Yan Liu. Soft contextual data augmentation for neural machine translation. In *ACL*, pages 5539–5544, 2019.
- [Guo *et al.*, 2020] Demi Guo, Yoon Kim, and Alexander Rush. Sequence-level mixed sample data augmentation. In *EMNLP*, pages 5547–5552, 2020.
- [He *et al.*, 2020] Junxian He, Jiatao Gu, Jiajun Shen, and Marc’Aurelio Ranzato. Revisiting self-training for neural sequence generation. In *ICLR*, 2020.
- [Hendrycks *et al.*, 2020] Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple method to improve robustness and uncertainty under data shift. In *ICLR*, 2020.
- [Kannan *et al.*, 2018] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.

- [Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.
- [Kumar *et al.*, 2020] Varun Kumar, Ashutosh Choudhary, and Eunah Cho. Data augmentation using pre-trained transformer models. In *AAACL*, pages 18–26, 2020.
- [Lample *et al.*, 2018] Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc’Aurelio Ranzato. Unsupervised machine translation using monolingual corpora only. In *ICLR*, 2018.
- [Li *et al.*, 2021] Bohan Li, Yutai Hou, and Wanxiang Che. Data augmentation approaches in natural language processing: A survey. *arXiv preprint arXiv:2110.01852*, 2021.
- [Miyato *et al.*, 2017] Takeru Miyato, Andrew M. Dai, and Ian J. Goodfellow. Adversarial training methods for semi-supervised text classification. In *ICLR*, 2017.
- [Papineni *et al.*, 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318, 2002.
- [Sennrich *et al.*, 2016a] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *ACL*, pages 86–96, 2016.
- [Sennrich *et al.*, 2016b] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *ACL*, pages 1715–1725, 2016.
- [Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.
- [Wang *et al.*, 2018] Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. SwitchOut: an efficient data augmentation algorithm for neural machine translation. In *EMNLP*, pages 856–861, 2018.
- [Wei and Zou, 2019] Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *EMNLP*, pages 6382–6388, 2019.
- [Xie *et al.*, 2017] Ziang Xie, Sida I Wang, Jiwei Li, Daniel Lévy, Aiming Nie, Dan Jurafsky, and Andrew Y Ng. Data noising as smoothing in neural network language models. *arXiv preprint arXiv:1703.02573*, 2017.
- [Xie *et al.*, 2020] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V. Le. Unsupervised data augmentation for consistency training. In *NeurIPS*, 2020.
- [Zhang *et al.*, 2017] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.