

# BJTU-WeChat’s Systems for the WMT22 Chat Translation Task

Yunlong Liang<sup>1\*</sup>, Fandong Meng<sup>2</sup>, Jinan Xu<sup>1†</sup>, Yufeng Chen<sup>1</sup> and Jie Zhou<sup>2</sup>

<sup>1</sup>Beijing Key Lab of Traffic Data Analysis and Mining,  
Beijing Jiaotong University, Beijing, China

<sup>2</sup>Pattern Recognition Center, WeChat AI, Tencent Inc, China  
{yunlongliang, jaxu, chenyf}@bjtu.edu.cn  
{fandongmeng, withtomzhou}@tencent.com

## Abstract

This paper introduces the joint submission of the Beijing Jiaotong University and WeChat AI to the WMT’22 chat translation task for English↔German. Based on the Transformer (Vaswani et al., 2017), we apply several effective variants. In our experiments, we utilize the pre-training-then-fine-tuning paradigm. In the first pre-training stage, we employ data filtering and synthetic data generation (i.e., back-translation, forward-translation, and knowledge distillation). In the second fine-tuning stage, we investigate speaker-aware in-domain data generation, speaker adaptation, prompt-based context modeling, target denoising fine-tuning (Meng et al., 2020), and boosted self-COMET-based model ensemble. Our systems achieve 0.810 and 0.946 COMET (Rei et al., 2020) scores<sup>1</sup> on English→German and German→English, respectively. The COMET scores of English→German and German→English are the highest among all submissions.

## 1 Introduction

We participate in the WMT 2022 shared task on chat translation in two language directions, English→German and German→English. In this year’s chat translation task, we apply the two-stage training strategy. In the first stage, we investigate model architecture and data augmentation. In the second stage, we mainly focus on exploiting speaker-aware in-domain data augmentation, speaker adaptation, prompt-based context modeling, target denoising fine-tuning (Meng et al., 2020), and model ensemble strategies. This task aims to build machine translation systems to translate conversational text and thus supports fluent communication between an agent speaking in En-

glish and a customer speaking in a different language (e.g., German), which is different from the first pre-training stage (Farajian et al., 2020; Liang et al., 2021a, 2022a; Liu et al., 2021; Gain et al., 2021, 2022; Buschbeck et al., 2022). Therefore, we mainly pay attention to the second fine-tuning stage.

In the first pre-training stage, we follow previous work (Meng et al., 2020; Zeng et al., 2021; Meng and Zhang, 2019; Yan et al., 2020) and utilize several effective Transformer variants. Specifically, we combine the Multi-Head-Attention (Vaswani et al., 2017), Average Attention Transformer (Zhang et al., 2018), and Talking-Heads Attention (Shazeer et al., 2020), which have shown significant model performance and diversity. For data augmentation, we employ the back-translation method to use the target-side monolingual data and apply the forward-translation to leverage the source-side monolingual data. To fully utilize the source-side of bilingual data, we use the sequence-level knowledge distillation method (Kim and Rush, 2016).

In the second fine-tuning stage, for speaker-aware in-domain data augmentation, based on the BConTrasT (Farajian et al., 2020) dataset of the WMT20 chat translation task, we firstly adapt our pre-trained model to each speaker by using the speaker tag as a pseudo token and then apply it to the Taskmaster-1 (Byrne et al., 2019) corpus to generate the speaker-aware in-domain data. For speaker adaptation, we follow previous work (Moghe et al., 2020) to prepend the corresponding speaker tag to each utterance on both the source and the target side to get a speaker-aware dataset. For prompt-based context modeling, we exploit the prompt learning to incorporate the bilingual context and then apply the target denoising fine-tuning method (Meng et al., 2020) to train our model. For the model ensemble, inspired by Zeng et al. (2021), we select high-potential can-

\*Work was done when Yunlong was interning at Pattern Recognition Center, WeChat AI, Tencent Inc, China.

†Jinan Xu is the corresponding author.

<sup>1</sup>The COMET is the official automatic evaluation metric.

didate models from two aspects, namely model performance (COMET scores) and model diversity (Self-COMET scores among all candidate models). Based on this, we design a search algorithm to gradually select the current best model of the model candidate pool for the final model ensemble.

## 2 Model Architectures

In this section, we describe the model architectures we used in two translation directions, where we mainly follow the previous state-of-the-art models (Zeng et al., 2021). We also refer readers to read the paper for details.

### 2.1 Model Configurations

Given the strong capacity of deeper and wider architectures, we use them in our experiments. Specifically, following Zeng et al. (2021), we use 20-layer encoders for deeper models and set the hidden size to 1024 for all models. We set the decoder depth to 10. For the wider ones, we adopt 12 encoder layers, 2048 for hidden size, and 8192 to 15000 for filter sizes.

### 2.2 Transformer Variants

**Average Attention Transformer.** Following Zeng et al. (2021), the average attention transformer (Zhang et al., 2018) are employed to add model diversity. In the AAN, the context representation  $g_i$  for each input embedding is calculated as follows:

$$g_i = \text{FFN}\left(\frac{1}{i} \sum_{k=1}^t y_k\right),$$

where  $y_k$  is the input embedding for step  $k$  and  $t$  is the current time step. FFN denotes the position-wise feed-forward network (Vaswani et al., 2017).

**Talking Heads Attention.** Similarly, talking-heads attention (Shazeer et al., 2020) also performs well in Zeng et al. (2021), which can transform the attention-logits and the attention scores and thus allow information interaction among attention heads by adding two linear projection layers  $W_l$  and  $W_a$ :

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{k}}W_l\right)W_aV.$$

## 3 System Overview

In this section, we describe our system used in the WMT 2022 chat translation shared task, which

includes two parts, namely general pre-training and in-domain fine-tuning. The pre-training part includes data filtering and synthetic data generation. The in-domain fine-tuning consists of speaker-aware in-domain data generation, speaker adaptation, prompt-based context modeling, the target denoising fine-tuning (Meng et al., 2020), and boosted Self-COMET-based model ensemble.

### 3.1 General Pre-training

#### 3.1.1 Data Filtering

We filter the bilingual training corpus (including synthetic parallel data) with the following rules (Zeng et al., 2021): 1) Normalize punctuation; 2) Remove the sentence whose length is more than 100 words or a single word that exceeds 40 characters; 3) Filter out the duplicated sentence pairs; 4) Delete the sentence whose word ratio between the source and the target words exceeds 1:4 or 4:1.

#### 3.1.2 Synthetic Data Generation

For data augmentation, we obtain the general domain synthetic data via back-translation, forward-translation, and knowledge distillation.

**Tagged Back-Translation.** Previous work has shown that different methods of generating pseudo corpus have a different influence on translation performance (Edunov et al., 2018; Hoang et al., 2018; Zeng et al., 2021). Following them, we attempt two generating strategies: 1) Beam Search: produce translation by beam search (beam size = 5). 2) Sampling Top-k: Select a word randomly from top-k ( $k = 15$ ) words when inference.

**Forward-Translation.** We then ensemble models to forward-translate the monolingual data of the source language to further enhance model performance. We obtain a stable improvement in both directions, which is consistent with previous work (Zeng et al., 2021).

**Knowledge Distillation.** Knowledge Distillation aims to transfer knowledge from the teacher model to student models, which has shown effective for NMT (Kim and Rush, 2016; Wang et al., 2021; Zeng et al., 2021). Specifically, we first use the teacher model to generate synthetic corpus in the forward direction (i.e., En→De). Then, we train our student models with the generated corpus.

Note that we prefix all the synthetic sentences by appending a pseudo tag <BT> when jointly training with genuine data.

## 3.2 In-domain Fine-tuning

### 3.2.1 Speaker-aware In-domain Data Generation

Inspired by Moghe et al. (2020), we prepend the corresponding speaker tag (the `<agent>` or the `<customer>`) to each utterance on both the source and the target side to get a speaker-aware dataset based on the BConTrasT dataset of the WMT20 chat translation task (Farajian et al., 2020). Secondly, we adapt our pre-trained model to each speaker on the speaker-aware dataset. Then, we apply the adapted model to the monolingual Taskmaster-1 (Byrne et al., 2019) corpus, which is the original source of BConTrasT (Farajian et al., 2020), to generate the speaker-aware in-domain data.

### 3.2.2 Speaker Adaptation

As a special characteristic of chat translation, distinguishing between the two speaker roles plays an important role as they both form the complete dialogue. And modeling the speaker characteristic has been demonstrated effective in previous work (Moghe et al., 2020; Liang et al., 2021c, 2022b, 2021b, 2022c). Therefore, our data used in the fine-tuning has a corresponding speaker tag (the `<agent>` or the `<customer>`) appended in the first token of each utterance.

### 3.2.3 Prompt-based Context Modeling

Previous studies (Wang et al., 2020; Moghe et al., 2020) have shown that the multi-encoder framework cannot improve the model performance after using the context in the chat translation task, while a unified model (Ma et al., 2020; Liang et al., 2021c) can. Therefore, we also investigate incorporating the context in the unified model with prompt learning (without modifying the model architecture). Specifically, we add two preceding bilingual contexts at the tail of each utterance with an indicator `<context begins>`, where we also use a special tag `<SEP>` to separate different utterances of the bilingual context. In this way, our model with context modeling can achieve a better COMET.

### 3.2.4 Target Denoising Fine-tuning

To bridge the exposure bias (Ranzato et al., 2016), we add noisy perturbations into decoder inputs when fine-tuning. Therefore, the model becomes more robust to prediction errors by target denoising fine-tuning (Zhang et al., 2019; Meng et al., 2020). Specifically, the fine-tuning data generator chooses

---

### Algorithm 1: Boosted Self-COMET-based Ensemble (BSCE)

---

**Input:**

- List of candidate models  $\mathbb{M} = \{m_1, \dots, m_n\}$
- Valid set COMET for each model  $\mathbb{C} = \{c_1, \dots, c_n\}$
- Average Self-COMET for each model  $\mathbb{S} = \{s_1, \dots, s_n\}$
- The number of models  $n$
- The number of ensemble models  $e$

**Output:** Selected Model Pool  $\mathbb{P}$ 

- 1: **for**  $i \leftarrow 1$  to  $n$  **do**
  - 2:    $weight = \frac{(max(\mathbb{S}) - min(\mathbb{S}))}{(max(\mathbb{C}) - min(\mathbb{C}))}$
  - 3:    $score_i = (c_i - min(\mathbb{C})) \cdot weight + (max(\mathbb{S}) - s_i)$
  - 4: **end for**
  - 5: Add the highest score model to candidates list  $\mathbb{P} = \{m_{top}\}$
  - 6: **while**  $|\mathbb{P}| < e$  **do**
  - 7:    $index = \arg \min_i \frac{1}{|\mathbb{M} - \mathbb{P}|} \sum_{i \in \mathbb{M} - \mathbb{P}, j \in \mathbb{P}} BLEU(i, j)$
  - 8:   Add  $m_{index}$  to candidate list  $\mathbb{P}$
  - 9: **end while**
  - 10: **return**  $\mathbb{P}$
- 

30% of utterance pairs (Note that we do not include the indicator word and the bilingual context) to add noise and keeps the remaining 70% of sentence pairs unchanged. For a chosen pair, we keep the source sentence untouched and replace the  $i$ -th token of the target sentence with (I) a random token of the current target sentence in 15% probability and (II) the unchanged  $i$ -th in 85% probability.

### 3.2.5 Boosted Self-COMET-based Model Ensemble (BSCE)

After we get plenty of fine-tuned models, how to search for the best combination for the ensemble model is a difficult question. Inspired by Zeng et al. (2021), we propose a Boosted Self-COMET-based Ensemble (BSCE) algorithm, as shown in algorithm 1. Since the existing boosted Self-BLEU-based pruning strategy (Zeng et al., 2021) is designed for achieving higher BLEU scores with high efficiency, it can not help obtain better COMET scores. Therefore, we adapt it to COMET scores. Then, we can obtain the best ensemble models from  $n$  top models by a greedy search strategy.

The algorithm takes as input a list of  $n$  strong single models  $\mathbb{M}$ , COMET scores on the development set for each model  $\mathbb{C}$ , average Self-COMET scores

for each model  $\mathbb{S}$ , the number of models  $n$ , and the expected number of ensemble models  $e$ . The algorithm returns a set  $\mathbb{P}$  consisting of  $e$  selected models. We calculate the weighted score for each model (line 2). The weight (line 3) calculated is a trade-off between the development set COMET score and the Self-COMET score since the performance and the diversity play the same key role in ensemble (Zeng et al., 2021). Then the set  $\mathbb{P}$  initially contains the model  $m_{top}$  has the highest weighted score. Next, we iteratively re-compute the average Self-COMET between the remaining models in ‘ $\mathbb{M} - \mathbb{P}$ ’ and selected models in  $\mathbb{P}$ , based on which we select the model that has a minimum Self-COMET score into  $\mathbb{P}$ .

## 4 Experiments and Results

### 4.1 Setting

The implementation of our models is based on Fairseq<sup>2</sup>. All the single models in the first pre-training stage are carried out on 8 NVIDIA V100 GPUs (32 GB memory of each). And all the models in the second fine-tuning stage are conducted on 4 NVIDIA V100 GPUs. We use the Adam optimizer with  $\beta_1 = 0.9$ ,  $\beta_2 = 0.998$ . The batch size are set to 8192 and 4096 tokens per GPU for pre-training and fine-tuning, respectively. We set the ‘‘update-freq’’ parameter to 2 and 1 for both stages. The learning rate is set to 0.0005 and 0.0004 for two stages, respectively. We use the warmup step to 4000. We calculate COMET<sup>3</sup> score for all experiments which is officially recommended.

English and German sentences are segmented by Moses<sup>4</sup>. We apply punctuation normalization and Truecasing. We use byte pair encoding BPE (Sennrich et al., 2016) with 32K operations. For the post-processing, we apply de-truecasing and de-tokenizing on the English and German translations with the scripts provided in Moses.

### 4.2 Dataset

The data statistics of the two stages are shown in Table 1. For the general pre-training, the bilingual data is the combination of all parallel data in WMT21. For monolingual data, we use the News Crawl, Common Crawl, and Extended Common Crawl. For synthetic data generation, we back-translate all the target monolingual data and

<sup>2</sup><https://github.com/pytorch/fairseq>

<sup>3</sup><https://unbabel.github.io/COMET/html/index.html>

<sup>4</sup><http://www.statmt.org/moses/>

	General pre-training	In-domain fine-tuning
Bilingual Data	74.8M	17,847
Source Mono Data	332.8M	302,079
Target Mono Data	237.9M	-

Table 1: Statistics of all training data.

Models	En→De	De→En
Chat baseline w/o context	0.403	0.588
Chat baseline w context	0.376	0.680
Pre-trained deeper model w/o context	0.544	0.865
+ in-domain genuine data w/ context (FT1)	0.772	0.905
+ in-domain pseudo data w/ context (FT2)	0.767	0.903
+ in-domain both data w/ context (FT3)	0.781	0.908
Pre-trained wider model w/o context	0.604	0.879
+ in-domain genuine data w/ context (FT4)	0.782	0.908
+ in-domain pseudo data w/ context (FT5)	0.779	0.906
+ in-domain both data w/ context (FT6)	<b>0.785</b>	<b>0.909</b>

Table 2: COMET scores on the Valid set for both pre-trained models, and each of fine-tuned on (i) in-domain genuine data, (ii) in-domain pseudo data, and (iii) both in-domain data.

forward-translate the source monolingual data. For the in-domain fine-tuning, we use all the training, valid, and testing data of the wmt20 chat task as our training data. For monolingual data, we select the Taskmaster-1 (Byrne et al., 2019) corpus to build the pseudo-paired data using the method described in Section 3.2.1.

### 4.3 Results

We report COMET scores (Rei et al., 2020) on the validation set (generally, beam size = 5 and length penalty = 0.6).

**Pre-training and Fine-tuning.** The results in Table 2 show that all pre-trained models outperform the baseline models trained on the chat training data. We observe that in-domain fine-tuning of the pre-trained models always gives large gains even on the in-domain pseudo data. We also find that the performance of different model architectures comes close after in-domain fine-tuning. Though these models perform similarly, as they have different architectures or are trained on different data, they generate diverse translations and show a cumulative effect when ensemble.

**Final Submissions.** Table 3 shows the results of our primary submission on both the validation and test set. Note that all candidate models with different architectures or trained with different data are used for the ensemble. We find that our BSCE

Models	En→De	De→En
Best Single Model	0.785	0.909
+ Normal Ensemble	0.788	0.908
+ BSCE	0.790	0.911
+ BSCE + Large beam (*)	<b>0.792</b>	<b>0.913</b>
Official results on the Test set		
+ BSCE + Large beam (*)	<b>0.810</b>	<b>0.946</b>
Best Official	<b>0.810</b>	<b>0.946</b>

Table 3: Valid set COMET scores for ensemble with different strategies and the official COMET results of our submissions. “\*” indicates the primary system of our submissions.

Models	En→De	De→En
FT6 + no tag	0.779	0.904
FT6 + speaker	<b>0.785</b>	<b>0.909</b>

Table 4: Valid set COMET scores for fine-tuning with speaker tags .

is effective in both directions (more analyses are shown in Section 5.3). Inspired by Wang et al. (2020), we also tried large beam size. Finally, our primary system achieves the highest results among all submissions<sup>5</sup>.

## 5 Analysis

### 5.1 Effect of Speaker Tags

As shown in Table 4, we observe that the performance in both directions improves with the addition of tags, which is consistent with Moghe et al. (2020). It shows that adding the speaker tag indeed can improve the chat translation performance.

### 5.2 Effect of Prompt-based Context Modeling (PCM)

As shown in Table 5, we investigate the effect of the context. The bilingual context involves the utterance in mixed language. Therefore, we investigate the different contexts with prompt learning. The results show that the models achieve slight performance gains with suitable context. And using context in the same language was more beneficial than the mixed context, which is consistent with previous work (Moghe et al., 2020).

### 5.3 Effect of Boosted Self-COMET-based Ensemble (BSCE)

Inspired by the boosted Self-BLEU-based ensemble (Zeng et al., 2021), we propose the Boosted

<sup>5</sup><https://wmt-chat-task.github.io/>

Models	En→De	De→En
FT6 + w/o context	<b>0.782</b>	<b>0.905</b>
using previous context (mix language)		
FT6 + w/ PCM (+ 1 prev)	0.781	<b>0.905</b>
FT6 + w/ PCM (+ 2 prev)	0.779	0.901
FT6 + w/ PCM (+ 3 prev)	0.775	0.897
using previous context (same language)		
FT6 + w/ PCM (+ 1 prev)	<b>0.785</b>	<b>0.909</b>
FT6 + w/ PCM (+ 2 prev)	0.784	<b>0.909</b>
FT6 + w/ PCM (+ 3 prev)	0.782	0.904

Table 5: Valid set COMET scores for fine-tuning with different contexts. The numbers before “prev” indicate the number of preceding utterances used as context.

Self-COMET-based Ensemble. To verify its superiority, we first select the top 10 models with different architecture and training data. The results are shown in the “+Normal Ensemble” of Table 3. For the BSCE, we need to get the translation result of every model to calculate the Self-COMET. After that, we only need to perform the inference process once. Then, we can select the best models for the ensemble. Here, we select 10 models and 4 models for En→De and De→En, respectively. The results are shown in “+BSCE” of Table 3. Based on it, we obtain better results after using the large beam (beam sizes of 9 and 8 for En→De and De→En, respectively). These results show the effectiveness of our BSCE method.

## 6 Conclusions

We investigate the pre-training-then-fine-tuning paradigm to build chat translation systems, which are some effective transformer-based architectures. Our systems are also built on several popular data augmentation methods such as back-translation, forward-translation, and knowledge distillation. In the fine-tuning, we enhance our system by speaker-aware in-domain data generation, speaker adaptation, prompt-based context modeling, target denoising fine-tuning (Meng et al., 2020), and boosted self-COMET-based model ensemble. Our systems achieve 0.810 and 0.946 COMET (Rei et al., 2020) scores on English→German and German→English, respectively. These COMET scores are the highest among all submissions.

## Acknowledgements

The research work described in this paper has been supported by the National Key R&D Program of China (2020AAA0108005) and the National Na-

ture Science Foundation of China (No. 61976015, 61976016, 61876198 and 61370130). The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve this paper.

## References

- Bianka Buschbeck, Jennifer Mell, Miriam Exel, and Matthias Huck. 2022. “hi, how can i help you?” improving machine translation of conversational content in a business context. In *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*, pages 189–198.
- Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Ben Goodrich, Daniel Duckworth, Semih Yavuz, Amit Dubey, Kyu-Young Kim, and Andy Cedilnik. 2019. [Taskmaster-1: Toward a realistic and diverse dialog dataset](#). In *Proceedings of EMNLP-IJCNLP*, pages 4516–4525.
- Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. [Understanding back-translation at scale](#). In *Proceedings of EMNLP*, pages 489–500, Brussels, Belgium.
- M. Amin Farajian, António V. Lopes, André F. T. Martins, Sameen Maruf, and Gholamreza Haffari. 2020. [Findings of the WMT 2020 shared task on chat translation](#). In *Proceedings of WMT*, pages 65–75.
- Baban Gain, Ramakrishna Appicharla, Asif Ekbal, Muthusamy Chelliah, Soumya Chennabasavraj, and Nikesh Garera. 2022. Low resource chat translation: A benchmark for hindi–english language pair. In *Proceedings of the 15th biennial conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 83–96.
- Baban Gain, Rejwanul Haque, and Asif Ekbal. 2021. Not all contexts are important: The impact of effective context in conversational neural machine translation. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.
- Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. [Iterative back-translation for neural machine translation](#). In *Proceedings of WMT*, pages 18–24, Melbourne, Australia.
- Yoon Kim and Alexander M. Rush. 2016. [Sequence-level knowledge distillation](#). In *Proceedings of EMNLP*, pages 1317–1327, Austin, Texas.
- Yunlong Liang, Fandong Meng, Yufeng Chen, Jinan Xu, and Jie Zhou. 2021a. [Modeling bilingual conversational characteristics for neural chat translation](#). In *Proceedings of ACL*, pages 5711–5724.
- Yunlong Liang, Fandong Meng, Jinan Xu, Yufeng Chen, and Jie Zhou. 2022a. [MSCTD: A multimodal sentiment chat translation dataset](#). In *Proceedings of ACL*, pages 2601–2613, Dublin, Ireland.
- Yunlong Liang, Fandong Meng, Jinan Xu, Yufeng Chen, and Jie Zhou. 2022b. [Scheduled multi-task learning for neural chat translation](#). In *Proceedings of ACL*, pages 4375–4388, Dublin, Ireland.
- Yunlong Liang, Fandong Meng, Ying Zhang, Yufeng Chen, Jinan Xu, and Jie Zhou. 2021b. Infusing multi-source knowledge with heterogeneous graph neural network for emotional conversation generation. In *Proceedings of AAAI*, volume 35, pages 13343–13352.
- Yunlong Liang, Fandong Meng, Ying Zhang, Yufeng Chen, Jinan Xu, and Jie Zhou. 2022c. Emotional conversation generation with heterogeneous graph neural network. *Artificial Intelligence*, 308:103714.
- Yunlong Liang, Chulun Zhou, Fandong Meng, Jinan Xu, Yufeng Chen, Jinsong Su, and Jie Zhou. 2021c. [Towards making the most of dialogue characteristics for neural chat translation](#). In *Proceedings of EMNLP*, pages 67–79.
- Siyu Liu, Yuqi Sun, and Longyue Wang. 2021. Recent advances in dialogue machine translation. *Information*, 12(11):484.
- Shuming Ma, Dongdong Zhang, and Ming Zhou. 2020. [A simple and effective unified encoder for document-level machine translation](#). In *Proceedings of ACL*, pages 3505–3511.
- Fandong Meng, Jianhao Yan, Yijin Liu, Yuan Gao, Xianfeng Zeng, Qinsong Zeng, Peng Li, Ming Chen, Jie Zhou, Sifan Liu, and Hao Zhou. 2020. [WeChat neural machine translation systems for WMT20](#). In *Proceedings of WMT*, pages 239–247, Online.
- Fandong Meng and Jinchao Zhang. 2019. [DTMT: A novel deep transition architecture for neural machine translation](#). In *Proceedings of AAAI*, pages 224–231.
- Nikita Moghe, Christian Hardmeier, and Rachel Bawden. 2020. [The university of edinburgh-uppsala university’s submission to the wmt 2020 chat translation task](#). In *Proceedings of WMT*, pages 471–476.
- Marc’Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. [Sequence level training with recurrent neural networks](#). In *Proceedings of ICLR*.
- Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. [COMET: A neural framework for MT evaluation](#). In *Proceedings of EMNLP*, pages 2685–2702, Online.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. [Neural machine translation of rare words with subword units](#). In *Proceedings of ACL*, pages 1715–1725, Berlin, Germany.
- Noam Shazeer, Zhenzhong Lan, Youlong Cheng, Nan Ding, and Le Hou. 2020. [Talking-heads attention](#).

- 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 *Proceedings of NIPS*, pages 5998–6008.
- Fusheng Wang, Jianhao Yan, Fandong Meng, and Jie Zhou. 2021. [Selective knowledge distillation for neural machine translation](#). In *Proceedings of ACL*, pages 6456–6466, Online.
- Longyue Wang, Zhaopeng Tu, Xing Wang, Li Ding, Liang Ding, and Shuming Shi. 2020. [Tencent ai lab machine translation systems for wmt20 chat translation task](#). In *Proceedings of WMT*, pages 481–489.
- Jianhao Yan, Fandong Meng, and Jie Zhou. 2020. [Multi-unit transformers for neural machine translation](#). In *Proceedings of EMNLP*, pages 1047–1059, Online.
- Xianfeng Zeng, Yijin Liu, Ernan Li, Qiu Ran, Fandong Meng, Peng Li, Jinan Xu, and Jie Zhou. 2021. [WeChat neural machine translation systems for WMT21](#). In *Proceedings of WMT*, pages 243–254, Online. Association for Computational Linguistics.
- Biao Zhang, Deyi Xiong, and Jinsong Su. 2018. [Accelerating neural transformer via an average attention network](#). In *Proceedings of ACL*, pages 1789–1798, Melbourne, Australia. Association for Computational Linguistics.
- Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. 2019. [Bridging the gap between training and inference for neural machine translation](#). In *Proceedings of ACL*, pages 4334–4343, Florence, Italy.