Multilingual Machine Translation Systems from Microsoft for WMT21 Shared Task

Jian Yang, Shuming Ma, Haoyang Huang, Dongdong Zhang, Li Dong, Shaohan Huang, Alexandre Muzio, Saksham Singhal, Hany Hassan Awadalla, Xia Song, Furu Wei
Microsoft Corporation
{t-jianya,shumma,haohua,dozhang,lidong1,shaohanh}@microsoft.com
{alferre,saksingh,hanyh,xiaso,fuwei}@microsoft.com

Abstract

This report describes Microsoft's machine translation systems for the WMT21 shared task on large-scale multilingual machine translation. We participated in all three evaluation tracks including Large Track and two Small Tracks where the former one is unconstrained and the latter two are fully constrained. Our model submissions to the shared task were initialized with DeltaLM[1], a generic pre-trained multilingual encoder-decoder model, and fine-tuned correspondingly with the vast collected parallel data and allowed data sources according to track settings, together with applying progressive learning and iterative back-translation approaches to further improve the performance. Our final submissions ranked first on three tracks in terms of the automatic evaluation metric.

1 Introduction

Recently, multilingual neural machine translation has attracted lots of attention because it enables one model to translate between multiple languages (Dong et al., 2015; Johnson et al., 2017; Arivazhagan et al., 2019; Dabre et al., 2020; Philip et al., 2020; Lin et al., 2021). To improve the performance of the multilingual translation models, there are various approaches on the training methods (Aharoni et al., 2019; Wang et al., 2020a,c), the model structures (Wang et al., 2018; Gong et al., 2021; Zhang et al., 2021a), and the data augmentation (Tan et al., 2019; Pan et al., 2021). M2M (Fan et al., 2020) leverages the large-scale data mined from the web data and explore the strategies to scale the model size and train the model effectively. Meanwhile, the multilingual pre-trained language models have proven beneficial for the multilingual machine translation models. mBART (Liu et al., 2020) pre-trains a multilingual model with the multilingual denoising objective to improve the multilingual machine translation.

[1] https://aka.ms/deltalm

In this work, we explore the effects of different advanced approaches for multilingual machine translation models, especially on the large-scale dataset. We first explore the way to leverage the pre-trained language models that have been trained with large-scale monolingual data. We use the public available DeltaLM-Large checkpoint to initialize the model. DeltaLM (Ma et al., 2021) is a multilingual pre-trained encoder-decoder model, which has been proven useful for multilingual machine translation.

We further explore the training methods and the data augmentation to improve the model. For efficient training, we apply progressive learning (Li et al., 2020; Zhou et al., 2021; Zhang et al., 2021b) to our model that continue-trains a shallow model into a deep model. Specifically, we first train a model with 24 encoder layers, and then continue-train it by adding 12 layers on the top of the encoder. As for the data augmentation, we implement iterative back-translation (Hoang et al., 2018; Dou et al., 2020) that back-translates the data for multiple rounds. Due to the limits of time and GPU memories of the shared task, we do not explore other approaches like mixture-of-experts (MOE) and model ensemble.

We participated in all three tracks including Large Track, Small Track #1, and Small Track #2. Our final submissions are fine-tuned from DeltaLM with the allowed data sources according to the track settings, followed by progressive learning and iterative back-translation. The submissions on three tracks all rank first in terms of the automatic evaluation metric.

2 Data

Large Track The monolingual and bilingual data are collected from multiple sources, including CCAligned (El-Kishky et al., 2020), CCMatrix (Schwenk et al., 2021), OPUS-100 (Zhang et al., 2020), JW300 (Agic and Vulic, 2019), Tatoeba

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 446–455
November 10–11, 2021.

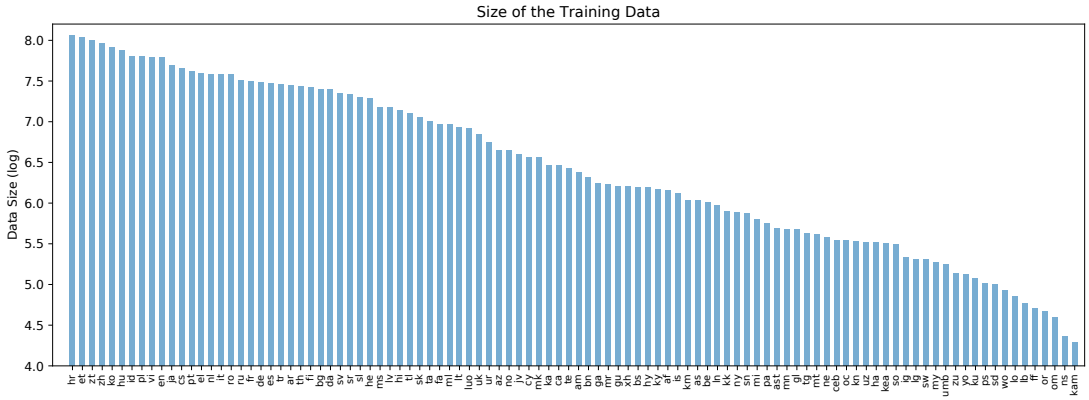

Figure 1: Dataset statistics of the bilingual data of the 102 languages. For better visualization, we apply the logarithmic function (base 10 logarithm) to the size of the training data. Each column denotes the data size of a language that was paired with the remaining 101 languages. For example, the first column denotes the number of bilingual sentence pairs that contain sentences from language hr.

(Tiedemann, 2012), WMT2021 news track[2], multilingual track data[3], and our in-house data. To improve the translation quality of non-English languages, we construct dual-pseudo parallel data (or dual-pseudo data briefly) in which the source and target sides per each sentence pair are translated from the same monolingual English sentence respectively. The Wikipedia English monolingual sentences are translated to other 70 languages by leveraging various machine translation models including in-house MT models, M2M (Fan et al., 2020), the multilingual model of small tracks, and our intermediate multilingual MT model.

Finally, the training data was split into three parts: the bitext data (1.7B parallel sentences from 394 language pairs), the back-translation (1.4B parallel sentences from 45 language pairs), and the dual-pseudo data (8.7B parallel sentences of 70 languages from 4830 language pairs). Figure 1 lists the statistics of the bilingual training data size of 102 languages.

Small Track #1 We use the constrained monolingual and bilingual data of 6 languages (Croatian, Hungarian, Estonian, Serbian, Macedonian, and English) provided by the shared task. According to the statistics, the bitext data contains 273M sentence pairs of all translation directions. Inspired by the previous work, we leverage the multilingual iterative back-translation method with one single multilingual model to generate parallel pseudo data. For En→X and X→En directions, we generate the back-translation data of 390M sentence pairs. As for X→Y directions, we generate the dual-pseudo data of 1.18B sentence pairs, where X and Y stand for any two non-English languages.

Small Track #2 The monolingual and bilingual corpora of 6 languages (Javanese, Indonesian, Malay, Tagalog, Tamil, and English) provided by the shared task are used for the multilingual model training, containing 98M bilingual data, 256M generated back-translation data, and 860M generated dual-pseudo data.

3 Large-scale Data Augmentation

In this section, we introduce details about how to perform the iterative back-translation method (Hoang et al., 2018) to augment data. We use different models for data augmentation according to different tracks. For the small tracks, the multilingual models were trained over the constrained data sets to generate data. For the large track, we leverage the M2M model (Fan et al., 2020), the intermediate multilingual MT models, and in-house MT models to generate different language pairs' data respectively, so as to play their respective advantages to enhance the data generation quality.

In practice, both the monolingual and bilingual corpora are effectively utilized in three ways: 1) For the back-translation data of X→En and En→X directions, we used the best model to generated X data accordingly by back-translating monolingual English Wikipedia data; 2) For the dual-pseudo data of X→Y directions, they are generated by back-translating the same English text to X and Y

[2] http://statmt.org/wmt21/translation-task.html

[3] http://data.statmt.org/wmt21/multilingual-task/

respectively. Alternatively, when the monolingual data of either X or Y is enough, we also directly perform back-translation between X and Y to obtain pseudo parallel data; 3) We try to augment existing bilingual corpora with the third language. Given the bilingual corpus (X_1, Y_1), we generate pseudo parallel corpus of (X_1, Y_2) and (X_2, Y_1) by back-translating X_1 to X_2 and Y_1 to Y_2, where X_2 and Y_2 are non-English languages.

4 Preprocessing

Filtering To enhance the model performance, we remove the noisy sentence pairs with the incorrect language identification or character encoding. More specifically, we remove the sentences longer than 1024 words and truncate the sentence to 512 tokens. We also construct three corpora after tokenization with different length ratio limitations, i.e. $\{1.5, 2.0, 2.5, 3.0\}$, between the source and the target sentence. Our multilingual model is first trained on the entire noisy data set and then continually tuned on cleaner data with descending length ratio, where the number of training directions is also gradually reduced by removing noisy language pairs. Therefore, we can progressively fine-tune the multilingual model in an efficient way (noisy corpora $\rightarrow$ clean corpora $\wedge$ numerous directions $\rightarrow$ selected directions $\wedge$ shallow encoder layers $\rightarrow$ deep encoder layers). Besides, to clean the back-translation corpora, we remove the sentences containing unknown tokens (`[UNK]`). Regarding the language Sr (Serbian), those sentences comprised of Latin characters in training data were also discarded since we found that the validation sets use Cyrillic script for this language instead.

Tokenization After data filtering, we use the SentencePiece (Kudo and Richardson, 2018) to tokenize all raw training, validation, and test data sets, where the SentencePiece model is consistent with the one used for DeltaLM (Ma et al., 2021). We shuffled the whole training dataset before launching the training of multilingual models. The input sentence is prefixed with the language tag to indicate the translation direction.

5 Model and Training

5.1 DeltaLM

We adopt the `DeltaLM_large` architecture as the backbone model for all our experiments, which has 24 Transformer encoder layers and 12 interleaved decoder layers with an embedding size of 1024, a dropout of 0.1, the feed-forward network size of 4096, and 16 attention heads. We directly initialize our model with the public available DeltaLM large checkpoint[4].

5.2 Multilingual Fine-tuning

The training data was split into the bitext corpora $D_b = \{D_b^1, \ldots, D_b^u\}$, the back-translation corpora $D_{bt} = \{D_{bt}^1, \ldots, D_{bt}^v\}$, and the dual-pseudo corpora $D_{dp} = \{D_{dp}^1, \ldots, D_{dp}^w\}$, where u, v, w represent the number of the corpora of different translation directions. The multilingual model with parameters Θ is jointly trained over the corpora to optimize the combined objective as below:

$$\begin{aligned}\mathcal{L}_{MT} = &-\lambda_1 \sum_{i=1}^{u} \mathbb{E}_{x,y \in D_b^i} \left[-\log P(y|x;\Theta)\right] \\ &-\lambda_2 \sum_{i=1}^{v} \mathbb{E}_{x,y \in D_{bt}^i} \left[-\log P(y|x;\Theta)\right] \\ &-\lambda_3 \sum_{i=1}^{w} \mathbb{E}_{x,y \in D_{dp}^i} \left[-\log P(y|x;\Theta)\right]\end{aligned} \tag{1}$$

where x, y denote the sentence pair in the bilingual corpus. $\mathcal{L}_{MT}$ is the combined translation objective of the multilingual model. $\lambda_1, \lambda_2, \lambda_3$ ($\lambda_1 + \lambda_2 + \lambda_3 = 1.0$) are used to balance the training objectives of the bitext corpora, the back-translation corpora, and the dual-pseudo corpora. In this work, we first set $\lambda_1 = 0.33, \lambda_2 = 0.33, \lambda_3 = 0.33$ and then reset $\lambda_1 = 0.6, \lambda_2 = 0.2, \lambda_3 = 0.2$ to focus more on the bitext corpora avoiding the noise introduced by pseudo data.

We follow the dynamic temperature-based data-sampling strategy (Fan et al., 2020; Wang et al., 2020b) to ease the underrepresentation of low-resource languages. The probability of picking a language is proportional to its number of sentences D_l, i.e., $p_l = \frac{D_l}{\sum_i D_i}$. We set the temperature $T = 5$ to rescale and control the distribution $p_l^{\frac{1}{T}}$. It can balance the samples between the high-resource languages and the low-resource languages.

5.3 Progressive Learning

We implement the progressive training method to train the model from shallow to deep (Li et al., 2020). The training process can be divided into two stages. In the first stage, the pre-trained DeltaLM model with 24 encoder layers and 12 decoder layers is directly adopted to initialize the multilingual

[4] https://aka.ms/deltalm

translation model with the same architecture. The shallow translation model with 24 encoder layers and 12 decoder layers is fine-tuned on all available multilingual corpora. In the second stage, we increase the depth of the encoder from 24 layers to 36 layers, where the bottom 24 layers of the encoder are initialized with the shallow model's encoder and the top 12 layers are randomly initialized. Then we perform continue training. The deeper encoders enlarge the model's capacity, but no much extra decoding cost is introduced.

5.4 Training Details

We train multilingual models with the Adam optimizer (Kingma and Ba, 2014) ($\beta_1 = 0.9$, $\beta_2 = 0.98$). The learning rate is set as 1e-4 with a warm-up step of $4,000$. The models are trained with the label smoothing with a ratio of 0.1. All experiments are conducted on 64 NVIDIA V100 or 32 A100 GPUs. The batch size is 1536 or 2048 tokens per GPU and the model is updated every 32 (for 64 V100 GPUs) or 64 (for 32 A100 GPUs) steps to simulate the large batch size.

5.5 Decoding

To enhance the performance of low-resource language pairs for X→Y directions, we adopt the pivot-based translation method (Kim et al., 2019). We use English as the pivot language and employ a unified model to perform the pivot-based translation. When the performance of X→Y directions on the validation set is better than the pivot-based translation X→En ∧ En→Y, we directly translate the language X into Y. Otherwise, we translate them in the pivot way. This approach is used for the submission to Large Track and Small Track #2. As for Small Track #1, we do not use the pivot-based translation.

6 Evaluation Results

Following the previous work (Goyal et al., 2021), we use the dev and the devtest of the FLORES-101 benchmark as our validation set and test set respectively. During the inference, the beam search strategy is performed with a beam size of 4 for the target sentence generation. We set the length penalty as 1.0 by default. The last N checkpoints ($N = \{1, 5, 10, 15, 20\}$) are averaged for evaluation and we select the best checkpoint based on the performance on the validation set. We report the SentencePiece-based BLEU using spBLEU[5].

6.1 Large Track

Given the unbalanced large-scale multilingual corpora, we use the hybrid strategy for the translation for Large Track. The pivot-based translation is more suitable for the low-resource translation direction between non-English languages since the corpora of X→Y are commonly scarce. Our model with the 36 encoder layers significantly outperforms the shallow counterpart with the 24 encoder layers, which indicates that using a deep encoder and shallow decoder is a good trade-off between the translation quality and the decoding speed. Table 1 shows that our model with the hybrid strategy gets the best performance with less inference cost than the pivot-based translation which costs double inference time compared to the direct translation. We build a massively multilingual neural machine model, which translates between any pair of 102 languages. In Figure 2 and Figure 3, we reported the spBLEU scores of the shallow model with 24 encoder layers and 6 decoder layers and our best multilingual model with 36 encoder layers and 12 decoder layers in all translation directions, where the languages are ordered alphabetically by the language code. Nearly 30% translation directions adopt the pivot-based translation, where the zero-resource and low-resource translation directions lack of supervised training data tend to be chosen for pivot-based translation.

6.2 Small Track #1

In Table 2, we compare the performance of M2M with our method in different architectures on any to English (X→En), English to any (En→X), and the translation between any non-English languages (X→Y). Both En→X and X→En contain 5 directions, while X→Y have 20 directions. Given the enormous bilingual and back-translation data of Small Track #1, we are able to perform the direct translation for all X→Y directions. Furthermore, we explore the deep encoder (36 encoder layers) and shallow decoder (12 decoder layers) considering the limited inference time. From Table 2, we can observe that the largest model (36 encoder layers and 12 decoder layers) has a significant improvement of +9.41 BLEU points over the strong M2M baseline.

[5] https://github.com/ngoyal2707/sacrebleu.git

	#Languages	#Params	#Layers	$\text{Avg}_{X\rightarrow En}$	$\text{Avg}_{En\rightarrow Y}$	$\text{Avg}_{X\rightarrow Y}$	Avg_{all}
M2M (Fan et al., 2020)	102	175M	6/6	15.43	12.02	5.85	6.00
	102	615M	12/12	20.03	16.21	7.66	7.86
DeltaLM + Zcode (Direct)	102	711M	24/6	30.39	23.52	11.21	11.52
	102	862M	24/12	33.09	27.21	13.56	13.89
	102	1013M	36/12	33.35	27.39	14.34	14.65
DeltaLM + Zcode (Pivot)	102	711M	24/6	31.32	24.04	14.74	14.99
	102	862M	24/12	33.09	27.21	17.20	17.45
	102	1013M	36/12	33.35	27.39	17.36	17.62
DeltaLM + Zcode (Hybrid)	102	711M	24/6	31.32	24.04	14.76	15.01
	102	862M	24/12	33.09	27.21	17.27	17.52
	102	1013M	36/12	**33.35**	**27.39**	**17.44**	**17.70**

Table 1: Evaluation results of Large Track for M2M and our method of 102 languages on the devtest of the FLORES-101 benchmark. $\text{Avg}_{X\rightarrow En}$ denotes the average score of directions between other languages to English. $\text{Avg}_{X\rightarrow En}$ denotes the average score of directions between English and other languages. $\text{Avg}_{X\rightarrow Y}$ denotes the average score of directions between non-English languages to other non-English languages. Avg_{all} denotes the average result of all translation directions.

	#Languages	#Params	#Layers	$\text{Avg}_{X\rightarrow En}$	$\text{Avg}_{En\rightarrow Y}$	$\text{Avg}_{X\rightarrow Y}$	Avg_{all}
M2M (Fan et al., 2020)	102	175M	6/6	24.60	20.83	20.80	21.44
	102	615M	12/12	31.58	29.62	26.66	27.98
DeltaLM + Zcode (Direct)	6	862M	24/12	43.78	41.02	34.38	37.06
	6	1013M	36/12	**44.34**	**41.32**	**34.68**	**37.39**

Table 2: Evaluation results of Small Track #1 for M2M and our method of 6 languages (Croatian, Hungarian, Estonian, Serbian, Macedonian, English) on the devtest of the FLORES-101 benchmark. **DeltaLM + Zcode (Direct)** denotes the strategy that we choose the direct translation for all translation directions, where the target language symbol is prefixed to the input sentence to indicate the translation direction. Our mutilingual translation model is only trained on the constrained corpora of 6 languages provided by the shared task.

6.3 Small Track #2

Compared to Small Track #1 (273M bilingual pairs), Small Track #2 contains smaller while more unbalanced training data (93M bilingual pairs). Therefore, we consider the hybrid strategy for the X→Y translation directions. We separately calculate the BLEU scores of the direct and the pivot-based translation on the validation set. For those directions satisfying $\text{BLEU}_{direct}(X, Y) \geq \text{BLEU}_{pivot}(X, Y)$, we employ the direct translation. Otherwise, we use the pivot-based translation direction by first translating the source language to English and then to the target language. According to Table 3, **DeltaLM + Zcode (Hybrid)** outperforms both the direct and pivot-based translation by about +0.5 BLEU points. It confirms that the hybrid strategy is essential since the training data of the X→En and En→Y is easy to obtain while the X→Y is hard to obtain. The deep model with the 36 encoder layers and 12 decoder layers has comparable performance with the shallow model with the 24 encoder layers and 12 decoder layers, which may be caused by the overfitting problem on the low-resource directions.

6.4 Discussion on Progressive Learning

Given the pre-trained model and large-scale parallel data, we adopt progressive learning as an alternative to fine-tune the multilingual model on the multilingual translation task. Our multilingual model is first trained on the large-scale noisy data and then continues to be tuned on the clean data (Noisy Data → Clean Data), where the model is denoted as ③. Since the training data of K languages in the dual-pseudo parallel data is generated by the same English monolingual data, we are able to adopt all possible $K \times (K-1)$ training directions on the clean data. The performance of many translation directions is improved by the additional dual-pseudo data while the performance of other directions has been degraded compared to the initial model ④, due to the poor quality of some languages in the dual-pseudo data. Therefore, only the part of $K \times (K-1)$ training directions is selected to continue training the multilingual model (Numerous Directions → Selected Directions), which we denoted as ②. To further enlarge the model's capacity, we extend the shallow model with 24 encoder layers to the deep model with 36

	#Languages	#Params	#Layers	$\text{Avg}_{X\rightarrow En}$	$\text{Avg}_{En\rightarrow Y}$	$\text{Avg}_{X\rightarrow Y}$	Avg_{all}
M2M (Fan et al., 2020)	102	175M	6/6	18.95	15.16	9.49	12.01
	102	615M	12/12	24.67	19.14	12.11	15.38
DeltaLM + Zcode (Direct)	6	862M	24/12	43.12	39.78	28.69	32.94
	6	1013M	36/12	43.56	39.04	28.60	32.83
DeltaLM + Zcode (Pivot)	6	862M	24/12	43.12	39.78	29.02	33.17
	6	1013M	36/12	43.56	39.04	28.63	32.85
DeltaLM + Zcode (Hybrid)	6	862M	24/12	**43.12**	**39.78**	**29.38**	**33.40**
	6	1013M	36/12	43.56	39.04	28.99	33.09

Table 3: Evaluation results of Small Track #2 for M2M and our method of 6 languages (Javanese, Indonesian, Malay, Tagalog, Tamil, English) on the devtest of the FLORES-101 benchmark. **DeltaLM + Zcode (Hybrid)** denotes the strategy that we choose the pivot-based translation (X→En, En→X) for low-resource X→Y directions and direct translation for high-resource X→Y directions.

ID	Large Track	Avg_{all}
①	DeltaLM + Zcode	14.65
②	① - Shallow Model → Deep Model	13.89
③	② - Numerous Directions → Selected Directions	13.09
④	③ - Noisy Data → Clean Data	12.24

Table 4: Ablation study of the large track on devtest. DeltaLM + Zcode is fine-tuned on the multilingual translation task via progressive learning.

Track	Submission Name	Avg_{all}
Large	DeltaLM + Zcode (Microsoft)	16.63
Small #1	DeltaLM + Zcode (Microsoft-Small)	37.59
Small #2	DeltaLM + Zcode (Microsoft-Small)	33.89

Table 5: Submission results based on the hidden test sets of our method on three tracks, including Large Track, Small Track #1, and Small Track #2.

encoder layers, where the top 12 encoder layers are initialized by random parameters (Shallow Model → Deep Model). Putting them all together, we obtain the final model ① **DeltaLM + Zcode**. Table 4 summarizes the results of the ablation study of these approaches. It shows that each approach has a significant contribution to the final model. This proves the effectiveness of progressive learning that can gradually improve performance in different aspects.

7 Submissions

Considering the trade-off between the decoding time and the performance, we submit the model (24 encoder layers and 12 decoder layers) with the hybrid strategy to both the Large Track and Small Track #2, while the deep model (36 encoder layers and 12 decoder layers) with the direct translation is submitted to Small Track #1. Table 5 summarizes the evaluation results of our model on the hidden test sets. According to the final results on the leaderboard, **DeltaLM + Zcode** ranks first across three tracks.

8 Conclusion

This paper describes Microsoft's submission to the large-scale multilingual machine translation of the WMT21 shared task. Our multilingual translation model achieves substantial improvement over the baseline systems by fine-tuning the pre-trained language model DeltaLM. We further enhance the model performance with the progressive learning and the iterative back-translation methods. As a result, our submitted systems get the top evaluation results on three tracks, including Large Track, Small Track #1, and Small Track #2.

References

Zeljko Agic and Ivan Vulic. 2019. JW300: A wide-coverage parallel corpus for low-resource languages. In *ACL 2019*, pages 3204–3210.

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *NAACL 2019*, pages 3874–3884.

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.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A survey of multilingual neural machine translation. *ACM Comput. Surv.*, 53(5):99:1–99:38.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *ACL 2015*, pages 1723–1732.

Zi-Yi Dou, Antonios Anastasopoulos, and Graham Neubig. 2020. Dynamic data selection and weighting for iterative back-translation. In *EMNLP 2020*, pages 5894–5904.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. Ccaligned: A massive collection of cross-lingual web-document pairs. In *EMNLP 2020*, pages 5960–5969.

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.

Hongyu Gong, Xian Li, and Dmitriy Genzel. 2021. Adaptive sparse transformer for multilingual translation. *CoRR*, abs/2104.07358.

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. *CoRR*, abs/2106.03193.

Cong Duy Vu Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *ACL 2018*, pages 18–24.

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.

Yunsu Kim, Petre Petrov, Pavel Petrushkov, Shahram Khadivi, and Hermann Ney. 2019. Pivot-based transfer learning for neural machine translation between non-english languages. In *EMNLP 2019*, pages 866–876.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *EMNLP 2018*, pages 66–71.

Bei Li, Ziyang Wang, Hui Liu, Yufan Jiang, Quan Du, Tong Xiao, Huizhen Wang, and Jingbo Zhu. 2020. Shallow-to-deep training for neural machine translation. In *EMNLP 2020*, pages 995–1005.

Zehui Lin, Liwei Wu, Mingxuan Wang, and Lei Li. 2021. Learning language specific sub-network for multilingual machine translation. In *ACL 2021*, pages 293–305.

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. *TACL*, 8:726–742.

Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, Alexandre Muzio, Saksham Singhal, Hany Hassan Awadalla, Xia Song, and Furu Wei. 2021. Deltalm: Encoder-decoder pre-training for language generation and translation by augmenting pretrained multilingual encoders. *CoRR*, abs/2106.13736.

Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural machine translation. In *ACL 2021*, pages 244–258.

Jerin Philip, Alexandre Berard, Matthias Gallé, and Laurent Besacier. 2020. Monolingual adapters for zero-shot neural machine translation. In *EMNLP 2020*, pages 4465–4470.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2021. Ccmatrix: Mining billions of high-quality parallel sentences on the web. In *ACL 2021*, pages 6490–6500.

Xu Tan, Yi Ren, Di He, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2019. Multilingual neural machine translation with knowledge distillation. In *ICLR 2019*.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *LREC 2012*, pages 2214–2218.

Xinyi Wang, Yulia Tsvetkov, and Graham Neubig. 2020a. Balancing training for multilingual neural machine translation. In *ACL 2020*, pages 8526–8537.

Yining Wang, Jiajun Zhang, Feifei Zhai, Jingfang Xu, and Chengqing Zong. 2018. Three strategies to improve one-to-many multilingual translation. In *EMNLP 2018*, pages 2955–2960.

Yiren Wang, ChengXiang Zhai, and Hany Hassan. 2020b. Multi-task learning for multilingual neural machine translation. In *EMNLP 2020*, pages 1022–1034.

Zirui Wang, Zachary C. Lipton, and Yulia Tsvetkov. 2020c. On negative interference in multilingual models: Findings and A meta-learning treatment. In *EMNLP 2020*, pages 4438–4450.

Biao Zhang, Ankur Bapna, Rico Sennrich, and Orhan Firat. 2021a. Share or not? learning to schedule language-specific capacity for multilingual translation. In *ICLR 2021*.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *ACL 2020*, pages 1628–1639.

Yikai Zhang, Songzhu Zheng, Pengxiang Wu, Mayank Goswami, and Chao Chen. 2021b. Learning with feature-dependent label noise: A progressive approach. In *ICLR 2021*.

Baohang Zhou, Xiangrui Cai, Ying Zhang, and Xiaojie Yuan. 2021. An end-to-end progressive multi-task learning framework for medical named entity recognition and normalization. In *ACL 2021*, pages 6214–6224.

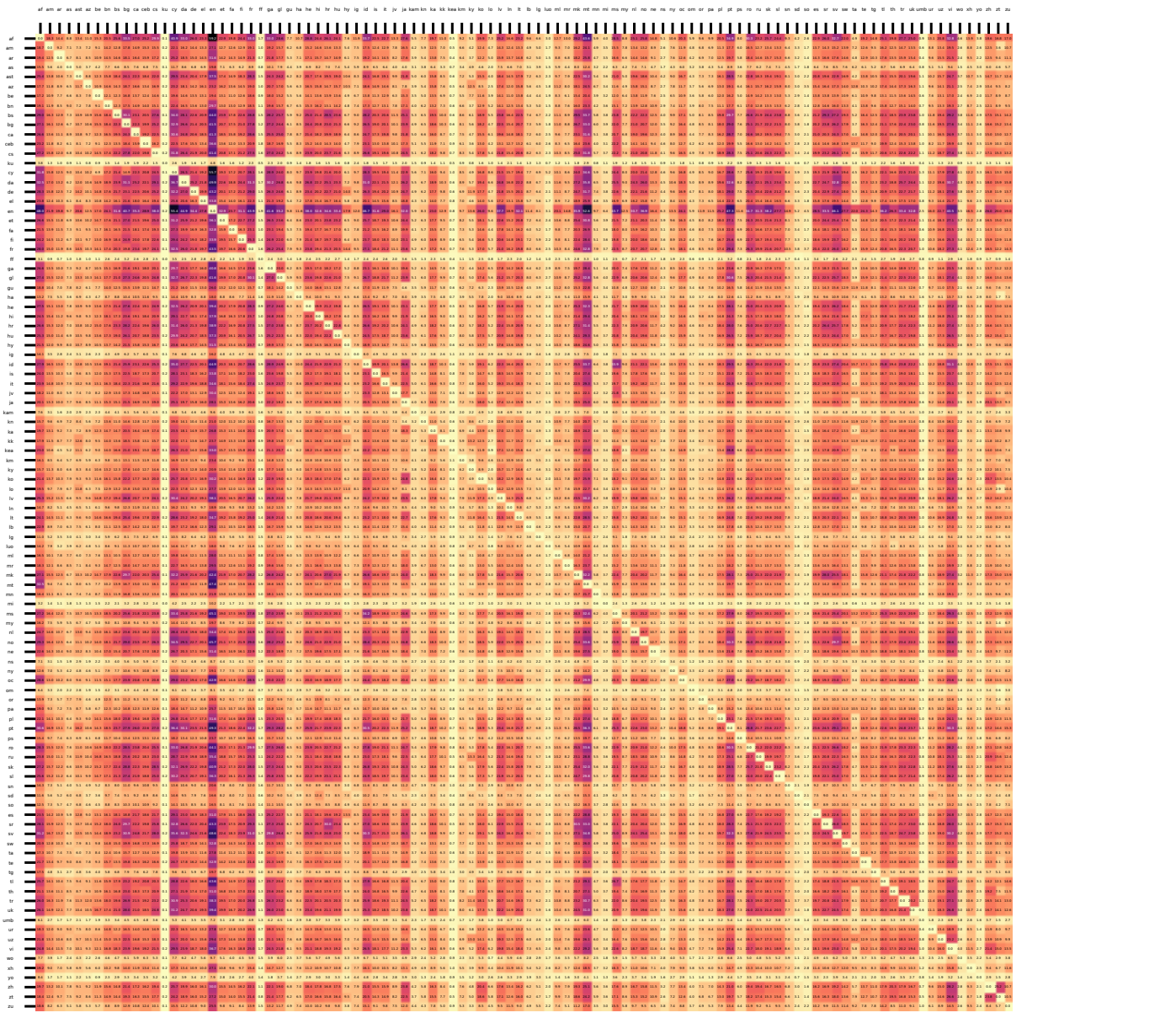

Figure 2: Evaluation results of our multilingual model (24 encoder layers and 6 decoder layers) on all translation directions on the FLORES-101 devtest set. The language x in the i-th row and language y in the j-th column denotes the translation direction from the language x to language y. For example, the cell of the 1-th row (af) and the 2-th column (am) represents the result of the translation direction af→am. The table shows the results of all translation directions of 102 languages.

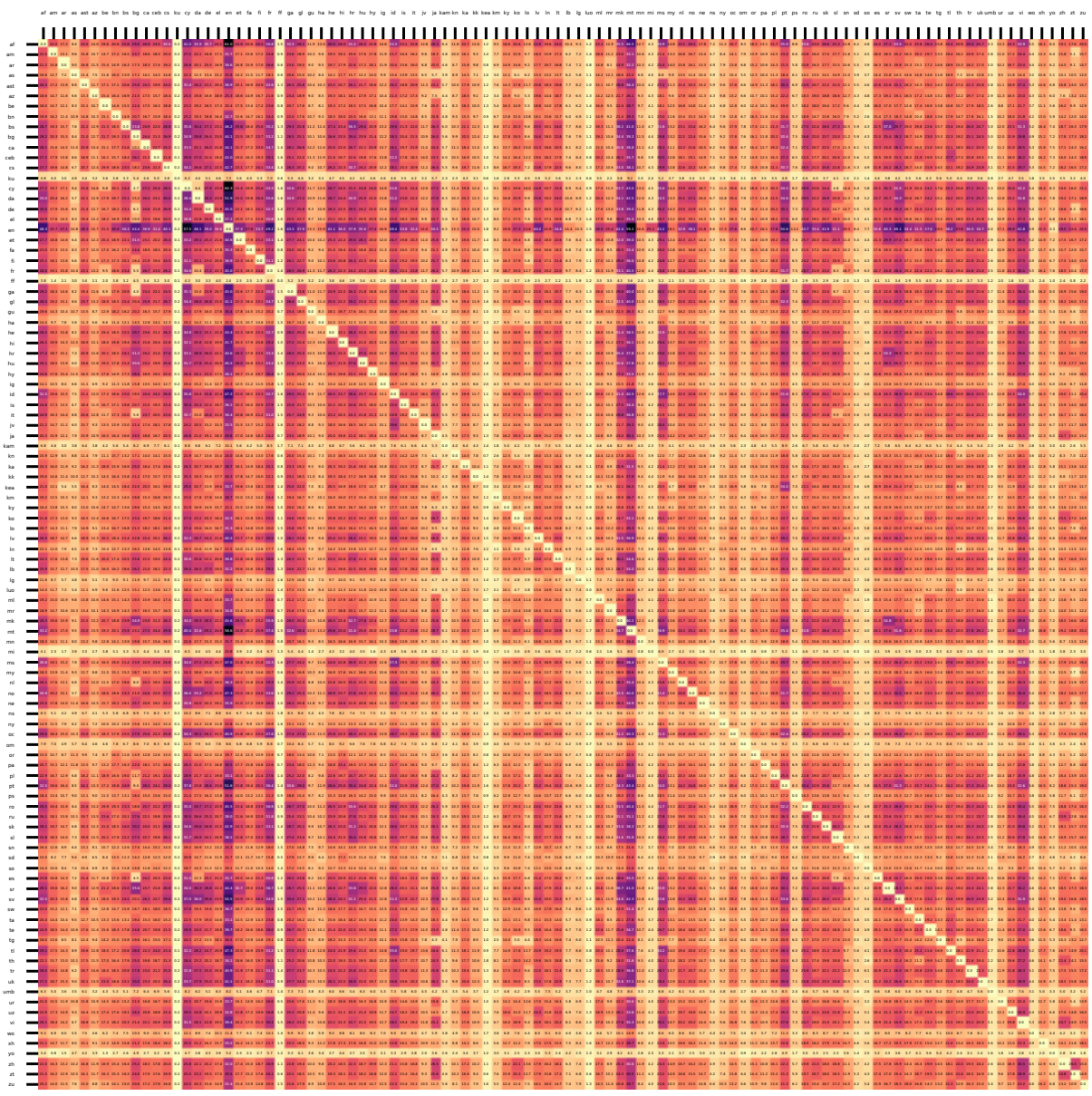

Figure 3: Evaluation results of our multilingual model (36 encoder layers and 12 decoder layers) on all translation directions on the FLORES-101 devtest set. The language x in the i-th row and language y in the j-th column denotes the translation direction from the language x to language y. For example, the cell of the 1-th row (af) and the 2-th column (am) represents the result of the translation direction af→am. The table shows the results of all translation directions of 102 languages.