# Revisiting Modularized Multilingual NMT to Meet Industrial Demands

**Sungwon Lyu**[1], **Bokyung Son**[1,2], **Kichang Yang**[1,3], **Jaekyoung Bae**[1]
[1]Kakao Enterprise / Seoul, Republic of Korea
[2]Department of Linguistics, Seoul National University
[3]School of Software, Soongsil University
{james.ryu, meta.mon, kevin.y, storm.b}@kakaoenterprise.com

## Abstract

The complete sharing of parameters for multilingual translation (1-1) has been the mainstream approach in current research. However, degraded performance due to the capacity bottleneck and low maintainability hinders its extensive adoption in industries. In this study, we revisit the multilingual neural machine translation model that only share modules among the same languages (M2) as a practical alternative to 1-1 to satisfy industrial requirements. Through comprehensive experiments, we identify the benefits of multi-way training and demonstrate that the M2 can enjoy these benefits without suffering from the capacity bottleneck. Furthermore, the interlingual space of the M2 allows convenient modification of the model. By leveraging trained modules, we find that incrementally added modules exhibit better performance than singly trained models. The zero-shot performance of the added modules is even comparable to supervised models. Our findings suggest that the M2 can be a competent candidate for multilingual translation in industries.

## 1 Introduction

With the current increase in the demand for neural machine translation (NMT), serving an increasing number of languages poses a practical problem for the industry. A naive approach for multilingual NMT is to have multiple single-directional models, which is unsustainable owing to the quadratic increase of models as more languages are introduced. A more practical approach is to limit the number of models by sharing the components among the models (Dong et al., 2015; Firat et al., 2016a; Ha et al.; Johnson et al., 2017). In addition to reducing the number of parameters, sharing the components is also regarded as an effective method to enhance the performance. A fully shared model (henceforth 1-1), which only uses one encoder and one decoder to translate all directions (Ha et al.; Johnson et al., 2017), has been the most popular method because of its compactness.

However, introduction of a significant number of tasks into a 1-1 model is known to cause capacity bottleneck. Aharoni et al. (2019) suggested that, given a fixed model capacity, a 1-1 model is bound to the tradeoff between the number of languages and translation accuracy. Zhang et al. (2020) explicitly identified the capacity bottleneck problem of the 1-1 model by showing a clear decrease in performance when translation directions are doubled. Moreover, data unbalance complicates the problem. Arivazhagan et al. (2019b) presented the transfer and interference dilemma among low and high resource languages in an unbalanced environment.

The capacity bottleneck observed in the 1-1 model is particularly undesirable for the industry. Unlimited scaling of the model size (Zhang et al., 2020) is impossible in practice, where inference cost and latency are crucial. With limited capacity, gain from multilingual translation training (henceforth multi-way training) without being subject to the losses of the capacity bottleneck is difficult to achieve. Furthermore, modification of the 1-1 model such as simple addition of a language is troublesome because the entire model must be retrained from the beginning as a single module, thus requiring a considerable amount of time and effort. This low maintainability makes 1-1 less attractable for industrial use. Still, the benefit from multi-way training is difficult to miss.

These problems lead us to revisit the multilingual neural machine translation model that share parameters among the same languages (Firat et al., 2016a). We named this architecture as the *modularized multilingual NMT model* (henthforth M2) since the model share language-specific modules (encoders or decoders) instead of the whole model.



*Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 5905–5918,
November 16–20, 2020. 

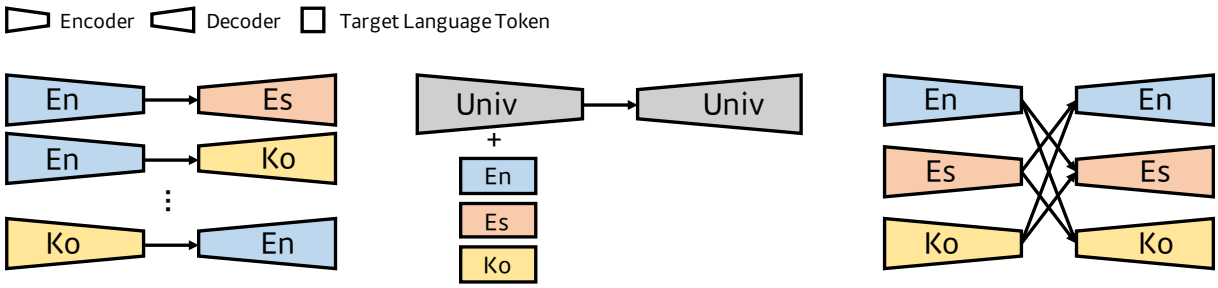


Figure 1: **Model overview** of three different types of multilingual NMT models for three languages: En, Es, Ko. *Left* is a collection of single models for 6 translation directions. *Middle* is the 1-1 model that share the whole parameters of the model for 6 directions. *Right* is the M2 model that only share language-specific modules.

Figure 1 illustrates the architectural overview of multilingual translation using single models, the 1-1 and the M2. Although the M2 has not been given substantial attention owing to the linear increase in its parameters as the number of languages increases, it is relatively free from the capacity bottleneck problem while maintaining a reasonable inference cost. In this study, we explore the possibility of M2 as an alternative to the 1-1 model in industrial settings.

To resolve the capacity bottleneck problem while enjoying the benefits, we identify the effects of multi-way training in a carefully controlled environment. We find that the *data-diversification* and *regularization* of multi-way training enable the M2 to outperform both single and 1-1 models with less suffering from capacity bottlenecks. Additionally, the M2 demonstrates a comparable performance increase to 1-1 for low resource pairs in an unbalanced environment.

Combined with its modularizable architecture, interlingual space learned by the M2 allows convenient and effective modification of the model. The simple addition of language-specific modules to the M2 outperformed an individually trained model. The zero-shot learning of the incremented language module outperforms English pivoted translation and is even comparable to a supervised model. Finally, we show that the language invariance of such space improves with more languages.

In summary, our contribution is threefold. 1) We conceptually specify the effects of multi-way training and verified them with comprehensive experiments. 2) We show that the M2 can leverage those effects as the 1-1 without the constraint of the capacity bottleneck. 3) Finally, we find that multi-way training of the M2 forms interlingual space which allows simple yet effective extension of languages.

## 2 Related works

### 2.1 Neural machine translation

The most popular framework for NMT is the encoder-decoder model (Cho et al., 2014; Sutskever et al., 2014; Bahdanau et al., 2014; Luong et al., 2015; Vaswani et al., 2017). Adopting attention module greatly improved the performance of encoder-decoder model by using context vector instead of fixed length vector (Bahdanau et al., 2014; Luong et al., 2015). By exploiting multiple attentive heads, the Transformer model has become the de-facto standard model in NMT (Vaswani et al., 2017; Ott et al., 2018; So et al., 2019).

### 2.2 Multilingual neural machine translation

Dabre et al. (2019) categorized the architectures of multilingual NMTs according to their degrees of parameter sharing. We briefly introduce the models under their criteria.

Early multilingual NMT models *minimally shared* the parameters by sharing language-specific encoder (Dong et al., 2015; Lee et al., 2017) or decoder (Zoph and Knight, 2016). Firat et al. (2016a) extended this to sharing both language-specific encoders and decoders with a shared attention module.

The 1-1 model, *fully shared*, uses only one encoder and decoder to translate all directions (Ha et al.; Johnson et al., 2017). The target language is indicated by prepending a reserved target language token to the source text. Being compact, the 1-1 model has become the mainstream of multilingual NMT research (Ha et al.; Johnson et al., 2017; Aharoni et al., 2019; Arivazhagan et al., 2019b; Wang et al., 2019; Liu et al., 2020), However, subsequent studies tried to solve the capacity bottleneck problem of the 1-1 through knowledge compression (Tan et al., 2019b), language clustering (Tan et al.,

2019a) or increased capacity (Zhang et al., 2020).

*Partially shared* models are extensively studied to compromise the capacity bottleneck and model size (Blackwood et al., 2018; Sachan and Neubig, 2018; Platanios et al., 2018; Zaremoodi et al., 2018; Bapna and Firat, 2019). Despite their popularity, we do not compare them in this work because *partially sharing* is essentially relaxing the capacity constraint of *fully sharing*. Also, Sachan and Neubig (2018) reported that the performance of *partially shared* models is language-specific, which is not the focus of our study. Instead, we focus on the general trade off of parameter sharing.

### 2.3 Interlingual representation

Building interlingual[1] representation is another interest in multilingual language modeling (Schwenk and Douze, 2017). Interlingual space is the ground for zero-shot translation (Johnson et al., 2017; Arivazhagan et al., 2019a; Al-Shedivat and Parikh, 2019) and incremental training (Escolano et al., 2019). Several explicit methods were suggested to build interlingual space including shared attention (Firat et al., 2016a), neural interlingua module (Lu et al., 2018), attention bridge (Vázquez et al., 2019), auxiliary loss (Arivazhagan et al., 2019a) and shared encoder (Sen et al., 2019).

We further extend the study of Firat et al. (2016a) which inspired our M2. Firat et al. (2016a) only shared English encoder and decoder as they used English-centered data (parallel corpus that include English). Instead we show that sharing modules of all languages using diverse directions of data further increases the performance and is the key to build interlingual representation without any explicit regularization.

Our motivation to rediscover the M2 is concurrently shared with Escolano et al. (2020). Escolano et al. (2020) empirically show that M2 is capable of quickly deploying new languages with incrementally added modules, and found it outperforms 1-1. We also experiment on incremental learning and get a similar conclusion, and further interpret the results as an indication that M2 effectively forms an interlingual space. Regarding comparison of M2 and 1-1 in general, we deliver an in-depth understanding of a less-studied model M2 focusing on how to maximize its utility in industry. Experiments on incremental learning are to check whether M2 is a maintainable alternative to 1-1 (which requires expensive re-training from scratch).

[1]We prefer the term 'interlingual' to 'language-agnostic' because we expect it to be better if the space is shared while maintaining the language-specific features instead of removing them.

## 3 Effects of multi-way training

Because of its complexity, the effects of multi-way training are yet to be identified. Various factors may affect the performance of multilingual translation: model size compared to the amount of data, the number of training directions, the degree of data imbalance among different directions, and the portion of multi-parallel data. In this section, we discuss the possible effects on performance resulting from these factors.

**Capacity bottleneck** A capacity bottleneck is the most plausible cause of performance degradation in multi-way training. For a fixed size model, the capacity bottleneck is more prominent with the increase in training directions (especially target languages) and the amount of data (Johnson et al., 2017; Aharoni et al., 2019; Arivazhagan et al., 2019b; Zhang et al., 2020).

**Cross-language effect** Cross-language effect occurs when multiple languages are shared in a module. Low resource languages reportedly benefit from multi-way training when trained along with high resource pairs (Zoph et al., 2016; Nguyen and Chiang, 2017; Neubig and Hu, 2018). The interaction among languages in a module can either be positive (transfer) or negative (interference) on the performance according to their similarity in linguistic patterns.

**Data-diversification** Data-diversification is associated with the portion of multi-parallel among multi-way data. If either the source-side or the target-side language is shared across two directions and data of the directions is not multi-parallel, the shared module learns more diverse samples of the language. For example, if an English encoder is shared between En-De and En-Fr directions (and English sentences of two are not completely shared), the encoder learns more diverse English sentences from both pairs. Few studies distinguished this effect (Firat et al., 2016a,b). We refer to the improvement resulting from this factor as the *data-diversification* effect.

**Regularization** Learning to encode or decode the same language in various directions may result

in better representation learning and less overfitting in a single direction. This effect has already been observed by Firat et al. (2016a) as the benefit of generalization and suggested by Aharoni et al. (2019) to benefit many-to-many models compared to many-to-one models.

## 4 Comparison of single models, 1-1 and M2

We compared the models with the same inference capacity in a series of conditions. Note that most of the multilingual NMT research was conducted in a joint one-to-many and many-to-one environment (JM2M): collected data are English-centered. Despite its simplicity, observations under such setting may be unreliable to speak for many-to-many (M2M) environment, which is also clearly in demand in the industry. Therefore, we set M2M training as the default.

We also distinguish between two different dataset compositions: the *sharing* case where all language pairs share the same sentence set, and the *non-sharing* case where there is no overlap between different pairs. To illustrate, a multiparallel set 'En - Es - Ko' can be shared for all possible three pairs (En - Es, En - Ko, Es - Ko) or used only once for one pair. Considering that multiparallel data is rare in practice, we compared the models in a strictly non-sharing environment.

### 4.1 Settings

**Dataset** We collected multi-parallel data from Europarl (Koehn, 2005) and selected four languages: German, English, Finnish, and French. To construct a completely balanced environment, we created 500K, 10K, and 10K (train, valid, and test) non-sharing pairs for every twelve possible directions from 1.56M multi-parallel data. For the unbalanced environment, we synthetically reduced the amount of data for some pairs to match a specific ratio of the data amounts for low, medium, and high resource pairs. For further details on data division, see appendix A.

**Model** For the 1-1 model, we used the model of Aharoni et al. (2019) which is transformer implementation of Johnson et al. (2017). For the M2, we modified Firat et al. (2016a) to not share the attention module. Language-specific embeddings are shared between the encoder and decoder. We implemented all models using transformer (Vaswani et al., 2017). We used the transformer with a hidden dimension of 256 and a feed-forward dimension of 1024 for our base model. The rest of the configuration follows the base model employed by Vaswani et al. (2017) except for the attention dropout and activation dropout of 0.1. The 1-1 model uses a joint vocabulary with 32K tokens, whereas the M2 uses a language-specific vocabulary with 16K tokens each, all processed using the BPE (Kudo, 2018) of the `sentencepiece` package[2] (Kudo and Richardson, 2018).

| Pairs | Single | 1-1 | M2 |
|---|---|---|---|
| De-En | 33.00 | 31.04 (-1.96) | 33.51 (0.51) |
| De-Fi | 15.20 | 13.08 (-2.12) | 15.93 (0.73) |
| De-Fr | 28.47 | 25.73 (-2.74) | 29.08 (0.61) |
| En-De | 25.87 | 23.83 (-2.04) | 26.46 (0.59) |
| En-Fi | 19.57 | 16.94 (-2.63) | 20.03 (0.46) |
| En-Fr | 35.74 | 32.99 (-2.75) | 36.09 (0.35) |
| Fi-De | 18.97 | 16.75 (-2.22) | 19.51 (0.54) |
| Fi-En | 29.26 | 27.32 (-1.94) | 30.24 (0.98) |
| Fi-Fr | 25.21 | 22.24 (-2.97) | 25.94 (0.73) |
| Fr-De | 22.23 | 20.09 (-2.14) | 22.64 (0.41) |
| Fr-En | 35.49 | 33.81 (-1.68) | 36.18 (0.69) |
| Fr-Fi | 15.42 | 13.6 (-1.82) | 16.15 (0.73) |
| Avg | 25.37 | 23.12 (-2.25) | **25.98 (0.61)** |

Table 1: SacreBLEU test scores of single models, 1-1, and M2 trained using a completely balanced, non-sharing dataset. Values in parentheses indicate the performance difference from single models.

**Training** We used the `fairseq` framework [3] (Ott et al., 2019) to train and test all models. We set the batch size so that every encoder/decoder module learned at a maximum of 6144 tokens/GPU. All models were trained using 4 NVIDIA Tesla V100 GPUs. We followed the default parameters of the Adam optimizer (Kingma and Ba, 2014). For the learning rate schedule, we used 2K warm-up steps until 1e-3, after which we used the inverse square root learning rate schedule (Vaswani et al., 2017). The best model was selected using the best validation loss within the same maximum number of epochs. All the performance was measured in `sacreBLEU4` (Post, 2018) using a beam size of 4 and a length penalty of 0.6. Appendix B provides more details of training.

| ID | Data sharing | Model size | Training pairs | Single | 1-1 | M2 |
|---|---|---|---|---|---|---|
| 1 | Non-sharing | Base | M2M(12) | 25.37 | 23.12 (-2.25) | **25.98 (0.61)** |
| 2 | Sharing | Base | M2M(12) | 25.34 | 23.27 (-2.07) | **25.65 (0.31)** |
| 3 | Non-sharing | Large | M2M(12) | 25.43 | 26.90 (1.47) | **27.17 (1.74)** |
| 4* | Non-sharing | Base | JM2M(6) | - | 27.50 (-2.32) | 29.70 (-0.12) |
| 5* | Non-sharing | Base | M2M(12) | 29.82 | 27.66 (-2.16) | **30.42 (0.6)** |

Table 2: Averaged SacreBLEU test scores of single models, 1-1, and M2 trained using a balanced dataset of different configurations. *M2M* indicates the training of full many-to-many directions among languages (12 directions), whereas *JM2M* represents the training of directions that only include English on one side(6 directions). * indicates that the score is averaged only on English-centric.

### 4.2 Balanced environment

We first compared the performance of multi-way directions in a balanced and non-sharing environment, which is the most strictly controlled.

The results are shown in Table 1. The 1-1 model performed worse than both the single models and the M2 in every direction, clearly indicating a capacity bottleneck. In contrast, **the M2 consistently outperformed not only the 1-1 model but also the single models in all directions**. As the M2 cannot benefit from *cross-language effect* due to the lack of a shared module between any languages, we hypothesize that the following two effects are in charge: *data-diversification* and *regularization*. We verify this hypothesis using ablation studies.

Note that the 1-1 model's variation of degradation is higher with target languages than with source languages, even though all the directions are trained using the same amount. The translation to English (-1.96, -1.94, and -1.68) consistently degraded the least, whereas that to French (-2.74, -2.75, and -2.97) degraded the most, given the same source languages. This finding is consistent with previous observations that the capacity bottleneck is more prominent in the decoders (Johnson et al., 2017; Arivazhagan et al., 2019b).

**Ablation** We compare models in a series of conditions (see IDs in Table 2). ① We denote the summarized performance demonstrated in Table 1 for reference. ② To establish whether *data-diversification* was responsible for the performance improvement of the M2, we experimented using fully shared data. ③ To observe the behavior under alleviated capacity constraints, we experimented using bigger models. We used a transformer with a hidden dimension of 512 and a feed-forward dimension of 2048 for our large model. The training settings are the same except for a larger batch size (x4). ④ Finally, we compared the models trained using the JM2M (6 directions instead of 12) to observe the behavior of the models with fewer directions. ⑤ We averaged scores of English-centric directions in ① to compare with ④. Appendix C presents the individual score for each direction.

Table 2 shows the results of each environment. When we completely shared the data(②), the performance gain of the M2 versus that of the single models (0.31) decreased. Given that ② eliminates the chance of *data-diversification*, the degraded performance (0.3) can be attributed to it. However, the fact that the M2 still outperforms the single models (0.31) implies that the M2 can still benefit from the *regularization* effect of multi-way training. The minor increase in performance of 1-1 (0.18) seems to imply that *data-diversification* can be detrimental under the severe capacity bottleneck.

③ shows the performance of a larger model trained using the same data. Single models barely improved with the use of larger models, indicating the absence of a capacity bottleneck. On the contrary, the 1-1 model and the M2 both showed an increase in performance. The 1-1 model exhibits a gain from multi-way training only with enough capacity (1.47). This indicates that the benefit of multi-way training can only be achieved with enough capacity for the 1-1 model. Although the M2 is less affected by capacity bottleneck, the larger capacity is also beneficial for the M2 (1.74) to fully leverage the benefits of multi-way training.

To compare the models trained with JM2M (④), ⑤ shows the score averaged only over directions from and to English ①. The JM2M scheme is likely to have mixed results: there is less pressure from the capacity bottleneck due to fewer training directions. However, possible gains from *data-diversification*

[2] https://github.com/google/sentencepiece
[3] https://github.com/pytorch/fairseq

| Resource | Pairs | 1:1:1 | | 1:2:4 | | 1:5:25 | |
|---|---|---|---|---|---|---|---|
| | | 1-1 | M2 | 1-1 | M2 | 1-1 | M2 |
| High | En-Fi | 16.94 (-2.63) | 20.03 (0.46) | 18.01 (-1.4) | 19.92 (0.51) | 18.66 (-0.75) | 19.82 (0.41) |
| | Fi-En | 27.32 (-1.94) | 30.24 (0.98) | 28.04 (-1.21) | 30.06 (0.81) | 28.51 (-0.74) | 29.9 (0.65) |
| | Avg | 22.13 (-2.28) | 25.14 (0.72) | 23.02 (-1.3) | **24.99 (0.66)** | 23.58 (-0.74) | **24.86 (0.53)** |
| Medium | En-Fr | 32.99 (-2.75) | 36.09 (0.35) | 32.73 (-1.24) | 35.26 (1.29) | 31.61 (1.14) | 33.66 (3.19) |
| | Fr-En | 33.81 (-1.68) | 36.18 (0.69) | 33.73 (-0.23) | 35.39 (1.43) | 33.1 (2.5) | 33.9 (3.3) |
| | Fi-Fr | 22.24 (-2.97) | 25.94 (0.73) | 23.35 (-0.11) | 25.27 (1.81) | 22.6 (3.37) | 24.08 (4.85) |
| | Fr-Fi | 13.6 (-1.82) | 16.15 (0.73) | 14.49 (0.47) | 15.58 (1.56) | 14.43 (3.78) | 14.19 (3.54) |
| | Avg | 25.66 (-2.31) | 28.59 (0.62) | 26.08 (-0.28) | **27.88 (1.52)** | 25.44 (2.7) | **26.46 (3.72)** |
| Low | De-En | 31.04 (-1.96) | 33.51 (0.51) | 30.31 (1.68) | 32.29 (3.66) | 28.45 (17.02) | 27.88 (16.45) |
| | En-De | 23.83 (-2.04) | 26.46 (0.59) | 22.69 (0.9) | 24.78 (2.99) | 18.61 (11.66) | 19.91 (12.96) |
| | De-Fi | 13.08 (-2.12) | 15.93 (0.73) | 13.72 (2.56) | 14.89 (3.73) | 12.76 (10.58) | 11.62 (9.44) |
| | Fi-De | 16.75 (-2.22) | 19.51 (0.54) | 16.8 (2.06) | 18.3 (3.56) | 14.01 (10.99) | 14.25 (11.23) |
| | De-Fr | 25.73 (-2.74) | 29.08 (0.61) | 25.8 (1.45) | 27.6 (3.25) | 23.37 (15.76) | 23.5 (15.89) |
| | Fr-De | 20.09 (-2.14) | 22.64 (0.41) | 19.76 (1.35) | 21.45 (3.04) | 16.18 (10.76) | 16.58 (11.16) |
| | Avg | 21.75 (-2.2) | 24.52 (0.57) | 21.51 (1.67) | **23.22 (3.37)** | 18.9 (12.8) | **18.96 (12.85)** |
| | Total Avg | 23.12 (-2.25) | 25.98 (0.61) | 23.29 (0.52) | **25.07 (2.3)** | 21.86 (7.17) | **22.44 (7.76)** |

Table 3: Test SacreBLEU test scores of single models, 1-1 model, and M2 trained using an unbalanced, completely non-sharing dataset. 1:1:1, 1:2:4, and 1:5:25 represent the ratios of the low, medium, and high resource pairs, respectively. Values in parentheses indicate the performance difference from single models in respective environments.

or *regularization* are also smaller. Both the 1-1 model and the M2 perform better when trained using M2M (⑤) than when trained using JM2M (④). However, the performance difference is more significant in the M2 (0.72) than in the 1-1 model (0.16). We assume that while both models benefit from *data-diversification* and *regularization* accompanied by training using more directions, the capacity bottleneck in 1-1 counterweighs those positive effects.

### 4.3 Unbalanced environment

We also compared the models with unbalanced training data, which is a natural condition in practice. To synthetically create an unbalanced environment, we first divided the pairs into low (De-En, De-Fi, De-Fr), medium (En-Fr, Fi-Fr), and high (En-Fr) resource pairs. Next, we reduced the amount of data for low and medium pairs, setting the ratio of low:medium:high = 1:2:4, and 1:5:25, respectively. The detailed division of the dataset can be found in appendix A. Note that the models learns with fewer data in the unbalanced environment. We first trained the models without up-sampling.

Table 3 shows the scores of the 1-1 model and the M2 in each setting (1:1:1, 1:2:4, 1:5:25). Both models show similar trends with unbalanced data. Compared to the balanced environment, medium and low resource pairs tend to benefit from multi-way training, with gains more prominent for lower resource pairs as the data get more unbalanced (12.8 by the 1-1 model and 12.85 by the M2). Interestingly, **the M2 exhibits a similar level of improvement to that of the 1-1 model in low and medium resource pairs.** Considering the M2 is not subject to the *cross-language transfer*, the performance increase in lower resource pairs may be better explained by *data-diversification* and *regularization*. This indicates that the *cross-language effect* of the 1-1 model may be more subtle than expected.

On the other hand, M2 barely showed the performance degradation in high resource pairs. This implies that the performance boost of low resource pairs and the drop of high resource pairs may not be necessarily trade-off without a capacity bottleneck.

| M | US | High | Medium | Low |
|---|---|---|---|---|
| 1-1 | × | 23.58 (-0.74) | 25.44 (2.7) | 18.9 (12.8) |
| | ○ | 20.5 (-3.83) | 24.31 (1.57) | **19.78 (13.68)** |
| M2 | × | **24.86 (0.53)** | **26.46 (3.72)** | 18.96 (12.85) |
| | ○ | 19.64 (-4.69) | 23.49 (0.75) | 16.88 (10.78) |

Table 4: Averaged test SacreBLEU scores of 1-1 and M2 trained with 1:5:25 dataset with and without up-sampling.

**Ablation** The sampling method in an unbalanced setting is known to affect the performance (Arivazhagan et al., 2019b). We compared two models in the most unbalanced environment (1:5:25) with and without up-sampling.

Table 4 shows the results. As previously reported, we confirm that up-sampling makes the results extreme in the 1-1 model: low resource pairs improve more (from 12.8 to 13.68), whereas

high resource pairs degrade more (from -0.74 to -3.83). On the other hand, up-sampling in the M2 harmed performance in all the low, medium, and high resource pairs. The difference in converge rates among modules may be the cause; models overfit in low-resource pairs, and underfit in high-resource pairs. This is supported by the changes in the M2's performance with more training epochs (Appendix C).

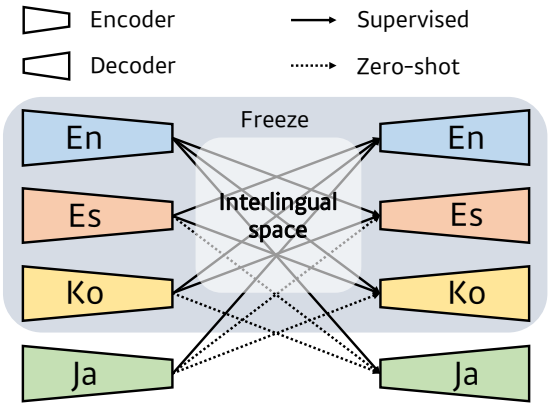


Figure 2: **Interlingual space** formed by the multi-way training of the M2 (En, Es and Ko). While freezing the M2, incrementally training a new language (Ja) with a single parallel corpus (En - Ja) adapt new modules to the interlingual space.

## 5 Interlingual space of the M2

Creating interlingual space has been an active research area (Lu et al., 2018; Sen et al., 2019; Arivazhagan et al., 2019a; Escolano et al., 2019; Vázquez et al., 2019) because it is critical to scaling out languages, such as incremental learning. Because input of M2 does not contain any information regarding the target language, encoders need to encode it so that any decoder can translate. At the same time, decoders of the M2 should be able to generate from output of any M2 encoder. For this reason, we assume that the output space of M2 encoders is interlingual.

Figure 2 illustrates the interlingual space of a M2. Multi-way training of 3 languages (En, Es and Ko) forms the interlingual space which is shared by 6 modules. This space is preserved as long as the weights of the M2 are frozen. Training a new module (Ja) with a single parallel corpus (En - Ja) using one of the frozen modules (En) adapt the module to the interlingual space. We speculate that the new module (Ja) would be compatible with the other modules (Es and Ko) if the interlingual space is formed well.

| ID | Model | En-Fr | Fr-En |
|---|---|---|---|
| 1 | M2(4) + En-Fr | 34.70 | 34.90 |
| 2 | M2(4) + En-Fr *with init* | 34.88 | 34.94 |
| 3 | M2(4) + En,De-Fr | 35.40 | 35.57 |
| 4 | M2(4) + En,De,Es-Fr | 35.41 | 35.70 |
| 5 | M2(4) + En,De,Es,Nl-Fr | **35.47** | **35.92** |
| 6* | Single | 34.48 | 34.11 |
| 7* | M2(5) | 36.24 | 36.35 |

Table 5: SacreBLEU test scores of a single model and incremented modules of the M2. Values in parentheses indicate the number of languages involved in the M2 (4: De, En, Es, Nl; 5: 4 + Fr). + indicates incremental training with the former model frozen. *with init* indicates that the incremented module is initialized using the weight of the English module. * represents the model is trained from scratch and not incrementally.

We verify this using incremental zero-shot learning. Additionally, we measure how the language invariance of the space changes as the number of languages involved in the M2 varies. Since maintainability is one of the critical needs in practice, high performance on incremental learning would be a desirable trait in industrial settings.

### 5.1 Setting

To increase the number of languages, we modified the multi-parallel corpus of Europarl differently. We selected six languages (German, English, Spanish, Finnish, French, and Dutch) and divided a 1.25M multi-parallel corpus into 250K for each direction without sharing. Other details are mostly the same as in former experiments. The detailed division of the dataset and training details can be found in appendix A and B.

### 5.2 Incremental training

We added French to an M2 model trained using all directions among four languages (German, English, Spanish, and Dutch). An additional French encoder and decoder were trained using English-French pairs while the parameters of English modules remained frozen (①). We also tested two methods to help incremental training as follows. 1) Initialize the new module using one of the modules trained using other languages. In the experiment, we used the weights copied from the English module as the initialization for French (②). Note that the English and French module does not share any information, such as embedding. 2) Train the module with

| Model | De-Fr | Fr-De | Es-Fr | Fr-Es | Nl-Fr | Fr-Nl |
|---|---|---|---|---|---|---|
| Pivot | 25.42 | 19.53 | 30.37 | 30.87 | 23.52 | 22.06 |
| 1 | 26.37 | 19.08 | 31.91 | 32.22 | 24.31 | 22.15 |
| 2 | **26.79** | **19.90** | 32.17 | 32.68 | 24.64 | 22.63 |
| 3 | - | - | **32.91** | **33.34** | 25.65 | 23.44 |
| 4 | - | - | - | - | **25.82** | **23.55** |
| 6* | 26.91 | 20.86 | 32.90 | 33.70 | 24.81 | 22.97 |
| 7* | 28.86 | 22.70 | 34.62 | 35.22 | 26.58 | 24.98 |

Table 6: SacreBLEU zero-shot test scores of the English-pivoted single models and incremented modules from Table 5. * means that the model is trained using the supervision of 250 thousand pairs.

auxiliary directions. We incrementally added auxiliary directions of De-Fr (③), Es-Fr (④), and Nl-Fr (⑤). We compared the models with a singly trained model (⑥), and the M2 models trained using five languages from scratch (⑦). ⑦ worked as an upper bound for the incremental training.

Table 5 shows the performance of En-Fr and Fr-En with incremental training. **The incrementally trained model without any additional method (①) outperformed a single model (⑥) even though half of the model was frozen.** This not only indicates that the language-agnostic space is well-formed but also shows that incremented direction can benefit from a well-trained frozen module.

We also found that our two methods are effective in incremental training. Even though French does not share any information with the trained English module, initializing the French module with the weights learned by the English module benefits the performance marginally. Incrementally training the new module using multiple directions helps as the number of directions increases. Note that the two methods can be applied orthogonally. Although none of the incrementally trained models outperform the M2 model trained from scratch, this still shows that simple incremental training for the M2 can be a good alternative for expensive training from scratch.

We examined whether an incremented module in one direction can generalize to the other directions. We compared the zero-shot performance of the models in Table 5 with the English-pivoted translation performance using two single models. We also denoted the supervised performance of single models, and jointly trained the M2 for reference (250K for each direction).

### 5.3 Incremental zero-shot learning

Table 6 shows the zero-shot performance of incrementally trained modules. Amazingly, **most of the incremented modules demonstrated better performance than the English-pivoted translation**. The only exception was in the Fr-De direction of the naively incremented module (①), which seemed to be marginal (-0.45). Our methods for incremental training were also effective for zero-shot performance. The results were even comparable to the single supervised models trained with 250K parallel corpus. This shows that multi-way training creates shared (interlingual) space instead of pair specific space.

### 5.4 The language invariance of the interlingual space

The interlingual space established by the M2 was confounding, considering no additional regularizations or methods were adopted. We measured the language invariance of the interlingual space while the varying the number of languages of the M2 model. We trained a series of M2 models that included 3 - 6 languages (6, 12, 20, and 30 directions) and found that the use of more languages to train the M2 also improved its performance in all directions (appendix D). We investigated with two metrics to measure the language invariance of interlingual space.

**Cosine Similarity** We measured the representation similarity of parallel sentences from a parallel corpus. To obtain the fixed-size representation, we average pooled the output of encoders through the time steps. We averaged the cosine similarity of 10K pairs from the test set.

**Mono-direction translation** When training the M2, mono-direction (where source and target languages are the same) is not trained because mod-

| Model | Cosine Similarity | | | | BLEU Score |
|---|---|---|---|---|---|
| | En-De | En-Nl | De-Nl | Avg | En-En |
| M2(3) | 0.7228 | 0.7062 | 0.7043 | 0.7111 | 75.55 |
| M2(4) | 0.7682 | 0.7425 | 0.7635 | 0.7581 | 82.55 |
| M2(5) | 0.7832 | 0.7603 | 0.7827 | 0.7754 | **83.13** |
| M2(6) | 0.8169 | 0.7905 | 0.8189 | **0.8088** | 82.80 |

Table 7: Cosine text similarity score of encoder outputs and SacreBLEU score of mono-direction translation(En-En). Values in parentheses indicate the number of languages involved in the M2 (3: De, En, Nl; 4: 3 + Es; 5: 4 + Nl; 6: 5 + Fi).

ules tend to learn to simply copy the input, which hinders translation training (Firat et al., 2016a). Meanwhile, interlingual output representation of the encoders should be able to be translated by any decoder, including the decoder of the source language. Therefore, the translation score of mono-direction translation shows how well the information of the source sentence is preserved.

Table 7 shows the cosine similarity and mono-direction translation scores of the M2. As the M2 trains using more languages, the cosine similarity of all three pairs increases, which implies higher language invariance in interlingual space. However, the gain from marginal languages decreases as the number of languages increases. Mono-direction translation scores mostly align with the number of languages except for the M2(6), which degraded a little from M2(5). As a result, we reasonably conclude that the language invariance of the interlingual space improves with more languages.

## 6 Conclusion

In this study, we re-evaluate the M2 model and suggest it as an appropriate choice for multilingual translation in industries. By extensively comparing the single models, 1-1 model, and M2 in varying conditions, we find that the M2 can benefit from multi-way training through *data-diversification* and *regularization* while suffering less from capacity bottlenecks. Additionally, we demonstrate that the M2 can also benefit low resource pairs in an unbalanced environment as a 1-1 model without being subject to *cross-language effect*. Next, we suggest that the M2 model is easily maintainable because of its interlingual space. The interlingual space not only enables incremental training in a simple manner, but also accompanies competitive incremental zero-shot performance. Furthermore, we validate that the language invariance of the space enhances as the number of languages in the M2 increases. We hope that this study sheds light on the relatively disregarded M2 model and provide a benchmark for selecting a model among varying levels of shared components.

## Acknowledgments

The authors of this paper would like to give special thanks to Sang-Woo Lee and Jihyung Moon for their honest and genuine feedback that helped improve the quality of the paper greatly. Also, we are extremely grateful to Jaehyeon Kim, Jaehun Jung and the anonymous reviewers for their suggestions and comments. Finally, we cannot express enough thanks to all of Context Part members of Kakao Enterprise for their continued support and encouragement.

## 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.

Maruan Al-Shedivat and Ankur Parikh. 2019. Consistency by agreement in zero-shot 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 1184–1197.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Roee Aharoni, Melvin Johnson, and Wolfgang Macherey. 2019a. The missing ingredient in zero-shot neural machine translation. *arXiv preprint arXiv:1903.07091*.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019b. Massively multilingual neural

machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548.

Graeme Blackwood, Miguel Ballesteros, and Todd Ward. 2018. Multilingual neural machine translation with task-specific attention. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3112–3122.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2019. A survey of multilingual neural machine translation. *arXiv preprint arXiv:1905.05395*.

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.

Carlos Escolano, Marta R Costa-jussà, and José AR Fonollosa. 2019. From bilingual to multilingual neural machine translation by incremental training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 236–242.

Carlos Escolano, Marta R Costa-jussà, José AR Fonollosa, and Mikel Artetxe. 2020. Multilingual machine translation: Closing the gap between shared and language-specific encoder-decoders. *arXiv preprint arXiv:2004.06575*.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016a. 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.

Orhan Firat, Baskaran Sankaran, Yaser Al-Onaizan, Fatos T Yarman Vural, and Kyunghyun Cho. 2016b. Zero-resource translation with multi-lingual neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 268–277.

Thanh-Le Ha, Jan Niehues, and Alexander Waibel. Toward multilingual neural machine translation with universal encoder and decoder. *Institute for Anthropomatics and Robotics*, 2(17/03):17.

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

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, 5:365–378.

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. *arXiv preprint arXiv:2001.08210*.

Yichao Lu, Phillip Keung, Faisal Ladhak, Vikas Bhardwaj, Shaonan Zhang, and Jason Sun. 2018. A neural interlingua for multilingual machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 84–92.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

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.

Toan Q Nguyen and David Chiang. 2017. Transfer learning across low-resource, related languages for neural machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 296–301.

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 the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9.

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.

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.

Devendra Sachan and Graham Neubig. 2018. Parameter sharing methods for multilingual self-attentional translation models. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 261–271.

Holger Schwenk and Matthijs Douze. 2017. Learning joint multilingual sentence representations with neural machine translation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 157–167.

Sukanta Sen, Kamal Kumar Gupta, Asif Ekbal, and Pushpak Bhattacharyya. 2019. Multilingual unsupervised nmt using shared encoder and language-specific decoders. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3083–3089.

David So, Quoc Le, and Chen Liang. 2019. The evolved transformer. In *International Conference on Machine Learning*, pages 5877–5886.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Xu Tan, Jiale Chen, Di He, Yingce Xia, QIN Tao, and Tie-Yan Liu. 2019a. 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)*, pages 962–972.

| | De | En | Fi | Fr |
|---|---|---|---|---|
| De | - | 1 | 2 | 3 |
| En | - | - | 3 | 2 |
| Fi | - | - | - | 1 |
| Fr | - | - | - | - |

Table 8: Division of multi-parallel parts for each pair in section 4

Xu Tan, Yi Ren, Di He, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2019b. Multilingual neural machine translation with knowledge distillation. In *International Conference on Learning Representations 2019*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Raúl Vázquez, Alessandro Raganato, Jörg Tiedemann, and Mathias Creutz. 2019. Multilingual nmt with a language-independent attention bridge. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 33–39.

Xinyi Wang, Hieu Pham, Philip Arthur, and Graham Neubig. 2019. Multilingual neural machine translation with soft decoupled encoding. In *International Conference on Learning Representations 2019*.

Poorya Zaremoodi, Wray Buntine, and Gholamreza Haffari. 2018. Adaptive knowledge sharing in multi-task learning: Improving low-resource neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 656–661.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. *arXiv preprint arXiv:2004.11867*.

Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 30–34.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575.

# A Dataset

## A.1 Division of multi-parallel dataset

In order to create completely non-sharing dataset and make the best use of multi-parallel corpus, we divide the 1.5K multi-parallel corpus into 3

| | De | En | Es | Fi | Fr | Nl |
|---|---|---|---|---|---|---|
| De | - | 1 | 2 | 3 | 4 | 5 |
| En | - | - | 3 | 4 | 5 | 2 |
| Es | - | - | - | 5 | 1 | 4 |
| Fi | - | - | - | - | 2 | 1 |
| Fr | - | - | - | - | - | 3 |
| Nl | - | - | - | - | - | - |

Table 9: Division of multi-parallel parts for each pair in section 5

| Resource | Pairs | 1:1:1 | 1:2:4 | 1:5:25 |
|---|---|---|---|---|
| High | En-Fi | 500K | 500K | 500K |
| Medium | En-Fr | 500K | 250K | 100K |
| | Fi-Fr | 500K | 250K | 100K |
| Low | De-En | 500K | 125K | 20K |
| | De-Fi | 500K | 125K | 20K |
| | De-Fr | 500K | 125K | 20K |

Table 10: The amount of data for each pair in section 4

parts(500K) for section 4 and 5 parts(250K) for section 5. And then, we assigned the parts to pairs so that no two directions of the same side share the same part. The assignment for section 4 and 5 are stated in table 8 and 9 respectively. Validation and test are divided with the same manner. For complete-sharing dataset, training data for all pairs only created from part 1. However, validation and test set remain the same with completely non-sharing dataset.

### A.2 Amount of data for each pairs

In order to create unbalanced environment in section 4, we limited the amount of data for some directions. Table 10 shows the amount of the data for each pair in balanced, and unbalanced environments in section 4. For section 5, the amounts of all directions are the same with 250K. All the validation and test set are the same with 10K.

Though our dataset can easily be reconstructed from the open dataset (Europarl) with described process, we also made our dataset available online[4] for convenience of readers. We only uploaded the dataset of the balanced environment since unbalanced environment can be made from them trivially. The dataset is binarized with `fairseq-preprocess` command of `fairseq` framework.

[4] `https://drive.google.com/file/d/1CmSzFI6h2cGYJshUWEPkF7Hx4UcL3DVl`

## B Training detail

### B.1 Batch size

We selected the batch size of 6144 max tokens with the best validation loss of a single model (En-De) among {1536, 3072, 6144, 12288, 24576} max tokens per GPU (4 GPUs). While the total number of parameters and the training directions is different among single model, 1-1 and M2, we set the batch size for each direction so that each module learns with the same batch size (6144 tokens). Specifically, one step of a single model includes a single direction, while that of 1-1(4) and M2(4) includes 12 directions. However, training directions per module between 1-1(4) and M2(4) is different with 12 and 3 directions. Therefore, the batch size per direction of 1-1 is 512 (1/12 of 6144) and that of M2 is 1536 (1/4 of 6144). Since we accumulate the gradients of all directions, all the compared modules learn with the same batch size of data.

### B.2 Sampling

To train balanced data, we used round robin scheduling of all directions. We compared two sampling methods in ablation of unbalanced environment: up-sampling and proportional sampling. Round robin scheduling is equivalent to up-sampling low-resource data in unbalanced environment. For efficient proportional sampling, we sampled several small batches of pairs proportional to the amount of total pairs. We accumulated gradients of several batches to make expected batch-size of each module to meet the total batch size.

### B.3 Early stopping

Since fixing the maximum tokens of a batch per module results in different step size among models, we stopped the training of models based on the maximum number of epochs. All the best models were chosen based on the best validation loss (averaged) within 100 epochs.

## C Detailed scores of ablations

This section provides detailed scores of the ablation part of the section 4 and 5.

Table 11 shows detailed scores under complete sharing (② of table 2) and increased capacity (③ of table 2). Table 12 shows detailed scores under JM2M(④ of table 2) and M2M(⑤ of table 2) training.

Table 13 shows detailed scores of the models under proportional sampling and up-sampling in table

| Pairs | Sharing | | | Large | | |
|---|---|---|---|---|---|---|
| | Single | 1-1 | M2 | Single | 1-1 | M2 |
| De-En | 32.86 | 30.61 (-2.25) | 33.14 (0.28) | 32.84 | 34.5 (1.66) | 34.59 (1.75) |
| De-Fi | 15.04 | 13.67 (-1.37) | 15.45 (0.41) | 15.21 | 16.96 (1.75) | 17.15 (1.94 |
| De-Fr | 28.27 | 26.39 (-1.88) | 28.93 (0.66) | 28.52 | 29.84 (1.32) | 30.24 (1.72) |
| En-De | 25.98 | 23.57 (-2.41) | 26.15 (0.17) | 25.97 | 27.23 (1.26) | 27.57 (1.6) |
| En-Fi | 19.52 | 17.19 (-2.33) | 19.63 (0.11) | 19.49 | 21.26 (1.77) | 21.39 (1.9) |
| En-Fr | 35.51 | 32.48 (-3.03) | 35.65 (0.14) | 35.65 | 36.39 (0.74) | 37.14 (1.49) |
| Fi-De | 18.63 | 17.33 (-1.3) | 19.38 (0.75) | 19.07 | 20.55 (1.48) | 20.88 (1.81) |
| Fi-En | 29.25 | 27.09 (-2.16) | 29.89 (0.64) | 29.39 | 31.35 (1.96) | 31.57 (2.18) |
| Fi-Fr | 25.53 | 23.18 (-2.35) | 25.78 (0.25) | 25.45 | 26.84 (1.39) | 27.39 (1.94) |
| Fr-De | 22.18 | 20.58 (-1.6) | 22.35 (0.17) | 22.28 | 23.73 (1.45) | 23.85 (1.57) |
| Fr-En | 35.58 | 33.2 (-2.38) | 35.63 (0.05) | 35.78 | 36.89 (1.11) | 37.12 (1.34) |
| Fr-Fi | 15.70 | 13.99 (-1.71) | 15.85 (0.15) | 15.55 | 17.28 (1.73) | 17.17 (1.62) |
| Avg | 25.34 | 23.27 (-2.06) | 25.65 (0.32) | 25.43 | 26.9 (1.47) | 27.17 (1.74) |

Table 11: Detailed scores of ② and ③ in table 2

| Pairs | Single | JM2M | | M2M | |
|---|---|---|---|---|---|
| | | 1-1 | M2 | 1-1 | M2 |
| De-En | 33.00 | 30.93 (-2.07) | 32.55 (-0.45) | 31.04 (-1.96) | 33.51 (0.51) |
| Fi-En | 29.26 | 27.18 (-2.08) | 29.08 (-0.18) | 27.32 (-1.94) | 30.24 (0.98) |
| Fr-En | 35.49 | 33.84 (-1.65) | 35.6 (0.11) | 33.81 (-1.68) | 36.18 (0.69) |
| En-De | 25.87 | 23.6 (-2.27) | 25.9 (0.03) | 23.83 (-2.04) | 26.46 (0.59) |
| En-Fi | 19.57 | 16.6 (-2.97) | 19.32 (-0.25) | 16.94 (-2.63) | 20.03 (0.46) |
| En-Fr | 35.74 | 32.86 (-2.88) | 35.77 (0.03) | 32.99 (-2.75) | 36.09 (0.35) |
| Avg | 29.82 | 27.5 (-2.32) | 29.7 (-0.12) | 27.66 (-2.17) | 30.42 (0.6) |

Table 12: Detailed scores of ④ and ⑤ in table 2

| Resource | Pairs | Proportional sampling | | Up-sampling | | |
|---|---|---|---|---|---|---|
| | | 1-1 | M2 | 1-1 | M2 | M2(+10) |
| High | En-Fi | 18.66 (-0.75) | 19.82 (0.41) | 15.7 (-3.71) | 14.76 (-4.65) | 16.86 (-2.55) |
| | Fi-En | 28.51 (-0.74) | 29.9 (0.65) | 25.3 (-3.95) | 24.52 (-4.73) | 26.81 (-2.44) |
| | Avg | 23.58 (-0.74) | 24.86 (0.53) | 20.5 (-3.83) | 19.64 (-4.69) | 21.84 (-2.5) |
| Medium | En-Fr | 31.61 (1.14) | 33.66 (3.19) | 31.27 (0.8) | 30.79 (0.32) | 32.72 (2.25) |
| | Fr-En | 33.1 (2.5) | 33.9 (3.3) | 31.75 (1.15) | 30.98 (0.38) | 32.39 (1.79) |
| | Fi-Fr | 22.6 (3.37) | 24.08 (4.85) | 21.54 (2.31) | 20.64 (1.41) | 22.43 (3.2) |
| | Fr-Fi | 14.43 (3.78) | 14.19 (3.54) | 12.67 (2.02) | 11.54 (0.89) | 12.86 (2.21) |
| | Avg | 25.44 (2.7) | 26.46 (3.72) | 24.31 (1.57) | 23.49 (0.75) | 25.1 (2.36) |
| Low | Ee-En | 28.45 (17.02) | 27.88 (16.45) | 28.31 (16.88) | 24.5 (13.07) | 24.24 (12.81) |
| | En-De | 18.61 (11.66) | 19.91 (12.96) | 21.03 (14.08) | 18.61 (11.66) | 18.32 (11.37) |
| | Ee-Fi | 12.76 (10.58) | 11.62 (9.44) | 11.8 (9.62) | 9.21 (7.03) | 9.31 (7.13) |
| | Fi-De | 14.01 (10.99) | 14.25 (11.23) | 15.06 (12.04) | 12.6 (9.58) | 12.27 (9.25) |
| | De-Fr | 23.37 (15.76) | 23.5 (15.89) | 24.34 (16.73) | 20.81 (13.2) | 20.27 (12.66) |
| | Fr-De | 16.18 (10.76) | 16.58 (11.16) | 18.12 (12.7) | 15.56 (10.14) | 14.94 (9.52) |
| | Avg | 18.9 (12.8) | 18.96 (12.85) | 19.78 (13.68) | 16.88 (10.78) | 16.56 (10.46) |
| | Total Avg | 21.86 (7.17) | 22.44 (7.76) | 21.41 (6.72) | 19.54 (4.86) | 20.28 (5.6) |

Table 13: Detailed scores of models of table 4. M2(+10) indicates the selected best model trained with addtional 10 epochs.

| Pair | M2(3) | M2(4) | M2(5) | M2(6) |
|---|---|---|---|---|
| De-En | 32.33 | 32.96 | 33.20 | 33.53 |
| En-De | 25.52 | 25.75 | 26.16 | 26.15 |
| De-Nl | 25.10 | 25.49 | 25.34 | 25.60 |
| Nl-De | 21.32 | 21.56 | 21.55 | 21.71 |
| En-Nl | 27.17 | 27.39 | 27.65 | 27.77 |
| Nl-En | 29.53 | 29.94 | 30.27 | 30.43 |
| Avg | 26.83 | 27.18 | 27.36 | 27.53 |

Table 14: Detailed scores of the models in 7

4. M2(+10) indictes the scores of the M2 trained 10 epochs after the best validation loss. M2(+10) shows the increased performance in medium and high resource pairs and degradation in low resource pairs. This indicates that up-sampling causes the difference in converge rates among pairs of different resources for M2.

## D Detailed scores of M2 with varying languages

Table 14 shows detailed scores of M2 trained with varying number of languages. This shows that M2 trained with more languages shows better performance.