SMaLL-100: Introducing Shallow
Multilingual Machine Translation Model for Low-Resource Languages

Alireza Mohammadshahi[*1,2,3]   **Vassilina Nikoulina**[1]   **Alexandre Berard**[1]
Caroline Brun[1]   **James Henderson**[2]   **Laurent Besacier**[1]

[1] NAVER LABS Europe   [2] IDIAP Research Institute   [3] EPFL

{first.last}@naverlabs.com
{alireza.mohammadshahi,james.henderson}@idiap.ch

Abstract

In recent years, multilingual machine translation models have achieved promising performance on low-resource language pairs by sharing information between similar languages, thus enabling zero-shot translation. To overcome the "curse of multilinguality", these models often opt for scaling up the number of parameters, which makes their use in resource-constrained environments challenging. We introduce *SMaLL-100*, a distilled version of the M2M-100 (12B) model, a massively multilingual machine translation model covering 100 languages. We train SMaLL-100 with uniform sampling across all language pairs and therefore focus on preserving the performance of low-resource languages. We evaluate SMaLL-100 on different low-resource benchmarks: FLORES-101, Tatoeba, and TICO-19 and demonstrate that it outperforms previous massively multilingual models of comparable sizes (200-600M) while improving inference latency and memory usage. Additionally, our model achieves comparable results to M2M-100 (1.2B), while being 3.6× smaller and 4.3× faster at inference.[1]

1 Introduction

Neural Machine Translation (NMT) systems are usually trained on datasets consisting of millions of parallel sentences, thus still performing poorly on low-resource languages, i.e., languages without a large amount of training data. Over the past few years, previous work has proposed several approaches to improve the quality of translations in low-resource languages, e.g., Multilingual Neural Machine Translation (MNMT) models (Johnson et al., 2017; Fan et al., 2020; Tang et al., 2021; Goyal et al., 2021), back-translation (Sennrich et al., 2016; Edunov et al., 2018) and unsupervised machine translation (Garcia et al., 2021; Ko et al.,

2021). Massively MNMT models are particularly interesting for low-resource languages as they benefit the most from knowledge transfer from related languages (Arivazhagan et al., 2019). However, it is also seen that *curse of multilinguality* hurts the performance of high-resource languages. So, previous work attempted to increase the model size to maintain the translation performance in both high and low-resource languages. This makes the use of these massively MNMT models challenging in real-world resource-constrained environments. To overcome this problem, we propose SMaLL-100, a **S**hallow **M**ultilingual **Ma**chine Translation Model for **L**ow-Resource **L**anguages covering 100 languages, which is a distilled alternative of M2M-100 (12B) (Fan et al., 2020), the most recent and biggest available multilingual NMT model. In this paper, we focus on very-low and low-resource language pairs as there is no reasonable-size universal model that achieves acceptable performance over a great number of low-resource languages. We do so by training SMaLL-100 on a perfectly balanced dataset.[2] While this leads to lower performance on the high-resource languages, we claim that this loss is easily recoverable through further fine-tuning. We evaluate SMaLL-100 on different low-resource benchmarks, e.g., FLORES-101 (Goyal et al., 2021), Tatoeba (Tiedemann, 2020), and TICO-19 (Anastasopoulos et al., 2020). To summarize, our contributions are as follows:

- We propose SMaLL-100, a shallow multilingual NMT model, focusing on low-resource language pairs.
- We evaluate SMaLL-100 on several low-resource NMT benchmarks.
- We show that our model significantly outperforms previous multilingual models of comparable size while being faster at inference. Additionally, it achieves comparable

---

[*]Work done during an internship at NAVER LABS Europe.
[1]The code and pre-trained SMaLL-100 model is available at https://github.com/alirezamshi/small100.

[2]All language pairs have the same sampling probability, regardless of their training data size.

results with M2M-100 (1.2B) model, with 4.3× faster inference and a 3.6× smaller size.

- While SMaLL-100 reaches 87.2% performance of the 12B teacher model, we show that this gap can be closed with a few fine-tuning steps both for low and high-resource languages.

2 Model and Training

2.1 SMaLL-100 Architecture

It has been shown by Kasai et al. (2021) that deep encoder / shallow decoder architectures can achieve good translation quality while being significantly faster at inference. Berard et al. (2021) have confirmed that this is also valid for multilingual NMT. Here, we use a 12-layer Transformer encoder (Vaswani et al., 2017) and 3-layer decoder. Table 8 in the Appendix B reports further details of the SMaLL-100 architecture. Different from M2M-100 model, we use language codes in the encoder side, as it is shown to perform better with shallow decoder architectures (Berard et al., 2021).

2.2 Training Strategy

SMaLL-100 is trained with a combination of two loss functions: a standard Cross Entropy loss (CE) and a Knowledge Distillation loss (KD). Given source sequence X and gold target translation $Y = (y_0,...,y_m)$, the CE loss is calculated as:

$$\mathcal{L}_{ce} = -\sum_{j=0}^{m}\sum_{z=1}^{|K|}\mathbb{1}\{y_j = z\}\log p(y_j = z|y_{<j}, X, \theta_S)$$

(1)

where $|K|$ is the target vocabulary size, $\mathbb{1}$ is the indicator function, and θ_S is the model parameters. $p()$ is the conditional probability function. We additionally use a word-level distillation loss, which is the Kullback–Leibler divergence between the output distributions of the student and teacher models (Hu et al., 2018). Specifically, it is calculated as:

$$\mathcal{L}_{kd} = -\sum_{j=0}^{m}\sum_{z=1}^{|K|}q(y_j = z|y_{<j}, X, \theta_T)$$
$$\times \log p(y_j = z|y_{<j}, X, \theta_S)$$

(2)

where θ_T is parameters of the teacher model. $q()$ is the conditional probability of the teacher model. The total loss is computed as:

$$\mathcal{L}_{total} = \mathcal{L}_{ce} + \alpha\mathcal{L}_{kd}$$

(3)

Resource Type	Criteria
Very-Low	$|K| \leq 100K$
Low	$100K < |K| \leq 1M$
Medium	$1M < |K| \leq 100M$
High	$100M < |K|$

Table 1: The criteria to split languages into different resource categories. $|K|$ is the amount of training data to/from English.

where α is a trainable parameter.

2.3 Training Data

Our training data includes parallel sentences from CCMatrix (Schwenk et al., 2019) and CCAligned (El-Kishky et al., 2020) datasets, which are part of the training data used by Fan et al. (2020) to train the M2M-100 models. As our goal is to maintain the performance of low-resource languages, we balance the training data across all language pairs; specifically, 100K sentence pairs are sampled for each language pair.[3] As a result, our training data contains nearly 456M parallel sentences, which is less than 6% of the original data on which M2M-100 (Fan et al., 2020) was trained. We use the same languages as M2M-100.

3 Experimental Setup

3.1 Evaluation Benchmarks

FLORES-101 is a multilingual NMT benchmark, containing 3,001 sentences from different domains, that are derived from English Wikipedia. Sentences are translated into 101 languages by human translators (Goyal et al., 2021). It mostly includes low and medium-resource languages. We use devtest subset for the evaluation.

Tatoeba is a crowd-sourced collection of user-provided translations in different languages (Tiedemann, 2020). We choose a subset of languages from test set of Tatoeba Challenge,[4] which are covered by M2M-100.

TICO-19 was created during the COVID-19 pandemic (Anastasopoulos et al., 2020). It contains sentences from 36 languages in the medical domain, including 26 low-resource languages.

---

[3]For language pairs with less than 100K sentence pairs, we repeat their data. We randomly select 100K sentences for language pairs with more than 100K training sentences.

[4]https://github.com/Helsinki-NLP/Tatoeba-Challenge

			Language Direction												
Model	params	Speed	VL2VL	VL2L	VL2M	VL2H	L2VL	L2L	L2M	L2H	M2VL	M2L	H2VL	H2L	AVG
FLORES-101															
FLORES-124	175M	5.3×	3.3	3.4	6.0	7.8	3.7	3.1	6.9	8.8	6.9	5.2	8.1	6.0	5.8
M2M-100	418M	3.1×	4.3	3.7	7.8	9.4	5.4	3.4	9.1	11.3	9.9	5.8	11.4	6.6	7.3
FLORES-124	615M	2.9×	5.1	5.1	9.2	11.2	5.8	4.7	10.6	13.1	10.3	7.6	11.5	8.5	8.6
Finetuned-100	330M	7.8×	6.1	5.4	8.7	11.3	5.7	4.1	9.0	11.8	10.4	6.8	13.0	8.0	8.4
SMaLL-100	330M	7.8×	7.9	7.0	10.3	12.6	8.4	6.1	11.6	14.3	13.7	9.0	16.7	10.2	10.7
M2M-100	1.2B	1.8×	6.7	6.1	10.8	12.8	8.7	6.1	13.0	15.9	13.6	8.8	15.4	9.7	10.6
M2M-100	12B	1×	**8.7**	**8.8**	**11.9**	**13.7**	**11.7**	**9.7**	**15.4**	**18.2**	**16.5**	**12.6**	**18.7**	**13.9**	**13.3**
Tatoeba															
FLORES-124	175M	5.3×	-	7.6	15.7	10.1	4.6	5.3	11.5	10.8	14.0	10.2	6.4	7.5	9.4
M2M-100	418M	3.1×	-	7.4	19.7	12.3	5.9	5.3	13.8	13.2	14.9	11.7	7.7	9.0	10.9
FLORES-124	615M	2.9×	-	**9.1**	19.4	11.4	6.9	7.6	12.7	13.7	14.4	13.3	8.0	9.7	11.4
Finetuned-100	330M	7.8×	-	4.0	21.1	14.4	7.7	5.2	15.3	14.2	14.0	12.1	8.9	8.3	11.4
SMaLL-100	330M	7.8×	-	4.6	22.1	**16.4**	8.7	7.0	16.7	15.8	16.3	14.5	10.6	11.2	13.1
M2M-100	1.2B	1.8×	-	8.8	19.5	13.1	8.7	7.2	16.3	17.0	17.2	13.4	**10.7**	11.1	13.0
M2M-100	12B	1×	-	8.6	**23.5**	13.1	9.8	10.2	17.8	17.9	18.5	15.2	10.7	13.2	14.4
TICO-19															
FLORES-124	175M	5.3×	4.6	5.5	8.1	11.5	4.4	5.6	9.7	12.2	3.9	8.0	4.2	8.7	7.2
M2M-100	418M	3.1×	4.0	5.5	9.8	13.7	4.2	5.7	11.6	14.9	4.1	8.8	5.3	9.4	8.1
FLORES-124	615M	2.9×	4.6	7.4	11.5	16.4	4.8	7.6	12.9	16.7	4.4	10.7	4.4	11.5	9.4
Finetuned-100	330M	7.8×	6.1	7.2	11.9	17.4	5.5	6.1	12.1	15.2	6.4	9.0	9.5	10.3	9.7
SMaLL-100	330M	7.8×	**7.8**	8.8	13.3	19.0	8.0	8.5	14.3	17.8	**8.3**	11.5	11.3	12.7	11.8
M2M-100	1.2B	1.8×	5.4	8.2	13.2	18.9	6.0	8.7	14.0	19.2	5.2	11.5	6.1	12.5	10.8
M2M-100	12B	1×	6.4	**10.9**	**15.4**	**20.6**	7.8	**11.9**	**16.6**	**21.4**	6.4	**15.4**	8.7	**16.4**	**13.1**

Table 2: Average spBLEU performance on FLORES-101, Tatoeba, and TICO-19 benchmarks for different language pair categories, defined in Appendix A. FLORES-101 results are computed on language pairs where M2M-100 12B has spBLEU scores higher than 3 to avoid polluting the analysis with meaningless scores. The first and second columns give the model size and speed-up ratios compared to M2M-100 (12B). Last column is the average spBLEU performance over all mentioned language directions. The best scores are shown in bold, and the second best results are shown with underline.

We evaluate on languages which are covered by M2M-100 (Fan et al., 2020).

Inspired by Goyal et al. (2021), we split the languages based on the amount of available training sentences aligned with English into 4 different categories: Very-Low (VL), Low (L), Medium (M), and High-resource (H). As the true amount of training data is both dependent on quality and quantity of parallel sentences, Goyal et al. (2021) suggested to estimate it by computing the number of bitext data aligned with English, that is calculated from statistics of OPUS corpora (Tiedemann, 2012). Table 1 illustrates the criteria for choosing the category of different languages. More details about the distribution of language pair categories in each benchmark are provided in Appendix A.

3.2 Baselines

M2M-100 Fan et al. (2020) is a recent many-to-many NMT model covering 100 languages. Fan et al. (2020) provide 3 variants with respectively 418M, 1.2B, and 12B parameters. We compare against these 3 variants.

FLORES-124 is an extension of M2M-100, covering additional 24 languages. Training data of the additional languages is derived from OPUS (Tiedemann, 2012). Goyal et al. (2021) provide two models with 175M and 615M parameters. We use both models as baselines.

FineTuned-100 uses the same architecture as defined in Section 2, but KD loss ($\mathcal{L}_{kd}$) is not used for training. For a fair comparison, it is trained for the same number of steps as SMaLL-100 model.

3.3 Implementation Details

SMaLL-100 contains nearly 330M parameters with 12 encoder and 3 decoder Transformer layers.[5] It is trained for 30 days on 16 TESLA V100-32GB GPUs,[6] with a batch size of 1K tokens and accumulated gradients over 9 batches. We

---

[5] It is initialized with M2M-100 (418M), using its first 3 decoder layers for the initialization of the student's decoder.

[6] 4 GPUs are used for the training of the student model. 12 GPUs are utilized to do the model parallelism of the teacher model.

Language pair	Language Type	Fine-tuned M2M-100 (418M)	SMaLL-100	steps	M2M-100 (12B)
Cebuano-English	Low	18.6	**29.8**	0.5K	27.7
English-Igbo	Low	9.2	**15.7**	1.5K	14.9
English-Malayalam	Low	16.7	**20.8**	3K	20.6
Georgian-Russian	Medium	6.9	**13.1**	0.5K	10.1
English-Italian	High	33.3	33.4	20K	**33.5**
French-Italian	High	31.3	31.5	20K	**32.0**
Italian-Spanish	High	26.6	26.8	20K	**27.0**

Table 3: spBLEU performance of fine-tuned SMaLL-100 and M2M-100 (418M) for the specified step, and M2M-100 (12B) on FLORES-101 devtest. The "step" column is the number of training steps required to reach M2M-100 (12B) performance. The type of each language pair is defined as the minimum of source and target language categories.

implement our model using fairseq repository.[7] We use `last-checkpoint`[8] of M2M-100 (12B) for the teacher model. For decoding, the beam search of 5 is applied. All hyper-parameters regarding the architecture and optimization strategy are provided in Appendix B.

For a faster convergence, we first fine-tune SMaLL-100 for 150k steps without distillation ($\mathcal{L}_{kd}$). Then, it is trained with both losses for 756K steps (nearly 1 epoch). For evaluation, we use SentencePiece BLEU (spBLEU), as it is shown to be a fair metric in multilingual settings (Goyal et al., 2021).[9] We use the same tokenizer and dictionary as M2M-100.

4 Results and Discussion

4.1 Low-Resource NMT Benchmarks

Table 2 shows the average spBLEU performance on FLORES-101, Tatoeba, and TICO-19 test sets for different categories of language directions.[10] SMaLL-100 outperforms all the models with comparable sizes while being smaller and faster at inference. Specifically, it outperforms M2M-100 418M both in terms of performance (+3.1 spBLEU) and inference speed (2.5× faster). We believe that Finetuned-100 outperforms M2M-100 418M for low-resource languages thanks to finetuning on the balanced dataset. The higher performance of

SMaLL-100 compared to Finetuned-100 across all benchmarks shows the benefit of KD loss which allows to distill knowledge from the teacher model. Additionally, SMaLL-100 achieves competitive results with M2M-100 (1.2B), while being 3.6× smaller and 4.3× faster at inference. Compared to the biggest M2M-100 model (12B), SMaLL-100 loses nearly 1.7 spBLEU but is 36× smaller and 7.8× faster. Regarding medium and high-resource language pairs (as shown in Appendix C.A), SMaLL-100 achieves better or similar performance compared to M2M-100 (418M) and FLORES-124 (615M), while it contains fewer parameters and is faster at the evaluation time. It under-performs for some medium and high-resource language pairs compared to the teacher model (M2M-100 12B), which could be easily recovered, as we describe in the remaining section.

4.2 Recovering Teacher Model Performance

To go further, we demonstrate that SMaLL-100 can easily recover the performance of the teacher model with just a few fine-tuning steps, both for low and high-resource language pairs. For comparison, we fine-tune M2M-100 (418M) model with the same number of steps.

Table 3 reports spBLEU performance for several language pairs, alongside the number of fine-tuning steps, required by SMaLL-100 model to reach M2M-100 (12B) performance.[11] We see that SMaLL-100 achieves better performance than M2M-100 (12B) after a few training steps on low-resource language pairs. For high-resource language pairs, SMaLL-100 is fine-tuned for 20K

---

[7]https://github.com/facebookresearch/fairseq
[8]https://github.com/facebookresearch/fairseq/tree/main/examples/m2m_100
[9]It utilizes a SentencePiece tokenizer with 256K tokens: https://github.com/facebookresearch/flores
[10]Complete spBLEU calculations of different language pairs on tested NMT benchmarks are provided in Appendix C. Speed is calculated on 2 TESLA V100-32GB GPUs with a batch size of 1 sentence over a subset of FLORES-101 devtest, containing nearly 10K sentences from all language pairs.

[11]More dataset and implementation details of these fine-tuning experiments are provided in Appendix D.

steps to reach the performance of M2M-100 (12B) model. Additionally, fine-tuned SMaLL-100 significantly outperforms fine-tuned M2M-100 (418M) model on low and medium-resource languages. This confirms that SMaLL-100 could be a powerful and lightweight initialization model for training on different language pairs.

5   Related Work

Compression and Distillation.   Over the past few years, pre-trained models lead to significant improvement by increasing the parameter size (Raffel et al., 2019; Fan et al., 2020; Zhang et al., 2022), which makes it challenging to use them in the resource-constraint environment. Previous work use several compression techniques e.g. knowledge distillation (Kim and Rush, 2016; Li et al., 2021), pruning (Behnke and Heafield, 2020; Zhang et al., 2021; Mohammadshahi et al., 2022), and quantization (Tao et al., 2022; Yao et al., 2022) to provide a reasonable-size model, while keeping the performance.

Multilingual NMT.   It provides a single model to translate between any pair of languages, which significantly improves performance on low-resource languages thanks to knowledge transfer (Haddow et al., 2021). Several works (Dong et al., 2015; Firat et al., 2016; Platanios et al., 2018; Fan et al., 2020; Berard et al., 2021) propose to include both language-specific, and language-independent parameters in MNMT models. Recently, massively MNMT models (Neubig and Hu, 2018; Arivazhagan et al., 2019; Aharoni et al., 2019; Fan et al., 2020; Zhang et al., 2020) have been proposed to translate between more than 100 languages. However, these models usually contain a huge number of parameters to maintain performance in both high and low-resource languages. Different from the previous work, we introduce SMaLL-100, which outperforms previous models with comparable size in low-resource language directions, while achieving better speed and being smaller.

6   Conclusion

We presented SMaLL-100 model, a shallow multilingual NMT model, focusing on low-resource languages. We evaluated our model on different NMT benchmarks. SMaLL-100 significantly outperforms multilingual models of comparable size on all of the tested benchmarks (FLORES-101,

Tatoeba, TICO-19) and is much faster at inference. It also achieves competitive results with M2M-100 1.2B (Fan et al., 2020), while being $4.3\times$ faster at inference and $3.6\times$ smaller. Compared to M2M-100 (12B), the biggest available MNMT model, SMaLL-100 loses nearly 1.7 spBLEU on average but it is significantly faster ($7.8\times$) and smaller ($36\times$), which makes it a good fit for resource-constrained settings. Additionally, we show that SMaLL-100 can achieve similar performance as M2M-100 (12B) with just a few steps of fine-tuning on specific language pairs.

Limitations

As mentioned in Section 4, SMaLL-100 model under-performs for some medium and high-resource languages, which could be resolved by further fine-tuning. Due to our computation constraint, we train SMaLL-100 model on nearly 6% of the original M2M-100 model. So, we encourage future research to increase the size of training data (especially for low-resource languages) to achieve better performance. Also, future research could apply different distillation strategies (Wu et al., 2020; Wang et al., 2021), as we just used word-level knowledge distillation loss (Hu et al., 2018).

Acknowledgement

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Antonios Anastasopoulos, Alessandro Cattelan, Zi-Yi Dou, Marcello Federico, Christian Federmann, Dmitriy Genzel, Francisco Guzmán, Junjie Hu, Macduff Hughes, Philipp Koehn, Rosie Lazar, Will Lewis, Graham Neubig, Mengmeng Niu, Alp Öktem, Eric Paquin, Grace Tang, and Sylwia Tur. 2020. TICO-19: the translation initiative for COvid-19. In *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*, Online. Association for Computational Linguistics.

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

Maximiliana Behnke and Kenneth Heafield. 2020. Losing heads in the lottery: Pruning transformer attention in neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2664–2674, Online. Association for Computational Linguistics.

Alexandre Berard, Dain Lee, Stephane Clinchant, Kweonwoo Jung, and Vassilina Nikoulina. 2021. Efficient inference for multilingual neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8563–8583, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5960–5969, Online. Association for Computational Linguistics.

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.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.

Xavier Garcia, Aditya Siddhant, Orhan Firat, and Ankur Parikh. 2021. Harnessing multilinguality in unsupervised machine translation for rare languages. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1126–1137, Online. Association for Computational Linguistics.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzman, and Angela Fan. 2021. The flores-101 evaluation benchmark for low-resource and multilingual machine translation.

Barry Haddow, Rachel Bawden, Antonio Valerio Miceli Barone, Jindřich Helcl, and Alexandra Birch. 2021. Survey of low-resource machine translation.

Minghao Hu, Yuxing Peng, Furu Wei, Zhen Huang, Dongsheng Li, Nan Yang, and Ming Zhou. 2018. Attention-guided answer distillation for machine reading comprehension. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2077–2086, Brussels, Belgium. Association for Computational Linguistics.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah Smith. 2021. Deep encoder, shallow decoder: Reevaluating non-autoregressive machine translation. In *International Conference on Learning Representations*.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Wei-Jen Ko, Ahmed El-Kishky, Adithya Renduchintala, Vishrav Chaudhary, Naman Goyal, Francisco Guzmán, Pascale Fung, Philipp Koehn, and Mona Diab. 2021. Adapting high-resource NMT models to translate low-resource related languages without parallel data. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 802–812, Online. Association for Computational Linguistics.

Bei Li, Ziyang Wang, Hui Liu, Quan Du, Tong Xiao, Chunliang Zhang, and Jingbo Zhu. 2021. Learning light-weight translation models from deep transformer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13217–13225.

Alireza Mohammadshahi, Vassilina Nikoulina, Alexandre Berard, Caroline Brun, James Henderson, and Laurent Besacier. 2022. What do compressed multilingual machine translation models forget?

Graham Neubig and Junjie Hu. 2018. Rapid adaptation of neural machine translation to new languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 875–880, Brussels, Belgium. Association for Computational Linguistics.

Emmanouil Antonios Platanios, Mrinmaya Sachan, Graham Neubig, and Tom Mitchell. 2018. Contextual parameter generation for universal neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 425–435, Brussels, Belgium. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, and Armand Joulin. 2019. Ccmatrix: Mining billions of high-quality parallel sentences on the web.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2021. Multilingual translation from denoising pre-training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3450–3466, Online. Association for Computational Linguistics.

Chaofan Tao, Lu Hou, Wei Zhang, Lifeng Shang, Xin Jiang, Qun Liu, Ping Luo, and Ngai Wong. 2022. Compression of generative pre-trained language models via quantization.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2214–2218, Istanbul, Turkey. European Languages Resources Association (ELRA).

Jörg Tiedemann. 2020. The tatoeba translation challenge – realistic data sets for low resource and multilingual MT. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1174–1182, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Fusheng Wang, Jianhao Yan, Fandong Meng, and Jie Zhou. 2021. Selective knowledge distillation for 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 (Volume 1: Long Papers)*, pages 6456–6466, Online. Association for Computational Linguistics.

Yimeng Wu, Peyman Passban, Mehdi Rezagholizadeh, and Qun Liu. 2020. Why skip if you can combine: A simple knowledge distillation technique for intermediate layers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1016–1021, Online. Association for Computational Linguistics.

Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. 2022. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models.

Tianfu Zhang, Heyan Huang, Chong Feng, and Longbing Cao. 2021. Enlivening redundant heads in multi-head self-attention for machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3238–3248, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Appendix A   Details of Evaluation benchmarks

A.A   Resource-type of Languages in Evaluated Datasets

af	Low	lg	Very-Low	lt	Medium	sn	Low	gl	Medium
am	Low	ka	Medium	luo	Low	sd	Very-Low	gd	Low
ar	Medium	de	High	lb	Medium	sk	Medium	ht	Low
hy	Low	el	Medium	mk	Medium	sl	Medium	su	Low
as	Very-Low	gu	Low	ms	Low	so	Low	ln	Very-Low
ast	Low	ha	Low	ml	Low	ku	Low	ilo	Low
az	Low	he	Medium	mt	Medium	es	High	mg	Medium
be	Very-Low	hi	Medium	mr	Low	sw	Low	tn	Very-Low
bn	Medium	hu	Medium	mi	Low	sv	Medium	br	Medium
bs	Low	is	Medium	mn	Low	tg	Low	ns	Very-Low
bg	Medium	ig	Low	ne	Very-Low	ta	Low	si	Medium
my	Low	id	Medium	nso	Very-Low	te	Low	yi	Low
ca	Medium	ga	Low	no	Medium	th	Medium	fy	Medium
ceb	Low	it	High	ny	Low	tr	Medium	sq	Medium
zh	Medium	ja	Medium	oc	Very-Low	uk	Medium	ss	Very-Low
hr	Very-Low	jv	Medium	or	Very-Low	umb	Low	fr	High
cs	Medium	kea	Very-Low	om	Low	ur	Low	ff	Very-Low
da	Medium	kam	Very-Low	ps	Low	uz	Very-Low	lo	Low
nl	Medium	kn	Low	fa	Medium	vi	Medium	lv	Medium
en	High	kk	Low	pl	Medium	cy	Low	ru	High
et	Medium	km	Low	pt	High	wo	Very-Low	sr	Medium
tl	Very-Low	ko	Medium	pa	Low	xh	Low	zu	Low
fi	Medium	ky	Low	ro	Medium	yo	Low	ba	Low

Table 4: ISO-639 code and resource type of languages used in evaluated NMT benchmarks.

A.B   FLORES-101

We use `devtest` subset of FLORES-101 for the evaluation. To better compare different models, we exclude evaluation of language pairs, in which the spBLEU performance of M2M-100 12B (Fan et al., 2020) model is below 3. This gives 5,934 language directions for the comparison. Table 5 shows the distribution of different categories of language pairs.

	VL2VL	VL2L	VL2M	VL2H	L2VL	L2L	L2M	L2H	M2VL	M2L	M2M	M2H	H2VL	H2L	H2M	H2H
No. lang. pairs	44	200	388	81	144	645	960	186	181	992	1330	259	35	190	257	42

Table 5: Distribution of resource categories for different language directions on FLORES-101 (Goyal et al., 2021).

A.C   Tatoeba Challenge

We use the `test` subset data, provided by Tiedemann (2020)[12] to evaluate all models. We choose a subset of dataset that includes languages which are covered by M2M-100 (Fan et al., 2020) model. This brings 1,844 language pairs for the evaluation. The distribution of different language pair categories is shown in Table 6.

---

[12]https://github.com/Helsinki-NLP/Tatoeba-Challenge

	VL2L	VL2M	VL2H	L2VL	L2L	L2M	L2H	M2VL	M2L	M2M	M2H	H2VL	H2L	H2M	H2H
No. lang. pairs	7	30	37	7	34	144	113	30	144	632	237	37	113	237	42

Table 6: Distribution of resource categories for different language directions on Tatoeba Challenge.

A.D TICO-19

We use the evaluation benchmark provided by Anastasopoulos et al. (2020)[13] to compare all models in multilingual medical domain. We utilize language pairs that are included in M2M-100 (Fan et al., 2020) model. This gives us 650 language pairs for the evaluation. Table 7 shows the number of language pairs in different resource types of language directions.

	VL2VL	VL2L	VL2M	VL2H	L2VL	L2L	L2M	L2H	M2VL	M2L	M2M	M2H	H2VL	H2L	H2M	H2H
No. lang. pairs	12	48	24	16	48	132	72	48	24	72	30	24	16	48	24	12

Table 7: Number of language pairs for different categories of language directions on TICO-19.

Appendix B Hyper-Parameters for Architecture and Optimization

Hyper-Parameter	Specification	Hyper-Parameter	Specification
Encoder Layer	12	Scheduler	inverse-sqrt
Decoder Layer	3	Optimizer	Adam
Encoder Emb dim.	1024	Clip Norm	1.0
Decoder Emb dim	1024	Learning Rate	1e-4
Encoder FFN Emb dim.	4096	Warmup init LR	1e-07
Decoder FFN Emb dim.	4096	Warmup Updates	40K
Number of attn heads	16	Adam Betas	0.9,0.98
Attention Dropout	0.1	Adam eps	1e-6
Share Encoder/Decoder Emb.	True	Label Smoothing	0.1
FP16	True	Dropout	0.1
Loss Scalar	2.0	Max Tokens (per GPU)	1,000

Table 8: List of hyper-parameters used for the architecture, and optimization.

---

[13] https://tico-19.github.io/

Appendix C   spBLEU Results

C.A   spBLEU of Remaining Language Directions

Model	params	Speed	Language Direction			
			M2M	M2H	H2M	H2H
FLORES-101						
FLORES-124	175M	5.3×	13.7	18.0	16.8	23.0
M2M-100	418M	3.1×	18.1	23.0	21.6	27.8
FLORES-124	615M	2.9×	19.3	24.1	22.7	28.9
Finetuned-100	330M	7.8×	16.4	21.2	19.7	26.0
SMaLL-100	330M	7.8×	19.3	24.2	22.6	28.8
M2M-100	1.2B	1.8×	22.3	28	25.8	32.7
M2M-100	12B	1×	23.9	29.5	27.6	34.3
Tatoeba						
FLORES-124	175M	5.3×	25.2	28.1	24.5	35.3
M2M-100	418M	3.1×	29.6	34.0	29.6	41.6
FLORES-124	615M	2.9×	31.7	35.5	31.0	43.0
Finetuned-100	330M	7.8×	28.3	34.2	28.1	39.2
SMaLL-100	330M	7.8×	31.9	36.3	31.1	42.4
M2M-100	1.2B	1.8×	33.8	39.0	34.2	47.4
M2M-100	12B	1×	33.1	39.0	34.2	48.7
TICO19						
FLORES-124	175M	5.3×	15.1	20.3	17.7	28.1
M2M-100	418M	3.1×	20.6	26.6	23.6	33.1
FLORES-124	615M	2.9×	20.2	27.0	23.3	33.9
Finetuned-100	330M	7.8×	19.7	24.4	23.4	31
SMaLL-100	330M	7.8×	21.7	27.4	25.2	33.7
M2M-100	1.2B	1.8×	21.1	30.2	24.8	37.7
M2M-100	12B	1×	24.8	33.1	28.3	39.4

Table 9: Average spBLEU performance of different models on FLORES-101, Tatoeba, and TICO-19 benchmarks for medium and high-resource language pair categories, defined in Section 4. The FLORES-101 results are computed on language pairs where M2M-100 12B has spBLEU scores higher than 3 to avoid polluting the analysis with meaningless scores. The first and second columns give the model size and speed-up ratios compared to M2M-100 (12B).
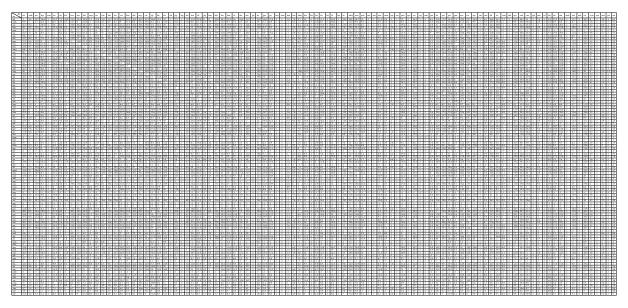
C.B FLORES-101

Table 10: spBLEU performance of last checkpoint of SMaLL-100 model on language pairs of FLORES-101.
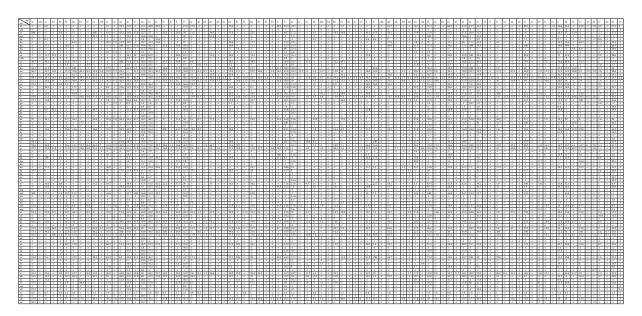
C.C Tatoeba

Table 11: spBLEU performance of last checkpoint of SMaLL-100 model on language pairs of Tatoeba.

C.D TICO19

tgt\src	am	ar	bn	my	zh	en	tl	fr	lg	ha	hi	id	km	ln	ms	mr	ne	ps	fa	ru	so	es	sw	ta	ur	zu
am	-1	7.5	12.7	1.5	7.6	18.3	11.3	13.7	2.4	6.9	16.2	14.2	5.5	2.4	14.4	5.9	6.1	7.3	11.8	10.6	0.5	15.7	13.7	6	10.4	5.2
ar	7.8	-1	16.8	1.5	12.3	29.9	3.8	23.6	1.5	7.4	24.9	26	9.8	1.8	23.9	4.2	0.3	6.4	20.5	18.8	0.4	29.5	21.3	1.7	13.3	2.5
bn	9.7	15.5	-1	4.4	13.8	33.8	23.9	21.6	3.9	10.7	31.1	27.9	12.6	4.4	27.3	11.3	16.4	10.4	21.1	19.4	1.6	27.9	22.7	9.9	18.3	9.9
my	6.7	6.3	11.4	-1	7.9	16.6	14.2	11.2	3.2	7.9	14.7	14.3	7.9	2.8	14.8	5.6	8	7.9	11.5	10.2	1.3	14.4	13.7	6.1	11	7.1
zh	8.6	15.6	18.3	4	-1	27.5	20.6	20.9	3.6	9.3	24.3	26.5	10.6	4	24.5	8.5	6.8	9.2	19.9	19	1.4	26.6	20.9	5.3	15.2	9
en	11.1	25.2	26.8	7	19.6	-1	35.6	36.3	5.6	15.4	39.3	47	17.9	3.6	44.7	12.1	19	11.8	29.9	29.7	2.1	47.5	32.4	9.3	20.8	15.7
tl	10.1	15.2	21.4	6	14.6	45.4	-1	26.1	5.7	14	28.5	33.9	14.7	6	33.2	10.1	14.6	10.6	21.9	22.5	1.6	33.3	26.2	7	17	14.6
fr	8.5	18.4	17.8	3.8	13.9	35.1	22.3	-1	3.5	9.7	26.1	30.7	11.6	3.9	27.6	8.3	2.6	7.9	20.8	22.5	1.1	34.3	21.3	1.7	14.3	8.8
lg	4.1	3.1	6.5	2.6	4.2	14.1	10.8	8.5	-1	7.4	7.2	10.2	3.9	4.6	9.8	3.9	4.5	4	5.7	8.7	1.5	11.3	9.7	2.4	5.2	6.8
ha	5.9	6.4	10.3	3.2	6.4	20.7	16.8	12.6	3.9	-1	13.5	16.2	8.1	4.8	16	5	6.8	7.2	11.7	11.7	1.4	16.6	14.7	4.3	9.8	8.3
hi	10.5	18	25.4	4.6	15.3	40.8	26.3	25	4.2	11.4	-1	31.9	13.5	4.5	31	13.5	20.4	11	23.6	22.1	1.6	32.6	25.5	10.4	21.4	10.5
id	9.7	19.5	22.2	4.8	17.4	43	22.9	28.6	4.7	12.7	30.2	-1	14.3	5.4	39.1	7.4	0.7	10.3	25.6	25	1.9	36.5	27.5	3.9	17.5	11.7
km	7.6	10.8	14.6	3.2	10	25	19.4	17	3.8	10.4	20	22.5	-1	4.1	22.2	6.8	9.4	8.8	15.9	14.7	1	21.5	18.9	6.6	13.1	8.6
ln	3.6	3.4	5.9	2.1	4.2	12	9.2	8.1	4.2	6.1	6.8	10.9	3	-1	10.1	3.4	3.4	4.3	5.9	8.2	1.2	10.7	9.4	2	5.2	6.3
ms	10.1	19.2	22.6	5	16.3	44.2	28	27.6	4.8	13.2	30.8	41.4	14.7	5.5	-1	8.9	1.1	10.6	25.2	23.8	1.9	35.2	28.2	6.6	17.8	12.4
mr	7.9	10.8	17.7	2.6	9.8	23.9	18.3	16.4	2.6	8.2	24.1	19.2	9.5	2.8	19.7	-1	12.3	8	15.8	14.1	0.9	20.4	16.4	8	14.9	7.3
ne	9.5	8.6	21.9	4	12.2	33.1	22.1	19.1	3.9	9.9	31.7	20.8	11.1	4.3	24.9	11.4	-1	10.5	14.8	17.3	1.7	25.1	20.4	8.4	17.4	9.5
ps	8	10.9	16.2	3.4	9.9	23.4	18.3	15.5	3.4	8.9	20.7	19.4	10	3.8	19.2	7.8	11.7	-1	17.1	13.9	1.4	20.1	17.1	7.6	14.7	8
fa	8.8	18.2	19.3	3.3	14.6	32	10.8	23.5	3.6	10.1	26	29	11.8	4	26.8	6.2	0.9	9.5	-1	20.9	1.2	29.7	22.6	3.5	15.8	7
ru	9.1	18.2	20.2	3.6	15.7	33.1	23	25.9	4	10.5	26.4	30	11.8	4.6	26.8	9.4	12.9	9.3	22	-1	1.4	32.7	22.6	4.6	15.9	10.5
so	1.1	0.2	1.1	0.9	0.3	2.1	3.4	1	1.6	3.3	1.8	1.2	1.4	1.5	1.5	0.8	0.8	2.4	0.6	0.3	-1	2.2	3.4	0.6	1.2	3
es	9.7	22.2	22.4	4.8	17.6	45.6	27.3	33.7	4.2	12.2	31.8	38	13.5	5.1	33.5	9.3	3.5	9.7	26	27.7	1.7	-1	26.5	2.4	17.7	9.8
sw	9.2	15.3	18.9	4.9	13	33.5	24.1	22.4	4.8	12.8	24.9	29.2	13.5	5.9	29.3	8.4	6.6	10.1	20.8	19.7	1.4	28.4	-1	6.3	16.1	8.1
ta	6	6.6	12	1.2	6.3	18.1	12.4	10.5	1.5	6.4	16.8	13.3	5.5	2.2	13.4	5.9	5.8	6.8	11	9.4	0.5	13.8	12.4	-1	10.9	4.3
ur	8.7	12.8	19.4	3.1	11.6	26.6	20	18	3.3	9.6	26.7	22.5	9.4	3.6	22.2	9.4	12.4	10	18.6	15.9	1.4	23.2	19.6	8.4	-1	7.5
zu	7.3	9.4	13.8	4.3	9.3	27.3	21.9	15.9	4.7	11.3	17.3	20.8	10.3	5.3	20.7	7	9.3	8.5	15.4	14.9	1.2	21.5	18.3	5.6	12.4	-1

Table 12: spBLEU performance of last checkpoint of SMaLL-100 model on language pairs of TICO19.

Appendix D   Details of Fine-Tuning on Low-Resource Language Pairs

For further fine-tuning of SMaLL-100 and M2M-100 (418M) models on selected language pairs, we use bilingual data provided by Tiedemann (2020) (release 2021.08.07)[14] as its training data is less noisy. We evaluate fine-tuned models on devtest subset of FLORES-101 (Goyal et al., 2021) benchmark with spBLEU metric (Goyal et al., 2021). We use the same hyper-parameters, as defined in Appendix B. We train each model on 2 TESLA V100-32GB GPUs.

---

[14] https://github.com/Helsinki-NLP/Tatoeba-Challenge/blob/master/data/README-v2021-08-07.md