

# WeChat Neural Machine Translation Systems for WMT20

Fandong Meng, Jianhao Yan, Yijin Liu, Yuan Gao, Xianfeng Zeng,  
Qinsong Zeng, Peng Li, Ming Chen, Jie Zhou, Sifan Liu and Hao Zhou

WeChat AI, Tencent, China

{fandongmeng,elliottyan,yijinliu,masongao,xianfzeng}@tencent.com  
{qinzzeng,patrickpli,mingchen,withtomzhou,stephenliu,harveyzhou}@tencent.com

## Abstract

We participate in the WMT 2020 shared news translation task on Chinese→English. Our system is based on the Transformer (Vaswani et al., 2017) with effective variants and the DTMT (Meng and Zhang, 2019) architecture. In our experiments, we employ data selection, several synthetic data generation approaches (i.e., back-translation, knowledge distillation, and iterative in-domain knowledge transfer), advanced finetuning approaches and self-bleu based model ensemble. Our constrained Chinese→English system achieves 36.9 case-sensitive BLEU score, which is the highest among all submissions.

## 1 Introduction

Our WeChat AI team participates in the WMT 2020 shared news translation task on Chinese→English. In this year’s translation task, we mainly focus on exploiting several effective model architectures, better data augmentation, training and model ensemble strategies.

For model architectures, we mainly exploit two different architectures in our approaches, namely Transformers and RNMT. For Transformers, we implement the Deeper transformer with Pre-Norm, the Wider Transformer with a larger filter-size and the average attention based transformer (Zhang et al., 2018). For the RNMT, we use the deep transition based DTMT (Meng and Zhang, 2019) model. We finally ensemble four kinds of models in our system.

For synthetic data generation, we explore various methods for out-of-domain and in-domain data generation. For out-of-domain data generation, we explore the back-translation method (Sennrich et al., 2016a) to leverage the target side monolingual data and the knowledge distillation method (Kim and Rush, 2016) to leverage the source side of golden parallel data. For in-domain data generation,

we employ iterative in-domain knowledge transfer to leverage the source-side monolingual data and golden parallel data. Furthermore, data augmentation methods, including noisy fake data (Wu et al., 2019) and sampling (Edunov et al., 2018), are used for training more robust NMT models.

For training strategies, we mainly focus on the parallel scheduled sampling (Mihaylova and Martins, 2019; Duckworth et al., 2019), the target denoising and minimum risk training (Shen et al., 2016; Wang and Sennrich, 2020) algorithm for in-domain finetuning.

We also exploit a self-bleu (Zhu et al., 2018) based model ensemble approach to enhance our system. As a result, our constrained Chinese→English system achieves the highest case-sensitive BLEU score among all submitted systems.

In the remainder of this paper, we start with an overview of model architectures in Section 2. Section 3 describes the details of our systems and training strategies. Then Section 4 shows the experimental settings and results. Finally, we conclude our work in Section 5.

## 2 Model Architectures

In this section, we first describe the model architectures we use in the Chinese→English Shared Task, including the Transformer-based (Vaswani et al., 2017) models and RNN-based (Bahdanau et al., 2014; Meng and Zhang, 2019) models.

### 2.1 Deeper Transformer

As shown in previous studies (Wang et al., 2019; Sun et al., 2019), deeper Transformers with pre-norm outperform its shallow counterparts on various machine translation benchmarks. In their work, increasing the encoder depth significantly improves the model performance, while they only introduce mild overhead in terms of speed in training and

inference, compared with increasing the decoder side depth.

Hence, we train deeper Transformers with a deep encoder aiming for a better encoding representation. In our experiments, we mainly adopt two settings, with the hidden size 512 (Base) and 1024 (Large). We adopt a 30-layer encoder for Base models and 20/24-layer encoders for Large models. Further increasing the encoder depth does not lead to a significant BLEU improvement. To keep the total trainable parameters the same among models, the filter sizes of Base and Large models are 16384 and 4096, respectively. For training, the batch size is 4,096 tokens per GPU, and we train each model using 8 NVIDIA V100 GPUs for about 7 days.

## 2.2 Wider Transformer

Inspired by last year’s Baidu system (Sun et al., 2019), we also train Wider Transformers with a larger inner dimension of the Feed-Forward Network than the standard Transformer Large system. Specifically, two settings are used in our experiments. With a filter size of 15,000, we set the number of encoder layers to 10, and with a filter size of 12,288, we set the number of encoder layers to 12. The number of total trainable parameters of the Wider Transformer is kept approximately the same as our Deeper Transformers.

In our experiments, we also set the batch size to be 4,096 and train the Wider Transformers with 8 NVIDIA V100 GPUs for about 7 days.

## 2.3 Average Attention Transformer

To introduce more diversity in our Transformer models, we use Average Attention Transformer (AAN) (Zhang et al., 2018) as one of our candidate architectures. The Average Attention Transformer replaces the decoder self-attention module in auto-regressive order with a simple average attention, and introduces almost no loss in model performance.

We believe that even though the performance of AAN does not drop in terms of BLEU, the output distributions of AAN networks should be different from the output distributions of original Transformers, which brings diversity for the final ensemble. This also complies with our findings in self-bleu experiments (Section 3.6).

In practice, AAN models are trained for both the Wider Transformer and Deeper Transformer. The batch size and other hyper-parameters are kept the same as its non-AAN counterpart.

## 2.4 DTMT

DTMT (Meng and Zhang, 2019) is the recently proposed deep transition RNN-based model for Neural Machine Translation, whose encoder and decoder are composed of the well-designed transition blocks, each of which consists of a linear transformation enhanced GRU (L-GRU) followed by several transition GRUs (T-GRUs). DTMT enhances the hidden-to-hidden transition with multiple non-linear transformations, as well as maintains a linear transformation path throughout this deep transition by the well-designed linear transformation mechanism to alleviate the vanishing gradient problem. This architecture has demonstrated its superiority over the conventional Transformer model and stacked RNN-based models in NMT (Meng and Zhang, 2019), and also achieves surprising performances on other NLP tasks, such as sequence labeling (Liu et al., 2019) and aspect-based sentiment analysis (Liang et al., 2019).

In our experiments, we use the bidirectional deep transition encoder, where each directional deep transition block consists of 1 L-GRU and 4 T-GRU. The decoder contains a query transition block and the decoder transition block, each of which consists of 1 L-GRU and 4 T-GRU. Therefore the DTMT consists of a 5 layer encoder and a 10 layer decoder, with a hidden size of 1,024. We use 8 NVIDIA V100 GPUs to train each model for about three weeks and the batch size is set to 4,096 tokens per GPU.

## 3 System Overview

In this section, we describe our system used in the WMT 2020 news shared task.

Figure 1 depicts the overview of our Wechat NMT. Our system can be divided into four parts, namely data filtering, synthetic data generation, in-domain finetuning, and ensemble. The synthetic generation part further includes the generation of out-of-domain and in-domain data. Next, we will illustrate these four parts.

### 3.1 Data Filter

Following previous work (Li et al., 2019), we filter the training bilingual corpus with the following rules:

- Normalize punctuation with Moses scripts.
- Filter out the sentences longer than 100 words, or exceed 40 characters in a single word.

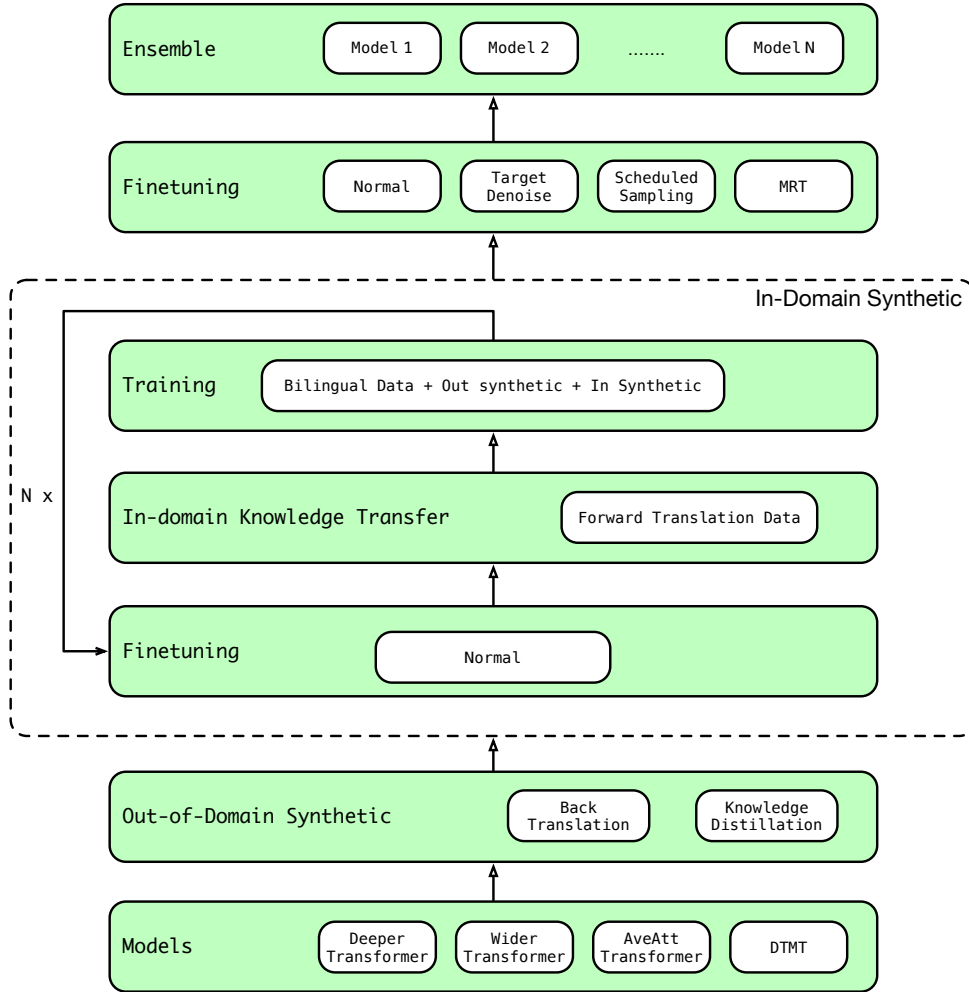


Figure 1: Architecture of WeChat NMT system. For simplicity, the data filtering module is ignored in the overview.

	NUM
Bilingual Data	20.7M
Chinese Monolingual Data	153.5M
English Monolingual Data	121.2M

Table 1: Statistics of all training data.

- Filter out the duplicated sentence pairs.
- The word ratio between the source and the target words must not exceed 1:4 or 4:1.

We also filter the monolingual corpus with the language model trained by the corresponding data of the bilingual training corpus.

In our experiments, the bilingual training data is a combination of News Commentary v15, Wiki Titles v2, WikiMatrix, CCMT and the UN corpus. The Chinese monolingual data includes News crawl, News Commentary, Common Crawl and Gigaword corpus. The English monolingual data includes News crawl, News discussions, Europarl

v10, News Commentary, Common Crawl, Wiki dumps and the Gigaword corpus. After data filtering, statistics of all training data are shown in Table 1.

### 3.2 Out-of-Domain Synthetic Data Generation

Now, we describe our techniques for constructing both out-of-domain and in-domain synthetic data. The out-of-domain synthetic corpus is generated via both large-scale back-translation and knowledge distillation to enhance the models’ performance for all domains. Then, we propose iterative in-domain knowledge transfer, which transfers in-domain knowledge to the huge monolingual corpus (i.e., Chinese), and builds our in-domain synthetic corpus. In the following sections, we elaborate above techniques in detail.

### 3.2.1 Large-scale Back-Translation

Back-translation is shown to be very effective to boost the performance of NMT models in both academic research (Hoang et al., 2018; Edunov et al., 2018) and previous years’ WMT competitions (Deng et al., 2018; Sun et al., 2019; Ng et al., 2019; Xia et al., 2019). Following their work, we also train baseline English-to-Chinese models with the parallel data provided by WMT2020. Both the Left-to-Right Transformer (L2R) and the Right-to-Left Transformer (R2L) are used to translate the filtered monolingual English corpus combined with the English side of golden parallel bitext to Chinese. Then the generated Chinese text and the original English text are regarded as the source side and target side, respectively.

In practice, it costs us 7 days on 5 NVIDIA V100 GPU machines to generating all back-translated data.

### 3.2.2 Knowledge Distillation

Knowledge distillation (KD) is proven to be a powerful technique for NMT (Kim and Rush, 2016) to transfer knowledge from the teacher model to student models. In particular, we first use the teacher models to generate synthetic corpus in the forward direction (i.e., Chinese→English). Then, we use the generated corpus to train our student models.

In this work, with baseline Chinese→English models (i.e., L2R and R2L) as teacher models, we translate the Chinese sentences of the parallel corpus to English to form our synthetic KD dataset. The knowledge distillation costs about 2 days on 2 NVIDIA V100 GPU machines to generate all synthetic data.

### 3.3 Iterative In-domain Knowledge Transfer

Since in-domain finetuning demonstrates substantial BLEU improvements (Sun et al., 2019; Li et al., 2019), we speculate that the parallel data and the dev/test sets fall in different domains. Therefore, adapting our models to the target domain in advance will provide gains over the dev/test sets and give a better initialization point for in-domain finetuning. To this end, we use knowledge transfer to inject more in-domain information into our synthetic data.

In particular, we first use normal finetuning (see Section 3.5) to equip our models with in-domain knowledge. Then, we ensemble these models and use the ensemble model to translate the Chinese monolingual corpus into English. For our ensemble

translator, we use 4 models with different architectures. Next, we pair original Chinese sentences with generated in-domain pseudo English sentences to form a pseudo parallel corpus. So far, the in-domain knowledge from ensembled models is transferred to the generated pseudo-parallel corpus. Finally, we retrain our model with both the in-domain pseudo-parallel and out-of-domain parallel data.

We refer to the above process as the in-domain knowledge transfer. In our experiments, we find that iteratively performing the in-domain knowledge transfer can further provide improvements (see Table 2). For each iteration, we replace the in-domain synthetic data and retrain our models, and it costs about 10 days on 8 NVIDIA V100 GPU machines. For the final submission, the knowledge transfer is conducted twice.

### 3.4 Data Augmentation

Aside from synthetic data generation, we also apply two data augmentation methods over our synthetic corpus. Firstly, adding synthetic/natural noises to training data is widely applied in the NLP fields (Li et al., 2017; Belinkov and Bisk, 2017; Cheng et al., 2019) to improve model robustness and enhance model performance. Therefore, we proposed to add token-level synthetic noises. Concretely, we perform random replace, random delete, and random permutation over our data. The probability of enabling each of the three operations is 0.1. We refer to this corrupted corpus as *Noisy* data.

Secondly, as illustrated in (Edunov et al., 2018), sampling generation over back-translation shows its potential in building robust NMT systems. Consequently, we investigate the performance of sampled synthetic data. For back-translated data, we replace beam search with sampling in its generation. For in-domain synthetic data, we replace the golden Chinese with the back sampled pseudo Chinese sentences. We refer to the data with sampling generation as *Sample* data.

As a special case, we refer to the without augmentation data as *Clean* data.

### 3.5 In-domain Finetuning

We train the model on large-scale out-of-domain data until convergence and then finetune it on small-scale in-domain data, which is widely used for domain adaption (Luong and Manning, 2015; Li et al., 2019). Specifically, we take Chinese→English test sets of WMT 17 and 18 as in-domain data, and filter out documents that are originally created in

English (Sun et al., 2019). We name above finetuning approach as *normal* finetuning. In all our finetuning experiments, we set the batch size to 4096, and finetune the model for around 400 steps<sup>1</sup> on the in-domain data.

Furthermore, the well-known problem of exposure bias in sequence-to-sequence generation becomes more serious under domain shift (Wang and Sennrich, 2020). To solve this issue, we further explore some *advanced* finetuning approaches and describe details in the following paragraphs.

**Parallel Scheduled Sampling.** We apply a two-pass decoding strategy for the Transformer decoder when finetuning, which is named as parallel scheduled sampling (Mihaylova and Martins, 2019; Duckworth et al., 2019). In the first pass, we obtain model predictions as a standard Transformer, and then mix the predicted sequence with the golden target sequence. In the second pass, we feed the above mixture of both golden and predicted tokens as decoder inputs for the final prediction. Thus the problem of the training-generation discrepancy is alleviated in the finetuning stage. According to our preliminary experiments, we set the proportion of predicted tokens in mixed tokens to 50%.

**Target Denoising.** In the training stage, the model never sees its own errors. Thus the model trained with teacher-forcing is prone to accumulated errors in testing (Ranzato et al., 2015). To mitigate this training-generation discrepancy, we add noisy perturbations into decoder inputs when finetuning. Thus the model becomes more robust to prediction errors by target denoising. Specifically, the finetuning data generator chooses 30% of sentence pairs to add noise, and keeps the remaining 70% of sentence pairs unchanged. For a chosen pair, we keep the source sentence unchanged, and replace the  $i$ -th token of the target sentence with (1) a random token of the current target sentence 15% of the time (2) the unchanged  $i$ -th token 85% of the time.

**Minimum Risk Training.** To further avoid the problem of exposure bias, we propose to use minimum risk training (Shen et al., 2016) in the finetuning stage, which directly optimizes the expected BLEU score instead of the Cross-Entropy loss, and

<sup>1</sup>According our experiments, finetuning with more steps will make the model easy to overfit on the small in-domain data.

naturally avoids exposure bias. Specifically, the objective is computed by,

$$R(\theta) = \sum_{s=1}^S \sum_{y \in S(x^{(s)})} Q(y|x^{(s)}; \theta, \alpha) \Delta(y, y^{(s)}), \quad (1)$$

where  $x^{(s)}$  and  $y^{(s)}$  are two paired sentences.  $\Delta$  denotes a risk function and  $S(x^{(s)}) \in Y$  is a sampled subset of full search space. Then, the distribution  $Q$  is defined over space  $S(x^{(s)})$ ,

$$Q(y|x^{(s)}; \theta, \alpha) = \frac{P(y|x^{(s)}; \theta)^\alpha}{\sum_{y' \in S(x^{(s)})} P(y'|x^{(s)}; \theta)^\alpha}. \quad (2)$$

In practice, we use 4 candidates for each source sentence  $x^{(s)}$ . Although the paper claimed that sampling generates better candidates, we find that the beam search performs better in our extremely large Transformer model. The risk function we used is the 4-gram sentence-level BLEU (Chen and Cherry, 2014) and we tune the optimal  $\alpha$  via grid search within  $\{0.005, 0.05, 0.5, 1, 1.5, 2\}$ . Each model is fine-tuned for a max of 1000 steps.

### 3.6 Ensemble

We split each training data into three shards among *Clean*, *Noisy* and *Sample* data respectively, which yields a total number of 9 shards. For each shard, we train seven varieties (two Deeper transformers, two Wider transformers, two AANs and one DTMT) with different model architecture. Then we apply four finetuning approaches on each model, thus the total number of models is quadrupled (about 200 models). For ensemble, it is difficult and inefficient to enumerate over all combinations of candidate models (e.g., grid search). Therefore a pruning strategy for model selection is necessary when ensemble. We try to greedily select the top-performing models for the ensemble. However, only slight improvement is obtained (less than 0.1 BLEU), as our models are too similar to each other after finetuning.

To further promote diversity among candidate models, we propose the self-bleu driven pruning strategy for *advanced* ensemble. Specifically, we take the translations of one model as hypotheses and translations of other models as references. Then we calculate the BLEU score for each model to evaluate its diversity among other models. Models with small BLEU scores are selected for ensemble, and vice versa. According to our experiments, we observe that (1) AAN and DTMT show

SETTINGS	DEEPER	WIDER	AVEATT	DTMT
Baseline	26.24	26.35	26.17	26.08
+ Back Translation	29.64	29.70	29.48	28.88
+ <i>Finetune</i>	35.71	35.89	35.80	35.03
+ 1st In-domain Knowledge Transfer	38.14	38.22	38.21	37.98
+ <i>Finetune</i>	38.36	38.25	38.13	37.85
+ 2nd In-domain Knowledge Transfer	38.32	38.29	38.34	38.05
+ <i>Finetune</i>	38.49	38.31	38.38	38.12
+ <i>Advanced Finetune</i>	39.08	39.12	38.93	38.66
+ Normal Ensemble	39.19			
+ Advanced Ensemble*	<b>39.89</b>			

Table 2: Case-sensitive BLEU scores (%) on the Chinese→English *newstest2019*, where ‘\*’ denotes the submitted system. For each model architecture, we report the highest score among the three shards of clean data.

FINETUNING APPROACH	DEEPER	WIDER	AVEATT	DTMT
Normal	38.49	38.31	38.38	38.12
Parallel Scheduled Sampling	38.76	38.84	<b>38.93</b>	–
Target Denoising	38.88	38.92	38.63	<b>38.66</b>
Minimum Risk Training	<b>39.08</b>	<b>39.12</b>	38.78	38.45

Table 3: Case-sensitive BLEU scores (%) on the Chinese→English *newstest2019* for different finetuning approaches after the 2nd in-domain knowledge transfer. For each model architecture, we report the highest score among three shards of clean data and bold the best result among different finetuning approaches.

a clear difference with other architectures; (2) data sharding is effective to promote diversity, especially for models trained with *Clean* data; (3) different finetuning approaches cannot bring diversity for the same model. Under the guidance of self-bleu scores, our *advanced* ensemble models consist of 20 single models with differences in model architectures, data types, shards and finetuning approaches. As shown in Table 2, the *advanced* ensemble achieves absolute improvements over the normal ensemble (up to 0.7 BLEU improvements).

## 4 Experiments

### 4.1 Settings

All of our experiments are carried out on 15 machines with 8 NVIDIA V100 GPUs each of which has 32 GB memory. We use cased BLEU scores calculated with Moses<sup>2</sup> mteval-v13a.pl script as evaluation metric. *newstest2019* is used as the development set. For all experiments, we use LazyAdam optimizer with  $\beta_1 = 0.9$ ,  $\beta_2 = 0.998$  and  $\epsilon = 10^{-9}$ . The learning rate is set to 2.0 and decay with training steps. We use warmup step = 8000. We set beam size to 4 and alpha to 0.6 during decoding.

<sup>2</sup><http://www.statmt.org/moses/>

### 4.2 Pre-processing and Post-processing

We segment the Chinese sentences with an in-house word segmentation tool. For English sentences, we successively apply punctuation normalization, tokenization and truecasing with the scripts provided in Moses. To enable open-vocabulary, we use byte pair encoding BPE (Sennrich et al., 2016b) with 32K operations for both Chinese and English sides.

For the post-processing, we apply de-truecasing and de-tokenizing on the English translations with the scripts provided in Moses.

### 4.3 Main Results

Table 2 shows that the translation quality is largely improved with proposed techniques. We observe a solid improvement of 2.8~3.4 BLEU for the baseline system after back translation. In-domain finetuning yields substantial improvements among all model architectures, which are 6.07~6.32 BLEU. The finetuned Transformer models achieve about 35.89 BLEU scores, and the DTMT achieves a 35.03 BLEU score. These findings demonstrate that the domain of training corpus is apart from the target domain, and hence domain adaptation has great potential in improving model performance in the target domain.

As described in Section 3.3, we inject the in-domain knowledge into our monolingual corpus.

Two In-domain knowledge transfers provide another up to 3.02 BLEU score gain (i.e., from about 35.03 to 38.05). The in-domain knowledge transfer brings more improvement compared with the normal finetuned models. Besides, we find that models further finetuned after in-domain transfer performs slightly better (about 0.1 BLEU). The improvement suggests that although in-domain transfer has already provided plenty of in-domain knowledge, it still has room for in-domain finetuning. We further apply *advanced* finetuning techniques to our models, as described in Section 3.5. The advanced finetuning further brings about 0.81 BLEU score gains, and we obtain our best single model with 39.12 BLEU scores.

In our preliminary ensemble experiments, we combine some top-performing models at each decoding step, but only achieve slight improvement over single models (about 0.1 BLEU). With our *advanced* ensemble strategies in section 3.6, further improvements are achieved over the normal ensemble (0.7 BLEU). As a result, our WMT 2020 Chinese→English submission achieves a cased BLEU score of 36.9 on *newstest2020*, which is the highest among all submissions.

#### 4.4 Effects of Advanced Finetuning Approaches

In this section, we describe our experiments on advanced finetuning. Here we take clean models as examples, but models trained with noisy data and sampled data show similar trends.

As shown in Table 3, all three advanced finetuning methods significantly outperform normal finetuning. For Wider and Deeper Transformers, Minimum Risk Training provides the highest BLEU gain, which is 0.81. For the Average Attention Transformer, Parallel Schedule Sampling improves the model performance from 38.38 to 38.93. For the DTMT model, Target Denoising performs the best, improving from 38.12 to 38.66. These findings are in line with the conclusion of Wang and Sennrich (2020) that links exposure bias with domain shift. For each type of model, we only keep the best-performing finetuned one for the final model ensemble.

## 5 Conclusion

In this paper, we introduce the system WeChat submitted for the WMT 2020 shared task on Chinese→English news translation. Our system

is based on the Transformer (Vaswani et al., 2017) with different variants and the DTMT (Meng and Zhang, 2019) architecture. Data selection, several effective synthetic data generation approaches (i.e., back-translation, knowledge distillation, and iterative in-domain knowledge transfer), advanced finetuning approaches (i.e., parallel scheduled sampling, target denoising, and minimum risk training) and self-bleu based model ensemble are employed and proven effective in our experiments. Our constrained Chinese→English system achieved 36.9 case-sensitive BLEU score which is the highest among all submissions.

## 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*.
- Boxing Chen and Colin Cherry. 2014. A systematic comparison of smoothing techniques for sentence-level bleu. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 362–367.
- Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust neural machine translation with doubly adversarial inputs. *arXiv preprint arXiv:1906.02443*.
- Yongchao Deng, Shanbo Cheng, Jun Lu, Kai Song, Jingang Wang, Shenglan Wu, Liang Yao, Guchun Zhang, Haibo Zhang, Pei Zhang, et al. 2018. Alibaba’s neural machine translation systems for wmt18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 368–376.
- Daniel Duckworth, Arvind Neelakantan, Ben Goodrich, Lukasz Kaiser, and Samy Bengio. 2019. Parallel scheduled sampling. *arXiv preprint arXiv:1906.04331*.
- Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.
- Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24.
- Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. *arXiv preprint arXiv:1606.07947*.

- Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, Kai Feng, Hexuan Chen, Tengbo Liu, Yanyang Li, Qiang Wang, Tong Xiao, and Jingbo Zhu. 2019. [The NiuTrans machine translation systems for WMT19](#). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 257–266, Florence, Italy. Association for Computational Linguistics.
- Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*.
- Yunlong Liang, Fandong Meng, Jinchao Zhang, Jinan Xu, Yufeng Chen, and Jie Zhou. 2019. [A novel aspect-guided deep transition model for aspect based sentiment analysis](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5569–5580, Hong Kong, China. Association for Computational Linguistics.
- Yijin Liu, Fandong Meng, Jinchao Zhang, Jinan Xu, Yufeng Chen, and Jie Zhou. 2019. [GCDT: A global context enhanced deep transition architecture for sequence labeling](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2431–2441, Florence, Italy. Association for Computational Linguistics.
- Minh-Thang Luong and Christopher D Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 76–79.
- Fandong Meng and Jinchao Zhang. 2019. DTMT: A novel deep transition architecture for neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 224–231.
- 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.
- Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair’s wmt19 news translation task submission. *arXiv preprint arXiv:1907.06616*.
- Marc’Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. [Improving neural machine translation models with monolingual data](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. [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.
- Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. [Minimum risk training for neural machine translation](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1692, Berlin, Germany. Association for Computational Linguistics.
- Meng Sun, Bojian Jiang, Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. [Baidu neural machine translation systems for WMT19](#). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 374–381, Florence, Italy. Association for Computational Linguistics.
- 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.
- Chaojun Wang and Rico Sennrich. 2020. [On exposure bias, hallucination and domain shift in neural machine translation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3544–3552, Online. Association for Computational Linguistics.
- Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019. [Learning deep transformer models for machine translation](#). pages 1810–1822.
- Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019. [Exploiting monolingual data at scale for neural machine translation](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4207–4216, Hong Kong, China. Association for Computational Linguistics.
- Yingce Xia, Xu Tan, Fei Tian, Fei Gao, Weicong Chen, Yang Fan, Linyuan Gong, Yichong Leng, Renqian Luo, Yiren Wang, et al. 2019. Microsoft research asia’s systems for wmt19. *arXiv preprint arXiv:1911.06191*.
- Biao Zhang, Deyi Xiong, and Jinsong Su. 2018. [Accelerating neural transformer via an average attention](#)



network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1789–1798, Melbourne, Australia. Association for Computational Linguistics.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Tegygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1097–1100.