Prediction Difference Regularization against Perturbation for Neural Machine Translation

Dengji Guo[1][2], **Zhengrui Ma**[1][2], **Min Zhang**[3], **Yang Feng**[1][2]*

[1] Key Laboratory of Intelligent Information Processing,
Institute of Computing Technology, Chinese Academy of Sciences (ICT/CAS)
[2] University of Chinese Academy of Sciences, Beijing, China
[3] Harbin Institute of Technology, Shenzhen, China
{guodengji19s,mazhengrui21b,fengyang}@ict.ac.cn
zhangmin2021@hit.edu.cn

Abstract

Regularization methods applying input perturbation have drawn considerable attention and have been frequently explored for NMT tasks in recent years. Despite their simplicity and effectiveness, we argue that these methods are limited by the under-fitting of training data. In this paper, we utilize prediction difference for ground-truth tokens to analyze the fitting of token-level samples and find that under-fitting is almost as common as over-fitting. We introduce prediction difference regularization (PD-R), a simple and effective method that can reduce over-fitting and under-fitting at the same time. For all token-level samples, PD-R minimizes the prediction difference between the original pass and the input-perturbed pass, making the model less sensitive to small input changes, thus more robust to both perturbations and under-fitted training data. Experiments on three widely used WMT translation tasks show that our approach can significantly improve over existing perturbation regularization methods. On WMT16 En-De task, our model achieves 1.80 SacreBLEU improvement over vanilla transformer.

1 Introduction

Neural machine translation models have achieved great success in recent years (Sutskever et al., 2014; Bahdanau et al., 2015; Gehring et al., 2017; Vaswani et al., 2017). Despite their efficiency and superb performance, NMT models are prone to over-fitting that universal regularization techniques such as dropout (Hinton et al., 2012) and label smoothing (Szegedy et al., 2016) have been indispensable. However, over-fitting is still a significant problem for NMT, especially for small and medium tasks, which motivates researchers to constantly explore more specialized and sophisticated regularization techniques.

Particularly, regularization methods applying input perturbation have been frequently explored for NMT models in recent years (Bengio et al., 2015; Wu et al., 2019; Sato et al., 2019; Takase and Kiyono, 2021). In these methods, neural models are trained to maximize the likelihood of perturbed samples that perturbed by a certain type of perturbations, with a primary intention to enhance model's robustness to perturbations, since neural models have been discovered fragile to small input noises (Szegedy et al., 2014; Liang et al., 2018; Belinkov and Bisk, 2018). In the past few years, many types of perturbations have been proposed to machine translation and been shown effective, including word-dropout (Gal and Ghahramani, 2016), word-replacement (Bengio et al., 2015; Wu et al., 2019) and adversarial perturbation (Miyato et al., 2017; Sato et al., 2019), etc.

In this paper, unlike previous works which are devoted to finding stronger perturbations and more appropriate perturbation schedules, we rethink the existing perturb-and-fit mechanism and prove that indiscriminate fitting of perturbed samples ignores and aggravates under-fitting, which dramatically limits the effectiveness of perturbation regularization. We further propose prediction difference regularization (PD-R), a simple and effective method that can alleviate over-fitting and under-fitting at the same time and significantly enhance the effectiveness of perturbation regularization.

Specifically, we use the prediction difference for ground-truth labels before and after input perturbation as an indicator of over-fitting and under-fitting for token-level samples. Quantitative analysis shows that a considerable part of token-level predictions get improved after input perturbation, indicating that the model is less fitted to those original samples compared to the perturbed samples, which has been ignored by previous works. We then divide labels in a batch into relatively under-fitted and over-fitted subsets according to real-time

[*] Yang Feng is the corresponding author of the paper.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 7665 - 7675
May 22-27, 2022 ©2022 Association for Computational Linguistics

prediction difference and train only one subset to fit the perturbed inputs and the other subset to fit the original inputs. Experiments show that training only the relatively under-fitted subset to further fit the perturbed inputs dramatically degrade the model performance, while the opposite gets better results than the existing indiscriminate way. This indicates that existing methods are hindered by the excessive fitting of perturbed data.

We further propose to use prediction difference as a regularization term, where the prediction difference is the divergence of prediction distribution caused by input perturbation. Since the value of prediction difference reflects the severity of over-fitting or under-fitting, both of which are cases we want to avoid for training models, regularizing prediction difference has been a natural solution to avoid above fitting problems. By combining cross-entropy loss and the prediction difference term, a model can be trained to fit training data with control of over-fitting and under-fitting.

We apply PD-R on simplest word dropout regularization and conduct experiments on three widely used WMT translation tasks covering small-scale, medium-scale, and large-scale data sets. Our method significantly improves over existing perturbation regularization methods. On WMT16 En-De translation task, our method achieves 1.80 SacreBLEU improvement over vanilla transformer model and 1.12 SacreBLEU over traditional word dropout regularization.

2 Background

In this section, we introduce basic principles of neural machine translation, representative types of perturbations as well as their training objectives.

2.1 Neural Machine Translation

For NMT, the probability of a target sentence $Y = y_{1:J}$ conditioned on its parallel source sentence $X = x_{1:I}$ is established based on chain rule:

$$p(Y|X, \boldsymbol{\theta}) = \prod_{j=1}^{J+1} p(y_j|y_{0:j}, X, \boldsymbol{\theta}), \quad (1)$$

where $\boldsymbol{\theta}$ represents the parameters of the model, y_0 and y_{J+1} are special tokens representing the beginning and end of a sentence respectively.

On this basis, NMT models are trained with the cross-entropy loss to minimize the negative log-likelihood of all samples in the training set $\mathcal{D}$:

$$\mathcal{L} = \mathcal{L}(\mathcal{D}, \boldsymbol{\theta}) = -\frac{1}{\mathcal{D}} \sum_{(X,Y)\in\mathcal{D}} \ell(X, Y, \boldsymbol{\theta}), \quad (2)$$

where $\mathcal{D} = \{(X_n, Y_n)\}_{n=1}^{|V|}$, $|V|$ is the size of the data set, $\ell(X, Y, \boldsymbol{\theta}) = \log p(Y|X, \boldsymbol{\theta})$.

2.2 Types of perturbations

Word Dropout and Replacement The simplest way to apply perturbation is to mask or replace one or more tokens of the original input sequence. The resulting sequence $\hat{x}$ is sampled from the original sequence and the perturbation sequence:

$$\hat{x_i} = \begin{cases} x_i, & \text{with probability 1 - } \alpha, \\ x_i^p, & \text{with probability } \alpha, \end{cases} \quad (3)$$

where $0 < \alpha < 1$ is the hyper-parameter of bernoulli sampling and x_i^p is the i-th word of the perturbation sequence. Note that the perturbation sequence x^p consists of zero vectors for word dropout (Gal and Ghahramani, 2016), and consists of random words sampled from the vocabulary with uniform or a particular distribution for word replacement (Bengio et al., 2015; Wu et al., 2019).

Adversarial Perturbation Adversarial Training (AdvT) tries to make perturbation that maximize the loss function, which is believed more effective for regularization. As described in Miyato et al. (2017) and Sato et al. (2019), the perturbed input embedding for x_i can be computed as follows:

$$\hat{\mathbf{e}}_i = \mathbf{e}_i + \kappa\hat{\mathbf{r}}_i, \quad (4)$$

where $\mathbf{e}_i$ is original embedding of i-th source word, κ is a scalar hyper-parameter that controls the norm of the perturbation, and $\hat{\mathbf{r}_i}$ is the worst case unit perturbation vector approximated by gradient back-propagation(Goodfellow et al., 2015):

$$\hat{\mathbf{r}}_i = \frac{\mathbf{g}_i}{||\mathbf{g}_i||_2}, \quad \mathbf{g}_i = \nabla_{\mathbf{e}_i}\mathcal{L}(\mathcal{D}, \boldsymbol{\theta}), \quad (5)$$

where $\mathbf{g}_i$ is the gradient of a model's loss function with respect to its input embedding $\mathbf{e}_i$.

In most cases, the inputs of the decoder side can also be perturbed in the same way as the encoder side. For scheduled sampling (Bengio et al., 2015) however, perturbation is limited at the decoder side.

2.3 Training Objectives of Perturbation Regularization

For word dropout and word replacement, the model is trained to fit the perturbed samples $\hat{X}$:

$$\mathcal{L} = \mathcal{L}(\hat{\mathcal{D}}, \boldsymbol{\theta}) = -\frac{1}{\mathcal{D}} \sum_{(\hat{X},Y)\in\hat{\mathcal{D}}} \ell(\hat{X}, Y, \boldsymbol{\theta}), \quad (6)$$

where $\hat{D}$ is the perturbed data set.

For adversarial training, two forward passes and two backward passes are required for computing perturbation vectors and then training with them. The model is trained to fit both the original samples and adversarial samples with loss function:

$$\mathcal{L} = \mathcal{L}(\mathcal{D}, \boldsymbol{\theta}) + \lambda L(\hat{\mathcal{D}}, \boldsymbol{\theta}), \quad (7)$$

where λ is a hyper-parameter. Here samples are perturbed at the embedding layer, rather than token-level perturbation at the input layer.

3 Prediction Difference Fitting Analysis

It is usually believed that a model's prediction for ground-truth target tokens will be hindered when the input is perturbed, which has been the initial motivation for perturbation regularization. However, this conclusion is not necessarily true in experiment for many reasons: Firstly, neural networks are complex and may not behave in an ideally logical way. Secondly, perturbations are randomly produced and may have complex properties. For example, word-replacement may induce synonyms or heteronyms, and high-dimension embedding perturbation is hard to interpret. Thirdly, the model is uncertain during training due to parameter dropout, which further brings uncertainty to the model's reaction to perturbations.

In this section, we analyze the influence of perturbations leveraging token-level prediction difference. Here the prediction difference is defined as the change of model's prediction probabilities for ground-truth target tokens (Gu and Tresp, 2019; Li et al., 2019a):

$$\begin{aligned}\Delta p(y_j) = & p(y_j|y_{0:j-1}, X, \boldsymbol{\theta}) \\ & - p(y_j|\hat{y}_{0:j-1}, \hat{X}, \boldsymbol{\theta}).\end{aligned} \quad (8)$$

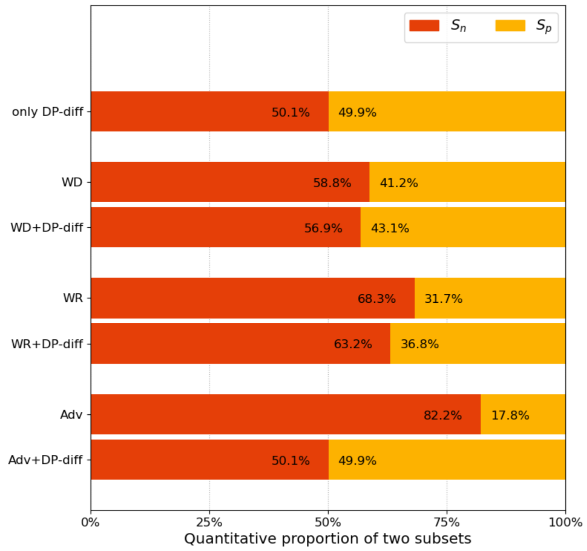
We apply random perturbations to samples in the test set and divide all target labels into two subsets according to their prediction change: positively influenced subset $\boldsymbol{S}_p$ containing labels whose prediction probabilities get bigger after input perturbation and negatively influenced subset $\boldsymbol{S}_n$ containing labels whose prediction probabilities get smaller. We compute the quantitative proportion and the average value of Δp for these two subsets to evaluate the influence of different perturbations.

Since the prediction difference is also under the influence of parameter dropout during training, we also conduct experiments both with and without parameter dropout difference. The original pass and the perturbed pass are carried out by two different sub-models if their parameter dropout mask is different. In experiment, We use a transformer base model trained on WMT16 En-De data set and conduct our experiments on a test set which is a combination of 5 test sets from WMT16 to WMT20. Our analysis covers different kinds of perturbations, including word-dropout, word-replacement, and adversarial perturbation. The word-dropout and word-replacement probabilities are set as 0.05, and all perturbations are applied on both sides of the model.

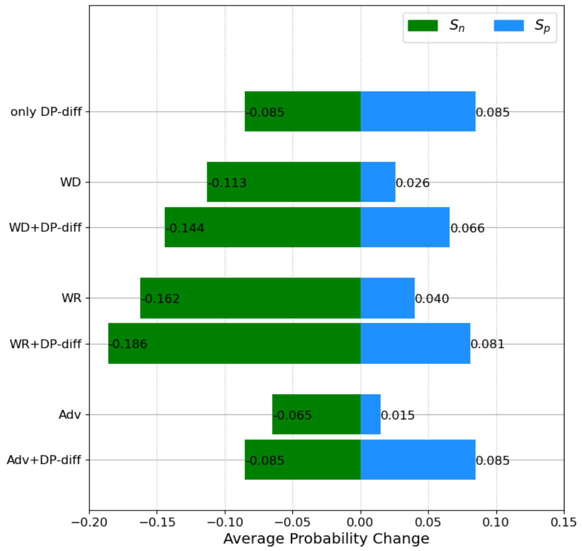
As illustrated in figure 1, for any certain type of perturbation, the negative impact is principal, especially for adversarial perturbation. However, the positive influence is non-negligible since the proportion of positively influenced tokens could reach 30%-40% for word-dropout and word-replacement. With parameter dropout difference, the positive influence could get further bigger and become more crucial.

We attribute prediction difference to relatively over-fitting and under-fitting of token-level samples. Since perturbations are very small, perturbed samples can be approximately viewed as good samples. For one target label, if its prediction probability gets smaller after a small input perturbation, it indicates the model is relatively over-fitted to the original sample, while the contrary case is the reflection of relatively under-fitting. With parameter dropout, predictions are carried out by sub-models, and these fitting problems also reflect the relative fitting bias of sub-models, which is also what we want to avoid.

As mentioned above, existing perturbation regularization methods are based on the motivation to enhance the model's performance against input perturbation and avoid over-fitting. However, experiments show that a model could be better fitted to the perturbed data rather than the original data, which is regarded by us as a sign of relatively under-fitting. This indicates that training a model



(a)



(b)

Figure 1: Influence of different perturbations on token-level label prediction. '$\boldsymbol{S}_n$' represents the negatively influenced set, while '$\boldsymbol{S}_p$' represents the positively influenced set. 'DP-diff', 'WD', 'WR' and 'Adv' represents parameter dropout difference, word-dropout, word-replacement and adversarial perturbation respectively. (a) Quantitative proportion of $\boldsymbol{S}_p$ and $\boldsymbol{S}_n$. (b) Average probability change normalized by subset size for $\boldsymbol{S}_p$ and $\boldsymbol{S}_n$.

	En→Ro	En→De
Transformer	33.4	32.55
only $\boldsymbol{S}_p$	31.59	30.16
only $\boldsymbol{S}_n$	**34.57**	**33.73**
both	34.53	33.20

Table 1: BLEU (Papineni et al., 2002) for selective training of word-dropout perturbation on WMT16 En-Ro and WMT16 En-De translation tasks. Evaluation set for En-De is a combination of 5 test sets form WMT16 to WMT20.

with perturbed data may not be necessary for some circumstances.

We further carry out selective training for word-dropout regularization, where one subset is trained to fit the perturbed inputs and the other subset is trained to fit the original inputs. As presented in table 1, training only $\boldsymbol{S}_n$ gets better results than existing indiscriminate training, while training only $\boldsymbol{S}_p$ gets worse results than vanilla transformer. This implies that the existing method suffers from degeneration caused by aggravated under-fitting.

4 Prediction Difference Regularization (PD-R)

Since both positive and negative prediction difference is a sign of improper fitting of samples, we therefore propose prediction difference regularization (PD-R), to regularize the model directly with the prediction difference:

$$\begin{aligned}\ell_{PD-R}(X,Y,\boldsymbol{\theta}) =&\mathcal{R}[P(*|X,Y_<,\boldsymbol{\theta}^{'}),\\ &P(*|\hat{X},\hat{Y}_<,\boldsymbol{\theta}^{''})],\end{aligned} \tag{9}$$

where $\mathcal{R}[\cdot]$ is the distance of two distributions, (X,Y) is a sample from data set $\mathcal{D}$, " $*$ " represents all prediction steps, $P(*|X,Y_<,\boldsymbol{\theta}^{'})$ is the prediction distributions for all steps conditioned on original source input X, target teacher forcing target input $Y_<$ and sub-model with parameters $\boldsymbol{\theta}^{'}$, and $P(*|\hat{X},\hat{Y}_<,\boldsymbol{\theta}^{''})$ is the prediction distributions for all steps conditioned on perturbed source input $\hat{X}$, perturbed target teacher forcing target input $\hat{Y}_<$ and sub-model with parameters $\boldsymbol{\theta}^{''}$. The total regularization loss is averaged over all samples in the data set:

$$\mathcal{L}_{PD-R}(\mathcal{D},\boldsymbol{\theta}) = \frac{1}{\mathcal{D}}\sum_{(X,Y)\in\mathcal{D}}\ell_{PD-R}(X,Y,\boldsymbol{\theta}). \tag{10}$$

The model is trained with a combination of cross-entropy loss and regularization term:

$$\mathcal{L} = \mathcal{L}(\mathcal{D},\boldsymbol{\theta}) + \gamma\mathcal{L}_{PD-R}(\mathcal{D},\boldsymbol{\theta}), \tag{11}$$

where γ is a hyper-parameter controlling the weight of regularization.

	WMT2016 En→Ro		WMT2017 Zn→En	
	2016	Δ	2017	Δ
Transformer (Vaswani et al., 2017)	33.16	–	23.98	–
Word-Drop	34.13	+ 0.97	24.20	+ 0.22
SSE-SE (Wu et al., 2019)	33.75	+ 0.59	24.14	+ 0.16
Scheduled Sampling (Bengio et al., 2015)	33.62	+ 0.46	23.74	– 0.24
AdvT (Sato et al., 2019)	33.65	+ 0.49	24.17	+ 0.19
R-Drop (Liang et al., 2021)	34.14	+ 0.96	**25.08**	+ **1.10**
Word-Drop + ST	34.24	+ 1.08	24.37	+ 0.39
Word-Drop (enc) + PD-R	34.22	+ 1.06	**24.98**	+ **1.00**
Word-Drop (dec) + PD-R	34.57	+ 1.41	24.76	+ 0.78
Word-Drop (both) + PD-R	**34.93**	+ **1.77**	24.86	+ 0.88

Table 2: SacreBLEU for different models on WMT16 En-Ro and WMT17 Zh-En tasks.

In experiment, we apply PD-R on simplest word-dropout perturbation with $\alpha = 0.05$ in Eq.(3) and $\gamma = 1.0$ in Eq.(11) without further hyper-parameter search. $\mathcal{R}[\cdot]$ in Eq.(9) is implemented as L1 distance, which performs slightly better than KL-divergence in our experiments.

5 Experiments

We evaluate PD-R on three public WMT machine translation tasks and compare it with representative related works.

5.1 Data Sets

To fully verify the effectiveness of our method on NMT, we conduct experiments on three machine translation tasks, including small-scale WMT16 English-Romanian(En-Ro), medium-scale WMT16 English-German (En-De), and large-scale WMT17 Chinese-English (Zh-En).

English-Romanian This data set contains about 0.6M processed parallel sentence pairs tokenized by Moses toolkit (Koehn et al., 2007) and segmented with 40K merge operations using BPE (Sennrich et al., 2016). We use news-dev 2016 and news-test 2016 as the validation set and test set respectively.

English-German The WMT16 En-De data set consists of about 4.5M parallel sentences pairs coded with 30K BPE merge-operations. For evaluation, we average the last 5 epochs and report results on all test sets from WMT2016 to WMT2020.

Chinese-English Our data set consists of over 20M parallel sentence pairs. The English and Chinese sentences are tokenized with Moses toolkit and Stanford Segmenter respectively, which are further applied 32K BPE segmentation. We use newsdev2016 for validation and newstest2017 for testing.

5.2 Configuration

To fairly compare each method, we reproduce all compared methods with transformer model (Vaswani et al., 2017) using open-source toolkit *Fairseq* (Ott et al., 2019), with the same model configuration and hardware facilities.

We use transformer `base` configuration for all experiments, with 6 encoder and decoder layers, 512 hidden dimensions, 8 attention heads and 2048 FFN dimensions. We train all models with 4000 warm-up steps, initial learning rate of $7e^{-4}$, label smoothing factor of 0.1, and Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 1e^{-9}$ as Vaswani et al. (2017). We set the dropout rate to 0.2 for small-scale En-Ro task and 0.1 for En-De and Zh-En tasks. All experiments are conducted on 4 GeForce RTX 3090 GPUs with a distributional batch-size of 4096 tokens each GPU and an overall accumulated batch-size of 4096 × 8 tokens. During inference, we use beam size of 4 and length penalty of 0.6 for all tasks.

For En-Ro and En-De translation tasks, we share the vocabulary for source and target and apply three-way weight tying(TWWT) (Press and Wolf, 2017) for training, the vocabulary sizes of both tasks are limited to 32768 tokens. We train models for 50 epochs for both tasks. For Zh-En translation task, the Chinese and English vocabulary sizes are 44K and 33K respectively, and models are trained for 300K steps.

5.3 Compared Methods

We reproduce four representative perturbation regularization methods and recently proposed R-Drop for comparison.

	WMT2016 En→De						
	2016	2017	2018	2019	2020	AVG	Δ
Transformer (Vaswani et al., 2017)	33.81	27.75	40.56	36.39	21.95	32.09	–
Word-Drop	34.14	28.00	41.07	38.04	22.62	32.77	+ 0.68
SSE-SE (Wu et al., 2019)	33.90	27.95	41.23	36.93	22.66	32.53	+ 0.44
Scheduled Sampling (Bengio et al., 2015)	33.96	28.12	41.13	37.39	22.58	32.63	+ 0.54
AdvT (Sato et al., 2019)	34.25	27.91	41.31	37.05	23.00	32.70	+ 0.61
R-Drop (Liang et al., 2021)	35.32	27.66	41.72	38.26	22.84	33.16	+ 1.07
Word-Drop + ST	34.85	28.23	42.10	38.13	22.74	33.21	+ 1.12
Word-Drop (enc) + PD-R	35.30	28.28	**42.92**	**39.09**	**23.88**	**33.89**	+ **1.80**
Word-Drop (dec) + PD-R	**35.39**	**28.34**	42.14	38.51	23.68	33.61	+ 1.52
Word-Drop (both) + PD-R	35.17	28.23	42.20	38.74	23.61	33.59	+ 1.50

Table 3: SacreBLEU for different models on WMT16 En-De task.

Word-Drop We implement word-dropout (Gal and Ghahramani, 2016) by randomly replace word embeddings with zero vectors with $\alpha = 0.05$ in Eq.(3).

SSE-SE The SSE-SE is a word-replacement method that randomly replaces input tokens with other tokens in vocabulary. As in Wu et al. (2019), we set $\alpha = 0.01$ in Eq.(3) and sample perturbation sequence with uniform distribution.

Scheduled Sampling Scheduled sampling (Bengio et al., 2015) is a word-replacement method that randomly replace target-side input tokens with model predictions. Each model prediction token is sampled using model's output distribution. The replacement rate α follows a curriculum learning strategy:

$$\alpha_i = \frac{k}{k + exp(i/k)}, \tag{12}$$

where i represents training steps, and k is a hyper-parameter depending on the speed of convergence. Our implementation of scheduled sampling for transformer is parallel as in Mihaylova and Martins (2019) and Duckworth et al. (2019). We set $k = (4590, 29350, 36150)$ for En-Ro, En-De and Zh-En tasks respectively. The hyper-parameter k is set to make sure that α_i is decayed to 0.9 at the end of training.

AdvT For adversarial training, we set $\kappa = 1$ in Eq.(4) and $\lambda = 1$ in Eq.(7) as Sato et al. (2019).

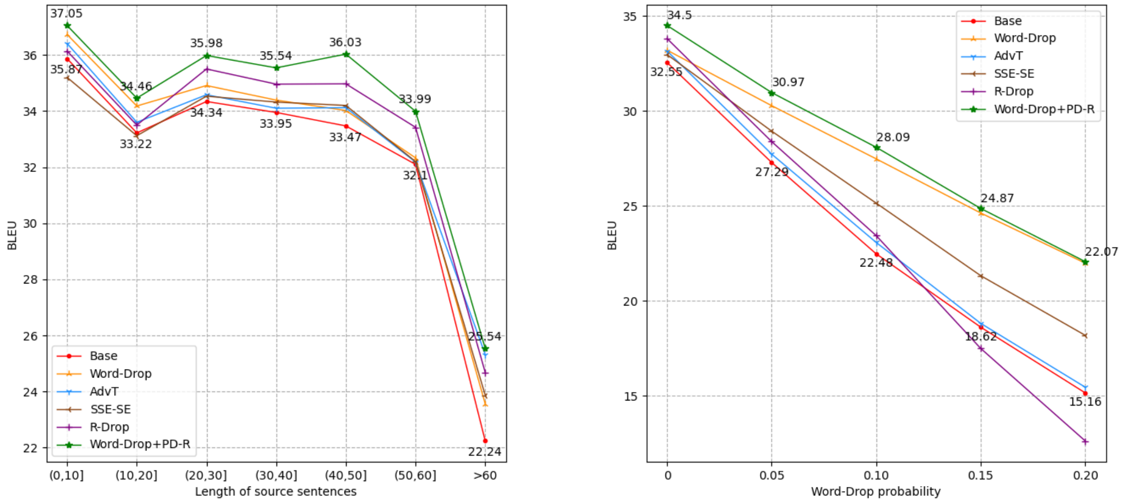
R-Drop R-Drop (Liang et al., 2021) is a very recent work whose implementation is similar to PD-R. However, its motivation is to restrict the freedom of parameters by reducing sub-model divergence, while ours is to avoid token-level sample fitting problems reflected by prediction difference. Since predictions are carried out by sub-models during training, the fitting bias of sub-models is also included in the prediction difference. From this point, R-Drop can be viewed as a sub-component of PD-R.

5.4 Main Results

SacreBLEU (Post, 2018) of compared methods and PD-R on three translation tasks are illustrated in table 2 and table 3. We apply PD-R on encoder-side word-dropout, decoder-side word-dropout, and both-side word-dropout. For all compared methods involving input perturbation, perturbation is applied on both sides of the model except scheduled sampling. Note that selective training of word-dropout regularization (only $\boldsymbol{S}_n$, referred as 'ST') is also presented for comparison with Word-Drop and PD-R.

Experiments show that existing perturbation regularization methods are similarly effective compared to each other, which is consistent with Takase and Kiyono (2021). R-Drop and selective training(ST) of word-dropout regularization are consistently better than existing perturbation regularization. Our PD-R against word-dropout significantly improves over word-dropout and other perturbation regularization methods on all three tasks, and also performs better than R-Drop on small-scale and medium-scale tasks. On WMT16 En-De, PDR achieves 1.80 SacreBLEU improvement over vanilla transformer, 1.12 SacreBLEU improvement over existing word-dropout perturbation regularization, and 0.73 SacreBLEU improvement over R-Drop.

On large scale WMT17 Zh-En task though, the improvement of perturbation regularization gets smaller compared to small and medium tasks, and



(a)

(b)

Figure 2: BLEU of models (a) on sentences of different lengths, (b) under different levels of word-dropout perturbation. Experiments are conducted on WMT16 En-De with a combined test set.

R-Drop performs better than PD-R. We attribute it to the fact that large-scale tasks are sufficient in data, regularization in data level has become a burden rather than help while regularizing sub-model bias is still beneficial.

6 Analysis

In this section, we analyze the robustness of our methods and distinguish the contribution of different components via ablation study.

6.1 Performance on long sentences

Longer sentences contain more complex word combinations that are unseen or seldom seen in the training set and suffer more from exposure bias (Ranzato et al., 2016; Zhang et al., 2019). Performance on long sentences reflects the model's robustness to unexpected inputs.

In experiment, we evaluate the performance of different models on WMT16 En-De task. We combine 5 test sets from WMT16 to WMT20 and divide samples into 7 subsets according to sentence length. As shown in figure 2a, PD-R achieves better results in all subsets, and the improvement tends to become larger as the sentence length grows, which implies that PD-R can better handle unexpected inputs of long sentences.

6.2 Robustness against perturbation

To better evaluate model's robustness to perturbations, we conduct perturbation attack for all models, similar as Michel and Neubig (2018) and Moradi and Samwald (2021). In experiment, we apply word-dropout on source sentences and generate target sentences based on perturbed source sentences. Experiment results in figure 2b show that PD-R against word-dropout and existing word-dropout regularization are consistently better than the base model, and the gap becomes larger as the proportion of perturbation grows, which confirms that our approach does improve the model's robustness to perturbation.

Note that our experiments on other types of perturbation attack conclude that a model is robust to a certain type of perturbation only if the model is trained on this kind of perturbation, so comparison of different perturbation regularization methods under one certain type of perturbation attack is not the focus of our discussion in this subsection.

6.3 Ablation Study

Ablation study in table 4 shows that training only the positively influenced subset S_p using PD-R is also effective, even more effective than training only S_n. This indicates that PD-R can properly handle both under-fitting and over-fitting.

As mentioned in section 3, sub-model bias is also a source of improper fitting problems. To dis-

	En→Ro	En→De
Transformer	33.40	32.55
only $\boldsymbol{S}_p$	34.83	34.22
only $\boldsymbol{S}_n$	34.59	34.16
both	35.11	34.50
only DP-diff	**34.70**	**34.10**
WD(enc) w/o DP-diff	33.97	33.97
WD(dec) w/o DP-diff	34.15	33.15
WD(both) w/o DP-diff	**34.40**	**34.02**
WD(enc) w/ DP-diff	34.63	**34.50**
WD(dec) w/ DP-diff	34.92	34.19
WD(both) w/ DP-diff	**35.11**	34.22

Table 4: BLEU for ablation study of PD-R, where 'WD' represents word-dropout, 'w/o DP-diff' represents that the two passes share the same parameter dropout mask, 'w/ DP-diff' represents that two different sub-models are used for the two passes.

tinguish the contribution of parameter dropout and word-dropout, we conduct experiments where the difference of two passes is restricted to only parameter dropout or only word-dropout. We also conduct experiments on the encoder side and decoder side separately. Experiment results show that parameter dropout is an important source of improvement, word-dropout is nearly as important as parameter dropout for PD-R, while using both of them gets the best results. As for the difference between the encoder side and decoder side, the decoder-side word-dropout contributes more on the En-Ro task, while on the En-De task the contribution of the encoder side is much bigger, this is also true when the two passes have no parameter dropout difference. The encoder side gets more important on larger data set, which is consistent with the main results.

7 Related Work

Works involving Input Perturbation Apart from the works mentioned above, some works introduce subword uncertainty at the subword segmentation stage, including sampling multiple subword candidates (Kudo, 2018), applying subword dropout (Park et al., 2020) or producing adversarial subword segmentation (Provilkov et al., 2020). For character-level tasks, there are also works using character-level perturbation including character-level random deletion, insertion, substitution and swap (Belinkov and Bisk, 2018; Karpukhin et al., 2019) and adversarial substitution (Ebrahimi et al., 2018). The mixup technique for NLP tasks can also be seen as a form of perturbation where samples are perturbed (mixed) with other samples for data augmentation or generation diversity (Guo et al., 2020; Li et al., 2021; Fang et al., 2022).

Our work can be regarded as one example of perturbation regularization. However, unlike previous perturbation regularization works which are focused on finding better perturbation, our work improves the training mechanism and can be applied to any type of perturbations.

Influence of Perturbation Perturbation is commonly considered as a negative factor for neural models by previous works (Szegedy et al., 2014; Liang et al., 2018; Belinkov and Bisk, 2018), which is generally correct with the fact that perturbation does degrade the training and inference accuracy of a model. Belinkov and Bisk (2018) demonstrates that the performance of NMT systems degrades monotonously as input modification increases, which is consistent with our observations. Based on the above facts, perturbation regularization is frequently studied to enhance models' robustness to unexpected inputs at the inference stage. From a data selection perspective, Khayrallah and Koehn (2018) and Briakou and Carpuat (2021) demonstrate that noisy or semantically divergent data is harmful to the training of NMT models. In this paper, we find that the interaction between perturbation and model is complicated and positive influence of perturbation is very common, which is further regarded by us as a sign of relatively under-fitting and a variable that needs to be restricted.

Prediction Difference Prediction difference is usually considered as a reflection of the relationship between input and output and is often used to analyze model behavior. Zintgraf et al. (2017) utilizes prediction difference to visualize the importance of a specific input image area to model decision. Li et al. (2019b) uses the prediction difference of a target word when a source word is removed to induce word alignment and find it more accurate than attention weights. Guo et al. (2019) finds that adversarial examples can be accurately and efficiently detected via prediction difference. Liang et al. (2021) proposes R-Drop and take prediction difference as a regularization term to regularize sub-model divergence. In this work, prediction difference is used as an analytical tool to detect improper fitting problems and also a regularization term to regularize the model's fitting bias to token-level samples.

8 Conclusion

In this paper, we propose to use probability difference for ground-truth tokens before and after input perturbation as an indicator to analyze the influence of different types of perturbations and attribute probability difference to improper fitting of token-level samples. We find that under-fitting is almost as common as over-fitting, which is totally ignored and further aggravated by existing perturbation regularization methods. To regularize both under-fitting and over-fitting, we use prediction difference as a regularization term (PD-R) and apply it on word-dropout regularization. Our method achieves significant improvement over existing methods on three WMT translation tasks and is proved more robust to input perturbation.

Acknowledgements

We thank all the anonymous reviewers for their insightful and valuable comments. This work was supported by National Key R&D Program of China (NO. 2017YFE0192900).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1171–1179.

Eleftheria Briakou and Marine Carpuat. 2021. Beyond noise: Mitigating the impact of fine-grained semantic divergences on neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7236–7249, Online. Association for Computational Linguistics.

Daniel Duckworth, Arvind Neelakantan, Ben Goodrich, Lukasz Kaiser, and Samy Bengio. 2019. Parallel scheduled sampling. *CoRR*, abs/1906.04331.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.

Qingkai Fang, Rong Ye, Lei Li, Yang Feng, and Mingxuan Wang. 2022. STEMM: Self-learning with Speech-text Manifold Mixup for Speech Translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 1019–1027.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252. PMLR.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Jindong Gu and Volker Tresp. 2019. Contextual prediction difference analysis. *CoRR*, abs/1910.09086.

Demi Guo, Yoon Kim, and Alexander Rush. 2020. Sequence-level mixed sample data augmentation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5547–5552, Online. Association for Computational Linguistics.

Feng Guo, Qingjie Zhao, Xuan Li, Xiaohui Kuang, Jianwei Zhang, Yahong Han, and Yu-an Tan. 2019. Detecting adversarial examples via prediction difference for deep neural networks. *Inf. Sci.*, 501:182–192.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580.

Vladimir Karpukhin, Omer Levy, Jacob Eisenstein, and Marjan Ghazvininejad. 2019. Training on synthetic noise improves robustness to natural noise in machine translation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 42–47, Hong Kong, China. Association for Computational Linguistics.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83, Melbourne, Australia. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Jicheng Li, Pengzhi Gao, Xuanfu Wu, Yang Feng, Zhongjun He, Hua Wu, and Haifeng Wang. 2021. Mixup decoding for diverse machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 312–320, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xintong Li, Guanlin Li, Lemao Liu, Max Meng, and Shuming Shi. 2019a. On the word alignment from neural machine translation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1293–1303. Association for Computational Linguistics.

Xintong Li, Guanlin Li, Lemao Liu, Max Meng, and Shuming Shi. 2019b. On the word alignment from neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1293–1303, Florence, Italy. Association for Computational Linguistics.

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2018. Deep text classification can be fooled. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4208–4215. ijcai.org.

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.

Paul Michel and Graham Neubig. 2018. MTNT: A testbed for machine translation of noisy text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 543–553, Brussels, Belgium. Association for Computational Linguistics.

Tsvetomila Mihaylova and André F. T. Martins. 2019. Scheduled sampling for transformers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 351–356, Florence, Italy. Association for Computational Linguistics.

Takeru Miyato, Andrew M. Dai, and Ian J. Goodfellow. 2017. Adversarial training methods for semi-supervised text classification. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Milad Moradi and Matthias Samwald. 2021. Evaluating the robustness of neural language models to input perturbations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 1558–1570. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

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*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Jungsoo Park, Mujeen Sung, Jinhyuk Lee, and Jaewoo Kang. 2020. Adversarial subword regularization for robust neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1945–1953, Online. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163, Valencia, Spain. Association for Computational Linguistics.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online. Association for Computational Linguistics.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Motoki Sato, 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, Florence, Italy. Association for Computational Linguistics.

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 (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826. IEEE Computer Society.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

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, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Liwei Wu, Shuqing Li, Cho-Jui Hsieh, and James L. Sharpnack. 2019. Stochastic shared embeddings: Data-driven regularization of embedding layers. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 24–34.

Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. 2019. Bridging the gap between training and inference for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4334–4343, Florence, Italy. Association for Computational Linguistics.

Luisa M. Zintgraf, Taco S. Cohen, Tameem Adel, and Max Welling. 2017. Visualizing deep neural network decisions: Prediction difference analysis. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.