Breaking Down Multilingual Machine Translation

Anonymous ACL submission

Abstract

While multilingual training is now an essential ingredient in machine translation (MT) systems, recent work has demonstrated that it has different effects in different multilingual settings, such as many-to-one, one-to-many, and many-to-many learning. These training settings expose the encoder and the decoder in a machine translation model with different data distributions. In this paper, we examine how different varieties of multilingual training contribute to learning these two components of the MT model. Specifically, we compare bilingual models with encoders and/or decoders initialized by multilingual training. We show that multilingual training is beneficial to encoders in general, while it only benefits decoders for low-resource languages (LRLs). We further find the important attention heads for each language pair and compare their correlations during inference. Our analysis sheds light on how multilingual translation models work and also enables us to propose methods to improve performance by training with highly related languages. Our many-to-one models for high-resource languages and one-to-many models for LRL outperform the best results reported by Aharoni et al. (2019).[1]

1 Introduction

Multilingual training regimens (Dong et al., 2015; Firat et al., 2016; Ha et al., 2016) are now a key element of natural language processing, especially for low-resource languages (LRLs) (Neubig and Hu, 2018; Aharoni et al., 2019). These algorithms are presumed to be helpful because they leverage syntactic or semantic similarities between languages, and transfer processing abilities across language boundaries.

In general, English is used as a central language due to its data availability, and three different multilingual training settings are considered: (1) *one-to-many*: training a model with languages pairs from English to many other languages. (2) *many-to-one*: training a model with languages pairs from many languages to English (3) *many-to-many*: training a model with the union of the above two settings' data. (1) and (3) can be used for English to other (En-X) translation, while (2) and (3) can be used for other to English (X-En) translation.

However, multilingual training has not proven equally helpful in every setting. Arivazhagan et al. (2019) showed that many-to-one training improves performance over bilingual baselines more than one-to-many does. In this paper we consider this result from the point of view of the components of the MT model. In the many-to-one setting, inputs of the model are from different language distributions so the encoder can be considered a multi-domain model, whereas the decoder is trained on a single distribution. In the one-to-many setting, it is the opposite: the encoder shares data, and the decoder is multi-domain. While there are recent studies analyzing multilingual translation models (Kudugunta et al., 2019; Voita et al., 2019a; Aji et al., 2020; Mueller et al., 2020), in general they do not (1) examine the impact of different multilingual training settings such as one-to-many and many-to-one, and (2) they do not examine the different components such as encoder and the decoder separately.

This motivates us to ask *"how do various types of multilingual training interact with learning of the encoder and decoder?"* To answer this question, we set up controlled experiments that decouple the contribution to the encoder and the decoder in various training settings. We first train multilingual models using many-to-one, one-to-many, or many-to-many training paradigms. We then compare training bilingual models with and without initializing the encoder or the decoder with parameters learnt by multilingual training. We find that, for LRLs, multilingual training is beneficial to both the encoder and the decoder. However, surprisingly, for high-resource languages (HRL), we found mul-

[1] We will release our scripts once accepted.

Lang.	az	be	gl	sk	ar	de	he	it
Size (K)	6	5	10	61	214	168	212	205

Table 1: Training data size.

tilingual training only beneficial to encoder but not to the decoder.

To further analyze the result, we examine *"to what degree are the learnt parameters shared across languages?"*. We use the head importance estimation method proposed by Michel et al. (2019) as a tool to identify the important attention heads in the model, and measure the consistency between the heads sets that are important for different language pairs. The results suggest that the encoder does share parameters across different languages in all settings. On the other hand, the decoder can treat the representation from the encoder in a language-agnostic way for X-En translation, and less parameter sharing is observed for En-X translation. Our analyses on parameter sharing also provides a possible explanation of Kudugunta et al. (2019)'s observation that the representation from the encoder is target-language-dependent .

Our investigation of how multilingual training works leads us to a method for improving MT models. With the comprehensive experiments in multilingual settings, for translation in HRL (Ar-En, De-En, He-En, It-En), we discover that fine-tuning multilingual model with target bilingual data outperforms the best results in Aharoni et al. (2019) by 2.99 to 4.63 BLEU score . With the analysis on the parameter sharing in the decoder, we are able to identify related languages. Fine-tuning jointly with the identified related languages boosts low-resource translation (En-Az, En-Be, En-Go, En-Sk) over the best results in Aharoni et al. (2019) by 1.66 to 4.44 BLEU score. Compared to Neubig and Hu (2018), our method does not require linguist knowledge, and thus may be more useful for less-studied low-resource languages.

In sum, our contributions are in three-fold. First, our experiments can be used as a diagnostic tool for multilingual translation to investigate how an encoder and a decoder benefit from multilingual training. Second, our results provide insights into how multilingual translation works. Third, we improve the translation models based on the findings from our analysis, showing a promising path for future research on multilingual machine translation.

2 Experimental Settings for Multilingual Training

Before stepping into our analysis, we first explain our experimental setup. The publicly available TED Talks Dataset (Qi et al., 2018) is used to train all our machine translation models. Following Neubig and Hu (2018), we break words into subwords with BPE jointly learnt over all source languages using the `sentencepiece` toolkit. The vocabulary size is 32000. We perform experiments with the Transformer architecture (Vaswani et al., 2017) using the hyper parameters same as in (Arivazhagan et al., 2019) [2]. All models are implemented and trained using *Fairseq* 0.10.0 (Ott et al., 2019). We trained multilingual translation models with 60 different languages on the TED Talks Dataset with the three settings described in Section 1: *one-to-many*, *many-to-one* and *many-to-many*. For *one-to-many* and *many-to-many* settings, we add a special language token to the input of the encoder to indicate the target language. Following Aharoni et al. (2019), we evaluate our models with BLEU score (Papineni et al., 2002; Post, 2018) on the selected 8 languages. They are representative for different language families (Qi et al., 2018). The size of the training is shown in Table 1.

3 How Multilingual Training Benefits Each Component

Previous studies have shown that the multilingual training results are generally stronger than the bilingual training (Arivazhagan et al., 2019). To understand how multilingual training benefits NMT, we analyze the effect of multilingual training on different components of an NMT model, specifically, the encoder and decoder.

3.1 Experiments Design

To study how multilingual training benefits each component, we train models on bilingual data with components initialized differently as follows:

- **Bilingual Only:** Models trained from scratch with no components initialized with parameters learnt from multilingual training.
- **Load encoder/decoder:** Models with trainable parameters of either encoder or decoder

[2] 6 layers in both the encoder and the decoder, 8 attention head, state dimension=512, ffn dimension=2048, label smoothing=0.1

Model		→ en							
		az	be	gl	sk	ar	de	he	it
All-En		9.1	15.2	27.4	25.4	23.9	28.3	27.9	31.5
All-All		8.1	12.6	22.8	24.6	21.7	27.1	26.1	31.1
Bilingual Only		2.1	1.4	2.8	18.5	28.5	32.0	34.8	35.7
All-En	Load Enc.	2.8	1.8	5.9	18.1	30.6	35.5	36.9	35.7
	Load Dec.	2.5	1.8	5.7	17.8	27.2	30.3	33.2	35.7
	Freeze Enc.	5.0	6.0	19.3	26.3	28.4	33.0	33.6	36.4
	Freeze Dec.	3.4	4.1	16.9	24.7	28.1	31.4	33.4	33.6
	Load Both	**11.5**	19.0	29.9	28.00	30.4	33.1	36.2	36.7
All-All	Load Enc.	5.4	7.0	20.6	28.0	30.9	35.7	37.1	38.1
	Load Dec.	1.4	0.5	0.9	20.4	28.9	32.2	34.0	35.3
	Freeze Enc.	3.3	5.0	9.3	23.8	25.9	32.4	32.2	34.2
	Freeze Dec.	2.0	6.2	20.1	26.9	30.1	34.4	35.9	36.8
	Load Both	11.3	**19.4**	**31.8**	**29.6**	**31.3**	**36.0**	**37.8**	**38.7**

Table 2: Results of translating into English. **All** in the model name refers to using all 60 languages.

initialized with parameters learnt from multilingual training.

- **Load both:** Models with parameters of both encoder and decoder initialized with parameters learnt from multilingual training. This can be seen as fine-tuning the multilingual model on bilingual data.

The motivation for this paradigm is that if multilingual training is beneficial to a component, then initializing the parameters of that component should result in improvements over random initialization and training on only bilingual data. If *load encoder* outperforms *bilingual only*, then we can say that multilingual training is beneficial for the encoder, and if *load decoder* outperforms we can make the analogous conclusion for the decoder. Thus comparing these models reveals how each component benefit from multilingual training.

We also consider a *load and freeze* setting (Thompson et al., 2018), where we initialize a component from a multilingual model and freeze its weights when fine-tuning on bilingual data. For example, in the *load decoder* setting, we train the loaded decoder with a randomly initialized encoder. We suspect that learning with randomly initialized component might ruin the other component which is well-trained with multilingual data, especially in the beginning of the training. Thus, we additionally experiment with this *load and freeze* setting to ensure the multilingual-trained component is not deteriorated.

3.2 Results and Discussion

The overall results of X-En and En-X are shown in Table 2 and Table 3, respectively. Because they are highly dependent on the training data size (Table 1), we discuss the results in two groups: HRL (HRL; referring to *ar*, *de*, *he*, and *it*) and LRL (LRL; referring to *az*, *be*, *gl*, *sk*).[3]

3.2.1 Low-Resource Language Results

For LRLs, we find that multilingual training is generally beneficial to both the encoders and the decoders in all of the three multilingual models. Both *load encoder* and *load and freeze decoder* can achieve performance better than the bilingual baseline. This suggests that the parameters in the encoder and the decoder learnt by multilingual training do contain information that is not effectively learnt from the smaller bilingual data.

The results also suggest that multilingual training is more beneficial for the encoders than for the decoders. In all cases, either *load encoder* or *freeze encoder* outperforms both *load decoder* and *load and freeze decoder*. However, multilingual training of the encoder and the decoder are complementary; loading both the encoder and the decoder can usually improve the performance over loading only one component.

[3] *sk* has intermediate size, and its behavior is not always consistent with the other LRL.

Model		en→							
		az	be	gl	sk	ar	de	he	it
En-All		4.9	9.0	24.2	21.9	15.1	27.9	24.1	33.3
All-All		3.1	6.2	20.5	18.4	12.7	24.5	21.1	30.5
Bilingual Baseline		1.3	1.9	3.9	13.1	15.6	27.1	25.4	32.0
En-All	Load Enc.	3.0	5.6	16.7	21.7	**17.2**	30.0	27.5	34.6
	Load Dec.	1.3	2.0	8.1	17.4	16.0	26.7	25.8	32.6
	Freeze Enc.	2.7	4.6	14.7	21.1	9.7	24.4	22.6	33.4
	Freeze Dec.	1.9	3.7	14.5	17.6	16.2	28.0	25.9	33.3
	Load All	**6.4**	**14.7**	**26.9**	**23.5**	17.1	**31.1**	**28.2**	**34.9**
All-All	Load Enc.	2.4	5.0	16.9	21.4	16.9	29.8	27.4	34.4
	Load Dec.	1.1	2.2	7.0	17.5	16.0	28.1	25.6	32.5
	Freeze Enc.	2.1	0.5	12.6	19.4	10.2	24.4	24.3	33.1
	Freeze Dec.	0.9	4.7	15.0	18.8	15.1	27.5	24.9	32.4
	Load All	6.1	13.0	26.4	23.2	17.0	30.3	27.9	34.6

Table 3: Results of translating from English. **All** in the model name refers to using all 60 languages.

3.2.2 High-Resource Language Results

On HRLs, we find that multilingual training is generally beneficial to the encoders in all of the three multilingual models, while it is not beneficial for the decoders in some settings. *Load encoder* always outperform the baseline models, but for the All-En model on X-En translation, and the All-All model on En-X translation, neither *load decoder* nor *load and freeze decoder* outperform the baseline model.

We also observe that multilingual training is generally more beneficial to the encoders than to the decoders. In all of the cases, *load encoder* can achieve performance competitive to *load both* (better or less by within 1 BLEU score). However, in all of the cases, both *load decoder* and *load and freeze decoder* have performance worse than *load both*. Therefore, multilingual training is not as beneficial to the decoders as to the encoders.

3.3 Discussion

For LRL, because the size of bilingual training data is small, it is not surprising that multilingual training is beneficial for both the encoder and the decoder. However, our results are somewhat more surprising for HRL — it is not trivial that multilingual training is not as beneficial. In the next section, we focus on explaining the phenomena observed on HRL by investigating how parameters are shared across languages.

4 How Multilingual Parameters are Shared in Each Component

Given the previous results, we are interested in exactly *how* parameters are shared among different language pairs. Given that we are using the Transformer architecture, for which multi-head attention is a fundamental component, we use the attention heads as a proxy to analyze how multilingual models work differently when translating between different languages. Specifically, we analyze our models by identifying the attention heads that are important when translating a language pair. Measuring the consistency between the sets of important attention heads for two language pairs gives us hints on the extent of parameter sharing.

4.1 Head Importance Estimation

First, we provide some background on head importance estimation, specifically the method proposed by Michel et al. (2019).

Given a set of multi-head attention modules, each of which can be written as

$$\text{MHAtt}(x) = \sum_{h=1}^{N_h} \xi_h \text{Att}_{W_q^{(h)}, W_k^{(h)}, W_v^{(h)}}(x), \quad (1)$$

where N_h is the number of attention heads, and $\xi_h = 1$ for all h.

The importance of a head can be estimated as

$$\tilde{I}_h = \mathbb{E}_{x \sim X} \left| \frac{\partial L(x)}{\partial \xi_h} \right|. \quad (2)$$

given a loss function L and input X. Then, the importance score of each head in an attention module is normalized

$$I_h = \frac{\tilde{I}_h}{\sqrt{\sum_i^{N_h} I_h^2}}. \quad (3)$$

Note that when the input X is different, the estimated importance score can be different. Therefore, when different language pairs are fed in, the important heads identified can be different. We denote the set of attention head scores estimated on translation from language l_a to language l_b as $H(l_a, l_b)$. We denote the scores of attention heads in a component by using superscript. For example, H^{enc} represents the scores of the heads in a encoder.

4.2 Measuring Parameter Sharing by Correlation of Head Scores

With the attention head importance scores estimated by Equation 3, we can investigate how parameters are shared across languages. For each of the En-All, All-En, All-All multilingual models, we estimated a set of head-importance scores $H(l_a, l_b)$ for each language pair (l_a, l_b) in the training setting. We calculate the head scores with the training loss function (MLE with label smoothing) and 100K randomly sampled sentences in the training set.

To investigate how much parameters are shared by two pairs of languages (l_a, l_b) and (l_c, l_d), we measure the agreement between $H(l_a, l_b)$ and $H(l_c, l_d)$. If a head is important for both of (l_a, l_b) and (l_c, l_d), then important parameters for translating are shared. Thus high agreement suggests high parameter sharing.

To quantify the agreement between two score sets, we use Spearman's rank correlation (Spearman, 1987). A rank-based correlation metric is used because the importance estimation was originally proposed to order attention heads in a model. Higher correlation implies higher agreement and thus implies higher parameter sharing. For each of the En-All, All-En, All-All models, we calculate the correlation between $H(l_a, l_b)$ and $H(l_c, l_d)$ for all language pairs (l_a, l_b) and (l_c, l_d) that are used to train the model. The detailed correlation computation process can be found in Appendix A. We plot the correlation matrices of the head scores (included in appendix) and summarize them in Table 4.

Model	Lang. Pair	H^{enc}	H^{dec}
All-En	X-En	.871 (.086)	.973 (.023)
En-All	En-X	.806 (.153)	.720 (.150)
All-All	X-En	.898 (.073)	.967 (.029)
All-All	En-X	.813 (.126)	.762 (.141)

Table 4: Correlation between the attention head scores when estimated using different language pairs.

4.3 How Multilingual Translation Models Share

Results in Table 4 combined with Section 3 provides the insights into how multilingual translation models work with respect to cross-lingual sharing:

Encoder for En-X: It is natural that the encoder from En-X likely benefit from multilingual training because it can generate representations tailored for different target languages with shared parameters. En-X is a set of language pairs where the source language is always English. Therefore, if the prepended target language token is ignored, the inputs of the encoders for all pairs in En-X are from one identical distribution. This is in contrast to X-En pairs, where the inputs are in different languages. However, for the encoders, we observe from Table 4 that the average correlation scores of En-X pairs (0.806 and 0.813), are lower than the correlation scores of X-En pairs (0.871 and 0.898). Kudugunta et al. discovers that the representation of the encoder is target-language-dependent. Thus we conjecture that some parameters may be used to generate representation tailored for the target languages. At the same time, since the inputs are from a single distribution (English) for different target languages, a large portion of parameters may still be shareable across target languages. Therefore, in this case, multilingual training is beneficial.

Encoder for X-En: For X-En language pairs, the input of the encoder is multilingual, which means the input from different X-En language pairs has distinct distribution. However, the correlation between different source languages is still high. It shows that high parameters sharing in the encoder is possible.

Decoder for En-X: The decoders for En-X have the lowest correlation. From the correlation matrix, we do see some parameter sharing between some language pairs. However, larger model capacity might be required for a model to be proficient in

all the languages.

Decoder for X-En: The decoder have average correlation as high as 0.973 and 0.967 for All-En and All-All models respectively. This suggests that to decode intermediate representation encoded by the encoder, the decoder use almost the same set of parameters. However, Kudugunta et al. shows that the representation encoded by the encoder is not language-agnostic. A possible explanation is that the important parameters of the decoder are highly determined by the target output, which is always in English. Therefore, even though the encoder representation is not language-agnostic, it is still difficult to learn parameters reflecting the difference. It suggests why multilingual training does not benefit the decoder in the X-En setting. The set of English sentences is almost the same for all the HRL pairs in the TED Talks dataset, so multilingual training can hardly provide more unique English sentences than bilingual training does. If the decoder is dedicated for generation, multilingual training cannot expose the decoder to more diverse data. Therefore the multilingually trained decoder does not perform better than the bilingual one.

5 Improving Translation Based on the Degree of Parameter Sharing

Insights from the previous section provide us with a new way to choose languages for multilingual training. In previous work (Lin et al., 2019), choosing on languages with similar linguistic properties is a popular practice. However, Mueller et al. (2020) found the effect is highly language-dependent. Sometimes training with similar languages might be worse than training on a set of unrelated languages. Here we otherwise propose an entirely model-driven way to find related languages to improve multilingual translation models. We explore choosing languages where parameters can be better shared.

5.1 Improving X-En by Related En-X Pairs

In the All-All model, we notice low parameter sharing between En-X and X-En pairs. The average correlation between $H^{enc}(En, X)$ and $H^{enc}(X, En)$ is 0.44 (std: 0.17). The average correlation between $H^{dec}(En, X)$ and $H^{dec}(X, En)$ is 0.49 (std: 0.13). It provides a possible explanation why training with both the En-X and the X-En pairs only brings little improvement over training with only En-X alone or with X-En alone.

The low correlation combined with results in Section 3 motivate us to experiment on improving X-En with related En-X pairs. Section 3 shows that the multilingual decoder has less advantage than the encoder. This may suggest the inefficiency of parameter sharing in the decoder. Therefore we experiment on choosing a set of related languages based on the degree of parameter in the decoder. We choose the language set L such that for all $l \in L$, the average correlation $\frac{1}{60}\sum_{l_i=1}^{60} \mathrm{Corr}(H^{dec}(En, l), H^{dec}(l_i, En))$ is higher than 0.60.

Results are shown in Table 5. Even though fine-tuning on related languages improves the overall performance, it is not better than fine-tuning on the All-En pairs only. Also, the average correlation between $H^{dec}(En, l_a)$ and $H^{dec}(l_b, En)$ is not improved. Our experiment demonstrates the difficulty of sharing parameters between All-En pairs and En-All pairs. We leave this problem for future work.

5.2 Improving En-X by Language Clusters

The low correlation between attention head scores of language pairs motivates us to improve the performance of En-X using related language pairs. As shown in Table 4, the decoders have the lowest correlation scores. We conjecture that it is due to the difficulty of sharing parameters between distant languages. Thus, we seek for finding related language sets, in each of which parameters can be shared.

Again, we resort to the attention head importance scores to find the related languages. Our intuition is that related languages would share many parameters in between and training a model on related languages would be helpful. As a sanity check of our idea, we first use t-SNE (Maaten and Hinton, 2008) to reduce the dimension of head-importance scores $H(l_a, l_b)$. We only focus on heads in the decoders, because the correlation score between $H_{(\mathrm{En},l_c)}$ and $H_{(\mathrm{En},l_d)}$ is lower in average for the decoders. The result visualized in Figure 1 illustrates that, the distance between $H_{(\mathrm{En},l_c)}$ and $H_{(\mathrm{En},l_d)}$ tend to be shorter if languages l_c and l_d are linguistically related. Hence, determining related languages with head score $H_{(\mathrm{En},l)}$ should be reasonable.

We then fine-tune multilingual models on related language clusters. Related languages clusters are determined by k-mean++ (Arthur and Vassilvitskii,

Model	az	be	gl	sk	ar	de	he	it
All-All	8.1	12.6	22.8	24.6	21.7	27.1	26.1	31.1
+ f.t. on All-En	10.5	17.5	29.7	28.1	25.9	31.3	30.5	34.0
+ f.t. on All-En & related	10.5	17.4	28.3	27.0	25.1	30.0	29.9	32.7

Table 5: Performance of All-All model fine-tuned on All-En pairs and fine-tuned on the union of All-En pairs and related En-All languages.

Model	az	be	gl	sk	ar	de	he	it
Bilingual Baseline	1.3	1.9	3.9	13.1	15.6	27.1	25.4	32.0
All-All	3.1	6.2	20.5	18.4	12.7	24.5	21.1	30.5
All-All w/ f.t. on related clusters	7.9	12.8	27.5	24.9	-	30.2	27.0	35.4
All-All w/ f.t. on random groups	6.9	13.3	22.5	24.3	-	-	27.5	35.2
En-All	4.9	9.00	24.2	21.9	15.1	27.9	24.1	33.3
En-All w/ f.t. on related clusters	7.9	13.9	21.0	26.2	16.7	30.4	27.1	35.4
En-All w/ f.t. on random groups	7.0	13.1	23.1	24.7	-	-	27.6	35.2
Load En-All w/ f.t. on closest	7.8	15.2	28.6					

Table 6: Performance of En-All model without and with fine-tuning on language clusters.

2007) with $k = 5$. We consider clusters that cover all of the four low-resource languages. For the All-All model, one of the cluster we consider contains Be, Gl, De, He, It, and the other one contains Az. For the En-All model, we also experiment with two clusters. One includes Ar, De, He, It, and the other includes Az, Be, Gl, Sk. As a baseline, we also experiment with random groups. They are groups generated by randomly splitting the 59 target languages.

The results are shown in Table 6. For both the En-All and the All-All model, except En-Gl, fine-tuning on clusters can improve performance on all the considered language pairs consistently. For LRLs, fine-tuning on related language clusters is also better than fine-tuning on random groups in general. To verify whether this improvement is brought by increased parameter sharing in the decoders, we check the correlation between H^{dec} after fine-tuning. The results shown in Table 7 shows improvements after fine-tuning on the clusters.

For low-resource language pairs En-Az, En-Be, En-Sk on the En-All model, we notice that only few languages are highly correlated with them (with correlation > 0.80). Therefore, we also experiment with fine-tuning the En-All model with only the language pairs with high correlation scores (> 0.80) for each of the three pairs , which boosts the performance of En-Be to 15.2 and En-Sk to 28.6.

Model	H^{dec} w/o f.t.	H^{dec} w/ f.t.
All-All	.762 (.141)	.894 (.069)
En-All (HL)	.855 (.066)	.866 (.065)
En-All (LL)	.826 (.096)	.834 (.091)

Table 7: Correlation between the decoder attention head scores when estimated using the language pairs in the cluster. HL and LL represent the cluster that includes HRL and the one that includes LRL respectively.

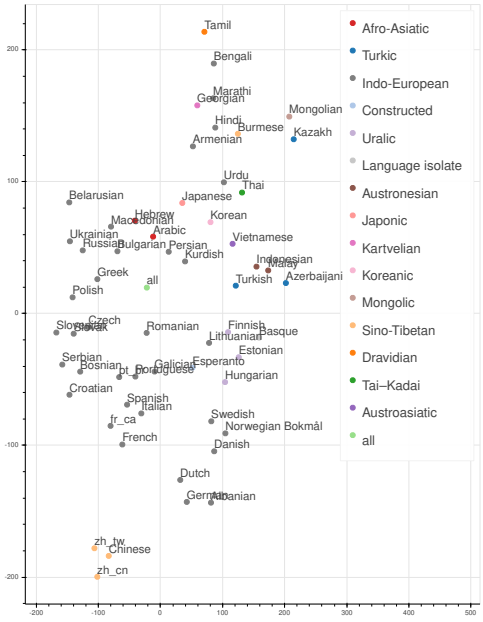

Figure 1: Visualization of the En-All decoder head scores of languages by t-SNE.

6 Related Work

The early attempts of multilingual training for machine translation use a single model to translate between multiple languages (Dong et al., 2015; Firat et al., 2016; Ha et al., 2016). Those works find multilingual NMT models are appealing because they not only give us a simple paradigm to handle mapping between multiple languages, but also improve performance on low and zero-resource languages pairs (Gu et al., 2018). However, how multilingual training contributes to components in the translation model still remains unknown.

There are some attempts at analyzing and explaining the translation models. Thompson et al. (2018) analyze the contribution of different components of NMT model to domain adaptation by freezing the weights of components during continued training. Arivazhagan et al. (2019) provide an comprehensive study on the state-of-the-art multilingual NMT model in different training and testing scenarios. Sachan and Neubig (2018) experiment with different parameter sharing strategies in Transformer models, showing that sharing parameters of embedding, key and query performs well for *one-to-many* settings. Artetxe et al. (2020) shows the strong transferability of monolingual representation to different languages. The intermediate representation of BERT can be language-agnostic if we freeze the embeddings during training. The deficiency of the *one-to-many* setting is explored in (Johnson et al., 2017). They find only the *many-to-one* setting consistently improves the performance across languages. Wang et al. (2018) also explore problems of the *one-to-many* setting, and show language-specific components are effective to improve the performance. Voita et al. (2019a) analyzes how generated sentences of NMT models are influenced by context in the encoder and decoder. The attempt to investigate encoder and decoder separately is similar to our work.

Multi-head attention has been shown effective in different NLP tasks. Beyond improving performance, multi-head attention can help with subject-verb agreement (Tang et al., 2018), and some heads are predictive of dependency structures (Raganato and Tiedemann, 2018). Htut et al. (2019) and Clark et al. (2019) report that heads in BERT attend significantly more to words in certain syntactic position. They show some heads seem to specialize in certain types of syntactic relations. Michel et al. (2019), Voita et al. (2019b), and Behnke and Heafield (2020) study the importance of different attention heads in NMT models, and suggest that we can prune those attention heads which are less important. Brix et al. (2020) also shows pruning NMT models can improve the sparsity level to optimize the memory usage and inference speed.

However, all previous works do not directly investigate how encoder and decoder of NMT models benefit from multilingual training, which is the key question of why multilingual training works. To our best knowledge, we are the first to tackle the question, and our analysis can be used to further improve multilingual NMT models.

7 Conclusion

In this work, we have the following findings: 1) In Section 3, we examine how multilingual training contributes to each of the components in a machine translation model. We discover that, while multilingual training is beneficial to the encoders, it is less beneficial to the decoders. 2) In Section 4, our analysis of important attention heads provides insight into the behavior of multilingual components. Results suggest that the encoder in the En-All model may generate target-language-specific representation, while the behavior of the decoder of the All-En model may be source-language-agnostic. In addition, in the All-All model, we observe indications of lower parameter sharing between X-En pairs and En-X pairs. 3) In Section 5, we explore approaches to improve the model based on our findings. On En-X translation, we outperform the best results in (Aharoni et al., 2019). With our proposed analysis as diagnostic tools, future work may further improve the multilingual systems.

Our findings provide some possible future directions. First, parameter sharing between En-X and X-En pairs in the All-All model seems low. Improving the sharing may improve the performance. Second, the decoder in the All-En model seems to behave in a source-language-agnostic way. It may not be optimal since the representation from the encoder is not source-language-agnostic (Kudugunta et al., 2019). To mitigate this issue, either the encoder is required to encode inputs into language-agnostic representation, or the decoder should behave in different ways according to the input representation. We leave the exploration in future work.

References

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

Alham Fikri Aji, Nikolay Bogoychev, Kenneth Heafield, and Rico Sennrich. 2020. In neural machine translation, what does transfer learning transfer? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7701–7710, Online. Association for Computational Linguistics.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

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.

David Arthur and Sergei Vassilvitskii. 2007. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, page 1027–1035, USA. Society for Industrial and Applied Mathematics.

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

Christopher Brix, Parnia Bahar, and Hermann Ney. 2020. Successfully applying the stabilized lottery ticket hypothesis to the transformer architecture. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3909–3915, Online. Association for Computational Linguistics.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*.

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

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

Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor O.K. Li. 2018. Universal neural machine translation for extremely low resource languages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 344–354, New Orleans, Louisiana. Association for Computational Linguistics.

Thanh-Le Ha, Jan Niehues, and Alexander Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. *arXiv preprint arXiv:1611.04798*.

Phu Mon Htut, Jason Phang, Shikha Bordia, and Samuel R Bowman. 2019. Do attention heads in bert track syntactic dependencies? *arXiv preprint arXiv:1911.12246*.

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.

Sneha Kudugunta, Ankur Bapna, Isaac Caswell, and Orhan Firat. 2019. Investigating multilingual NMT representations at scale. 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 1565–1575, Hong Kong, China. Association for Computational Linguistics.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. 2019. Choosing transfer languages for cross-lingual learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3125–3135, Florence, Italy. Association for Computational Linguistics.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, pages 14014–14024.

Aaron Mueller, Garrett Nicolai, Arya D. McCarthy, Dylan Lewis, Winston Wu, and David Yarowsky. 2020. An analysis of massively multilingual neural machine translation for low-resource languages. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3710–3718, Marseille, France. European Language Resources Association.

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

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. 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.

Ye Qi, Devendra Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. 2018. When and why are pre-trained word embeddings useful for neural machine translation? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 529–535, New Orleans, Louisiana. Association for Computational Linguistics.

Alessandro Raganato and Jörg Tiedemann. 2018. An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 287–297, Brussels, Belgium. Association for Computational Linguistics.

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, Brussels, Belgium. Association for Computational Linguistics.

Charles Spearman. 1987. The proof and measurement of association between two things. *The American journal of psychology*, 100(3/4):441–471.

Gongbo Tang, Mathias Müller, Annette Rios, and Rico Sennrich. 2018. Why self-attention? a targeted evaluation of neural machine translation architectures. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4263–4272, Brussels, Belgium. Association for Computational Linguistics.

Brian Thompson, Huda Khayrallah, Antonios Anastasopoulos, Arya D. McCarthy, Kevin Duh, Rebecca Marvin, Paul McNamee, Jeremy Gwinnup, Tim Anderson, and Philipp Koehn. 2018. Freezing subnetworks to analyze domain adaptation in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 124–132, Brussels, Belgium. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30:5998–6008.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019a. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019b. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Yining Wang, Jiajun Zhang, Feifei Zhai, Jingfang Xu, and Chengqing Zong. 2018. Three strategies to improve one-to-many multilingual translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2955–2960, Brussels, Belgium. Association for Computational Linguistics.

A Correlation of Head Scores

Here we detail the computation of the correlation of head scores for two pairs of languages (l_a, l_b) and (l_c, l_d). The steps are as follow:

1. The the two language pairs' head importance scores $H(l_a, l_b)$ and $H(l_c, l_d)$ are estimated with Equation 3. Since there are many heads in a Transformer model, both $H(l_a, l_b)$ and $H(l_c, l_d)$ are vectors.

2. We flatten the scores in $H(l_a, l_b)$ and $H(l_c, l_d)$ into two arrays of scalars. We treat the two arrays as the observations of two variables. Then we use Spearman correlation to compute the correlation between the two variables. In other words, the input of the Spearman correlation function is the two arrays.

B Related Related Language Pairs

The related language pairs used in Section 5 are: en-zh_cn en-it en-es en-vi en-zh_tw en-nl en-fr en-fr_ca en-th en-pt_br en-ru.

C Language Clusters

En-All model:

- en-ja en-ko en-zh en-zh-cn en-zh-tw
- en-az en-be en-bs en-cs en-da en-eo en-et en-eu en-fi en-gl en-hr en-hu en-lt en-mk en-nb en-pl en-sk en-sl en-sq en-sr en-sv en-tr en-uk
- en-bn en-hi en-hy en-ka en-ku en-mr en-my en-ta en-th en-ur
- en-ar en-bg en-de en-el en-es en-fa en-fr en-fr-ca en-he en-id en-it en-ms en-nl en-pt en-pt-br en-ro en-ru en-vi
- en-kk en-mn

All-All:

- en-be, en-bg, en-bs, en-cs, en-de, en-el, en-es, en-fr, en-fr-ca, en-gl, en-he, en-hr, en-it, en-lt, en-mk, en-pl, en-pt, en-pt-br, en-ro, en-ru, en-sk, en-sl, en-sq, en-sr, en-uk
- en-ar, en-fa, en-ja, en-ko, en-th, en-vi, en-zh, en-zh-cn, en-zh-tw
- en-bn, en-hi, en-hy, en-ka, en-ku, en-mr, en-my, en-ur
- en-az, en-da, en-eo, en-et, en-fi, en-hu, en-id, en-ms, en-nb, en-nl, en-sv, en-tr
- en-eu, en-kk, en-mn, en-ta

D Random Clusters

- en-pt en-fa en-fr en-kk en-hi en-da en-hu en-de en-nl en-ar en-hy en-zh-cn
- en-sr en-fi en-be en-ko en-ru en-ur en-it en-id en-el en-eu en-sq en-zh en-bs en-bn en-sv en-bg en-my en-ro en-ta en-sl en-et en-ku en-mn en-uk en-he en-tr
- en-mk en-mr
- en-ms en-pl en-pt-br en-cs en-zh-tw en-es
- en-vi en-eo en-hr en-nb en-fr-ca en-az en-sk en-ka en-lt en-th en-ja en-gl

Theses random clusters are generated by (1) shuffling the 59 languages, (2) randomly selecting positions. The results 5 segments separated by the 4 positions are the 5 clusters.

E Closest Languages

The closest languages used in Section **??** are:

- Az: en-az en-eu en-fi en-tr
- Be: en-be en-it en-uk
- Gl: en-gl en-pt en-es en-lt en-it en-pt_br

F Experimental Details

- Infrastructure: All the experiments can be conducted on one single RTX 2080Ti GPU.
- Evaluation: We report the BLEU score calculated by FairSeq.
- Version of FairSeq: We use v0.10.0 (https://github.com/pytorch/fairseq/tree/v0.10.0)
- Dataset: It can be downloaded from https://github.com/neulab/word-embeddings-for-nmt.

Figure 2: Correlation matrix between language pairs. The top-left corner is the correlation between the encoder head scores H^{enc}, while the bottom-right corner is the correlation between the decoder head scores H^{dec}. The top matrix is the correlation matrix of the All-All model, while the bottom-left and the bottom-right ones are the correlation matrices of the All-En and the En-All models respectively.

Figure 3: Correlation matrix between language pairs after fine-tuning on related languages. The top-left corner is the correlation between the encoder head scores H^{enc}, while the bottom-right corner is the correlation between the decoder head scores H^{dec}.

Figure 4: Correlation matrix between language pairs after fine-tuning on the languages clusters. The first figure is the matrix of the fine-tuned All-All model. The second and the third ones are the matrix of the En-All model fine-tuned on the language clusters containing the high-resource and the LRL respectively. The top-left corner is the correlation between the encoder head scores H^{enc}, while the bottom-right corner is the correlation between the decoder head scores H^{dec}.