Extending Multilingual Machine Translation through Imitation Learning

Anonymous ACL submission

Abstract

Despite the growing variety of languages supported by existing multilingual neural machine translation (MNMT) models, most of the world's languages are still being left behind. We aim to extend large-scale MNMT models to a new language, allowing for translation between the newly added and all of the already supported languages in a challenging scenario: using only a parallel corpus between the new language and English. Previous approaches, such as continued training on parallel data including the new language, suffer from catastrophic forgetting (i.e., performance on other languages is reduced). Our novel approach **Imit-MNMT** treats the task as an imitation learning process, which mimicks the behavior of an expert, a technique widely used in the computer vision area, but not well explored in NLP. More specifically, we construct a pseudo multi-parallel corpus of the new and the original languages by pivoting through English, and imitate the output distribution of the original MNMT model. Extensive experiments show that our approach significantly improves the translation performance between the new and the original languages, without severe catastrophic forgetting. We also demonstrate that our approach is capable of solving the copy and off-target problems, which are two common issues in current large-scale MNMT models.

1 Introduction

Recent advancements in multilingual machine translation (MNMT) have marked a significant leap towards supporting a large number of languages in a single model. For example, the *m2m_100* model (Fan et al., 2021) supports the translation between 100 languages and the *nllb* model (Costa-jussà et al., 2022) even supports translation for over 200 languages. However, there are currently around 7,000 languages spoken in the world[1] and the majority of language pairs suffer from a scarcity of resources required for training machine translation models. How to extend the existing MNMT models is a significant problem. Thus motivated, we raise the following two research questions:

Q1: Can we extend the existing MNMT models to a new language using only parallel data between the new language and English?

Q2: If yes, can the extended MNMT model achieve performance improvements in the new language pairs, while preserving the performance of the original pairs?

Existing methods for extending MNMT models can be divided into three groups. i) Continuing the training process with as much of the available corpus as possible (Costa-jussà et al., 2022; Ebrahimi and Kann, 2021; Berard, 2021). ii) Extending (Ko

[1] https://www.ethnologue.com

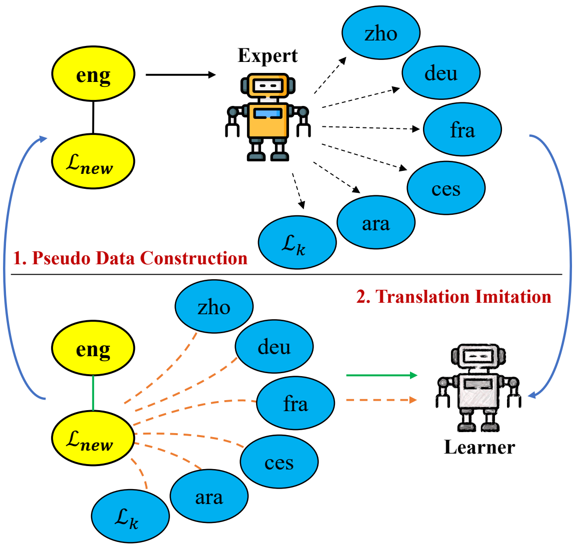


Figure 1: Our proposed framework for extending MNMT models using only a parallel dataset between the new language and English. Given an expert MNMT model (i.e., standard MNMT model which we download and it does not know the new language), we treat the extension task as a process of imitation learning, by cloning the behavior of the expert model on a pseudo multi-parallel corpus.

et al., 2021) or substituting the vocabulary (Garcia et al., 2021) from the new languages. iii) Introducing additional small layers in existing MNMT models to train additional parameters that adapt to new languages (Marchisio et al., 2022; Pfeiffer et al., 2021; Artetxe et al., 2020). Although promising, i) and ii) improve the performance of the new language at the expense of the performance of the original language pairs, while iii) increases the number of model parameters with each language added, which limits its applicability.

To address the aforementioned challenges and tackle the two research questions, we aim to leverage imitation learning to extend an MNMT model in a challenging scenario, i.e., using only a parallel corpus between the new language and English. Imitation learning, also known as "learning from demonstrations" (Hussein et al., 2017), aims to mimic the behavior of an expert from its demonstrations, and has been shown to be effective in various research areas, including robot learning (Fang et al., 2019) and computer vision (Qin et al., 2022). However, there is less work applying it to NLP tasks directly (Shi et al., 2022; Yao et al., 2020) due to its reliance on large vocabularies, which poses challenges for large-scale models. As depicted in Figure 1, the framework we propose can be intuitively divided into two distinct parts. a) Given an expert MNMT model (i.e., standard MNMT model which we download and it does not know the new language), we randomly select k languages that are already supported by the expert MNMT model. We then translate the English side of the parallel corpus to the k languages using the expert model and beam search (Freitag and Al-Onaizan, 2017), resulting in a pseudo multi-parallel corpus including the new language. b) The learner model is trained to mimic the translation behavior of the expert between English and the k languages, however we use the new language side of the gold parallel data instead of English, thus learning translation between the new language and the k selected languages. Additionally, we weight the importance of the k languages based on the expert's performance on them. Note that our approach differs from other machine translation models that use a pseudo-corpus in that our approach uses separate expert and learner models for pseudo-corpus generation and model parameter updating. Our experiments show that the use of separate expert and learner models is crucial to avoid catastrophic forgetting and achieve good learning performance on the new languages.

In summary, we make the following contributions: i) We present a novel framework **Imit-MNMT** which allows large-scale MNMT models to be extended to a new language. ii) Experiments on 8 new languages show that our method improve the performance of the new languages, while maintaining the performance of the original language pairs. iii) We demonstrate that our proposed method can be seen as a promising solution for addressing the common copy problem (Liu et al., 2021) and off-target problem (Zhang et al., 2020) in MNMT models. iv) To the best of our knowledge, this is the first work that extends the MNMT model using imitation learning.

2 Background and Related Work

2.1 Multilingual Machine Translation

MNMT learns a many-to-many mapping function to translate from one language to another (Johnson et al., 2017). With the rapid advancement of computational resources, the development of MNMT has experienced significant leaps and bounds across three distinct levels. Firstly, there has been a gradual shift from English-centric models (Johnson et al., 2017) to models that prioritize non-English languages (Fan et al., 2021). Secondly, MNMT has progressed from supporting translation of dozens (Liu et al., 2020) to enabling translation of hundreds of languages (Fan et al., 2021; Costa-jussà et al., 2022). Lastly, the number of model parameters employed in MNMT has expanded from millions (Liu et al., 2020) to billions (Fan et al., 2021; Costa-jussà et al., 2022), indicating a substantial increase in capacity and capability.

Given L languages, a MNMT model supports translation between $L \times (L-1)$ language pairs. Our goal is to extend the MNMT model to a new language, so that it supports the translation between $(L+1) \times L$ language pairs. Furthermore, current approaches aimed at expanding MNMT suffer from severe catastrophic forgetting. Hence, another goal of our method is to maintain the performance of the original $L \times (L-1)$ language pairs.

2.2 Imitation Learning

Imitation learning (i.e., learning from expert demonstrations), is a method that allows a learner to make decisions as intelligently as an expert. Behavior cloning (BAIN, 1995) and inverse reinforcement learning (NG, 2000), as the two representative approaches of imitation learning, have proved to be effective in several areas. The former attempts to minimize the action differences between the learner and the expert, while the latter mimics the expert behavior by constructing and maximizing an adversarial reward function. Various variants of imitation learning were developed based on the ideas of these two algorithms (Ho and Ermon, 2016; Brantley et al., 2020). While imitation learning and knowledge distillation (Gou et al., 2021) share similarities, they are fundamentally distinct concepts. The former focuses on learning from observed behavior, while the latter focuses on transferring knowledge from a well-trained model to a smaller model. We opt for imitation learning over knowledge distillation because our goal is to extend the MNMT model while maintaining the translation performance of the original language pairs, which aligns with the objective of imitation learning.

2.3 Improving NMT using Synthetic Data

The use of synthetic data to enhance machine translation performance has been widely used, especially in low-resource language scenarios. Among them, back translation (Sennrich et al., 2016) is the earliest and most successful approach. Subsequently, more and more approaches investigate how to utilize synthetic data more effectively to enhance machine translation performance, such as unsupervised machine translation (Lample et al., 2018) and more efficient back translation (Niu et al., 2018; Xu et al., 2022). Although effective, these methods typically perform the generation of pseudo-data and the enhancement of NMT models jointly with a single model, in an *On-the-Fly* manner. This makes it difficult to ensure the quality of pseudo-data generated by the model with updated parameters, and can easily cause interference with the original model. In contrast, our method separates the generation of pseudo-corpora and the updating of model parameters into separate expert models and learner models and integrates them into the imitation learning process, effectively blocking noise in the pseudo-corpora from damaging the learner model.

3 Method

Imit-MNMT contains two parts which are applied iteratively: pseudo multi-parallel data construction (Section 3.1) and imitation learning (Section 3.2). The imitation learning process can further be divided into language weighting (Section 3.2.1) and translation behavior imitation (Section 3.2.2), respectively. Algorithm 1 shows the complete algorithm of Imit-MNMT.

Algorithm 1 Imit-MNMT

Input: Expert MNMT model π^{E}; original languages L; Parallel data $\mathcal{D}_{\ell_{new}}^{\ell_{eng}}$

1: initialize $\pi = \pi^{\mathrm{E}}$
2: **while** not converged **do**
3: $\quad \mathscr{L}_{k-lang}$ =uniform($|L|$)
4: $\quad$ Construct pseudo detaset $\hat{\mathcal{D}}_{\ell_{new}}^{\mathscr{L}_{k-lang}}$ (1)
5: $\quad$ Minimize $\mathcal{L}_{total}$ (5)
6: **end while**
7: **return** Learner model π

3.1 Online Pseudo Multi-Parallel Data Construction

English, as the most resourceful language in the world, often has an easily accessible parallel corpus with other languages. Therefore, our scenario is to extend the MNMT model using only the parallel corpus between the new language and English. As a foundation for imitation learning, we first construct multi-parallel data between the new language and the original languages in an online mode.

Given a parallel corpus $\mathcal{D}_{\ell_{new}}^{\ell_{eng}}$ between a new language ℓ_{new} and English ℓ_{eng}, we randomly select k languages ($\mathscr{L}_{k-lang}$) already supported by the expert MNMT model to construct a pseudo k-way parallel dataset $\hat{\mathcal{D}}_{\ell_{new}}^{\mathscr{L}_{k-lang}} = \{\hat{\mathcal{D}}_{\ell_{new}}^{\ell_k} : k \in \mathscr{L}_{k-lang}\}$ between the new language and the k languages, utilizing beam search from the MNMT model. More specifically, for a parallel sentence pair $(X^{\ell_{new}}, X^{\ell_{eng}}) \in \mathcal{D}_{\ell_{new}}^{\ell_{eng}}$, we generate pseudo parallel sentences by using the English sentences and the expert model. The construction process of $\hat{\mathcal{D}}_{\ell_{new}}^{\ell_k}$ can be formulated as:

$$\bigcup_{\boldsymbol{x}^{\ell_{new}}|\boldsymbol{x}^{\ell_{eng}} \in \mathcal{D}_{\ell_{new}}^{\ell_{eng}}} gen\left(\pi^{E}, \boldsymbol{x}^{\ell_{eng}}, \ell_k\right) \quad (1)$$

where $gen(\cdot)$ is the beam search function and π^E denotes the parameters of the expert model. Note that the parameters of π^E are not updated during the generation process. The k languages are resampled in each batch.

3.2 Extending MNMT as an Imitation Game

After constructing the pseudo k-way parallel data, an intuitive idea is to use this data to update the parameters of the expert MNMT model. This is known as the *On-the-Fly* approach, which involves using the same model to construct the pseudo-corpus and updating the parameters. However, this approach faces the following challenges: i) The pseudo corpus introduces noise that has a significant impact on the training process, particularly when dealing with low-resource languages. Related experiments can be found in Figure 2 and the results will be discussed in Section 5. ii) Similarly, the introduction of noisy data directly affects the representation of the selected k original languages in the MNMT model, leading to a substantial impact on the performance of the original language pairs. Our results regarding this aspect can be found in Figure 3 and will also be discussed in Section 5. To mitigate these challenges, we treat the original MNMT model as an expert and keep it frozen, while we train a separate learner model with the ultimate objective of acquiring the capability to translate between the new language and the original languages[2] by weighting the language (Section 3.2.1) and mimicking translation behavior (Section 3.2.2) of the expert model.

3.2.1 Language Weighting

The expert MNMT model is trained on a set of parallel corpora consisting of multiple language pairs. However, the sizes of these corpora are imbalanced, which leads to poor performance on some languages. To account for this in the learner model, we reduce the importance of low performing languages during imitation learning.

In general, we assume the importance of a given language during training is closely aligned with the performance of the expert model on it. We assume that language pairs demonstrating exceptional performance in the expert MNMT model also yield good quality pseudo data when the source or target side is replaced with the new language that is being added to the MNMT model (and vice versa for low performing language pairs). To accomplish this, we compute the BLEU score of the expert model for each original language paired with English using the FLORES-101 devtest dataset (Goyal et al., 2022). Subsequently, we assign a higher weight to those original languages which have superior BLEU score, thereby emulating their data distribution in the expert model during the training process of the learner.

More specifically, the weight of a non-English language ℓ_t can be calculated as:

$$W\left(\ell_t\right)=\frac{B\left(\ell_{eng}, \ell_t\right)}{\sum_{i=1}^{k} B\left(\ell_{eng}, \ell_i\right)} \cdot k \qquad (2)$$

where $B\left(\ell_s, \ell_t\right)$ is the BLEU score for language pair from ℓ_s to ℓ_t. The weight distribution is used in the next step.

3.2.2 Translation Behavior Imitation

Given an expert MNMT model π^{E} that supports translation between L languages, our goal is to imitate its behaviour and train a new learner model π that supports translation between a new language ℓ_{new} and the L original languages. Our training objective consists of two parts: i) training π on the gold $\mathcal{D}_{\ell_{new}}^{\ell_{eng}}$ and ii) imitating π^{E} on the pseudo $\hat{\mathcal{D}}_{\ell_{new}}^{\mathcal{L}_{k-lang}}$, thus we define two cross-entropy loss functions $\mathcal{L}_{gold}\left(\ell_1, \ell_2\right)$ as:

$$\mathbb{E}_{\boldsymbol{y} \mid \boldsymbol{x} \sim \mathcal{D}_{\ell_1}^{\ell_2}}\left[\sum_{t=1}^{T}-\log \pi\left(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x}, \ell_1, \ell_2\right)\right] \qquad (3)$$

and $\mathcal{L}_{imit}\left(\ell_1, \ell_2\right)$ as:

$$\mathbb{E}_{\hat{\boldsymbol{y}} \mid \boldsymbol{x} \sim \hat{\mathcal{D}}_{\ell_1}^{\ell_2}}\left[\sum_{t=1}^{T}-\log \pi\left(\hat{y}_t \mid \hat{\boldsymbol{y}}_{<t}, \boldsymbol{x}, \ell_1, \ell_2\right)\right] \qquad (4)$$

respectively. Where t indicates a time-step during imitation learning.

Given parallel data with a new language and English, we define the overall training objective for extended model trained from new language to the original languages as:

$$\begin{aligned}\mathcal{L}_{total} = & \mathcal{L}_{gold}\left(\ell_{new}, \ell_{eng}\right)+ \\ & \sum_{i=1}^{k} W\left(\ell_k\right) \cdot \mathcal{L}_{imit}\left(\ell_{new}, \ell_k\right)\end{aligned} \qquad (5)$$

The objective function when trained in the reverse direction can be defined similarly.

4 Experiments

Datasets. We experiment with the following new languages[3]: Akan (aka), Dinka (dik), Bambara (bam), Chokwe (cjk), Dyula (dyu), Balinese (ban),

[2] We consider two directions: either train the extended model from the new to the original languages or train the extended model from the original languages to the new language.

[3] We use ISO 639-2 language codes: https://en.wikipedia.org/wiki/List_of_ISO_639-2_codes

Bemba (bem) and Banjar (bjn). All training data is taken from the mined *nllb* dataset[4] provided by *AllenNLP*. We filter out sentences longer than 120 tokens and preprocess all data using sentencepiece (Kudo and Richardson, 2018). More details of the data can be found in Appendix A.

Baselines. We compare our method to the following baselines. i) **m2m_100**: Using the original *m2m_100* model (Fan et al., 2021). ii) **Finetune**: Fine-tuning m2m_100 model on the parallel data between new language and English. iii) **Extend_Vocab**: Extending the vocabulary of the original m2m_100 model with tokens of the new language, then continue training using the same data as for *Finetune* (Wang et al., 2020). iv) **Adapter**: Train an additional language-specific layer for the new language (Philip et al., 2020). v) **On-the-Fly**: We use the same pseudo parallel data as our method to implement an *On-the-Fly* finetuning on the m2m_100 model. Compared to our method, it uses a single model as both expert and learner, while our method uses two separate models (keeping the expert fixed).

Implementation. We use the m2m_100 model as the basis of the baselines and Imit-MNMT, released in the HuggingFace repository (Wolf et al., 2020). For *Adapter* training, we use the implementation from (Lai et al., 2022). We implemented *Extend_Vocab* based on Wang et al. (2020); To ensure a fair comparison, we maintained a consistent vocabulary size of 23,288 and the extended model size of 507.75 MB for each new language. This size is 23.53 MB larger than the original m2m_100 model. For *On-the-Fly* method, we use the same setting ($k = 5$ and $k = 10$) as our proposed method. It is worth to highlight that both the *Adapter* and *Extend_Vocab* baselines introduce additional parameters to the original m2m_100 model. More details of the model configuration can be found in Appendix B.

Evaluation. We measure case-sensitive detokenized BLEU with SacreBLEU[5] (Post, 2018). Recently, the BLEU score was criticized as an unreliable automatic metric (Kocmi et al., 2021; Zerva et al., 2022). Therefore, we also evaluate our approach using chrF++ (Popović, 2017). The corresponding chrF++ scores are shown in Appendix E. Inspired by Mohammadshahi et al. (2022), we split the languages based on the amount of available training sentences aligned with English into 3 different categories: Low(L), Mid(M) and High(H). All results are evaluated on the FLORES-200 benchmark[6].

5 Results

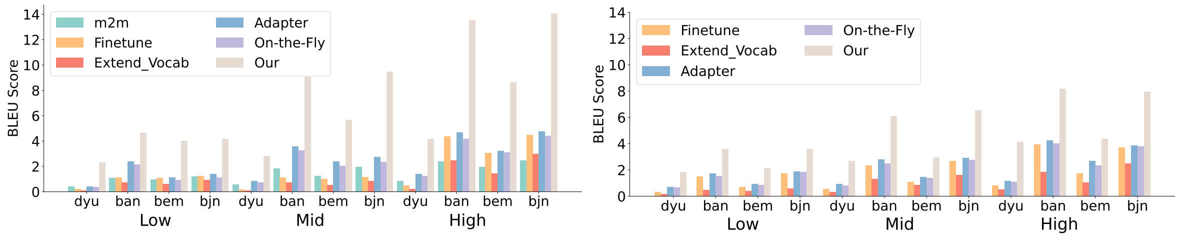
Figure 2 and 3 present the corresponding answers to the two research questions defined in Section 1. For the results of all language pairs used in the experiments, please refer to Table 6 and 7, Table 8 and 9 in the Appendix E.

Q1: successfully extending to a new language

Baselines. *Finetune*, *Adapter* and *On-the-Fly* methods demonstrate certain improvements over the original m2m_100 model, but *extend_vocab* significantly underperforms in comparison to the original m2m_100 model. This discrepancy arises due to the insufficient integration of the newly extended vocabulary into the original tokenizer, a conclusion also highlighted in (Ebrahimi and Kann, 2021). Interestingly, we observe that *On-the-Fly* exhibited inferior performance compared to both *Finetune* and *Adapter*. We hypothesize that this discrepancy arises from the fact that when the performance of the selected language pair is poor, the quality of the corresponding generated pseudo-corpus also suffers. Consequently, updating both the already trained parameters and the pseudo-corpus adversely impacts the overall performance. Our approach outperforms all baselines in both translation directions, achieving the best performance. For instance, when compared to the strongest baseline, *Adapter*, our extended model trained from the new language to the original languages exhibited an average improvement of 3.28. The improvement was 2.12 in the reverse direction.

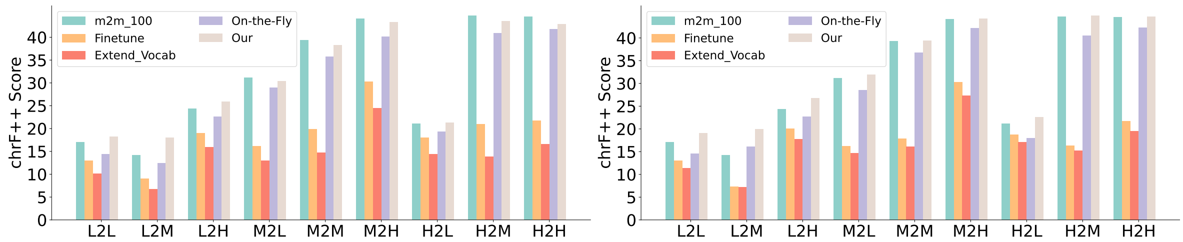
Training directions. Comparing (a) and (b) in Figure 2, we find that the translation performance when training from the new language to the original language is better than in the opposite direction. There are two reasons for this phenomenon: Firstly, decoding the new language does not perform as well as decoding the original language. Secondly, the new language is not well represented in the subword vocabulary, leading to a high frequency of unknown tokens (UNKs) when the fine-grained subword model is used for generating the new language (Pfeiffer et al., 2021). We believe that the first reason is the main factor. He et al. (2019)

[4] https://huggingface.co/datasets/allenai/nllb
[5] https://github.com/mjpost/sacrebleu
[6] https://github.com/facebookresearch/flores/tree/main/flores200

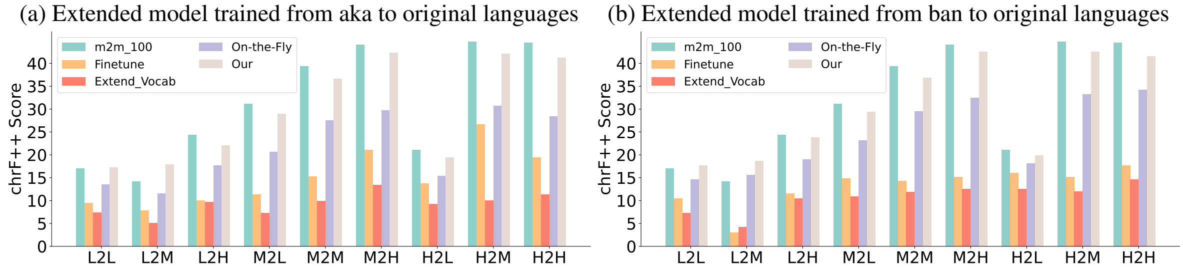


(a) Translation from new languages to original languages (b) Translation from original languages to new languages

Figure 2: **Main Results (the answer of Q1)**: Average BLEU scores for different categories in two directions on the FLORES-200 benchmark. The original languages in (a) and (b) indicates the languages already supported in m2m_100. We do not include the results of m2m_100 method in (b), because the original m2m_100 model does not support the translation from original languages to new languages.



(a) Extended model trained from aka to original languages (b) Extended model trained from ban to original languages



(c) Extended model trained from original languages to aka (d) Extended model trained from original languages to ban

Figure 3: **Main Results (the answer of Q2)**: chrF++ scores of the extended model on 9 original language pairs grouped by available resources on both source and target sizes (**L**ow, **M**id, **H**igh). In each language pair classification, we random select two example pairs and we show the average chrF++ scores. (a) and (b) evaluate the extended model trained from the new language to the original languages. (c) and (d) evaluate the extended model trained from the original languages to the new language.

discovered that the decoder of machine translation is more sensitive to noisy inputs than the encoder. Compared to the original language, the new language has less data available, making it difficult to train a proper decoder from scratch. This can introduce additional noise into the decoder during machine translation, negatively impacting overall translation performance. This problem is further illustrated in Table 1. However, translation from the new language to the original language does not suffer from these issues, because the new language can share some vocabulary with the original language on the source-side.

Corpora sizes. Our investigation reveals that our method demonstrates enhanced effectiveness when applied to larger corpora. Notably, the translation performance in the *bjn* language exhibits the most significant improvement in both translation directions, outperforming other baselines by a considerable margin. The reason behind this observation is that with a larger corpus, the model can be trained more extensively, allowing it to reach a sufficient level of proficiency. On the other hand, when the corpus is small, the model may converge prematurely, leading to suboptimal performance.

Different language categories. We observe that our approach achieves better performance in language pairs where the original language is a high-resource language, compared to language pairs involving low- and mid-resource languages. This discrepancy can be attributed to the superior performance of the original m2m_100 model when translating between high-resource languages and English. As a result, our approach can effectively imitate high-quality translations between high-resource languages and new languages. Conversely, the original m2m_100 model exhibits poor performance when translating between low-resource languages and English, with BLEU scores mostly below 5. Consequently, when attempting to imitate translations to a new language using these low-resource languages, the presence of significant noise leads to even worse translation quality.

Q2: avoiding catastrophic forgetting

Baselines. Figure 3 shows that all baselines suffered from severe catastrophic forgetting, wherein the training process prioritized adaptation to the new language at the expense of the original language pair. Consequently, the performance on the original language pairs deteriorated significantly.

Impact of extended model performance. As depicted in Table 6 and 7, the performance of the model extended with the *ban* language outperforms the performance of the model extended with *aka*. By comparing (a) and (b) as well as (c) and (d) in Figure 3, we find that the extended model for the *ban* language has a smaller impact on the original language pairs compared to the extended model for the *aka* language. For instance, when comparing (a) and (b) in the *eng2srp* language pair, the *aka* extended model performs -1.92 lower, whereas it achieves an increase of $+0.37$ in case of *ban* (Please refer to Table 8 and 9 for detailed scores). This observation can be attributed to language transfer in the MNMT model. When the extended model demonstrates good performance, it indicates a stronger integration of the new language into the MNMT model and an enhanced ability to transfer between the original languages. As a result, the extended model with improved performance has a smaller impact on the original language pair. For instance, as shown in Figure 3, (b) performs significantly better than (a), and similarly, (d) outperforms (c).

Training directions. Comparing (a) and (c), as well as (b) and (d) in Figure 3, it becomes apparent that extending the source side yields better results compared to extending the target side. For example, in the case of the *eng2deu* language pair, the extended model trained from the new language to the original language has a smaller impact compared to the extended model trained in the reverse direction. This is evident from the differences observed between our method and the original m2m_100 model when comparing (a) and (c) as well as (b) and (d) in Figure 3. Specifically, the differences are -3.77 versus -5.96 and $+0.27$ versus -4.85 (Please refer to Table 8 and 9 for detailed scores). This phenomenon is similar to the conclusion drawn in Figure 2, i.e., the performance of the extended model trained from the new language to the original language surpasses that of the model trained in the opposite direction, reinforcing the consistency of the findings.

Different language categories. Our findings indicate that language pairs including high-resource target languages (e.g., *L2H*, *M2H*, and *H2H*) consistently exhibit better performance compared to the other six translation categories. Notably, the *H2H* direction stands out as particularly strong in terms of translation quality. The reason behind this observation is that the *H2H* language pairs already demonstrate strong performance in the original m2m_100 model, due to abundant training data. As a result, the imitation process assigns higher weights to these language pairs, as indicated by Eq. 2, further enhancing their overall performance.

6 Analysis

6.1 Ablation Study

To investigate the importance of the dynamic language weight allocation proposed in Section 3.2.1 and the superiority of our designed imitation learning framework (i.e., separating the expert model and the learning model instead of the on-the-fly in a mixing mode), we conduct a detailed ablation analysis and the results are shown in Table 2. By comparing #2 with #3, and #4 with #5, we find that *LW* demonstrate its advantages in both Imit-MNMT and *On-the-Fly* methods, with the advantage being particularly in Imit-MNMT. Furthermore, the comparison between #3 and #5 highlights the advantage of our imitation learning, i.e., separating the expert model from the learning model and instead updating the weights individually within the learner model.

6.2 Copy and Off-Target Problems

Our analysis focuses on two common problems in large-scale MNMT models. The copying problem (Liu et al., 2021) refers to the phenomenon where certain words are excessively copied by the models from the source side to the target side instead of being accurately translated. On the other hand, the off-target problem (Zhang et al., 2020) arises when the MNMT model translates the text into an incorrect language.

	CR_from_aka						CR_to_aka					
	aka			ban			aka			ban		
	low	mid	high	low	mid	high	low	mid	high	low	mid	high
m2m_100	0.38	0.31	0.48	0.44	0.47	0.50	-	-	-	-	-	-
Finetune	**0.24**	0.27	0.24	0.25	0.25	0.24	0.29	0.27	0.25	0.40	0.41	0.38
Extend_Vocab	0.47	0.39	0.50	0.31	0.30	0.30	0.33	0.32	0.28	0.34	0.36	0.31
Adapter	0.38	0.33	0.37	0.25	0.16	0.19	0.27	0.26	0.29	0.37	0.34	0.28
On-the-Fly (k=5)	0.46	0.37	0.42	0.31	0.20	0.23	0.30	0.29	0.32	0.38	0.36	0.31
On-the-Fly (k=10)	0.48	0.46	0.47	0.37	0.22	0.25	0.32	0.31	0.35	0.41	0.37	0.34
Our (k=5)	0.31	0.20	0.25	0.25	0.07	0.11	**0.15**	0.25	**0.23**	0.24	**0.17**	0.11
Our (k=10)	0.29	**0.18**	**0.15**	**0.22**	**0.05**	**0.05**	**0.15**	**0.23**	**0.23**	**0.23**	**0.17**	**0.10**

	OTR_from_aka						OTR_to_aka					
	aka			ban			aka			ban		
	low	mid	high	low	mid	high	low	mid	high	low	mid	high
m2m_100	0.44	0.52	0.46	0.70	0.84	0.90	-	-	-	-	-	-
Finetune	0.93	0.84	0.78	0.99	0.99	0.85	0.29	0.21	0.23	0.16	0.12	0.12
Extend_Vocab	0.95	0.93	0.83	0.99	0.99	0.85	0.33	0.30	0.31	0.33	0.22	0.17
Adapter	0.31	0.28	0.13	0.36	0.25	0.10	0.15	0.18	0.20	0.27	0.24	0.23
On-the-Fly (k=5)	0.43	0.35	0.37	0.42	0.37	0.26	0.21	0.25	0.28	0.30	0.25	0.25
On-the-Fly (k=10)	0.56	0.47	0.42	0.50	0.43	0.33	0.25	0.33	0.30	0.31	0.28	0.27
Our (k=5)	**0.26**	**0.05**	**0.02**	0.28	0.33	**0.00**	0.11	0.16	0.16	0.04	**0.04**	0.04
Our (k=10)	**0.26**	**0.05**	**0.02**	**0.23**	**0.02**	**0.00**	**0.08**	**0.12**	**0.13**	**0.03**	**0.04**	**0.03**

Table 1: **Copy and Off-Target Problem**: results of copy ratio (CR) and off-target ratio (OTR). A lower value indicates better performance of the model. The ‘A_B_C’ in header indicates ‘A’ (CR and OTR) problem for extended model translate from (to) *aka* to (from) the original languages. **Bold** and underlined numbers indicates the best and second-best results respectively.

		New→Original			Original→New		
		low	mid	high	low	mid	high
#1	m2m	1.20	1.94	2.47	-	-	-
#2	On-the-Fly (k=5)	1.11	2.37	4.39	1.84	2.75	3.81
	On-the-Fly (k=10)	1.06	2.18	4.21	1.78	2.64	3.70
#3	On-the-Fly with LW (k=5)	1.87	2.65	5.14	1.88	2.86	3.89
	On-the-Fly with LW (k=10)	2.02	2.83	5.36	2.06	2.93	3.26
#4	Imit-MNMT w/o LW (k=5)	2.15	3.14	5.47	2.08	2.91	4.05
	Imit-MNMT w/o LW (k=10)	2.47	3.62	5.88	2.27	2.84	2.97
#5	Imit-MNMT(k=5)	3.71	9.26	13.99	2.99	5.81	7.11
	Imit-MNMT(k=10)	**4.16**	**9.45**	**14.04**	**3.60**	**6.52**	**7.94**

Table 2: Ablation study of Imit-MNMT with/without using language weighting (LW) on extending the m2m_100 model on ‘bjn’.

In contrast to (Liu et al., 2021), we consider two distinct types of copying behaviors: i) the proportion of tokens copied from the source sentence; ii) the ratio of consecutively repeated words in the generated target sentences. The total copying ratio (CR) can be formulated as follows:

$$CR = \frac{\sum_{i=1}^{T} cs(i)}{\sum_{i=1}^{T} count(i)} + \frac{\sum_{i=1}^{T} rt(i)}{\sum_{i=1}^{T} count(i)} \quad (6)$$

where $cs(\cdot)$ is number of tokens copied from the source sentence (i), $rt(\cdot)$ is the number of consecutive repeated tokens in the generated target sentences and $count(\cdot)$ is the number of tokens in the generated target sentence. T is the number of sentences in the test set.

To quantify the extent of off-target behaviors, we compute the ratio of off-target sentences in the translation outputs using the following formula:

$$OTR = \frac{\sum_{i=1}^{T} ot(i)}{T} \quad (7)$$

where $ot(\cdot)$ is a function that judges whether a sentence belongs to an incorrect language[7].

We conducted experiments to demonstrate the effectiveness of our proposed methods in addressing these two challenges in Table 1. We observe the following findings: i) Our approach has demonstrated effectiveness in tackling both of these challenges, which shows a reduction in *CR* and *OTR*. ii) All our findings align with the four comparisons presented in Figure 2 and 3. These two figures show that our method exhibits superior performance in extending the new language compared to other baselines. This suggests that the representation information of the new language is more effectively integrated into the MNMT model, resulting in a decrease in both *CR* and *OTR*. We show the complete results for the copy and off target problems in Appendix C.

6.3 Further Investigation

In addition, we conducted an evaluation of the domain transfer ability of our approach. We find that our method outperforms other baseline methods in zero-shot scenarios. For additional detailed results, please refer to the Appendix D.

7 Conclusion

We introduce **Imit-MNMT**, an innovative approach that extends MNMT to new languages without compromising the translation performance of the original language pairs. More specifically, we present a novel perspective on extending a MNMT model by framing it as an imitation game. Remarkably, our approach leverages only a parallel corpus between the new language and English. Our approach outperforms several robust baseline systems, showcasing its superior performance. Furthermore, it exhibits zero-shot domain transfer capabilities and provides notable advantages in addressing the copy and off-target problems.

[7]We use language identification from NLLB: https://dl.fbaipublicfiles.com/nllb/lid/lid218e.bin

8 Limitations

This work has two limitations. i) We conducted evaluations solely on the m2m_100 model. However, our approach is expected to be applicable to other models such as mt5, mbart, etc., and can be extended to various other NLP tasks, including question answering and text generation. ii) We specifically focused on the scenario of utilizing a parallel corpus only from the new language to English. However, it is worth noting that there might exist parallel sentence pairs between the new language and other languages as well. We believe that incorporating additional corpora from other languages has the potential to further enhance the overall performance.

References

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

M BAIN. 1995. A framework for behavioral cloning. *Machine Intelligence*.

Alexandre Berard. 2021. Continual learning in multilingual NMT via language-specific embeddings. In *Proceedings of the Sixth Conference on Machine Translation*, pages 542–565, Online. Association for Computational Linguistics.

Kiante Brantley, Wen Sun, and Mikael Henaff. 2020. Disagreement-regularized imitation learning. In *International Conference on Learning Representations*.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Abteen Ebrahimi and Katharina Kann. 2021. How to adapt your pretrained multilingual model to 1600 languages. 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 4555–4567, 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, et al. 2021. Beyond english-centric multilingual machine translation. *The Journal of Machine Learning Research*, 22(1):4839–4886.

Bin Fang, Shidong Jia, Di Guo, Muhua Xu, Shuhuan Wen, and Fuchun Sun. 2019. Survey of imitation learning for robotic manipulation. *International Journal of Intelligent Robotics and Applications*, 3:362–369.

Markus Freitag and Yaser Al-Onaizan. 2017. Beam search strategies for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 56–60, Vancouver. Association for Computational Linguistics.

Xavier Garcia, Noah Constant, Ankur Parikh, and Orhan Firat. 2021. Towards continual learning for multilingual machine translation via vocabulary substitution. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1184–1192, Online. Association for Computational Linguistics.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Tianyu He, Xu Tan, and Tao Qin. 2019. Hard but robust, easy but sensitive: How encoder and decoder perform in neural machine translation. *arXiv preprint arXiv:1908.06259*.

Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29.

Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. 2017. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35.

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.

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.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 478–494, Online. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

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, Brussels, Belgium. Association for Computational Linguistics.

Wen Lai, Alexandra Chronopoulou, and Alexander Fraser. 2022. m^4 adapter: Multilingual multi-domain adaptation for machine translation with a meta-adapter. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4282–4296, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*.

Xuebo Liu, Longyue Wang, Derek F. Wong, Liang Ding, Lidia S. Chao, Shuming Shi, and Zhaopeng Tu. 2021. On the copying behaviors of pre-training for neural machine translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4265–4275, Online. Association for Computational Linguistics.

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. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Kelly Marchisio, Patrick Lewis, Yihong Chen, and Mikel Artetxe. 2022. Mini-model adaptation: Efficiently extending pretrained models to new languages via aligned shallow training. *arXiv preprint arXiv:2212.10503*.

Alireza Mohammadshahi, Vassilina Nikoulina, Alexandre Berard, Caroline Brun, James Henderson, and Laurent Besacier. 2022. SMaLL-100: Introducing shallow multilingual machine translation model for low-resource languages. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8348–8359, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

AY NG. 2000. Algorithms for inverse reinforcement learning. In *Proceedings of 17th International Conference on Machine Learning, 2000*, pages 663–670.

Xing Niu, Michael Denkowski, and Marine Carpuat. 2018. Bi-directional neural machine translation with synthetic parallel data. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 84–91, Melbourne, Australia. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2021. UNKs everywhere: Adapting multilingual language models to new scripts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10186–10203, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jerin Philip, Alexandre Berard, Matthias Gallé, and Laurent Besacier. 2020. Monolingual adapters for zero-shot neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4465–4470, Online. Association for Computational Linguistics.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

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, Brussels, Belgium. Association for Computational Linguistics.

Yuzhe Qin, Yueh-Hua Wu, Shaowei Liu, Hanwen Jiang, Ruihan Yang, Yang Fu, and Xiaolong Wang. 2022. Dexmv: Imitation learning for dexterous manipulation from human videos. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*, pages 570–587. Springer.

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.

Ning Shi, Bin Tang, Bo Yuan, Longtao Huang, Yewen Pu, Jie Fu, and Zhouhan Lin. 2022. Text editing as imitation game. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1583–1594, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zihan Wang, Karthikeyan K, Stephen Mayhew, and Dan Roth. 2020. Extending multilingual BERT to low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages

2649–2656, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Jiahao Xu, Yubin Ruan, Wei Bi, Guoping Huang, Shuming Shi, Lihui Chen, and Lemao Liu. 2022. On synthetic data for back translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 419–430, Seattle, United States. Association for Computational Linguistics.

Ziyu Yao, Yiqi Tang, Wen-tau Yih, Huan Sun, and Yu Su. 2020. An imitation game for learning semantic parsers from user interaction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6883–6902, Online. Association for Computational Linguistics.

Chrysoula Zerva, Taisiya Glushkova, Ricardo Rei, and André F. T. Martins. 2022. Disentangling uncertainty in machine translation evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8622–8641, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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.

A Datasets

All the corpora used in our experiments are publicly available through the NLLB corpus, which was mined by AllenAI. We conducted the following cleaning and filtering prepossessing steps on the corpus: i) elimination of duplicated sentences; ii) exclusion of sentences exceeding 120 tokens in length; iii) removal of sentences identified as incorrect language using a langid model. Table 3 shows the detailed statistics of the corpus after preprocessing. Our approach is evaluated using the FLORES-200 dataset.

Language	#Size	Language	#Size
Akan (aka)	133,151	Dyula (dyu)	286,391
Dinka (dik)	159,128	Balinese (ban)	324,936
Bambara (bam)	180,936	Bemba (bem)	427,159
Chokwe (cjk)	214,973	Banjar (bjn)	766,894

Table 3: Data statistics (number of sentences) of parallel data between new languages and English.

B Model Configuration

Our training process consists of two steps. In each batch, we initially generate a pseudo-parallel corpus online, using an expert model, between the new language and the selected k languages. Subsequently, we employ imitation learning to mimic the translation between the new language and the original all-language pair. To maintain consistency with other baseline systems, we set our batch size to 16, learning rate to 5e-5, and dropout to 0.1. Furthermore, we tune the number of iterations on the dev set and update the numbers later.

C Full Results for Copy and Off-Target Problem

We show the complete results for the copy and off-target problems in Table 5.

D Zero-Shot Domain Transfer

To explore the zero-shot domain transfer capacity of our models, we utilize the extended model for the *dyu* language in different domains. All experiments are conducted on the FLORES-200 multi-domain dataset. The corresponding results can be found in Table 4. Our approach yielded the following findings: i) Our approach demonstrates strong domain transfer capabilities, surpassing the baseline systems, even when applied to the original language pairs such as *eng-rus* and *eng-wol*. ii) Our approach exhibits superior transfer capabilities in *eng-dyu* language pair compared to other baselines. This observation, to a certain extent, suggests the extendability of our MNMT model to new languages. One possible explanation is that our approach possesses stronger general multilingual properties, which is helpful for domain transfer. This observation aligns with the findings of Lai et al. (2022), who demonstrated that MNMT models can be transferred across different domains in the same language.

E Detailed Results

In addition to BLEU, we also use chrF++ (Popović, 2017) as an evaluation metric. The results in Tables 7 and 9 correspond to Tables 6 and 8, respectively. We show that Imit-MNMT is more effective than all baseline systems in terms of chrF++, which is consistent with the BLEU scores.

	eng-dyu			eng-rus			eng-wol		
	chat	health	news	chat	health	news	chat	health	news
m2m_100	-	-	-	**18.97**	31.46	22.69	0.23	0.84	0.65
Finetune	0.68	0.19	0.34	0.04	0.04	0.02	0.04	0.05	0.08
Extend_Vocab	0.79	0.09	0.24	0.05	0.02	0.03	0.05	0.03	0.08
Adapter	0.30	1.23	2.81	10.55	26.40	17.90	0.43	0.83	1.19
On-the-Fly ($k = 5$)	0.25	0.86	2.47	8.82	22.06	15.25	0.33	0.75	0.92
On-the-Fly ($k = 10$)	0.14	0.72	2.09	8.30	21.73	14.77	0.28	0.62	0.81
Our ($k = 5$)	1.29	1.54	2.75	18.02	31.05	23.02	1.01	1.34	1.45
Our ($k = 10$)	**1.50**	**1.87**	**3.28**	18.63	**31.84**	**23.41**	**1.34**	**1.63**	**1.76**

Table 4: **Domain Transfer**: evaluate the zero-shot domain transfer on the extended model for the *dyu* language.

	aka			dik			bam			cjk			dyu			ban			bem			bjn		
	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high
m2m_100	0.38	0.31	0.48	0.38	0.42	0.49	0.34	0.38	0.45	0.40	0.42	0.47	0.37	0.41	0.35	0.44	0.47	0.50	0.40	0.45	0.54	0.43	0.45	0.45
Finetune	**0.24**	0.27	0.24	0.34	0.32	0.35	**0.26**	0.31	0.31	0.36	0.12	0.14	0.42	0.40	0.42	0.25	0.25	0.24	0.23	0.24	0.21	0.10	0.13	0.19
Extend_Vocab	0.47	0.39	0.50	0.39	0.41	0.35	0.38	0.40	0.35	0.34	0.34	0.35	0.48	0.48	0.57	0.31	0.30	0.30	0.27	0.26	0.23	0.10	0.13	0.16
Adapter	0.38	0.33	0.37	**0.31**	0.36	0.40	0.29	0.27	0.36	0.25	0.14	0.13	0.32	0.31	0.41	0.25	0.16	0.19	0.34	0.27	0.22	0.31	0.24	0.21
On-the-Fly (k=5)	0.46	0.37	0.42	0.38	0.42	0.42	0.35	0.32	0.37	0.28	0.27	0.17	0.37	0.36	0.45	0.32	0.20	0.23	0.36	0.31	0.28	0.35	0.25	0.24
On-the-Fly (k=10)	0.48	0.46	0.47	0.41	0.48	0.46	0.39	0.37	0.40	0.31	0.29	0.20	0.41	0.42	0.48	0.37	0.22	0.25	0.39	0.34	0.31	0.38	0.29	0.28
Our (k=5)	0.31	0.20	0.25	0.34	0.34	0.33	0.33	0.30	0.31	0.25	0.12	0.05	0.38	0.24	0.37	0.25	0.07	0.11	0.20	0.17	0.17	0.05	0.09	0.09
Our (k=10)	0.29	**0.18**	**0.15**	0.33	**0.30**	**0.31**	0.30	**0.22**	**0.25**	**0.20**	**0.08**	**0.01**	**0.32**	**0.20**	**0.33**	**0.22**	**0.05**	**0.05**	**0.15**	**0.11**	**0.13**	**0.05**	**0.08**	**0.06**

(a) Copy Ratio: Extended model translate from *aka* to original languages

	aka			dik			bam			cjk			dyu			ban			bem			bjn		
	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high
m2m_100	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Finetune	0.29	0.27	0.25	0.34	0.38	0.40	0.39	0.36	0.38	0.37	0.33	0.35	0.35	0.31	0.29	0.40	0.41	0.38	0.34	0.38	0.39	0.21	0.23	0.13
Extend_Vocab	0.33	0.32	0.28	0.42	0.44	0.43	0.40	0.38	0.37	0.29	0.24	0.14	0.33	0.32	0.35	0.34	0.36	0.31	0.38	0.40	0.43	0.28	0.31	0.30
Adapter	0.27	0.26	0.29	0.36	0.41	0.40	0.42	0.38	0.35	0.33	0.28	0.28	0.30	0.29	0.30	0.37	0.34	0.28	0.35	0.33	0.32	0.26	0.27	0.28
On-the-Fly (k=5)	0.30	0.29	0.32	0.38	0.43	0.41	0.45	0.40	0.36	0.35	0.30	0.30	0.32	0.31	0.31	0.38	0.36	0.31	0.35	0.37	0.37	0.28	0.29	0.30
On-the-Fly (k=10)	0.32	0.31	0.35	0.40	0.44	0.45	0.46	0.41	0.39	0.36	0.32	0.31	0.33	0.35	0.33	0.41	0.37	0.34	0.38	0.39	0.39	0.30	0.35	0.34
Our (k=5)	**0.15**	0.25	**0.23**	0.29	**0.28**	0.25	0.29	0.26	0.21	0.14	0.18	0.19	0.16	0.17	0.22	0.24	**0.17**	0.11	0.17	0.19	0.16	0.15	0.12	0.14
Our (k=10)	**0.15**	**0.23**	**0.23**	**0.26**	**0.28**	**0.22**	**0.27**	**0.21**	**0.15**	**0.13**	**0.14**	**0.14**	**0.14**	**0.12**	**0.14**	**0.23**	**0.17**	**0.10**	**0.14**	**0.15**	**0.12**	**0.10**	**0.09**	**0.09**

(b) Copy Ratio: Extended model translate from original languages to *aka*

	aka			dik			bam			cjk			dyu			ban			bem			bjn		
	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high
m2m_100	0.44	0.52	0.46	0.42	0.54	0.47	0.38	0.47	0.39	0.64	0.8	0.82	0.57	0.73	0.78	0.70	0.84	0.90	0.66	0.81	0.86	0.70	0.85	0.90
Finetune	0.93	0.84	0.78	0.95	0.91	0.81	0.98	0.96	0.85	0.98	0.95	0.84	0.99	0.96	0.85	0.99	0.99	0.85	0.99	0.99	0.85	0.99	0.99	0.85
Extend_Vocab	0.95	0.93	0.83	0.98	0.96	0.84	0.99	0.97	0.86	0.99	0.97	0.86	0.99	0.98	0.86	0.99	0.99	0.85	0.99	0.99	0.85	0.99	0.99	0.85
Adapter	0.31	0.28	0.13	0.33	0.20	0.14	0.42	0.28	0.22	0.43	0.20	0.23	0.36	0.25	0.16	0.36	0.25	0.10	0.36	0.20	0.13	0.48	0.19	0.12
On-the-Fly (k=5)	0.43	0.35	0.37	0.44	0.37	0.28	0.45	0.35	0.30	0.48	0.34	0.35	0.43	0.28	0.27	0.42	0.37	0.26	0.49	0.38	0.25	0.52	0.28	0.22
On-the-Fly (k=10)	0.56	0.47	0.42	0.51	0.40	0.32	0.57	0.44	0.38	0.52	0.39	0.42	0.49	0.32	0.30	0.50	0.43	0.33	0.54	0.42	0.31	0.55	0.34	0.29
Our (k=5)	**0.26**	**0.05**	**0.02**	0.28	**0.05**	**0.02**	0.29	**0.04**	**0.02**	0.28	0.06	0.03	0.29	**0.08**	0.06	0.28	0.03	**0.00**	0.19	0.04	**0.01**	**0.26**	**0.03**	**0.00**
Our (k=10)	**0.26**	**0.05**	**0.02**	**0.24**	**0.05**	**0.02**	**0.25**	**0.04**	**0.02**	**0.26**	**0.05**	**0.02**	**0.26**	**0.08**	**0.05**	**0.23**	**0.02**	**0.00**	**0.13**	**0.03**	**0.01**	**0.23**	**0.03**	**0.00**

(c) Off-Target Ratio: Extended model translate from *aka* to original languages

	aka			dik			bam			cjk			dyu			ban			bem			bjn		
	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high
m2m_100	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Finetune	0.29	0.21	0.23	0.23	0.21	0.19	0.16	0.15	0.14	0.34	0.23	0.22	0.25	0.21	0.19	0.16	0.12	0.12	0.24	0.19	0.16	0.22	0.17	0.17
Extend_Vocab	0.33	0.30	0.31	0.40	0.27	0.21	0.29	0.18	0.17	0.58	0.50	0.50	0.31	0.22	0.20	0.33	0.22	0.17	0.31	0.29	0.28	0.60	0.50	0.46
Adapter	0.15	0.18	0.20	0.18	0.15	0.16	0.23	0.14	0.16	0.30	0.27	0.25	0.25	0.20	0.20	0.27	0.24	0.23	0.28	0.26	0.24	0.31	0.26	0.24
On-the-Fly (k=5)	0.21	0.25	0.28	0.24	0.18	0.20	0.28	0.19	0.20	0.35	0.30	0.28	0.28	0.27	0.22	0.30	0.25	0.25	0.31	0.28	0.25	0.33	0.29	0.27
On-the-Fly (k=10)	0.25	0.33	0.30	0.28	0.24	0.23	0.30	0.25	0.23	0.39	0.33	0.31	0.30	0.29	0.26	0.31	0.28	0.27	0.34	0.31	0.27	0.35	0.31	0.31
Our (k=5)	0.11	0.16	0.16	0.08	0.07	0.06	0.06	**0.05**	0.05	0.12	0.15	0.16	**0.09**	0.16	0.14	**0.01**	**0.01**	**0.01**	0.04	**0.04**	0.04	0.12	0.10	0.11
Our (k=10)	**0.08**	**0.12**	**0.13**	**0.07**	**0.05**	**0.05**	**0.04**	**0.05**	**0.04**	**0.10**	**0.14**	**0.14**	**0.09**	**0.13**	**0.10**	**0.01**	**0.01**	**0.01**	**0.03**	**0.04**	**0.03**	**0.10**	**0.07**	**0.08**

(d) Off-Target Ratio: Extended model translate from original languages to *aka*

Table 5: **Copy and Off-Target Problem**: results of copy ratio (CR) and off-target ratio (OTR). A lower value indicates better performance of the model.

	aka			dik			bam			cjk			dyu			ban			bem			bjn		
	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high
m2m_100	0.67	0.78	1.20	0.55	0.76	1.08	0.45	0.55	0.70	0.69	0.88	1.27	0.41	0.56	0.85	1.07	1.82	2.41	0.97	1.26	1.95	1.20	1.94	2.47
Finetune	0.83	0.88	2.01	0.76	0.78	1.97	0.76	0.76	1.97	0.47	0.49	1.13	0.21	0.18	0.49	1.12	1.12	4.35	1.07	1.00	3.08	1.23	1.18	4.50
Extend_Vocab	0.33	0.34	0.83	0.39	0.38	0.94	0.38	0.34	0.83	0.22	0.21	0.47	0.09	0.09	0.20	0.74	0.74	2.48	0.61	0.52	1.45	0.93	0.85	3.00
Adapter	0.95	1.24	2.40	0.97	1.05	2.32	1.00	1.38	2.85	0.58	1.33	2.19	0.41	0.86	1.40	2.40	3.56	4.69	1.13	2.40	3.23	1.41	2.75	4.77
On-the-Fly (k=5)	0.91	1.03	2.14	0.84	0.95	2.15	0.87	1.17	2.33	0.57	1.02	1.94	0.37	0.75	1.26	2.16	3.25	4.17	0.94	2.05	3.09	1.11	2.37	4.39
On-the-Fly (k=10)	0.82	0.96	1.91	0.79	0.87	2.06	0.75	1.03	2.15	0.41	0.89	1.82	0.28	0.66	1.10	2.03	3.07	4.02	0.81	1.93	2.91	1.06	2.18	4.21
Our (k=5)	1.97	2.94	4.80	1.91	3.48	4.94	1.86	3.88	5.87	1.45	2.49	3.87	1.83	2.37	3.74	3.88	8.62	12.93	3.56	5.41	8.29	3.71	9.26	13.99
Our (k=10)	**2.37**	**3.11**	**5.17**	**2.26**	**3.87**	**5.37**	**2.04**	**4.12**	**6.27**	**1.94**	**2.84**	**4.25**	**2.30**	**2.82**	**4.18**	**4.63**	**9.06**	**13.53**	**4.02**	**5.66**	**8.63**	**4.16**	**9.45**	**14.04**

(a) Translation from new languages to original languages

	aka			dik			bam			cjk			dyu			ban			bem			bjn		
	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high
m2m_100	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		-
Finetune	0.33	0.59	0.82	0.47	0.77	1.12	0.87	1.80	2.31	0.27	0.35	0.48	0.31	0.57	0.82	1.51	2.33	3.94	0.69	1.09	1.75	1.75	2.69	3.72
Extend_Vocab	0.16	0.24	0.33	0.17	0.34	0.50	0.35	0.73	0.90	0.11	0.24	0.36	0.17	0.34	0.53	0.49	1.32	1.85	0.41	0.85	1.04	0.61	1.61	2.48
Adapter	0.53	0.68	0.95	0.61	0.89	1.37	0.98	1.93	2.37	0.63	0.82	0.95	0.72	0.93	1.16	1.74	2.79	4.26	0.94	1.45	2.67	1.87	2.93	3.86
On-the-Fly (k=5)	0.41	0.58	0.87	0.57	0.82	1.28	0.91	1.85	2.24	0.56	0.63	0.87	0.68	0.82	1.07	1.53	2.51	4.02	0.85	1.39	2.36	1.84	2.75	3.81
On-the-Fly (k=10)	0.35	0.46	0.75	0.51	0.73	1.11	0.82	1.71	2.16	0.47	0.59	0.76	0.61	0.74	0.93	1.41	2.46	3.83	0.72	1.26	2.25	1.78	2.64	3.70
Our (k=5)	1.54	1.88	2.66	1.98	2.37	3.73	2.21	2.93	3.57	1.15	1.79	2.35	1.35	2.27	3.29	2.96	5.57	7.15	1.69	2.47	3.46	2.99	5.81	7.11
Our (k=10)	**2.15**	**2.47**	**3.34**	**2.24**	**2.79**	**4.16**	**2.54**	**3.41**	**4.21**	**1.84**	**2.47**	**3.17**	**1.86**	**2.68**	**4.13**	**3.62**	**6.09**	**8.18**	**2.17**	**2.96**	**4.35**	**3.60**	**6.52**	**7.94**

(b) Translation from original languages to new languages

Table 6: **Main Results (the answer of Q1)**: Average BLEU scores for different categories in two directions on the FLORES-200 benchmark. The original languages in (a) and (b) indicates the languages already supported in m2m_100. k indicates the number of expert language pairs described in Section 3.2.1. Results in bold are significant over original m2m_100 model at 0.01, evaluated by booststrap resampling (Koehn, 2004).

	aka			dik			bam			cjk			dyu			ban			bem			bjn		
	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high
m2m_100	15.25	15.97	18.20	14.36	15.85	17.63	13.52	14.36	15.46	15.39	16.57	18.52	13.14	14.44	16.39	17.58	20.66	22.50	17.07	18.48	21.10	18.20	21.06	22.67
Finetune	16.28	16.57	21.29	15.85	15.97	21.16	15.85	15.85	21.16	13.70	13.87	17.88	10.72	10.23	13.87	17.83	17.83	26.92	17.58	17.22	24.24	18.34	18.11	27.20
Extend_Vocab	12.30	12.41	16.28	12.94	12.84	16.90	12.84	12.41	16.28	10.88	10.72	13.70	8.29	8.29	10.57	15.72	15.72	22.70	14.82	14.12	19.28	16.85	16.39	24.05
Adapter	16.96	18.39	22.47	17.07	17.48	22.24	17.22	18.99	23.67	14.60	18.78	21.85	13.14	16.45	19.08	22.47	25.33	27.54	17.88	22.47	24.59	19.12	23.42	27.68
On-the-Fly (k=5)	16.74	17.38	21.70	16.34	16.96	21.73	16.51	18.07	22.27	14.52	17.33	21.06	12.74	15.78	18.48	21.76	24.64	26.57	16.90	21.42	24.26	17.78	22.38	26.99
On-the-Fly (k=10)	16.22	17.01	20.96	16.03	16.51	21.45	15.78	17.38	21.73	13.14	16.63	20.66	11.70	15.18	17.73	21.36	24.21	26.28	16.16	21.03	23.82	17.53	21.82	26.65
Our (k=5)	21.16	23.90	27.73	20.96	25.15	27.98	20.80	26.00	29.48	19.28	22.72	25.98	20.69	22.38	25.71	26.00	33.13	37.47	25.33	28.76	32.74	25.65	33.86	38.38
Our (k=10)	**22.38**	**24.31**	**28.37**	**22.06**	**25.98**	**28.70**	**21.39**	**26.48**	**30.08**	**21.06**	**23.65**	**26.73**	**22.18**	**23.60**	**26.59**	**27.43**	**33.64**	**37.99**	**26.28**	**29.16**	**33.14**	**26.56**	**34.07**	**38.42**

(a) Translation from new languages to original languages

	aka			dik			bam			cjk			dyu			ban			bem			bjn		
	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high	low	mid	high
m2m_100	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		-
Finetune	12.30	14.67	16.22	13.70	15.91	17.83	16.51	20.59	22.21	11.57	12.52	13.78	12.07	14.52	16.22	19.52	22.27	26.12	15.39	17.68	20.42	20.42	23.26	25.67
Extend_Vocab	9.87	11.17	12.30	10.06	12.41	13.95	12.52	15.65	16.68	8.81	11.17	12.63	10.06	12.41	14.20	13.87	18.74	20.76	13.14	16.39	17.43	14.82	19.90	22.70
Adapter	14.20	15.32	16.96	14.82	16.63	18.95	17.12	21.03	22.38	14.97	16.22	16.96	15.59	16.85	18.02	20.38	23.52	26.75	16.90	19.28	23.21	20.83	23.87	25.96
On-the-Fly (k=5)	13.14	14.60	16.51	14.52	16.22	18.57	16.74	20.76	22.00	14.44	14.97	16.51	15.32	16.22	17.58	19.60	22.78	26.28	16.39	19.04	22.36	20.73	23.42	25.86
On-the-Fly (k=10)	12.52	13.61	15.78	14.04	15.65	17.78	16.22	20.27	21.76	13.70	14.67	15.85	14.82	15.72	16.85	19.12	22.64	25.90	15.59	18.48	22.03	20.52	23.13	25.63
Our (k=5)	19.64	20.86	23.18	21.20	22.38	25.69	21.91	23.87	25.35	17.97	20.56	22.33	18.87	22.09	24.73	23.95	29.02	31.30	20.20	22.67	25.11	24.02	29.39	31.25
Our (k=10)	**21.73**	**22.67**	**24.84**	**22.00**	**23.52**	**26.56**	**22.86**	**25.00**	**26.65**	**20.73**	**22.67**	**24.45**	**20.80**	**23.24**	**26.50**	**25.46**	**29.81**	**32.61**	**21.79**	**23.95**	**26.92**	**25.41**	**30.44**	**32.32**

(b) Translation from original languages to new languages

Table 7: **Main Results (the answer of Q1)**: Average chrF++ scores for different categories in two directions on the FLORES-200 benchmark.

	L2L		L2M		L2H		M2L		M2M		M2H		H2L		H2M		H2H	
	afr2tam	ibo2pan	guj2slk	pan2kor	tgk2eng	gle2spa	hin2msa	fas2mon	dan2est	fas2hun	ara2eng	fas2spa	eng2tam	fra2hau	eng2srp	deu2mkd	eng2deu	fra2deu
m2m_100	**1.46**	0.60	0.47	**1.82**	**7.99**	0.83	**20.51**	**1.45**	**16.63**	**13.77**	**29.21**	**16.28**	**2.28**	1.67	**25.45**	**20.84**	**22.79**	**22.85**
Finetune	0.68	0.21	0.06	0.03	3.69	0.34	1.53	0.36	1.50	1.72	17.47	1.50	0.83	1.59	2.86	1.21	2.31	2.02
Extend_Vocab	0.31	0.09	0.02	0.02	1.64	0.29	0.72	0.19	0.45	0.76	7.46	0.98	0.49	0.61	0.54	0.45	1.03	0.76
Adapter	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
On-the-Fly (k=5)	0.84	0.35	0.34	1.23	4.25	1.25	15.28	1.26	11.54	10.59	20.41	12.75	1.23	1.74	18.53	15.93	18.54	18.35
On-the-Fly (k=10)	0.73	0.27	0.31	1.05	3.87	1.07	13.71	1.03	10.93	8.84	18.43	10.34	1.05	1.41	16.29	12.72	16.31	15.73
Our (k=5)	1.14	1.07	1.08	1.62	6.54	1.71	17.64	1.27	14.61	12.29	27.74	14.79	1.57	2.21	23.15	18.49	18.73	20.78
Our (k=10)	1.21	**1.19**	**1.13**	1.68	6.78	**1.90**	18.75	1.34	15.28	12.62	28.08	15.02	1.69	**2.37**	23.53	18.87	19.02	21.36
Δ	-0.25	+0.59	+0.66	-0.14	-1.21	+1.07	-1.76	-0.11	-1.35	-1.15	-1.13	-1.26	-0.59	+0.70	-1.92	-1.97	-3.77	-1.49

(a) Extended model trained *from aka* to original languages

	L2L		L2M		L2H		M2L		M2M		M2H		H2L		H2M		H2H	
	afr2tam	ibo2pan	guj2slk	pan2kor	tgk2eng	gle2spa	hin2msa	fas2mon	dan2est	fas2hun	ara2eng	fas2spa	eng2tam	fra2hau	eng2srp	deu2mkd	eng2deu	fra2deu
m2m_100	1.46	0.60	0.47	1.82	**7.99**	0.83	20.51	1.45	16.63	13.77	29.21	16.28	2.28	1.67	25.45	20.84	22.79	22.85
Finetune	0.64	0.23	0.02	0.13	3.93	0.50	1.43	0.41	1.33	0.95	20.87	1.00	0.99	1.69	0.94	0.75	2.36	1.93
Extend_Vocab	0.44	0.13	0.01	0.17	2.08	0.50	1.01	0.30	0.88	0.71	13.91	0.85	0.80	1.15	0.76	0.58	1.71	1.34
Adapter	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
On-the-Fly (k=5)	0.93	0.32	0.42	1.37	4.95	1.04	16.37	0.93	12.93	11.46	25.47	13.75	1.04	1.28	17.33	16.22	19.94	18.44
On-the-Fly (k=10)	0.81	0.25	0.36	1.14	3.86	0.92	15.71	0.75	11.04	10.08	23.09	12.47	0.85	1.07	15.91	14.85	17.31	16.39
Our (k=5)	1.34	1.14	1.22	1.79	7.92	1.64	20.95	1.46	16.55	13.53	28.89	16.19	2.23	2.10	25.41	20.85	22.43	22.76
Our (k=10)	**1.57**	**1.23**	**1.27**	**2.05**	7.45	**2.15**	**21.27**	**1.71**	**16.84**	**13.83**	**29.62**	**16.46**	**2.56**	**2.35**	**25.82**	**21.08**	**23.06**	**23.01**
Δ	+0.11	+0.63	+0.80	+0.23	-0.54	+1.32	+0.76	+0.26	+0.21	+0.06	+0.41	+0.18	+0.28	+0.68	+0.37	+0.24	+0.27	+0.16

(b) Extended model trained *from ban* to original languages

	L2L		L2M		L2H		M2L		M2M		M2H		H2L		H2M		H2H	
	afr2tam	ibo2pan	guj2slk	pan2kor	tgk2eng	gle2spa	hin2msa	fas2mon	dan2est	fas2hun	ara2eng	fas2spa	eng2tam	fra2hau	eng2srp	deu2mkd	eng2deu	fra2deu
m2m_100	**1.46**	0.60	0.47	**1.82**	**7.99**	**0.83**	**20.51**	**1.45**	**16.63**	**13.77**	**29.21**	**16.28**	**2.28**	**1.67**	**25.45**	**20.84**	**22.79**	**22.85**
Finetune	0.23	0.08	0.09	0.06	0.46	0.04	0.62	0.07	0.71	0.65	3.10	1.14	0.45	0.49	4.42	3.98	1.64	1.33
Extend_Vocab	0.11	0.03	0.01	0.03	0.24	0.09	0.13	0.02	0.20	0.13	0.72	0.24	0.13	0.13	0.16	0.17	0.29	0.21
Adapter	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
On-the-Fly (k=5)	0.42	0.47	0.12	0.52	2.03	0.51	4.24	0.59	4.57	4.81	6.61	5.44	0.73	0.64	6.57	6.91	4.69	5.80
On-the-Fly (k=10)	0.37	0.31	0.08	0.36	1.88	0.44	3.86	0.46	4.22	4.50	6.03	5.03	0.58	0.51	6.19	6.47	4.25	5.36
Our (k=5)	0.88	0.82	0.84	1.24	5.01	0.70	15.16	1.08	12.85	10.2	25.86	13.26	1.22	1.19	20.59	16.30	16.49	17.63
Our (k=10)	1.03	**0.97**	**0.94**	1.37	5.25	0.73	15.47	1.25	13.35	10.82	26.05	13.86	1.69	1.29	21.04	16.97	16.83	18.54
Δ	-0.43	+0.37	+0.47	-0.45	-2.74	-0.10	-5.04	-0.20	-3.28	-2.95	-3.16	-2.42	-0.59	-0.38	-4.41	-3.87	-5.96	-4.31

(c) Extended model trained *from* original languages to *aka*

	L2L		L2M		L2H		M2L		M2M		M2H		H2L		H2M		H2H	
	afr2tam	ibo2pan	guj2slk	pan2kor	tgk2eng	gle2spa	hin2msa	fas2mon	dan2est	fas2hun	ara2eng	fas2spa	eng2tam	fra2hau	eng2srp	deu2mkd	eng2deu	fra2deu
m2m_100	**1.46**	0.60	0.47	**1.82**	**7.99**	0.83	**20.51**	**1.45**	**16.63**	**13.77**	**29.21**	**16.28**	**2.28**	**1.67**	**25.45**	**20.84**	**22.79**	**22.85**
Finetune	0.29	0.12	0.03	0.00	0.30	0.24	1.54	0.17	0.87	0.31	0.78	0.55	0.47	1.25	0.84	0.50	1.23	0.96
Extend_Vocab	0.13	0.02	0.01	0.01	0.16	0.23	0.53	0.07	0.47	0.17	0.46	0.27	0.26	0.46	0.41	0.22	0.67	0.50
Adapter	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
On-the-Fly (k=5)	0.76	0.44	0.62	0.83	3.25	0.43	6.28	0.81	6.22	5.51	8.83	7.31	1.17	1.21	9.74	7.70	9.84	9.37
On-the-Fly (k=10)	0.62	0.38	0.51	0.61	2.84	0.37	5.52	0.75	5.85	5.27	7.36	6.29	0.87	1.04	8.20	6.62	7.73	8.52
Our (k=5)	0.96	0.85	1.01	1.31	5.71	1.03	15.73	1.12	13.10	10.68	26.15	13.61	1.41	1.38	21.74	16.61	17.60	18.20
Our (k=10)	1.16	**1.04**	**1.20**	1.44	5.88	**1.17**	15.89	1.36	13.67	10.90	26.59	14.16	1.77	1.44	22.36	17.26	17.94	18.49
Δ	-0.30	+0.44	+0.73	-0.38	-2.11	+0.34	-4.62	-0.09	-2.96	-2.87	-2.62	-2.12	-0.51	-0.23	-3.09	-3.58	-4.85	-4.36

(d) Extended model trained *from* original languages to *ban*

Table 8: **Main Results (the answer of Q2):** BLEU scores of the extended model on 9 original language pairs grouped by available resources on both source and target sizes (**L**ow, **M**id, **H**igh). In each language pair classification, we random select two example pairs. (a) and (b) evaluate the extended model trained from the new language to the original languages. (c) and (d) evaluate the extended model trained from the original languages to the new language. Δ indicates the difference between *m2m_100* and our approach. **Bold** and underlined numbers indicates the best and second-best results respectively. We do not include the results of *Adapter* method, because the results are the same as in (Fan et al., 2021).

	L2L		L2M		L2H		M2L		M2M		M2H		H2L		H2M		H2H	
	afr2tam	ibo2pan	guj2slk	pan2kor	tgk2eng	gle2spa	hin2msa	fas2mon	dan2est	fas2hun	ara2eng	fas2spa	eng2tam	fra2hau	eng2srp	deu2mkd	eng2deu	fra2deu
m2m_100	**19.32**	14.75	13.70	**20.66**	**32.38**	16.28	**43.11**	**19.28**	**40.45**	**38.20**	**48.00**	**40.19**	**22.12**	20.13	**46.03**	**43.32**	**44.51**	**44.55**
Finetune	15.32	10.72	7.33	5.94	25.61	12.41	19.60	12.63	19.48	20.31	41.06	19.48	16.28	19.83	23.70	18.25	22.21	21.32
Extend_Vocab	12.07	8.29	5.25	5.25	20.02	11.83	15.59	10.40	13.52	15.85	31.71	17.12	13.87	14.82	14.28	13.52	17.38	15.85
Adapter	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
On-the-Fly (k=5)	16.34	12.52	12.41	18.34	26.73	18.43	39.42	18.48	36.20	35.27	43.05	37.31	18.34	20.38	41.80	39.92	41.81	41.68
On-the-Fly (k=10)	15.65	11.57	12.07	17.48	25.98	17.58	38.15	17.38	35.61	33.39	41.73	35.01	17.48	19.12	40.20	37.29	40.21	39.77
Our (k=5)	17.92	17.58	17.63	19.94	30.47	20.27	41.18	18.52	38.89	36.90	47.25	39.03	19.75	21.91	44.72	41.77	41.94	43.28
Our (k=10)	18.25	**18.16**	**17.88**	20.16	30.80	**20.93**	41.95	18.83	39.42	37.20	47.42	39.22	20.20	**22.38**	44.95	42.03	42.13	43.64
Δ	-1.07	+3.41	+4.18	-0.50	-1.58	+4.65	-1.16	-0.46	-1.03	-1.00	-0.57	-0.97	-1.92	+2.26	-1.08	-1.29	-2.38	-0.90

(a) Extended model trained *from aka* to original languages

	L2L		L2M		L2H		M2L		M2M		M2H		H2L		H2M		H2H	
	afr2tam	ibo2pan	guj2slk	pan2kor	tgk2eng	gle2spa	hin2msa	fas2mon	dan2est	fas2hun	ara2eng	fas2spa	eng2tam	fra2hau	eng2srp	deu2mkd	eng2deu	fra2deu
m2m_100	19.32	14.75	13.70	20.66	**32.38**	16.28	43.11	19.28	40.45	38.20	48.00	40.19	22.12	20.13	46.03	43.32	44.51	44.55
Finetune	15.04	11.02	5.25	9.27	26.10	13.95	19.20	13.14	18.78	16.96	43.34	17.22	17.17	20.20	16.90	15.78	22.36	21.03
Extend_Vocab	13.42	9.27	4.25	10.06	21.51	13.95	17.28	11.95	16.57	15.52	38.31	16.39	16.10	17.97	15.85	14.60	20.27	18.83
Adapter	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
On-the-Fly (k=5)	16.85	12.19	13.24	18.95	28.00	17.43	40.26	16.85	37.47	36.12	46.04	38.18	17.43	18.57	40.96	40.14	42.74	41.74
On-the-Fly (k=10)	16.16	11.31	12.63	17.92	25.96	16.79	39.76	15.78	35.72	34.74	44.69	37.06	16.39	17.58	39.91	39.08	40.94	40.27
Our (k=5)	18.83	17.92	18.30	20.56	32.29	20.02	43.39	19.32	40.39	37.99	47.84	40.12	21.97	21.58	46.01	43.32	44.30	44.49
Our (k=10)	**19.75**	**18.34**	**18.52**	**21.42**	31.70	**21.73**	**43.59**	**20.27**	**40.60**	**38.25**	**48.20**	**40.32**	**22.91**	**22.33**	**46.23**	**43.47**	**44.67**	**44.64**
Δ	+0.43	+3.59	+4.83	+0.76	-0.68	+5.46	+0.48	+0.99	+0.15	+0.05	+0.20	+0.13	+0.79	+2.20	+0.20	+0.15	+0.16	+0.09

(b) Extended model trained *from ban* to original languages

	L2L		L2M		L2H		M2L		M2M		M2H		H2L		H2M		H2H	
	afr2tam	ibo2pan	guj2slk	pan2kor	tgk2eng	gle2spa	hin2msa	fas2mon	dan2est	fas2hun	ara2eng	fas2spa	eng2tam	fra2hau	eng2srp	deu2mkd	eng2deu	fra2deu
m2m_100	**19.32**	14.75	13.70	**20.66**	**32.38**	**16.28**	**43.11**	**19.28**	**40.45**	**38.20**	**48.00**	**40.19**	**22.12**	**20.13**	**46.03**	**43.32**	**44.51**	**44.55**
Finetune	11.02	8.00	8.29	7.33	13.61	6.48	14.90	7.68	15.52	15.11	24.29	17.92	13.52	13.87	27.05	26.20	20.02	18.78
Extend_Vocab	8.81	5.94	4.25	5.94	11.17	8.29	9.27	5.25	10.57	9.27	15.59	11.17	9.27	9.27	9.87	10.06	11.83	10.72
Adapter	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
On-the-Fly (k=5)	13.24	13.70	9.05	14.12	21.36	14.04	26.71	14.67	27.32	27.75	30.57	28.81	15.65	15.04	30.51	30.98	27.54	29.38
On-the-Fly (k=10)	12.74	12.07	8.00	12.63	20.86	13.42	25.96	13.61	26.67	27.20	29.72	28.13	14.60	14.04	29.96	30.37	26.73	28.68
Our (k=5)	16.57	16.22	16.34	18.39	28.10	15.46	39.33	17.63	37.40	34.87	46.25	37.76	18.30	18.16	43.16	40.20	40.35	41.17
Our (k=10)	17.38	**17.07**	**16.90**	18.95	28.50	15.65	39.57	18.43	37.84	35.50	46.36	38.27	20.20	18.61	43.44	40.70	40.60	41.81
Δ	-1.94	+2.32	+3.21	-1.71	-3.88	-0.62	-3.54	-0.85	-2.61	-2.70	-1.64	-1.92	-1.92	-1.52	-2.58	-2.62	-3.91	-2.74

(c) Extended model trained *from* original languages to *aka*

	L2L		L2M		L2H		M2L		M2M		M2H		H2L		H2M		H2H	
	afr2tam	ibo2pan	guj2slk	pan2kor	tgk2eng	gle2spa	hin2msa	fas2mon	dan2est	fas2hun	ara2eng	fas2spa	eng2tam	fra2hau	eng2srp	deu2mkd	eng2deu	fra2deu
m2m_100	**19.32**	14.75	13.70	**20.66**	**32.38**	16.28	**43.11**	**19.28**	**40.45**	**38.20**	**48.00**	**40.19**	**22.12**	**20.13**	**46.03**	**43.32**	**44.51**	**44.55**
Finetune	11.83	9.05	5.94	0.00	11.95	11.17	19.64	10.06	16.51	12.07	15.97	14.36	13.70	18.43	16.34	13.95	18.34	17.01
Extend_Vocab	9.27	5.25	4.25	4.25	9.87	11.02	14.20	7.68	13.70	10.06	13.61	11.57	11.44	13.61	13.14	10.88	15.25	13.95
Adapter	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
On-the-Fly (k=5)	15.85	13.42	14.90	16.28	24.64	13.33	30.09	16.16	30.01	28.92	33.37	31.51	18.07	18.25	34.38	32.02	34.49	33.98
On-the-Fly (k=10)	14.90	12.84	14.04	14.82	23.65	12.74	28.94	15.78	29.45	28.53	31.58	30.11	16.51	17.43	32.63	30.58	32.05	33.01
Our (k=5)	17.01	16.39	17.28	18.70	29.24	17.38	39.77	17.83	37.62	35.36	46.41	38.06	19.12	18.99	43.88	40.43	41.15	41.57
Our (k=10)	18.02	**17.43**	**18.20**	19.24	29.50	**18.07**	39.89	18.91	38.11	35.58	46.65	38.52	20.49	19.24	44.25	40.91	41.39	41.77
Δ	-1.30	+2.68	+4.51	-1.42	-2.88	+1.79	-3.22	-0.37	-2.34	-2.62	-1.35	-1.67	-1.64	-0.89	-1.77	-2.41	-3.12	-2.77

(d) Extended model trained *from* original languages to *ban*

Table 9: **Main Results (the answer of Q2)**: ChrF++ scores of the extended model on 9 original language pairs grouped by available resources on both source and target sizes (Low, Mid, High).