TenTrans Large-Scale Multilingual Machine Translation System for WMT21

Wanying Xie^{1,2} Bojie Hu¹ Han Yang¹ Dong Yu² Qi Ju^{1*} ¹ TencentMT Oteam, China

² Beijing Language and Culture University, China

xiewanying07@gmail.com, yudong@blcu.edu.cn

{bojiehu, sharryyang, damonju}@tencent.com

Abstract

This paper describes TenTrans large-scale multilingual machine translation system for WMT 2021. We participate in the Small Track 2 in five South East Asian languages, thirty directions: Javanese, Indonesian, Malay, Tagalog, Tamil, English. We mainly utilized forward/back-translation, in-domain data selection, knowledge distillation, and gradual fine-tuning from the pre-trained model FLORES-101. We find that forward/backtranslation significantly improves the translation results, data selection and gradual fine-tuning are particularly effective during adapting domain, while knowledge distillation brings slight performance improvement. Also, model averaging is used to further improve the translation performance based on these systems. Our final system achieves an average BLEU score of 28.89 across thirty directions on the test set.

1 Introduction

We participate in the WMT 2021 large-scale multilingual machine translation task small track 2 in 6 languages: English, Indonesian, Javanese, Malay, Tamil, Tagalog (briefly, En, Id, Jv, Ms, Ta, Tl). Any two of these languages translated into each other produces a total of 30 directions, including English↔Indonesian, English↔Javanese, English \leftrightarrow Malay, English \leftrightarrow Tamil, English \leftrightarrow Tagalog, Indonesian↔Javanese, Indonesian↔Malay, Indonesian↔Tamil, Indonesian↔Tagalog, Javanese \leftrightarrow Malay, Javanese \leftrightarrow Tamil, Javanese \leftrightarrow Tagalog, Malay↔Tamil, Malay↔Tagalog and Tamil↔Tagalog. To meet the requirements for data restrictions, our systems are all built with constrained data sets. For all systems, we adopt a universal encoder-decoder architecture that shares

parameters across all languages (Johnson et al., 2017).

Our systems are based on several techniques and approaches. We experiment with base and deeper Transformer (Vaswani et al., 2017) architectures to get reliable baselines, fine-tune the pre-training model FLORES-101 (Goyal et al., 2021) to further improve the baseline system. Moreover, we generate pseudo bilingual sentences from the large-scale monolingual data, apply sequence level knowledge distillation (Kim and Rush, 2016) on partial language pairs, and try a more effectively fine-tuning strategy to domain adaptation (Gu et al., 2021). Particularly in the language pairs with inferior translations, we specifically improve their performance. All of these technologies have improved our systems, particularly data selection and gradual finetuning. We carefully rethought this strategy and found the main gain may come from in-domain knowledge adaptation.

This paper was structured as follows: Section 2 describes the data set. Then, we present a detailed overview of our systems in Section 3. The experiment settings and main results are shown in Section 4. Finally, we conclude our work in Section 5.

2 Data Prepration

We use FLORES-101 SentencePiece (SPM)¹ tokenizer model with 256K tokens to tokenize bitext and monolingual sentences². Since it is important to clean data strictly (Wang et al., 2018), we follow m2m-100 data preprocessing procedures³ to filter bitext data. The rules are as follows:

• Remove sentences with more than 50% punctuation.

^{*}Corresponding author: Qi Ju.

Our code, data, and model can be obtained at https://github.com/TenTrans/TenTrans

¹https://github.com/google/sentencepiece

²https://dl.fbaipublicfiles.com/flores101/pretrained_models/ flores101_mm100_615M.tar.gz

³https://github.com/pytorch/fairseq/tree/master/examples/ m2m_100

	En⇔Id	En⇔Jv	En⇔Ms	En⇔Ta	En⇔Tl	$Id{\leftrightarrow}Jv$	$Id{\leftrightarrow}Ms$	Id⇔Ta
No filter	36.15M	1.52M	7.43M	1.19M	6.97M	0.75M	4.23M	0.46M
Filtered	33.67M	1.41M	7.03N	1.06M	6.15M	0.67M	3.97M	0.41M
	Id⇔Tl	Jv↔Ms	Jv⇔Ta	Jv⇔Tl	Ms⇔Ta	Ms⇔Tl	Ta⇔Tl	
No filter	Id⇔Tl 2.56M	$Jv \leftrightarrow Ms$ 0.41M	Jv⇔Ta 0.06M	Jv⇔Tl 0.74M	Ms⇔Ta 0.33M	Ms↔Tl 1.27M	Ta⇔Tl 0.53M	

Table 1: Number of sentences in bitext data sets.

	En	Id	Jv	Ms	Та	Tl
No filter	126.44M	5.46M	0.41M	1.87M	2.06M	0.41M
Filtered	113.36M	5.26M	0.38N	1.85M	2.03M	0.39M

Table 2: Number of sentences in monolingual data sets.

- Deduplicate training data.
- Remove all instances of evaluation data from the training data.
- Filter sentences that are longer than 250 tokens or length ratio upper than 3.

For monolingual data, we still employ those rules except the length ratio filter. See Table 1 for the statistics of bitext data sets and Table 2 for monolingual data sets.

3 System Overview

3.1 Base Systems

Our systems are based on the Transformer architecture (Vaswani et al., 2017) as implemented in TenTrans ⁴, a unified end-to-end multilingual and multi-task training platform. We first train a model following the Transformer *base* setup to jointly training all language pairs as our base system. Then, inspired by Wang et al. (2019), we experiment with raising network capacity by increasing encoder/decoder layers and feed-forward networks. We found that using a deeper encode layer (24) and a larger feed-forward network size (4096) can provide reasonable performance improvements while maintaining manageable network size and not increasing inference time.

Because of the recent popularity of using largescale pre-training models to fine-tune specific languages and tasks (Fan et al., 2020; Liu et al., 2020), we use the pre-trained model FLORES-101 released by the organizer to fine-tune on the bitext data. This system has further improved our translation performance in all thirty translation directions. Note that to fine-tune FLORES-101 we train our models using FAIRSEQ (Ott et al., 2019).

3.2 Forward-Translation and Back-Translation

Back-translation is an effective and common way to boost translation quality by using monolingual data to produce pseudo training parallel data. As opposed to back-translation, forward translation use source-side monolingual data to translate into the target language, and can be quite effective in some cases (Bogoychev and Sennrich, 2019). Wu et al. (2019) has shown that when monolingual data from source and target languages are used together to produce pseudo data, the translation quality is best, and the experimental performance will be improved with the increase of data.

In this work, considering the excellent performance of forward-translation and back-translation, we use both methods together. For translation directions with more than 5 million bitext data, such as En \leftrightarrow Id, En \leftrightarrow Ms, En \leftrightarrow Tl, we separately train an individual model for each direction and use it for the pseudo-corpus generation. For other translation directions with less than 5 million bitext data, we use the baseline system of all language pairs jointly training for translating pseudo sentences. Due to a large amount of English monolingual data, English monolingual sentence was randomly divided into 13.36M, 25M, 25M, 25M, and 25M for $En \rightarrow Id$, $En \rightarrow Jv$, $En \rightarrow Ms$, $En \rightarrow Ta$, and $En \rightarrow Tl$ translation respectively. All monolingual data of Id, Jv, Ms, Ta, and Tl are used in translation to all other directions.

3.3 In-domain Data Selection

The training data is provided by the publicly available Opus repository, which contains data of various quality from a variety of domains, while the hidden test set is the same domain as the provided dev and devtest datasets. After fine-tuning on a

⁴https://github.com/TenTrans/TenTrans

	$En \rightarrow Id$	$En {\rightarrow} Jv$	$En{\rightarrow}Ms$	En→Ta	$En \rightarrow Tl$	$Id{\rightarrow}En$	$Id{\rightarrow}Jv$	$Id{\rightarrow}Ms$	Id→Ta	$Id{\rightarrow}Tl$
0.7	3.83M	7.62K	1.15M	0.21M	0.36M	12.32M	0.05M	2.11M	0.23M	0.43M
0.8	3.44M	6.93K	1.05M	0.19M	0.34M	12.12M	0.05M	2.08M	0.22M	0.42M
0.9	2.82M	5.99K	0.89M	0.16M	0.30M	11.82M	0.04M	2.02M	0.22M	0.41M
0.99	1.24M	3.14K	0.44M	0.09M	0.17M	10.73M	0.03M	1.84M	0.20M	0.37M
	Jv → En	Jv→Id	Jv → Ms	Jv→Ta	$Jv \rightarrow Tl$	$Ms {\rightarrow} En$	$Ms{\rightarrow}Id$	$Ms {\rightarrow} Jv$	Ms→Ta	$Ms{\rightarrow}Tl$
0.7	59.99K	41.90K	15.82K	9.98K	14.77K	3.65M	2.20M	24.77K	0.18M	0.33M
0.8	57.12K	40.74K	14.83K	9.78K	14.29K	3.59M	2.17M	23.32K	0.18M	0.33M
0.9	53.05K	39.07K	13.76K	9.47K	13.61K	3.49M	2.11M	21.24K	0.17M	0.32M
0.99	41.59K	34.07K	10.47K	8.37K	11.39K	3.14M	1.91M	15.36K	0.16M	0.29M
	Ta→En	Ta→Id	Ta→Jv	Ta→Ms	Ta→Tl	Tl→En	Tl→Id	$Tl \rightarrow Jv$	$Tl \rightarrow Ms$	Tl→Ta
0.7	0.72M	0.28M	17.60K	0.21M	0.20M	1.12M	0.43M	17.95K	0.31M	0.15M
0.8	0.71M	0.27M	16.83K	0.20M	0.19M	1.11M	0.42M	17.32K	0.30M	0.15M
0.9	0.69M	0.27M	15.73K	0.20M	0.19M	1.09M	0.41M	16.38K	0.30M	0.15M
0.99	0.63M	0.24M	12.49K	0.18K	0.16M	1.03M	0.38M	0.01M	0.28M	0.13M

Table 3: Data filtered at different thresholds for all language pairs.

mixture of authentic bitext and pseudo-data, we select domain-specific data from the bitext and continue to fine-tune to further improve translation quality.

Due to the scarcity of in-domain data, we utilize pre-trained language model multilingual BERT (Devlin et al., 2019) to train a domain classifier for extracting in-domain sentences from authentic bilingual sentences. To train the domain classifier, we consider all available dev data as positive data, and randomly sample bilingual data as negative samples. At the same domain test set, the domain classifier recognition accuracy is achieved at 93.97%. We select sentences predicted to be positive with a probability greater than threshold 0.7 to form an in-domain corpus.

3.4 Knowledge Distillation

Knowledge distillation (Hinton et al., 2015) is a way to train a smaller network of students to perform better by learning from a larger teacher model. On this basis, sequence-level knowledge distillation trains the student model on the new data generated by the teacher model to further improve the performance of the student (Kim and Rush, 2016).

A multilingual translation model that trains too many languages at the same time may degrade performance (Xie et al., 2021), especially involving 30 translation directions in this work. It makes it harder for the model to accommodate all language pairs. Based on this, we fine-tune the FLORES-101 model on five language pairs with En \rightarrow Ta, Id \rightarrow Ta, Jv \rightarrow Ta, Ms \rightarrow Ta, Tl \rightarrow Ta to produce an Any-to-Ta specific translation model (Tan et al., 2019). These five language pairs are chosen because they do not perform very well and have more room for improvement. We used this model as the teacher model to translate the training data of the five language pairs. The new data was then combined with data of other language pairs to train the student model.

3.5 Gradual Fine-tuning

Fine-tuning can improve the machine translation model by adapting the initial model trained on abundant but less domain-specific examples to the data in the target domain. This domain adaptation is usually accomplished with a phase of finetuning. While Xu et al. (2021) prove that gradual fine-tuning over a multi-stage process can yield substantial further gains. Intuitively, the model is iteratively trained to convergence on data whose distribution progressively approaches that of the in-domain data, similar to the curriculum learning strategy (Bengio et al., 2009; Kocmi and Bojar, 2017).

In this work, we use gradual fine-tuning combined with in-domain data selection. After training the domain classifier, authentic bilingual sentences with positive predictions and probabilities greater than the thresholds of 0.7, 0.8, 0.9, and 0.99 are selected to form in-domain corpora with different similarity degrees. Data statistics with different thresholds are shown in the Table 3. The higher the threshold, the more the selected data fits into the domain of the dev set and test set. We started with a gradual fine-tuning on the domain-specific data selected at the 0.7 thresholds, followed by the 0.8 thresholds, and so on.

System	Average BLEU
Transformer	22.25
+ F&B	25.05
+ deep (24)	25.43
FLORES-101	15.38
+ Fine-tuning	24.23
+ F&B	26.50
+ Data Selection	27.24
+ Gradual FT	28.03
+ KD	28.15
+ Recover 12	28.32
Averaging	28.94

Table 4: Average BLEU (%) scores of all systems. The '+' means the approach added to the system over the previous line.

To further improve performance, we selected 12 language pairs that are significantly better than the baseline system. We consider them BLEU-sensitive and performance-friendly language pairs, which include $En \rightarrow Ta$, $Id \rightarrow Ta$, $Jv \rightarrow En$, $Jv \rightarrow Ta$, $Jv \rightarrow Tl$, $Ms \rightarrow Ta$, $Ta \rightarrow En$, $Ta \rightarrow Id$, $Ta \rightarrow Jv$, $Ta \rightarrow Ms$, $Ta \rightarrow Tl$ and $Tl \rightarrow Ta$. After the gradual fine-tuning, we recover all the authentic bilingual sentences of these 12 language pairs, while the training data when the threshold is 0.99. We continue to fine-tune the multilingual translation model. We find that the results still improve on these 12 language pairs and the performance of other language pairs is almost unchanged.

3.6 Model Averaging

Model averaging is typically used between 5 or 10 adjacent checkpoints on the same system. It is almost impossible to average different systems because neurons or parameters at the same location in different systems may be responsible for completely different knowledge or responsibilities. Our systems kept the random seeds consistent, and the training data did not differ too much, so we tried a variety of model averaging methods to see whether the performance was improved. We finally chose average multiple checkpoints in a single system, and then averaged on different systems. In this way, the translation result can be further improved.

4 Experiments

4.1 Experiment Settings

Except for the FLORES-101 fine-tuning experiments training on 48 NVIDIA P40 GPUs, the rest of our experiments are carried out with 16 NVIDIA P40 GPUs. Our model apply Adam (Kingma and Ba, 2015) as optimizer with $\beta_1 = 0.9, \beta_2 = 0.98$, and $\epsilon = 10^{-9}$. We set the label smoothing to 0.2 and the dropout rate to 0.3. The initial learning rate is set to 5e-4 varied under a warm-up strategy with 4000 steps. For training, the batch size is 4096 tokens per GPU. For fine-tuning FLORES-101, we apply a temperature sampling strategy with sampling temperature T = 1.5 (Arivazhagan et al., 2019). During inference, we decode with beam search and set beam size to 4 for all language pairs. The translation results we reported is detokenized and then the quality is evaluated using the 4-gram case-sensitive BLEU (Papineni et al., 2002) with the SacreBLEU tool (Post, 2018).⁵

4.2 Main Results

Results for all of our systems are shown in Table 4. For convenience, we only report the average BLEU for 30 language pairs. The detailed BLEU scores for each language pair of systems implemented by TenTrans tool are shown in Table 5, and the relevant systems for fine-tuning FLORES-101 are shown in Table 6.

As shown in Table 4, we found that the baseline system with fine-tuning FLORES-101 performed better than the baseline system with no pre-training model (24.23 vs. 22.25). Forward-translation and back-translation (F&B) greatly improved the translation performance in both TenTrans (25.05 vs. 22.25) and FLORES-101 (26.50 vs. 24.23) frameworks. The results of individual models for forward-translation and back-translation and back-translation and back-translation and back-translation are shown in Table 7. Deep Transformer with 24 encoder layers further improves translation results, but still not as high as fine-tuning FLORES-101 systems. Given the excellent performance of the pre-trained model, our subsequent series of approaches are based on fine-tuning FLORES-101.

In-domain data selection is restricted to indomain data size (threshold 0.7), but we also obtain a solid improvement of 0.74 BLEU on average. Gradual fine-tuning (Gradual FT) is also effective, which enables the model to potentially better fit

⁵BLEU+case.mixed+numrefs.1+smooth.exp+tok.spm +version.1.5.0

System	$En \rightarrow Id$	$En \rightarrow Jv$	$En \rightarrow Ms$	En→Ta	$En \rightarrow Tl$	Id→En	$Id{\rightarrow}Jv$	$Id{\rightarrow}Ms$	Id→Ta	$Id \rightarrow Tl$
Transformer	43.37	18.12	40.44	15.99	29.21	36.39	17.93	33.18	13.98	24.47
+ F&B	42.93	23.54	41.73	22.18	30.65	36.58	21.65	33.70	17.93	24.98
+ deep (24)	44.13	23.61	42.59	22.76	31.38	37.95	21.50	34.34	18.13	25.69
	$Jv{\rightarrow}En$	Jv→Id	$Jv { ightarrow} Ms$	Jv→Ta	$Jv {\rightarrow} Tl$	$Ms {\rightarrow} En$	$Ms {\rightarrow} Id$	$Ms {\rightarrow} Jv$	Ms→Ta	$Ms{\rightarrow}Tl$
Transformer	23.38	24.33	21.60	9.27	16.06	35.21	34.29	16.61	13.72	23.46
+ F&B	26.20	26.04	24.55	14.43	19.39	37.08	34.50	21.09	18.15	24.10
+ deep (24)	26.25	25.72	24.08	14.76	19.95	38.02	34.73	21.13	18.65	24.95
	Ta→En	Ta→Id	Ta→Jv	Ta→Ms	Ta→Tl	Tl→En	Tl→Id	$Tl {\rightarrow} Jv$	$Tl{\rightarrow}Ms$	Tl→Ta
Transformer	14.61	13.12	5.83	13.21	14.55	34.28	28.05	13.53	26.35	12.93
+ F&B	21.04	15.76	10.62	16.39	17.55	36.52	27.87	18.48	27.75	18.17
+ deep (24)	21.19	17.02	9.70	17.04	18.20	36.52	28.85	17.89	27.93	18.35

Table 5: Results of the systems implemented by TenTrans on the devtest set.

~~~										
System	En→ld	En→Jv	En→Ms	En→Ta	En→Tl	Id→En	ld→Jv	ld→Ms	ld→Ta	ld→Tl
FLORES-101	37.28	15.35	33.40	3.38	6.38	33.75	16.55	29.45	1.36	8.07
+ Fine-tuning	44.79	17.60	41.39	20.14	30.71	38.86	18.26	34.56	16.64	26.50
+ F&B	44.06	23.43	43.01	22.67	32.49	40.29	21.53	35.31	18.86	26.62
+ Data Selection	44.73	22.02	43.46	24.32	33.14	41.05	20.81	35.66	18.60	28.28
+ Gradual FT	45.30	23.13	43.74	25.77	33.43	41.53	22.04	35.88	19.96	28.68
+ KD	45.49	24.40	43.97	28.21	33.90	40.99	22.07	35.95	23.52	28.99
+ Recover 12	45.52	24.46	44.02	28.89	34.27	40.94	22.08	35.93	23.67	28.82
Averaging	46.15	24.16	44.47	27.24	34.19	42.00	22.64	36.55	22.87	29.45
	$Jv{\rightarrow}En$	$Jv{\rightarrow}Id$	$Jv{\rightarrow}Ms$	Jv→Ta	$Jv{\rightarrow}Tl$	$Ms{\rightarrow}En$	$Ms{\rightarrow}Id$	$Ms{\rightarrow}Jv$	$Ms{\rightarrow}Ta$	$Ms{\rightarrow}Tl$
FLORES-101	20.90	22.77	18.98	3.73	12.07	34.24	32.18	15.18	2.22	9.64
+ Fine-tuning	25.38	25.95	23.08	9.78	17.95	38.46	35.87	17.46	16.61	25.02
+ F&B	27.23	26.69	24.98	15.46	20.93	39.38	36.20	21.73	19.34	27.17
+ Data Selection	30.67	28.92	26.57	15.10	21.68	41.60	36.37	21.21	20.33	27.25
+ Gradual FT	31.63	29.69	27.16	16.82	22.40	41.68	36.72	21.63	21.41	27.40
+ KD	30.79	29.44	26.84	17.28	22.93	41.55	36.73	20.80	23.30	28.50
+ Recover 12	30.76	29.55	26.84	17.30	23.05	41.78	36.76	21.57	23.63	28.08
Averaging	31.82	30.22	27.93	18.47	23.74	42.54	37.27	22.15	23.47	28.34
	Ta→En	Ta→Id	Ta→Jv	Ta→Ms	Ta→Tl	Tl→En	Tl→Id	Tl→Jv	Tl→Ms	Tl→Ta
FLORES-101	8.41	5.36	3.11	4.89	3.30	26.10	20.43	11.52	18.00	3.46
+ Fine-tuning	19.91	16.44	7.76	15.96	17.04	36.61	30.85	12.68	28.71	15.80
+ F&B	23.68	17.75	10.83	17.61	18.91	39.80	30.84	19.48	29.71	18.97
+ Data Selection	25.11	19.14	10.00	18.91	20.20	41.42	32.71	16.27	30.97	20.61
+ Gradual FT	25.29	20.07	12.86	19.68	20.85	41.42	33.55	18.12	31.33	21.75
+ KD	25.04	19.03	11.27	18.91	21.20	40.95	32.83	17.06	31.19	21.33
+ Recover 12	25.04	19.59	11.08	19.07	20.92	41.17	32.96	18.58	31.30	21.93
Averaging	26.19	21.21	12.75	20.25	21.87	42.06	34.17	19.00	31.93	23.04

Table 6: Results of the systems about FLORES-101 on the devtest set. The '+' means the approach added to the system over the previous line. "Average" stands for averaging the model of the three best checkpoints of the "+ Gradual FT" system and the "+ Recover 12" system respectively.

System	$En \rightarrow Id$	$Id{\rightarrow}En$	$En \rightarrow Ms$	$Ms \rightarrow En$	$En \rightarrow Tl$	Tl→En
Individual	46.08	40.72	42.95	38.76	31.84	37.94

Table 7: Results of the individual models for forward-translation and back-translation.

System	AVE	En→Ta	Id→Ta	Jv→Ta	Ms→Ta	Tl→Ta
FLORES-101 + Fine-tuning	15.79	20.14	16.64	9.78	16.61	15.80
Any-to-Ta	22.83	28.75	23.40	17.02	23.27	21.70

Table 8: Results of FLORES-101 fine-tuning on Any-to-TA language pairs only.

Dataset	sp_bleu
flores101-small2-test	28.89
flores101-small2-dev	29.25
flores101-small2-devtest	28.94

Table 9: Flores MT Evaluation (Small task 2) results: https://dynabench.org/models/460

the distribution of the target domain. The knowledge distillation, however, has not brought much improvement (28.15 vs. 28.03). The translation performance of the teacher model is shown in Table 8. We guess that it may be because the translation quality of the teacher model is not excellent enough, which leads to the improvement of the student model is not satisfactory. We then recovered bilingual sentences for 12 BLEU-sensitive language pairs. As shown in Table 6, the performance of these 12 language pairs improved significantly, while the results of the other language pairs barely changed, so our average BLEU improved further. For model averaging, we tried different combinations and finally found that averaging the three best checkpoints in "+ Gradual FT" and "+ Recover 12" will produce the best performance (28.94).

# 4.3 Submitted Results

As shown in Table 9, we ultimately chose the bestperforming model on devtest to submit to Dynabench ⁶ and achieve 28.89 in the hidden test set.

# 5 Conclusion

This paper introduced our TenTrans submissions on WMT21 large-scale multilingual machine translation small task 2. Our main exploration is using more diversified architectures and fine-tuning strategy, utilizing forward-translation and back translation and approaches including in-domain data selection, knowledge distillation, and gradual finetuning. We experimented with these methods and continuously improve our system performance. On the whole, all of our systems performed competitively and ranked 3rd on the leaderboard.

#### References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George F. Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR*, abs/1907.05019.

- Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009, pages 41– 48.
- Nikolay Bogoychev and Rico Sennrich. 2019. Domain, translationese and noise in synthetic data for neural machine translation. *CoRR*, abs/1911.03362.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 4171–4186.
- Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation. *CoRR*, abs/2010.11125.
- Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2021. The flores-101 evaluation benchmark for low-resource and multilingual machine translation.
- Shuhao Gu, Yang Feng, and Wanying Xie. 2021. Pruning-then-expanding model for domain adaptation of neural machine translation. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021, pages 3942–3952. Association for Computational Linguistics.
- Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.
- Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *TACL*, 5:339–351.
- Yoon Kim and Alexander M. Rush. 2016. Sequencelevel knowledge distillation. In *Proceedings of the* 2016 Conference on Empirical Methods in Natural

⁶https://dynabench.org/flores

Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016, pages 1317–1327. The 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.
- Tom Kocmi and Ondrej Bojar. 2017. Curriculum learning and minibatch bucketing in neural machine translation. In Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017, Varna, Bulgaria, September 2 - 8, 2017, pages 379–386.
- Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Trans. Assoc. Comput. Linguistics*, 8:726–742.
- Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA, pages 311–318.
- Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.
- Xu Tan, Jiale Chen, Di He, Yingce Xia, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with language clustering. 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 2019, Hong Kong, China, November 3-7, 2019, pages 963–973.
- 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, 4-9 December 2017, Long Beach, CA, USA, pages 5998–6008.
- Qiang Wang, Bei Li, Jiqiang Liu, Bojian Jiang, Zheyang Zhang, Yinqiao Li, Ye Lin, Tong Xiao, and Jingbo Zhu. 2018. The niutrans machine translation system for WMT18. In Proceedings of the Third Conference on Machine Translation: Shared Task Papers, WMT 2018, Belgium, Brussels, October 31 -

*November 1, 2018*, pages 528–534. 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. In Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pages 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 2019, Hong Kong, China, November 3-7, 2019, pages 4205– 4215. Association for Computational Linguistics.
- Wanying Xie, Yang Feng, Shuhao Gu, and Dong Yu. 2021. Importance-based neuron allocation for multilingual neural machine translation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021, pages 5725– 5737. Association for Computational Linguistics.
- Haoran Xu, Seth Ebner, Mahsa Yarmohammadi, Aaron Steven White, Benjamin Van Durme, and Kenton W. Murray. 2021. Gradual fine-tuning for low-resource domain adaptation. *CoRR*, abs/2103.02205.