Manifold Adversarial Augmentation for Neural Machine Translation

Guandan Chen, Kai Fan, Kaibo Zhang, Boxing Chen, Zhongqiang Huang
Alibaba DAMO Academy
{guandan.cgd, k.fan, kaibo.zkb}@alibaba-inc.com
{boxing.cbx, z.huang}@alibaba-inc.com

Abstract

Improving the robustness of neural machine translation models on variations of input sentences is an active area of research. In this paper, we propose a simple data augmentation approach by sampling virtual sentences from the vicinity distributions in higher-level representations, constructed either from individual training samples via adversarial learning or pairs of training samples through mixup. By simplifying and extending previous work that operates at the token level, our method can construct virtual training samples in a broader space and achieve improved translation accuracy compared to the previous state-of-the-art. In addition, we present a simple variation of the mixup strategy to better utilize the pseudo training samples created from back-translation, obtaining further improvement in performance.

1 Introduction

In recent years, neural machine translation (NMT) models (Sutskever et al., 2014; Bahdanau et al., 2014; Vaswani et al., 2017) have dramatically improved the quality of machine translation, especially with the introduction of the seminal Transformer architecture (Vaswani et al., 2017) that has become the de facto modeling choice. NMT training aims to learn a parameterized function that models the prediction of the translation in a target language given a source language sentence from labeled training data, which is often limited in volume especially for low-resource domains or languages. Similar to other fields in deep learning, model robustness is an area of concern for NMT as a minor change in the input sentence may result in a different or incorrect translation. In practice this can happen with spelling or grammar errors (Provilkov et al., 2019), speech recognition errors (Ruiz et al., 2019; Di Gangi et al., 2019), or even a sentence of the same meaning but with a slightly different use of words or expressions. Some studies (Belinkov and Bisk, 2017) have shown that the performance of NMT models can drop significantly when small perturbations are added to input sentences.

This problem can be attributed to overfitting as it is difficult to reliably model the translation distribution for the part of input space that has little or no training samples. There have been several attempts to address this problem by filling the space via data augmentation. One direction is to create new training samples by adding perturbations at the token level (Wang et al., 2018; Belinkov and Bisk, 2017; Sperber et al., 2017; Ebrahimi et al., 2018; Li et al., 2019; Cheng et al., 2018, 2019, 2020; Levy et al., 2019), through either token insertion, deletion, and substitution operations or introducing noises to token embedding vectors. Among these approaches, Cheng et al. (2019) demonstrated the effectiveness of incorporating adversarial training samples that are natural sentences, with their semantic relevance to the original sentence safeguarded by language modeling. Cheng et al. (2020) achieved further improvement by creating more diverse but virtual sentences by mixing up actual training samples or synthesized adversarial samples via interpolation of word embeddings, but again at the token level.

Inspired by the success of manifold mixup in computer vision (Verma et al., 2019) and the recent evidence of separable manifolds in deep language representations (Mamou et al., 2020), we propose to simplify and extend previous work on adversarial learning and mixup augmentation to operate in high-level hidden representations, and as such we name the method *manifold adversarial augmentation*. Specifically, we create adversarial representations on a randomly selected hidden layer to attack the NMT model by adding perturbations based on gradients at a random scale to

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 3184–3189

some randomly selected positions. Because the adversarial representations diverge slightly from the original representation but in many different ways, they can be viewed as many diverse sentences that are different in expression but have similar meanings. We also create virtual samples by mixing up the hidden representations of two randomly selected samples at a randomly selected hidden layer. Similarly, the mixup presentations can be viewed as many diverse sentences that fill the semantic space between the two original samples, which can help obtain smoother decision boundaries in the data space that is less populated. We further extend the mixup strategy to back-translation, another effective data augmentation method for machine translation, creating virtual samples to bridge the gap between pseudo samples and gold samples.

Experiments on the LDC Chinese-English and IWSLT English-French benchmark tasks demonstrate that our method can significantly improve the vanilla Transformer model by more than 4 and 3 BLEU respectively, averaged over multiple data sets for each task. Compared to the recent state-of-the-art AdvAug method in (Cheng et al., 2020), our method achieves an average improvement of 0.39 and 1.10 BLEU respectively. Further improvement can be achieved with the use of back-translation data.

2 Method

As our manifold adversarial augmentation method is closely related to the AdvAug method (Cheng et al., 2020), we start by highlighting, and also depicting in Figure 1, their similarities and differences.

AdvAug uses both adversarial learning and mixup augmentation at the token level. The adversarial samples are obtained by randomly replacing a small subset of input words (on either source or target side) with words adjacent in the direction of the gradient that can also fit in context based on language modeling. The generated adversaries tend to be natural sentences, however, the variation is limited as it cannot deal with word insertion, deletion, reordering, and more general variations in language expression. Their mixup operation creates virtual samples by interpolating the word embedding vectors of two randomly selected training samples or adversarial samples. While it can generate more training samples, it is hard to interpret the virtual samples as representions of natural sentences, limiting its potential in dealing with natural texts.

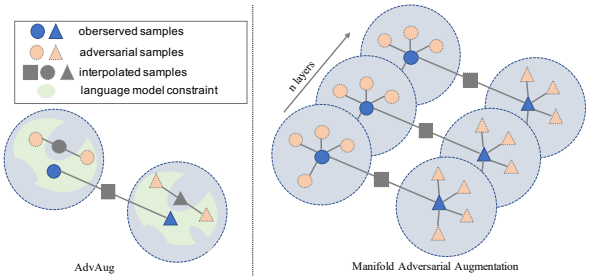

Figure 1: Comparison of the AdvAug method and our manifold adversarial augmentation method.

In contrast, our method operates on higher-level hidden represention for both adversarial learning and mixup augmentation, relying on multiple neural layers to extract semantic meanings, which makes it easier to perform arithmetic operations on semantics. Although we do not explicitly construct adversarial samples that are natural texts, we conjecture that our method has the potential of covering more variations that can occur naturally. We next describe the details of our approach.

2.1 Adversarial Learning

Let $\boldsymbol{x}$ be the input sequence to our model, which could be either a source language sentence or a target language text representing the translation history. We use $\boldsymbol{h}^{(j)}$ and $\boldsymbol{z}^{(k)}$ to denote the hidden representations at the j-th encoder layer and the history portion of the k-th decoder layer, respectively. $\texttt{Enc}_{>j}$ denotes the function composed of the encoder layers higher than j, and $\texttt{Dec}_{>k}$ the function composed of the decoder layers higher than k plus the output layer, which computes the generation distribution of output words[1]. We generate perturbation $\delta_{\boldsymbol{h}^{(j)}}$ to the encoder representations $\boldsymbol{h}^{(j)}$ as follows:

$$\delta_{\boldsymbol{h}^{(j)}} = \gamma * \boldsymbol{\eta} * \boldsymbol{g}^{(j)}$$

where $\boldsymbol{g}^{(j)}$ is the gradient with respect to the NMT training loss L_{nmt} back-propagated at $\boldsymbol{h}^{(j)}$, γ is a hyper-parameter controlling the maximum amount of perturbation, and $\boldsymbol{\eta} = [\eta_i \sim Beta(\alpha_{adv}, \beta_{adv}); 0 < i \leq |\boldsymbol{x}|]$ is a random variable providing more fine-gained control of the perturbation. By setting $\alpha_{adv} < 1$ and $\beta_{adv} < 1$, η_i

[1]Instead of performing manifold adversarial augmentation on a predetermined hidden layer, we choose to randomly select $j \in [0, K_{src}]$ and $k \in [0, K_{tgt}]$ among a range of encoder and decoder layers, allowing more variations. Appendix A.2 examines the effect of different K_{src} and K_{tgt} values on model performance.

can concentrate close to 0 or 1 and act like a gate independently controlling whether to add perturbation at a specific position, mimicking the random selection of positions for word replacement in AdvAug. Similarly, we generate perturbation $\delta_{\boldsymbol{z}^{(k)}}$ to $\boldsymbol{z}^{(k)}$ on the decoder side. The manifold adversarial learning loss L^m_{adv} is computed by:

$$\tilde{\boldsymbol{h}} = \texttt{Enc}_{>j}(\boldsymbol{h}^{(j)} + \delta_{\boldsymbol{h}^{(j)}})$$
$$L^m_{adv} = \mathbb{E}[\texttt{KL}(\texttt{Dec}_{>k}(\tilde{\boldsymbol{h}}, \boldsymbol{z}^{(k)} + \delta_{\boldsymbol{z}^{(k)}}), \boldsymbol{\omega})]$$

Here $\boldsymbol{\omega}$ represents the prediction distribution of NMT model on the original training sample, and we base the adversarial loss on KL-divergence instead of MLE, following the VAT work in (Miyato et al., 2018).

2.2 Mixup Augmentation

Verma et al. (2019) investigated manifold mixup augmentation as a way to leverage semantic interpolations at hidden representations as additional training signals for the image classification task. They demonstrated that it results in neural models with smoother decision boundaries at multiple layers, avoiding being overly confident in the space with little or no training samples, and can improve model performance and robustness. Inspired by this work, we attempt to extend the mixup augmentation method in AdvAug (Cheng et al., 2020) from word embeddings to hidden representations at higher layers for NMT training. Specifically, given two training samples, we first compute their hidden representations $\boldsymbol{h}^{(j)}$ and $\boldsymbol{h'}^{(j)}$ at the j-th encoder layer, hidden representations $\boldsymbol{z}^{(k)}$ and $\boldsymbol{z'}^{(k)}$ at the history portion of the k-th decoder layer, and their output distributions $\boldsymbol{\omega}$ and $\boldsymbol{\omega'}$. We then construct the hidden representations and the output distribution of the virtual mixup sample as follows:

$$\tilde{\boldsymbol{h}}^{(j)} = m_\lambda(\boldsymbol{h}^{(j)}, \boldsymbol{h'}^{(j)})$$
$$\tilde{\boldsymbol{z}}^{(k)} = m_\lambda(\boldsymbol{z}^{(k)}, \boldsymbol{z'}^{(k)})$$
$$\tilde{\boldsymbol{\omega}} = m_\lambda(\boldsymbol{\omega}, \boldsymbol{\omega'})$$

where $m_\lambda(\boldsymbol{x}, \boldsymbol{y}) = \lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}$ denotes the interpolation of two vectors, with an interpolation weight $\lambda \sim Beta(\alpha_{mixup}, \beta_{mixup})$ randomly sampled from a Beta distribution for each pair of training samples. The manifold mixup augmentation loss L^m_{mixup} is computed by:

$$L^m_{mixup} = \mathbb{E}[\texttt{KL}(\texttt{Dec}_{>k}(\texttt{Enc}_{>j}(\tilde{\boldsymbol{h}}^{(j)}), \tilde{\boldsymbol{z}}^{(k)}), \tilde{\boldsymbol{\omega}})]$$

Finally, our manifold adversarial augmentation method optimizes on the combination of original NMT training loss, adversarial learning loss, and the mixup augmentation loss:

$$L = L_{nmt} + L^m_{adv} + L^m_{mixup}$$

2.3 Extention to Back-Translation

Back translation is an effective data augmentation method for machine translation. However, it is well known that pseudo training samples created from back translation have different characteristics from the gold training samples, due to factors such as domain mismatch and translation errors. To bridge this gap, we extend the manifold mixup augmentation strategy to create virtual training samples that are interpolated between a pseudo training sample and a gold training sample, again at hidden representations. We can adjust the parameters of the distribution $Beta(\alpha_{bt}, \beta_{bt})$ for generating the interpolation weight, biasing it toward the gold training sample to alleviate the aforementioned problems with back translation. Let $\tilde{\boldsymbol{h}}^{(j)}_{\boldsymbol{bt}}$, $\tilde{\boldsymbol{z}}^{(k)}_{\boldsymbol{bt}}$, and $\tilde{\boldsymbol{\omega}}_{\boldsymbol{bt}}$ be interpolation results, we define an additional training loss:

$$L^{m,bt}_{mixup} = \mathbb{E}[\texttt{KL}(\texttt{Dec}_{>k}(\texttt{Enc}_{>j}(\tilde{\boldsymbol{h}}^{(j)}_{\boldsymbol{bt}}), \tilde{\boldsymbol{z}}^{(k)}_{\boldsymbol{bt}}), \tilde{\boldsymbol{\omega}}_{\boldsymbol{bt}})]$$

3 Experiments

3.1 Setup

We conduct experiments on two language pairs: Chinese-English and English-French. For the Chinese-English translation task, we use the LDC corpus with 1.2M sentence pairs for training, NIST06 for validation, and NIST02, NIST03, NIST04, NIST05, NIST08 as the test sets. For the English-French translation task, we use the IWSLT 2016 corpus with 230k sentence pairs for training, test2012 for validation, and test2013 and test2014 as the test sets. All models are based on the Transformer architecture. Details of the data processing, model configuration, and training settings can be found in the appendix.

We compare with the following methods:

- The vanilla Transformer model (Vaswani et al., 2017).
- The virtual adversarial regularization method in (Sano et al., 2019), which adds a proportion of normalized gradient to the source and target word embeddings for adversarial training.

METHOD	Chinese-English						English-French	
	MT06	MT02	MT03	MT04	MT05	MT08	test13	test14
Vaswani et al. (2017)	44.57	45.49	44.55	46.20	44.96	35.11	40.88	37.79
Sano et al. (2019)	45.75	46.37	45.02	46.49	45.88	35.90	41.67	38.72
Cheng et al. (2019)	46.95	47.06	46.48	47.39	46.58	37.38	41.76	39.46
Cheng et al. (2020)	49.26	49.03	47.96	48.86	**49.88**	**39.63**	43.03	40.91
Our method	**49.43**	**49.54**	**50.34**	**49.46**	49.04	39.19	**44.58**	**41.56**

Table 1: Comparison of main results with different robust training methods.

METHOD	Chinese-English						English-French	
	MT06	MT02	MT03	MT04	MT05	MT08	test13	test14
$L_{nmt} + L^{w}_{mixup}$	48.14	48.75	48.80	48.45	47.69	38.55	43.72	40.37
$L_{nmt} + L^{m}_{mixup}$	48.45	49.55	49.69	49.47	48.95	39.40	44.24	40.46
$L_{nmt} + L^{w}_{adv}$	47.65	48.34	48.40	48.48	47.88	38.59	44.17	40.05
$L_{nmt} + L^{m}_{adv}$	47.90	49.05	48.57	48.88	48.39	38.68	44.35	40.34
$L_{nmt} + L^{w}_{mixup} + L^{w}_{adv}$	48.18	49.37	49.59	48.90	49.03	39.01	44.52	40.87
$L_{nmt} + L^{m}_{mixup} + L^{m}_{adv}$	49.43	49.54	50.34	49.46	49.04	39.19	44.58	41.56

Table 2: Ablation study result of different loss functions. L^{w}_{mixup} and L^{w}_{adv} corresponds to adversarial agumentation at the word embedding level, compared with augmentation at the hidden representation level for L^{m}_{mixup} and L^{m}_{adv}.

- The doubly adversarial inputs method in (Cheng et al., 2019), which performs adversarial learning with word substitutions in the source and target text based on language modeling and gradients at word embeddings.

- The AdvAug method in (Cheng et al., 2020), a state of the art adversarial learning method for NMT, also described in Section 2.

3.2 Main Results

Table 1 shows that our method is very competitive in comparison with other methods in the literature, achieving the overall best results. Compared to the vanilla Transformer, our method achieves more than 4 BLEU points of improvement on average on the Chinese-English task and more than 3 BLEU points of improvement on the English-French task. Compared to AdvAug, the previous state of the art, our method outperforms on 4 out of 6 test sets on the Chinese-English task, yielding up to 2.38 BLEU points of improvement on MT03 and an average improvement of 0.39 BLEU points over the 6 test sets. On the English-French task, our approach yields 1.55 and 0.65 BLEU points of improvement on the two test sets respectively.

METHOD	test13	test14
Vaswani et al. (2017)	40.88	37.79
+back-translation	43.55	40.20
Our method	44.58	41.56
+back-translation	44.81	41.92
+back-translation, $L^{m,bt}_{mixup}$	45.46	42.13

Table 3: Back-translation results on the English-French task.

3.3 Ablation Study

Table 2 presents the ablation study results of different loss functions. In addition to two manifold adversarial augmentation loss functions described in Section 2, we also include their counterparts computed at the word embeddings for comparison. First, we always achieve better MT results with loss functions computed at the hidden representions than at the word embeddings, further validating our motivation that operating at higher hidden layer is superior. Second, we observe that adversarial learning and mixup augmentation are complementary to each other, with the combination of the two achieving the best performance.

3.4 Results with Back Translation

We conduct back-translation experiments on the English-French task as it has a smaller training set

and can potentially benefit more from back translation. 25M French sentences from newscrawl07-11[2] are used as additional monolingual data, and are translated to English using a Transformer model trained with only the parallel training data. As shown in Table 3, both the Transformer baseline and our method can benefit from back-translation, although our method obtains a smaller improvement compared to the Transformer baseline as it has a significantly higher BLEU score to start with (actually higher than the Transformer baseline with back-translation). With the addition of our specially designed mixup loss $L^{m,bt}_{mixup}$ that biases toward the gold training samples in mixup augmentation, our method is able to achieve an extra gain of 0.65 and 0.21 BLEU improvement on the two test sets.

4 Conclusion

In this paper, we present a simple yet effective manifold adversarial augmentation method for NMT. By training on virtual samples constructed through adversarial learning and mixup augmentation at higher-level hidden representations, our method can train more robust NMT models with improved translation performance.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation. *arXiv preprint arXiv:1711.02173*.

Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust neural machine translation with doubly adversarial inputs. *arXiv preprint arXiv:1906.02443*.

Yong Cheng, Lu Jiang, Wolfgang Macherey, and Jacob Eisenstein. 2020. Advaug: Robust adversarial augmentation for neural machine translation. *arXiv preprint arXiv:2006.11834*.

Yong Cheng, Zhaopeng Tu, Fandong Meng, Junjie Zhai, and Yang Liu. 2018. Towards robust neural machine translation. *arXiv preprint arXiv:1805.06130*.

Mattia Antonino Di Gangi, Robert Enyedi, Alessandra Brusadin, and Marcello Federico. 2019. Robust neural machine translation for clean and noisy speech transcripts. *arXiv preprint arXiv:1910.10238*.

[2]http://www.statmt.org/wmt12/translation-task.html

Javid Ebrahimi, Daniel Lowd, and Dejing Dou. 2018. On adversarial examples for character-level neural machine translation. *arXiv preprint arXiv:1806.09030*.

Omer Levy, Jacob Eisenstein, Marjan Ghazvininejad, et al. 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.

Xian Li, Paul Michel, Antonios Anastasopoulos, Yonatan Belinkov, Nadir Durrani, Orhan Firat, Philipp Koehn, Graham Neubig, Juan Pino, and Hassan Sajjad. 2019. Findings of the first shared task on machine translation robustness. *arXiv preprint arXiv:1906.11943*.

Jonathan Mamou, Hang Le, Miguel Del Rio, Cory Stephenson, Hanlin Tang, Yoon Kim, and SueYeon Chung. 2020. Emergence of separable manifolds in deep language representations.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2019. Bpe-dropout: Simple and effective subword regularization. *arXiv preprint arXiv:1910.13267*.

Nicholas Ruiz, Mattia Antonino Di Gangi, Nicola Bertoldi, and Marcello Federico. 2019. Assessing the tolerance of neural machine translation systems against speech recognition errors. *arXiv preprint arXiv:1904.10997*.

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.

Matthias Sperber, Jan Niehues, and Alex Waibel. 2017. Toward robust neural machine translation for noisy input sequences. In *International Workshop on Spoken Language Translation (IWSLT)*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

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

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and

Yoshua Bengio. 2019. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pages 6438–6447. PMLR.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018. Switchout: an efficient data augmentation algorithm for neural machine translation. *arXiv preprint arXiv:1808.07512*.

A Appendix

A.1 Experiment details

For the IWSLT English-French translation task, the training sets are preprocessed with BPE with 20k joint operations, and English and French share a vocabulary of 20k sub-words. For the NIST Chinese-English translation task, the training sets are preprocessed with BPE with 60k joint operations, and the vocabulary size is 60k and 30k for Chinese and English respectively.

We follow the network settings in the original Transformer work. The total numbers of the parameters of the model are 64757760 and 83247104 for French-English and Chinese-English translation tasks. The dropout ratio is 0.3. The model is optimized with Adam. We use inverse square root as the learning rate schedule, with the peak learning rate of 5e-4, warm-up steps of 4000. During decoding, the beam size is 4 and the length penalty is 0.6. We search hyper-parameters for producing adversarial examples according to BLEU [3] on the validation set. Finally, the maximum number of layers for manifold data augmentation at source side K_{src} and target side K_{tgt} are both set to 3. We let $\alpha_{adv} = 0.5$ and $\beta_{adv} = 0.5$ on the source side, and $\alpha_{adv} = 0.3$ and $\beta_{adv} = 0.7$ on the target side. When mixing training example pairs, the hyper-parameter α_{mixup} and β_{mixup} are both set to 8 for the English-French translation task, and set to 0.2 for the Chinese-English translation task. When we mix parallel sentence with back-translated parallel sentence, we let $\alpha_{bt} = 8$ and $\beta_{bt} = 4$.

We use 1 V100 GPU for the IWSLT English-French translation task, and 4 V100 GPU for the NIST Chinese-English translation task. It takes about 24 hours and 72 hours for these two tasks respectively.

K_{src}	MT06	K_{tgt}	MT06
1	48.55	1	48.40
2	48.72	2	48.78
3	49.43	3	49.43
4	49.15	4	48.78
5	48.76	5	48.65
6	48.60	6	48.32

Table 4: Effect of K_{src} and K_{tgt} for manifold adversarial augmentation on NIST Chinese-English translation task

μ_1	μ_2	test13	test14
0	0	40.88	37.79
0	0.3	42.60	40.22
0	0.7	43.17	40.41
0	1	44.24	40.46
0.3	0	43.58	40.09
0.7	0	43.46	39.94
1	0	44.35	40.34
1	1	44.58	41.56

Table 5: Effect of different weights for losses on the IWSLT16 English-French translation task

A.2 Effect of K_{src} and K_{tgt} for manifold adversarial augmentation

Instead of performing manifold adversarial augmentation on a predetermined hidden layer, we choose to randomly select $j \in [0, K_{src}]$ and $k \in [0, K_{tgt}]$ among a range of encoder and decoder layers, allowing more variations. We study their effect on the validation set of the NIST Chinese-English translation task. We fix $K_{src} = 3$ (or $K_{tgt} = 3$), when change the value of K_{tgt} (or K_{src}). As shown in Table 4, large or small K_{src} and K_{tgt} will make the model performs worse down to about 1 BLEU.

A.3 Impact of different weights for losses

To further study the impact of different losses, we set the training loss of our model as $L = L_{nmt} + \mu_1 L_{adv}^{(m)} + \mu_2 L_{mixup}^{(m)}$, and compare the performance when we set different μ_1 and μ_2. We conduct experiments on the IWSLT English-French translation task. As shown in Table 5, too small μ_1 and μ_2 will make the model perform worse.

[3] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/mteval-v13a.pl