Mixup Decoding for Diverse Machine Translation

Jicheng Li[12†], **Pengzhi Gao**[3], **Xuanfu Wu**[12], **Yang Feng**[12*], **Zhongjun He**[3], **Hua Wu**[3], and **Haifeng Wang**[3]

[1] Key Laboratory of Intelligent Information Processing,
Institute of Computing Technology, Chinese Academy of Sciences (ICT/CAS)
[2] University of Chinese Academy of Sciences
[3] Baidu Inc. No. 10, Shangdi 10th Street, Beijing, 100085, China
{lijicheng,wuxuanfu20s,fengyang}@ict.ac.cn
{gaopengzhi,hezhongjun,wu_hua,wanghaifeng}@baidu.com

Abstract

Diverse machine translation aims at generating various target language translations for a given source language sentence. To leverage the linear relationship in the sentence latent space introduced by the mixup training, we propose a novel method, *MixDiversity*, to generate different translations for the input sentence by linearly interpolating it with different sentence pairs sampled from the training corpus during decoding. To further improve the faithfulness and diversity of the translations, we propose two simple but effective approaches to select diverse sentence pairs in the training corpus and adjust the interpolation weight for each pair correspondingly. Moreover, by controlling the interpolation weight, our method can achieve the trade-off between faithfulness and diversity without any additional training, which is required in most of the previous methods. Experiments on WMT'16 en→ro, WMT'14 en→de, and WMT'17 zh→en are conducted to show that our method substantially outperforms all previous diverse machine translation methods.

1 Introduction

Neural machine translation (NMT) (Sutskever et al., 2014; Wu et al., 2016; Gehring et al., 2017; Vaswani et al., 2017; Ott et al., 2018) has achieved significant success in improving the quality of machine translation. Despite these successes, NMT still faces problems in translation diversity (Vanmassenhove et al., 2019; Gu et al., 2020). Due to the existence of lexical diversity, syntactic diversity and synonymous words in the target language, one source language sentence usually corresponds to multiple proper translations. However, existing NMT models mostly consider the one-to-one mapping but neglects the one-to-many mapping between the source and target languages.

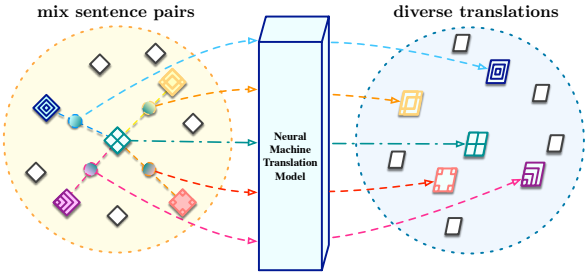


Figure 1: Illustration of the proposed method, *MixDiversity*, which linearly interpolates the input sentence with various sentence pairs sampled from the training corpus so as to generate diverse translations.

Many studies have been proposed to tackle such issues by exploiting the diversity in the model space, such as using different experts (Shen et al., 2019), applying different multi-head attentions (Sun et al., 2020), and utilizing different models (Wu et al., 2020). Although the model-oriented methods have been well studied, the data-oriented method still lacks exploration.

In this work, we focus on improving the translation diversity by exploiting the diversity in the sentence space. Since different translations of one source sentence share the same semantics, their sentence-level embeddings will gather in the same region in the target sentence space. In other words, each sentence in this region is a translation of the source sentence. By sampling different sentences from this region, we can obtain various translations. To sample different translations from this region, we propose a simple but effective method, *MixDiversity*. As aforementioned, the NMT model learns a one-to-one mapping between the source and target languages. Given the source sentence and the generated tokens in the decoder, the NMT model can map the source sentence into a corresponding target sentence. Therefore, to obtain various translations on the target side, we need to find the corresponding inputs for the NMT model. By mixing the source sentence with the sampled sentence pairs in the training corpus via linear interpolation, we

[†]This work was done when Jicheng Li was interning at Baidu Inc., China.
[*]Yang Feng is the corresponding author of the paper.

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 312–320
November 7–11, 2021.

can obtain mixed sentences as inputs for the NMT model and map them into a corresponding sentence in the target sentence space. By assigning a larger interpolation weight for the source sentence, the mixed sentence then has similar semantics, and the corresponding translation has higher faithfulness to the source sentence. In this way, by mixing the source sentence with different sentence pairs during decoding, we can obtain diverse mixed sentences as inputs for the NMT model and map them to different translations for the source sentence.

Given that NMT models are non-linear functions, the interpolation weight of the input sentences could decline, and the semantic of the output could shift to the randomly sampled sentence pairs. To guarantee the consistency of the interpolation weight during decoding, we force the NMT model to learn to maintain the proportion between the mixed sentences with the mixup training strategy (Guo et al., 2020) which linearly interpolates two randomly sampled sentence pairs in both encoder and decoder during training. The main idea of our approach is illustrated in Figure 1, where we mix one source sentence with four different sentence pairs sampled from the training corpus to obtain four variant mixed samples as inputs for the NMT model and map the mixed sentences to four diverse sentences in the target space.

2 MixDiversity

2.1 Overview

During training, we linearly interpolate word embeddings of two randomly sampled sentence pairs on both the source and target sides. During inference, since the corresponding target sentence of the input can not be obtained in advance, we interpolate word embeddings of previously generated tokens and the sampled target sentence in the decoder. Note that the MixDiversity can also be used without the Mixup Training.

2.2 Mixup Training for NMT

We apply the mixup training (Guo et al., 2020) to encourage the NMT model to learn the linear relationship in the latent space of the input sentences. Consider a pair of training samples $(\mathbf{x}^i, \mathbf{y}^i)$ and $(\mathbf{x}^j, \mathbf{y}^j)$ in the parallel corpus $\mathcal{S}$, where $\mathbf{x}^i$ and $\mathbf{x}^j$ denote the source sentences, and $\mathbf{y}^i$ and $\mathbf{y}^j$ denote the target sentences. The synthetic sample $(\mathbf{x}^{ij}, \mathbf{y}^{ij})$ is generated as follows.

$$\mathbf{x}^{ij} = \lambda \mathbf{x}^i + (1-\lambda)\mathbf{x}^j, \;\; \mathbf{y}^{ij} = \lambda \mathbf{y}^i + (1-\lambda)\mathbf{y}^j,$$

where λ is drawn from a Beta distribution $\text{Beta}(\alpha, \alpha)$ with a hyper-parameter α. The synthetic sample $(\mathbf{x}^{ij}, \mathbf{y}^{ij})$ is then fed into the NMT model for training to minimize the empirical risk:

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

where ℓ denotes the cross entropy loss, θ is a set of model parameters, $f(*)$ is the probability predictions of the NMT model,

$$\ddot{\mathbf{y}}^{ij} = \lambda \ddot{\mathbf{y}}^i + (1-\lambda)\ddot{\mathbf{y}}^j, \quad (2)$$

and $\ddot{\mathbf{y}}^i$ and $\ddot{\mathbf{y}}^j$ are the sequences of one-hot label vectors for $\mathbf{y}^i$ and $\mathbf{y}^j$ respectively.

2.3 Mixup Decoding for Diverse MT

At inference, assume $\mathbf{x} = x_1, ..., x_I$ that corresponds to the source sentence with length I. We mix it with K different sentence pairs $(\mathbf{x}^1, \mathbf{y}^1), \ldots, (\mathbf{x}^K, \mathbf{y}^K)$ selected from the training corpus to generate K different translations of $\mathbf{x}$. Specifically, for the i^{th} translation, we first interpolate the token embeddings of $\mathbf{x}$ with the token embeddings of $\mathbf{x}^i$ in the encoder side:

$$\hat{e}(x_t^i) = \lambda_t^i e(x_t) + (1-\lambda_t^i) e(x_t^i), \;\; \forall t \in [1, I]. \quad (3)$$

The encoder then maps the mixed token embeddings $\hat{e}(\mathbf{x}_1^i), \ldots \hat{e}(x_I^i)$ into the corresponding hidden representations $\mathbf{h}^i$.

In the decoder side, at step t, we mix the embedding of the token y_{t-1}, which is predicted by the NMT model at step $t-1$, with the embedding of y_{t-1}^i as follows:

$$\hat{e}(y_{t-1}^i) = \lambda_t^i e(y_{t-1}) + (1-\lambda_t^i) e(y_{t-1}^i), \quad (4)$$

where y_0 and y_0^i are the special beginning-of-sentence symbol $\langle bos \rangle$. The predicted token y_t is then calculated by

$$y_t = \mathop{\text{argmax}}_{y \in \mathcal{V}_y} P(y | \mathbf{h}^i, \hat{e}(y_{\leqslant t-1}^i); \theta), \; t \geqslant 1, \quad (5)$$

where $\mathcal{V}_y$ is the vocabulary of the target language. Note that λ_t^i's in (3) and (4) are drawn from the Beta distribution $\text{Beta}(\alpha, \alpha)$ with the same α for different t and i.

Select Sentence Pairs by Source Length We first group sentence pairs in the training corpus by their source sentence lengths and then randomly select K sentence pairs $(\mathbf{x}^1, \mathbf{y}^1), \ldots (\mathbf{x}^K, \mathbf{y}^K)$ from

the groups that have similar length compared with the input sentence. Specifically, given an input sentence with length I, we sample sentence pairs from the groups with lengths in the range of $[I-1, I]$.

Adjust Interpolation Weight by Similarity In order to correctly translate the semantic of the input sentence, $\mathbf{x}$ needs to dominate the mixed samples. Different sentences in $(\mathbf{x}^1, \mathbf{y}^1), \ldots (\mathbf{x}^K, \mathbf{y}^K)$ may have different similarity with $\mathbf{x}$, and a higher similarity between $\mathbf{x}^i$ and $\mathbf{x}$ implies a looser constraint on the interpolation weight between them. Thus, taking the similarity between $\mathbf{x}^i$ and $\mathbf{x}$ into account, we sample the interpolation weight λ_t^i from the Beta distribution as follows.

$$\lambda_t^i \sim \text{Beta}(\alpha_i, \alpha_i), \quad \alpha_i = \tau + \frac{\tau}{d(\mathbf{x}, \mathbf{x}^i)}, \qquad (6)$$

where τ is a hyper-parameter to control the interpolation weight, and $d(*)$ is the Euclidean distance between the embeddings of two sentences, which are defined as the average among all token embeddings in the sentence. In our implementation, λ_t^t is actually set to be $\max(\lambda_t^t, 1-\lambda_t^t)$. The larger distance between $\mathbf{x}$ and $\mathbf{x}^i$ is, the larger interpolation weight λ_t^t we have, which leads to dynamically adjusting on the interpolation weight based on the sentence similarity.

3 Experimental Setup

3.1 Data Description

Our experiments consider three translation datasets: WMT'16 English-Romanian (en→ro), WMT'14 English-German (en→de), and WMT'17 Chinese-English (zh→en). All sentences are prepossessed with byte-pair-encoding (BPE) (Sennrich et al., 2016). For WMT'16 en→ro, we use the preprocessed dataset released in Lee et al. (2018) which contains 0.6M sentence pairs. We use newsdev-2016 as the validation set and newstest-2016 as the test set. We build a shared vocabulary with 40K BPE types. For WMT'14 en→de, it consists of 4.5M training sentence pairs, and we use newstest-2013 for validation and newstest-2014 for test. We build a shared vocabulary with 32K BPE types. For WMT'17 zh→en, it consists of 20.1M training sentence pairs, and we use devtest-2017 as the validation set and newstest-2017 as the test set. We build the source and target vocabularies with 32K BPE types separately.

Strategy	Baseline BLEU $\mathcal{R}$		
	en→ro	en→de	zh→en
Vanilla	32.80	27.43	24.07
Mixup	33.75	27.70	24.40

Table 1: The baseline BLEU of different training strategy in each dataset.

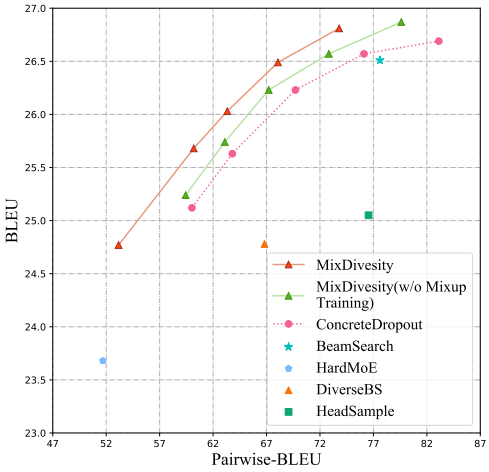


Figure 2: Illustration of the trade-off between reference BLEU and pair-wise BLEU in WMT'14 en→de with different τ.

3.2 Model Configuration

We apply a standard 6-layer Transformer Base model (Vaswani et al., 2017) with 8 attention heads, embedding size 512, and FFN layer dimension 2048. We use the label smoothing (Szegedy et al., 2016) with $\epsilon = 0.1$ and Adam (Kingma and Ba, 2015) optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$. We set learning rate as 0.0007 with 4000 warmup steps from the initialized learning rate of 10^{-7}. The NMT model is trained with dropout 0.1 and max tokens 4096. When adopting the mixup training strategy, we set α as 1.0, 0.1 and 0.1 for en→ro, en→de and zh→en respectively. We train our model on 4 NVIDIA V100 GPUs until it converges. At the inference time, we set beam size as 4 with length penalty 0.6.

3.3 Evaluation Metrics

Referring to Wu et al. (2020), we adopt the average BLEU with reference (rfb) to measure the faithfulness of different translations to the input sentence and the average pairwise-BLEU (pwb) to measure the pair-wise similarity between different translations. The higher rfb, the better accuracy of the translations. The lower pwb, the better diversity of the translations. In our experiments, given one input sentence, we generate five different translations

Method	WMT'16 en→ro			WMT'14 en→de			WMT'17 zh→en		
	rfb⇑	pwb⇓	EDA⇓	rfb⇑	pwb⇓	EDA⇓	rfb⇑	pwb⇓	EDA⇓
BeamSearch (BS)	31.99	80.82	26.62	26.51	77.61	21.55	23.69	81.36	19.64
DiverseBS (Vijayakumar et al., 2016)	30.65	76.46	25.92	24.78	66.81	20.71	22.43	66.93	17.49
HardMoE (Shen et al., 2019)	31.13	68.42	23.01	23.68	**51.69**	19.69	21.77	**49.13**	15.20
HeadSample (Sun et al., 2020)	26.94	**59.38**	26.42	25.05	76.55	22.71	21.24	73.96	21.33
ConcreteDropout (Wu et al., 2020)	31.20	65.24	21.94	25.12	60.02	18.49	**23.10**	55.61	13.97
MixDiversity	**31.50**	59.57	**21.10**	**25.50**	57.50	**17.79**	22.96	51.52	**13.88**
w/o Mixup Training	31.48	66.44	22.16	25.24	59.43	18.15	22.87	54.01	13.92

Table 2: The best result of each method on WMT'16 en→ro, WMT'14 en→de, and WMT'17 zh→en. For DiverseBS, HardMoE, and HeadSample, we select the result under the best settings described in their papers. For ConcreteDropout and MixDiversity, we validate the model under different hyper-parameter settings on the validation set to find the best settings for the model, and we report the result on the test set under the best settings. We get the best results of MixDiversity with $\tau = 0.3$, 0.3, and 0.25 in en→ro, en→de and zh→en respectively. ⇑ means the higher, the better. ⇓ means the lower, the better.

for all methods.

When we calculate Diversity Enhancement per Quality (DEQ) (Sun et al., 2020) to evaluate the overall performance of different methods, we find that the DEQ results are not stable. For instance, the DEQ scores of ConcreteDropout in Figure 2 (from the leftmost point to the rightmost point) are 12.65, 15.69, 28.21, -24.83, and 30.61, where positive and negative scores appear alternately. We thus propose a new metric, Euclidean Distance from the ultimate Aim (EDA), to evaluate the overall quality of the results synthetically.

Consider rfb and pwb as the abscissa and the ordinate of a coordinate system, where $0 \leqslant \text{rfb} \leqslant \mathcal{R}$, and $0 \leqslant \text{pwb} \leqslant \mathcal{P}$. $\mathcal{R}$ is the baseline BLEU, which is defined as the BLEU score of the top one translation by beam search decoding with beam size 4 in our experiments. $\mathcal{P} = 100$ is the maximal pwb. Different results with specific rfb and pwb scores could be mapped to different points in this coordinate system. The ultimate aim of the diverse machine translation task is to reach the point $(\mathcal{R}, 0)$. By measuring the Euclidean distance between $(\mathcal{R}, 0)$ and the result, we can evaluate the overall quality of the result.

We, however, notice that rfb and pwb have different ranges ($\mathcal{P} > \mathcal{R}$), and pwb decreases much faster than rfb with the changing of τ. As a consequence, the calculated EDA is biased to the results with the lower pwb scores. To alleviate such bias, we normalize the value of rfb and pwb to $[0, 1]$ by dividing $\mathcal{R}$ and $\mathcal{P}$ respectively and add a weight $\omega = \frac{\mathcal{R}}{\mathcal{P}}$ on the pwb term shown as follows:

$$\text{EDA} = 100\% \cdot \sqrt{(\frac{\mathcal{R} - \text{rfb}}{\mathcal{R}})^2 + \omega^2 (\frac{0 - \text{pwb}}{\mathcal{P}})^2}.$$

Note that different training strategies lead to different baseline BLEU $\mathcal{R}$. Table 1 shows the baseline BLEU of Transformer in each dataset. When we use EDA to evaluate the performance of ConcreteDropout in Figure 2, we get 18.49, 18.69, 19.61, 21.1, and 22.95. This result shows that EDA is a better and more stable overall evaluation metric than DEQ for the diverse machine translation.

4 Experimental Results

4.1 Main Results

We show the results of different methods on generating diverse translations in Table 2. We compare our method with the conventional beam search decoding (BeamSearch) and the existing model-oriented methods, including DiverseBS, HardMoE, HeadSample, and ConcreteDropout. For each method, we exhibit its best result with the lowest EDA score. We can see that MixDiversity gets lower EDA scores than all existing methods in all three datasets, and the performance of MixDiversity without the mixup training also outperforms other competitors on WMT'14 en→de and WMT'16 zh→en with lower EDA scores.

Figure 2 shows the trade-off results between the reference BLEU and the pair-wise BLEU on WMT'14 en→de. We can see that mixup training or not, MixDiversity generally performs better than all other methods without additional training or finetuning, which is required in most previous methods, such as HardMoE.

4.2 Ablation Study

The results of the ablation study are shown in Table 3, which consists of three experiments. In the

	Method	rfb⇑	pwb⇓	EDA⇓
1	MixDiversity ($\tau = 0.3$)	25.50	57.50	**17.79**
	w/o Mixup Training	25.24	59.43	18.15
	w/o LenSelection	25.58	65.13	19.09
	w/o SimWeight	25.77	69.99	20.12
2	MixDiversity ($\tau = 0.3$)			
	+ Mixup Samples	25.50	57.50	**17.79**
	+ Mixup Noises	21.58	43.78	25.20
	+ Mixup SynSents	11.44	13.30	58.81
3	MixDiversity ($\tau = 0.3$)			
	+ Both Sides Mixup	25.50	57.50	**17.79**
	+ Only Encoder Mixup	25.83	66.09	19.51
	+ Only Decoder Mixup	25.41	60.69	18.73

Table 3: Ablation study on WMT'14 en→de.

first experiment, we evaluate the performance of our method with different settings: training NMT models without mixup strategy (*w/o* Mixup Training), decoding by randomly selecting K sentence pairs from the entire training corpus (*w/o* LenSelection), and sampling the interpolation weights without considering similarities between $\mathbf{x}$ and $\mathbf{x}^i$ (*w/o* SimWeight). In the second experiment, we not only attempt to mix the input sentence with Gaussian noise drawn from $\mathcal{N}(0, 2)$, but we also mix the input sentence with synthetic sentence pairs which are made up of tokens that are randomly sampled from the vocabulary. In both cases, we observe remarkable increases in EDA. Such a phenomenon indicates that the potential linguistic features in training samples could assist MixDiversity in generating different translations of high diversity and faithfulness. In the last experiment, we verify the rationality and effectiveness of the mixup operations in both encoder and decoder.

4.3 Applications of Diverse Translation

In Table 4, we compare MixDiversity with BeamSearch (BS) to show the application of diverse translation methods on boosting the performance of both Back Translation and Knowledge Distillation. We generate sentences with a beam size of 5 for all methods. For BeamSearch (Top 5) and MixDiversity, we generate five different translations. In the Back Translation experiment, we randomly sample 4M sentences from the German monolingual corpus distributed in WMT'18 and combine the original parallel corpus with the back-translated parallel corpus to train the NMT model. In the Data Distillation experiment, we train the student NMT model with the generated sentences of the teacher NMT model.

	Back Trans.	Knowledge Distill.
Baseline	27.43	–
BS (Top 1)	28.81	27.28
BS (Top 5)	28.82	27.46
MixDiversity	**29.19**	**27.83**

Table 4: Results of the Back Translation and the Knowledge Distillation experiments on WMT'14 en→de.

5 Related Work

Many studies have been proposed to improve the translation diversity by exploiting the diversity in the model space. Li et al. (2016) and Vijayakumar et al. (2016) adopt various regularization terms in the beam search decoding to encourage generating diverse outputs. He et al. (2018) generates different translations by incorporating condition signals of different models. Shen et al. (2019) proposes to training NMT models with the mixture of experts method and generates diverse translations using different latent variables of different experts. Shu et al. (2019) generates diverse translation conditioned on different sentence codes. Sun et al. (2020) discovers that encoder-decoder multi-head attention in Transformer learns multiple target-source alignments and generates diverse translations by sampling different heads in the attention modules. Wu et al. (2020) samples different models from a posterior model distribution and employs variational inference to control the diversity of translations.

6 Conclusion

In this work, we propose a novel method, *MixDiversity*, for the diverse machine translation. Compared with the previous model-oriented methods, MixDiversity is a data-oriented method that generates different translations of the input sentence by utilizing the diversity in the sentence latent space. We also propose two simple but effective methods to select the mixup samples and adjust the mixup weights for each sample. To evaluate the overall performance synthetically, we design a new evaluation metric, *EDA*. Experimental results show that MixDiversity outperforms all previous methods in the field of diverse machine translation.

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

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.

Shuhao Gu, Jinchao Zhang, Fandong Meng, Yang Feng, Wanying Xie, Jie Zhou, and Dong Yu. 2020. Token-level adaptive training for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1035–1046, Online. Association for Computational Linguistics.

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.

Xuanli He, Gholamreza Haffari, and Mohammad Norouzi. 2018. Sequence to sequence mixture model for diverse machine translation. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 583–592, Brussels, Belgium. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Brussels, Belgium. Association for Computational Linguistics.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. A simple, fast diverse decoding algorithm for neural generation. *ArXiv preprint*, abs/1611.08562.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 371–376, Berlin, Germany. Association for Computational Linguistics.

Tianxiao Shen, Myle Ott, Michael Auli, and Marc'Aurelio Ranzato. 2019. Mixture models for diverse machine translation: Tricks of the trade. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5719–5728. PMLR.

Raphael Shu, Hideki Nakayama, and Kyunghyun Cho. 2019. Generating diverse translations with sentence codes. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1823–1827, Florence, Italy. Association for Computational Linguistics.

Zewei Sun, Shujian Huang, Hao-Ran Wei, Xin-yu Dai, and Jiajun Chen. 2020. Generating diverse translation by manipulating multi-head attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8976–8983.

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.

Eva Vanmassenhove, Dimitar Shterionov, and Andy Way. 2019. Lost in translation: Loss and decay of linguistic richness in machine translation. In *Proceedings of Machine Translation Summit XVII: Research Track*, pages 222–232, Dublin, Ireland. European Association for Machine Translation.

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.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *ArXiv preprint*, abs/1610.02424.

Xuanfu Wu, Yang Feng, and Chenze Shao. 2020. Generating diverse translation from model distribution with dropout. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1088–1097, Online. Association for Computational Linguistics.

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

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

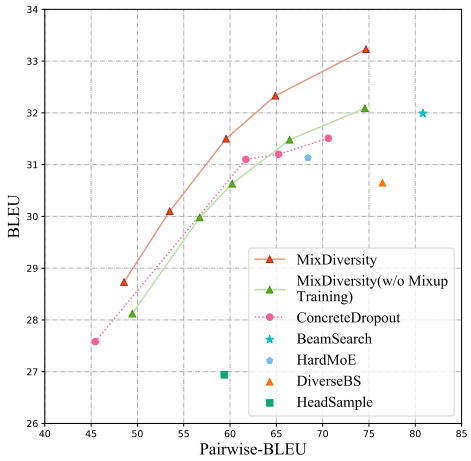
A Methods for Comparison

In our experiments, we set $k = 5$ and compare our method with the following works:

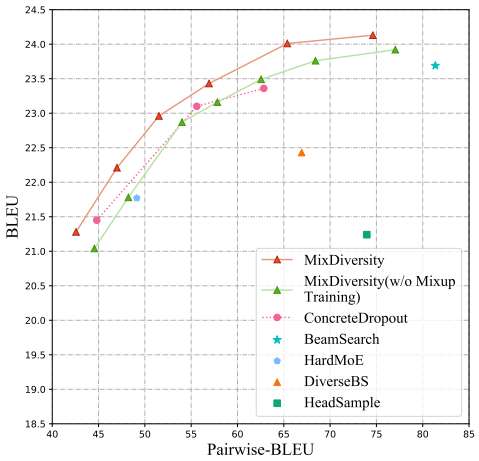
- **BeamSearch (BS)**: In our experiments, we choose the top k sentences generated by beam search decoding as the result.
- **DiverseBeamSearch (DiveseBS)** (Vijayakumar et al., 2016): It generates diverse translations by grouping sentences in the beam search decoding with a regularization term to guarantee the diversity between different groups. We set the number of groups as k, and each group includes two sentences in our experiments.
- **HardMoE** (Shen et al., 2019): It first trains the model with k different hidden states and then generates different translations with different hidden states.
- **HeadSample** (Sun et al., 2020): It generates different outputs by sampling different heads in multi-head attention modules. In our experiments, we set the number of heads to be sampled as 3.
- **ConcreteDropout** (Wu et al., 2020): It generates different outputs by sampling different models from the model distribution using variational inference.

B Trade-off between reference BLEU and pair-wise BLEU

Figure 3 shows the trade-off results between reference BLEU and pair-wise BLEU in WMT'16 en→ro and WMT'17 zh→en. From results in both en→ro and zh→en, we find that the lines of the MixDiversity and the ConcreteDropout overlap with each other. In addition, the ConcreteDropout needs to finetune the translation model under different configurations to achieve different trade-off results between the BLEU and the pair-wise BLEU. While the HardMoE needs to retrain the whole model with different settings of the number of experts so as to achieve the trade-off between the two BLEU scores. Besides, the performance of the HeadSample is unstable with different number of the sampled heads. In contrast, the MixDiversity can achieve the trade-off between the two BLEU scores by the hyper-parameter τ without any additional training or finetuning time.



(a) en→ro



(b) zh→en

Figure 3: Illustration of the trade-off between reference BLEU and pair-wise BLEU in WMT'16 en→ro and WMT'17 zh→en with different τ.

C Case Study

In Table 6, we illustrate a case of outputs from the MixDiversity and the BeamSearch in WMT'17

	rf-BERTscore⇑	pw-BERTscore⇓	EDA-BERTscore⇓
Beam Search (BS)	**85.50**	95.87	96.95
HeadSample (Sun et al., 2020)	84.99	96.29	97.45
ConcreteDropout (Wu et al., 2020)	84.93	95.52	96.69
MixDiversity (w/o Mixup Training)	84.61	**92.26**	**93.53**

Table 5: The evaluation result using BERT-Score in WMT'14 en→de. ⇑ means the higher, the better. ⇓ means the lower, the better.

Source	因此， 人类 希望 有朝一日 在 火星 建立 居住 基地， 最终 向 火星 移民， 把 它 变成 人类 的 第二 家园 。
Reference	Therefore , the human beings hope that one day on the Mars to establish a base of residence , and ultimately to Mars immigration , it turned into a second home of mankind .
BeamSearch	Therefore , human beings hope that one day they will establish a residence base on Mars and eventually emigrate to Mars , making it their second home .
	Therefore , the human race hopes one day to establish a residence base on Mars and eventually emigrate to Mars , making it the second home of the human race .
	Therefore , the human race hopes one day to establish a residence base on Mars and eventually emigrate to Mars , turning it into the second home of mankind .
	Therefore , human beings hope that one day they will establish a residence base on Mars and eventually emigrate to Mars , making it the second home of human beings .
	Therefore , human beings hope that one day they will establish a residence base on Mars and eventually emigrate to Mars , turning it into the second home of mankind .
MixDiversity ($\tau = 0.15$)	Therefore , man hopes one day to establish a residence base on Mars , and eventually emigrate to Mars and turn it into a second home .
	So humans hope to one day establish a residence base on Mars and eventually emigrate to Mars and turn it into a second home for humanity .
	So man wants one day to establish a residence base on Mars and eventually emigrate to Mars and make it his second home .
	So man hopes one day to build a living base on Mars and eventually emigrate to make it a second home for humanity .
	So man wants to be able to build a living base on Mars and eventually emigrate to Mars , turning it into a second home .
MixDiversity ($\tau = 0.35$)	So . one day , humans want to build a living base on Mars and eventually emigrate to Mars and turn it into a second home .
	The human race , therefore , hopes that one day it will establish a residence base on Mars and eventually immigrate to Mars to make it a second home .
	So man hopes one day to establish a base on Mars and eventually emigrate to Mars and turn it into a second home for man .
	Thus , mankind hopes that one day it will establish a living base on Mars and eventually immigrate to Mars , becoming a second home for humanity .
	So man wants to be able to build a residence base on Mars and eventually emigrate to Mars , making it thesecond home of man .

Table 6: Example outputs of BeamSearch and MixDiversity in WMT'17 zh→en.

zh→en. For the MixDiversity, we show the translation results under different τ. When $\tau = 0.15$, the 5 outputs of the MixDiversity follow a similar sentence pattern "So man/human hopes one day to ...". When the value of τ increase from 0.15 to 0.35, both the number of sentence pattern and the number of subjects in the 5 generated translations are expanded and the differences between translations also becomes more obvious.

D Evaluation Results of the BERT-Score

As aforementioned, reference BLEU and pairwise BLEU have been used to measure faithfulness and diversity in this work. However, BLEU simply counts n-gram overlap between the inference and the reference, which can not account for meaning-preserving lexical and compositional diversity, e.g., synonyms and paraphrases. In contrast, the BERT-Score (Zhang et al., 2020) seems to be a better measure, which computes a similarity score for each token in the inference sentence with each token in the reference sentence and correlates better with human judgments.

We apply the BERT-Score to evaluate the performance of different methods in WMT'14 en→de, as shown in Tabel 5. we adopt the average BERT-

score with reference (denoted as rf-BERTscore) to measure the faithfulness and the average pairwise BERT-score among generated sentences (denoted as pw-BERTscore) to measure the diversity. At last, we calculate the EDA using the BERT-Score (denoted as EDA-BERTscore) by substituting the BLEU score with the BERT-Score. We can see that the MixDiversity (w/o Mixup training) gets the best pw-BERTscore and the best EDA-BERTscore.