A Closer Look at Transformer Attention for Multilingual Translation

Jingyi Zhang[1]**, Hongfei Xu**[2]**, Kehai Chen**[3] **and Gerard de Melo**[1]
[1]Hasso Plattner Institute, University of Potsdam, Germany
[2]Zhengzhou University, Henan, China
[3]School of Computer Science and Technology, Harbin Institute of Technolgy, Shenzhen, China
Jingyi.Zhang@hpi.de, hfxunlp@foxmail.com, chenkehai@hit.edu.cn
gerard.demelo@hpi.de

Abstract

Transformers are the predominant model for machine translation. Recent studies also showed that a single Transformer model can be trained to learn translation for multiple different language pairs, achieving promising results. In this work, we investigate how multilingual Transformer models pay attention when translating different language pairs. To achieve this, we first conduct automatic pruning to eliminate a large number of noisy heads and then assess the functions and behaviors of the remaining heads in both self-attention and cross-attention. We find that different language pairs, in spite of having different syntax and word orders, tend to share the same heads for the same functions, such as syntax heads and reordering heads. However, the different characteristics of different language pairs can clearly cause interference in function heads and affect head accuracies. Additionally, we reveal an interesting behavior of the Transformer cross-attention: the deep-layer cross-attention heads work in a cooperative way to learn different options for word reordering, which may be caused by the nature of translation tasks having multiple different gold translations in the target language for the same source sentence.[1]

1 Introduction

For traditional statistical machine translation, such as phrase-based translation (Koehn et al., 2003), the translation process is very clear: source phrases are translated into target phrases according to translation rules and then target phrases are reordered to ensure the fluency of the target sentence. However, in state-of-the-art neural translation models (Bahdanau et al., 2014; Vaswani et al., 2017; Chen et al., 2018), how the model learns to translate is substantially less obvious. The behavior of the Transformer model (Vaswani et al., 2017) remains particularly hazy, as it contains many different self- and cross-attention heads in different layers.

A number of existing studies conducted analyses of functions and behaviors of attention heads in Transformer translation models. Voita et al. (2019b) found that the Transformer attention is noisy, as most of the Transformer heads can be pruned away without significant loss in translation quality. They also identified three important functions of self-attention in the Transformer encoder, such as heads focusing on syntax. Ferrando and Costa-jussà (2021) demonstrated that the cross-attention of the Transformer model frequently attends to uninformative source words to balance the contribution of source and target context for predicting the next word. Chen et al. (2020) showed that some cross-attention heads learn alignment for the current target word, achieving higher accuracies than cross-attention heads that learn alignment for the next target word. However, these methods only analyzed attention in bilingual models, not for multilingual Transformer models.

Multilingual translation, i.e., training a single Transformer to learn translation for multiple different language pairs, has received much attention in recent years and obtained promising results (Wang et al., 2020; Kim et al., 2021; Pires et al., 2023). A number of studies investigated how a multilingual Transformer learns to translate different language pairs. Several of these (Lin et al., 2021; Wang et al., 2020; Xie et al., 2021) learned language-dependent weight masks to identify language-dependent subnetworks. Pires et al. (2023) trained the multilingual Transformer to learn language-specific layers and improved translation quality. Chiang et al. (2022) and Kim et al. (2021) assessed how different language pairs share important heads in multilingual Transformer models.

However, prior work has not yet studied the specific functions and behaviors of different attention heads in multilingual Transformer models. In this

[1] Code and scripts for reproducing our results can be found https://github.com/jingyiz/multilingual-translation-attention-head-analysis.

Proceedings of the Eighth Conference on Machine Translation (WMT), pages 496–506
December 6–7, 2023. ©2023 Association for Computational Linguistics

paper, we investigate functions and behaviors[2] of both self-attention and cross-attention for multilingual translation. We find that different language pairs with different syntax and different word orders tend to share the same heads for the same functions (such as syntax heads and reordering heads), but the different characteristics of different language pairs can clearly cause interference in function heads and affect head accuracies compared to bilingual models. We further obtain an interesting finding about how the Transformer learns word reordering: different cross-attention heads in deep layers work in a cooperative way to learn different options for reordering. This may result from the fact that there are multiple different gold translations (reorderings) in the target language for the same source sentence[3].

2 Related Work

There are a number of studies on analyzing layer representations of different Transformer layers. Voita et al. (2019a) used canonical correlation analysis and mutual information estimators to study how information flows across Transformer layers for different learning objectives. Kudugunta et al. (2019) used Singular Value Canonical Correlation Analysis (SVCCA) to analyze how representations evolve in a multilingual translation model. Xu et al. (2021b) analyzed how word translation evolves in Transformer layers and showed that translation already happens progressively in encoder layers and even in the input embeddings, by measuring word translation accuracy of different Transformer layers. These methods did not analyze the specific functions of attention heads.

Other prior work analyzed Transformer attention to better understand a particular aspect of the translation process. Tang et al. (2021) analyzed Transformer attention for negation translation and showed that negation is often rephrased during training, which can make it more difficult for the model to learn a reliable link between source and target negation. Tang et al. (2018) analyzed Transformer cross-attention for learning word sense disambiguation (WSD) and showed that cross-attention is likely to distribute more attention to the ambiguous noun itself rather than context tokens, in comparison to other nouns, which suggests that the Transformer learns to encode contextual information necessary for WSD in the encoder hidden states. Additionally, Tang et al. (2018) also noticed that, from shallow layers to deep layers, the cross-attention accuracy for aligning the next target word first increases and then decreases. However, we our study is the first to reveal the cooperative behavior of cross-attention heads.

There is also prior work that studied representation sharing in multilingual translation. Firat et al. (2016) proposed a multiway, multilingual model with language-specific encoders and decoders and showed result quality improvements over models trained on only one language pair. Several authors (Zhang et al., 2021; Bapna and Firat, 2019; Zhu et al., 2021) considered language-dependent gating and adaptation for layer representations. Xu et al. (2021a) proposed parallel encoder and decoder layers with language-dependent weighted layer aggregation. Wang et al. (2019) presented a universal representer to replace both encoder and decoder models to enable parameter sharing between encoder and decoder and they made the representer sensitive for specific languages using language-sensitive embedding, attention, and discriminator. Zhu et al. (2020) incorporated a language-aware interlingua into the encoder–decoder architecture, which enables the model to learn a language-independent representation from the semantic spaces of different languages, while still allowing for language-specific specialization of a particular language pair. Additionally, Shaham et al. (2023) showed that controlling the proportion of each language pair in the training data can balance the amount of interference between languages in multilingual models. Yuan et al. (2023) developed a detachable model by assigning each language (or group of languages) to an individual branch that supports plug-and-play training and inference with a novel efficient training recipe. Xu et al. (2023) investigated how to utilize intra-distillation to learn more language-specific parameters and then showed the importance of these language-specific parameters. However, these methods did not investigate the head functions in multilingual models.

[2] Following Voita et al. (2019b)'s work, we use a weight-based method for analyzing attention head behaviors. It is also possible to use a norm-based method (Kobayashi et al., 2020), which may provide a more detailed interpretation of the inner workings of Transformers compared to weight-based methods in some cases.

[3] In the training data of translation models, it is rather rare that the same source sentence has multiple different translated target sentences, but it is very common that the same source phrase has multiple different translated target phrases. Therefore, translation models are able to learn to translate a source sentence into different target sentences.

	DeEn	FrEn	RoEn	EnDe	EnFr	EnRo	Average
R-bi	25.90	29.47	31.46	21.94	31.08	25.74	27.59
R-multi	25.88	29.85	34.07	21.70	31.00	26.17	28.11
R-finetune	26.48	29.91	34.96	22.61	31.79	27.21	28.82
R-prune ($\lambda = 25$)	26.31	29.76	34.87	22.31	31.46	27.12	28.63
R-prune ($\lambda = 35$)	26.23	29.56	34.82	22.05	31.40	27.21	28.54

Table 1: Translation results (BLEU) on the test sets.

Train		
DeEn	2.5M	Europarl v7, TED2020, News-Commentary v11
FrEn	2.5M	Europarl v7, TED2020, News-Commentary v11
RoEn	0.9M	Europarl v8, TED2020, SETIMES2
Valid		
DeEn	5,014	newstest2009, newstest2010
FrEn	5,014	newstest2009, newstest2010
RoEn	1,999	newsdev2016
Test		
DeEn	9,006	newstest2011, newstest2012, newsdev2013
FrEn	9,006	newstest2011, newstest2012, newsdev2013
RoEn	1,999	newstest2016

Table 2: Datasets and their number of sentence pairs.

3 How Do Transformers Pay Attention for Multilingual Translation?

3.1 Methodology and Experimental Setup

Bilingual Baseline. We used the original Transformer model in its base setting (Vaswani et al., 2017) (i.e., the same model parameters, training parameters and inference parameters) as our bilingual baseline model and conducted translation experiments for six translation directions[4]: German↔English (De↔En), French↔English (Fr↔En), and Romanian↔English (Ro↔En). For each translation direction (such as De→En), we trained a Transformer model using the training data and validation data for this translation direction as shown in Table 2.[5] Following Vaswani et al. (2017), we trained each model for 100k training steps. However, because our training data size for a single translation direction is smaller than in their work, 100k training steps caused overfitting in our models. Therefore we computed the validation loss after each training epoch and then chose the best validation checkpoint for evaluation. Translation results on the test sets are given in Table 1 as R-bi.

	$\lambda = 25$	$\lambda = 35$
DeEn	74	54
FrEn	58	49
RoEn	74	52
EnDe	85	64
EnFr	65	53
EnRo	81	56
Shared	55	45
Total	144	

Table 3: Number of remaining heads after automatic pruning for different translation directions. "Shared" means the number of heads that remain for all six translation directions. "Total" refers to the original number of all heads before automatic pruning.

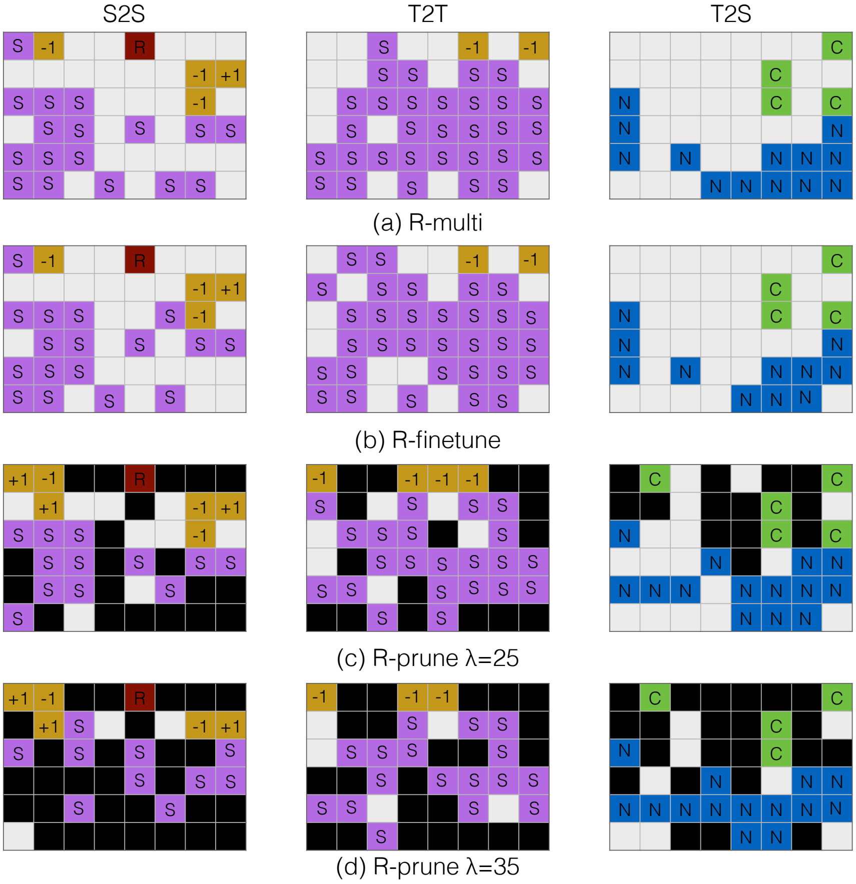
Multilingual Translation. For multilingual translation, we trained a single Transformer to learn translation for all six translation directions. We combined all training data in Table 2 together and added a special token at the beginning of each source sentence to indicate which target language we desire the model to generate, following Johnson et al. (2017). We used the same base setting of the original Transformer with 100k training steps for our multilingual model. During training of the multilingual model, we computed the validation loss for the combined validation data after each training epoch and found that the validation loss continuously decreased, so we used the final checkpoint of the multilingual model for evaluation. The evaluation results of the multilingual model are given in Table 1 as R-multi. In the results, we can observe that the multilingual model obtained comparable or higher translation quality compared to our bilingual baseline for different language pairs.

Finetuning. We then finetuned[6] the multilingual model for each translation direction using direction-

[4] We chose these language pairs because parallel sentences with gold-standard word alignments are available (Zhang and van Genabith, 2021) for these language pairs, which can be used to analyze target-to-source attention (alignment).

[5] For subword segmentation, we applied byte pair encoding (Sennrich et al., 2016) and learned a joint vocabulary of size 32k for all languages in our experiments.

[6] Finetuning a multilingual model for a given translation direction (i.e., multilingual pretraining) is very popular for low-resource language pairs and can significantly improve translation quality. We find that finetuning generally did not change the functions of different heads (see Figure 1) but did improve the accuracies of function heads for the given translation direction (see Table 12).



(a) R-multi

(b) R-finetune

(c) R-prune λ=25

(d) R-prune λ=35

Figure 1: Functional heads (red: rare word head (R); yellow: positional head (-1 and $+1$); purple: syntactical head (S); green: C-alignment head (C); blue: N-alignment head (N)) contained in (a) the multilingual model (R-multi); (b) finetuned models (R-finetune); (c) pruned models with $\lambda = 25$; (d) pruned models with $\lambda = 35$. From left to right, the three columns of figures represent S2S, T2T, and T2S attention. Each figure shows attention heads from the first layer to the last layer (top-down) and each layer contains 8 heads. Black denotes heads that are pruned away.

specific training and validation data.[7] During finetuning, we set the maximum number of finetuning steps to 50k and computed the validation loss after each training epoch, and finally used the best validation checkpoint for evaluation. We found that models for all six directions converged during finetuning (the best validation checkpoint is not the last checkpoint). The results of the finetuned models are given in Table 1 as R-finetune. As shown in Table 1, finetuning a pre-trained multilingual translation model for a specific translation direction can improve the translation quality for the given translation direction (i.e., R-finetune > R-multi). Table 1 also shows that the finetuned models can achieve higher translation quality compared to the bilingual models (R-bi) for all translation directions in our experiments.

Head Pruning. As shown by Voita et al. (2019b), Transformer attention is noisy, i.e., many attention heads carry no important function and can be pruned away without significant loss in translation quality. Following them, we conduct automatic pruning to identify important heads and analyze their functions. For each translation direction, we continue to finetune the already converged model (R-finetune) with a regularization loss (Louizos et al., 2017) along with the original translation loss to prune away useless heads. With the regularization loss, the model learns a 0/1 gate for each head. Heads with a 0 gate are pruned away. A weight λ is assigned to the regularization loss to control the amount of heads to be pruned, i.e., a higher weight for the regularization loss will result in more heads being pruned away. Translation results after head pruning are given in Table 1 and the number of remaining heads for each translation direction is

[7] For example, when we finetuned the multilingual model for the De→En direction, we only used training and validation data with German in the source and English in the target.

	DeEn	FrEn	RoEn	EnDe	EnFr	EnRo
Accu	0.31	0.36	0.19	0.33	0.36	0.26

Table 4: Accuracy of the rare word head.

listed in Table 3.[8] At $\lambda = 35$, roughly $2/3$ of all heads were pruned away and the average BLEU only decreased by 0.28. Table 3 also shows that different language pairs tended to share important heads, as most of the remaining heads remained for all six translation directions.

3.2 Head Function Analysis

We analyzed the behavior of the remaining heads to understand their functions.

Source-to-source Rare-word Heads. We find that one source-to-source (S2S) attention head in the first encoder layer tends to attend to the most infrequent word of the input sentence, which agrees with the bidirectional findings of Voita et al. (2019b). The maximum weight of this head is assigned to one of the least two frequent words in the input sentence roughly 30% of the time, as shown in Table 4 for most language pairs. We also find that this behavior of attending to rare words does not occur in target-to-target (T2T) and target-to-source (T2S) attention, as all T2T and T2S heads achieved less than 10% accuracy at attending to the two least frequent words. The S2S rare word head is marked in red in Figure 1.[9]

Self-attention Positional Heads. We find that some self-attention heads in both the encoder and the decoder tend to attend to neighbors (+1 or -1 position). We call a self-attention head "positional" if its maximum attention weight is assigned to neighbors at least 80% of the time. For example, if the maximum weight of a head is assigned to the -1 relative position more than 80% of the time, then this head is identified as a positional -1 head, as shown in Figure 1. Table 5 shows positional heads found in the finetuned models (R-finetune). We find that different language pairs generally share the same positional heads, and positional heads only occur in shallow encoder and decoder layers. As shown in Figure 1, positional heads essentially remain unchanged during finetuning. However, during pruning, some positional heads are eliminated and some new positional heads emerge, mostly because positional attention is easy to learn and therefore this function tends to migrate from one head to another during automatic pruning.

	head	directions
S2S (-1)	1:6	DeEn,FrEn,RoEn,EnDe,EnFr,EnRo
	2:6	DeEn,FrEn,RoEn,EnDe,EnFr,EnRo
	0:1	EnDe,EnFr
S2S (+1)	1:7	DeEn,FrEn,RoEn,EnDe,EnFr,EnRo
T2T (-1)	0:5	DeEn,FrEn,RoEn,EnDe,EnFr,EnRo
	0:7	DeEn,FrEn,RoEn,EnDe,EnFr,EnRo

Table 5: Positional heads in the finetuned models (R-finetune). "1:6" denotes the 6th head in the 1st layer.

	German	French	Romanian	English
`obj`	1	-2	-1	-2
`nsubj`	1	1	2	1
`advmod`	1	1	1	1
`amod`	1	-1	-1	1

Table 6: The highest-probability relative distance for different dependency relationships (forward direction).

Self-attention Syntactical Heads. We find that some self-attention heads in both the encoder and the decoder learn syntactical dependencies, i.e., the maximum attention weight is assigned to a syntactically related word of the current word. We call a self-attention head "syntactical" if it learns a dependency relationship with an accuracy at least 10% higher than the baseline accuracy of this relationship. The baseline accuracy of a dependency relationship is the accuracy of a fictional head that always attends to the most likely relative position of this relationship. For example, for the `obj` dependency relationship in English, the correct dependency typically is encountered at the -2 relative position (38% of cases), which is the most likely relative position for this relationship. Hence, a fictional head that always attends to the -2 relative position will achieve 38% accuracy for this relationship, and 38% can serve as the baseline accuracy for the English `obj` relationship. For different languages, the most likely relative position of the `obj` relationship is different, as shown in Table 6. We look at four important dependency relation-

[8] The base Transformer model contains 144 attention heads in total: 48 self-attention heads in the encoder, 48 self-attention heads and 48 cross-attention heads in the decoder. Cross-attention and self-attention in both the encoder and the decoder have 6 layers and each layer contains 8 heads.

[9] Figure 1 shows functional heads identified for at least one translation direction. For example, all syntactical heads identified for different translation directions as shown in Table 7 and Table 9 are marked as S heads in Figure 1. Black heads are heads that were pruned away for all translation directions during automatic pruning.

✓	3:6	**amod-f**	DeEn,FrEn
		amod-b	FrEn,RoEn
		advmod-f	FrEn
		nsubj-f	EnDe
×	3:2	**obj-b**	DeEn,RoEn,EnDe,EnFr
		nsubj-f	DeEn
✓	2:0	**obj-b**	RoEn,EnDe,EnFr
		nsubj-f	DeEn,EnDe
✓	2:2	**obj-f**	RoEn,EnDe,EnFr,EnRo
×	5:5	**obj-b**	DeEn,RoEn,EnRo
		nsubj-f	DeEn
✓	4:2	**obj-f**	DeEn,RoEn,EnDe
×	2:1	**amod-f**	DeEn
		amod-b	RoEn
		obj-b	RoEn
×	4:1	**nsubj-f**	DeEn,EnDe,EnFr
✓	3:4	**obj-f**	RoEn,EnDe
		nsubj-b	DeEn
×	5:1	**obj-f**	RoEn
		advmod-b	RoEn
✓	5:0	**amod-b**	FrEn,RoEn
✓	0:0	**nsubj-f**	FrEn,EnDe
✓	3:7	**nsubj-b**	FrEn
×	2:5	**advmod-f**	FrEn
×	5:3	**obj-f**	RoEn
×	4:0	**nsubj-b**	RoEn
×	3:1	**obj-b**	RoEn

Table 7: Dependency relationships learned by S2S syntactical heads in the finetuned models. × means the head is pruned away with automatic pruning at $\lambda = 35$, while ✓ means the head remains after pruning.

head	DeEn	FrEn	RoEn	EnDe	EnFr	EnRo
2:2	0.06	0.43	0.44	**0.65**	**0.51**	**0.54**
4:2	**0.46**	**0.44**	**0.48**	0.53	0.45	0.48

Table 8: Accuracy of S2S heads for the obj-f relationship in the finetuned models. Among all S2S heads, head “2:2” achieved the highest obj-f accuracy for EnDe, EnFr, and EnRo; head “4:2” achieved the highest obj-f accuracy for DeEn, FrEn, and RoEn.

ships[10]: obj (v→o), nsubj (v→s), advmod (v→a), and amod (n→a). For each of these 4 relationships, we consider both the forward and the backward directions. Ultimately, we thus investigate whether a head learns any of the 8 relationships obj-f, obj-b, nsubj-f, nsubj-b, advmod-f, advmod-b, amod-f, and amod-b. We find a head can learn different dependency relationships, as shown in Tables 7 and 9. The most important dependency relationship learned by the Transformer is obj, as more than half of all syntactical heads mainly learn the obj relationship. Tables 7 and 9 further show that some translation directions share some syntactical heads (e.g., DeEn, RoEn, and EnDe share the

✓	3:6	**obj-f**	DeEn,FrEn,RoEn,EnDe,EnRo
		advmod-f	DeEn,EnDe,EnFr
		amod-f	EnRo
		amod-b	EnRo
		nsubj-f	EnDe
✓	2:6	**obj-f**	DeEn,FrEn,RoEn,EnDe,EnFr,EnRo
		nsubj-f	EnDe
		advmod-f	EnDe
✓	3:5	**obj-f**	DeEn,FrEn,RoEn,EnDe,EnFr,EnRo
		nsubj-f	EnDe
		advmod-f	EnDe
✓	2:3	**nsubj-b**	DeEn,FrEn,RoEn,EnDe,EnFr,EnRo
		amod-f	EnRo
×	3:3	**nsubj-b**	DeEn,FrEn,RoEn,EnDe,EnFr,EnRo
✓	2:2	**obj-f**	DeEn,EnDe,EnRo
		nsubj-f	EnDe
		advmod-f	EnDe
✓	1:5	**amod-f**	EnFr,EnRo
		amod-b	EnRo
		advmod-f	EnFr
		obj-f	EnRo
✓	1:6	**obj-f**	DeEn,RoEn,EnDe,EnFr,EnRo
×	1:2	**advmod-f**	DeEn,EnDe,EnFr
		nsubj-f	EnDe
		obj-f	EnDe
✓	2:1	**amod-f**	EnFr,EnRo
		nsubj-f	EnDe
		advmod-f	EnDe
		obj-f	EnDe
✓	3:4	**obj-f**	DeEn,EnDe,EnRo
		nsubj-b	EnRo
×	5:5	**obj-f**	DeEn,FrEn,RoEn,EnDe
×	2:5	**obj-f**	EnDe,EnRo
		advmod-b	EnRo
×	0:2	**obj-f**	EnDe,EnRo
✓	4:5	**obj-f**	EnDe,EnRo
×	5:0	**obj-f**	DeEn,FrEn
×	4:4	**obj-f**	EnRo
		amod-f	EnRo
✓	1:0	**obj-f**	EnRo
		amod-b	EnRo
×	2:4	**nsubj-b**	DeEn
		amod-b	EnRo
✓	1:3	**nsubj-f**	EnDe
		advmod-f	EnDe
✓	4:0	**obj-f**	EnDe
		nsubj-f	EnDe
✓	3:7	**obj-f**	EnRo
		amod-b	EnRo
✓	4:6	**obj-f**	EnRo
✓	4:7	**obj-f**	EnDe
×	5:1	**obj-f**	EnDe
✓	4:1	**obj-f**	EnRo
×	5:3	**obj-f**	EnRo
×	5:6	**nsubj-b**	EnFr
×	3:1	**obj-f**	EnRo
×	0:1	**obj-f**	EnDe
×	2:7	**obj-f**	EnDe
✓	3:2	**obj-f**	EnRo

Table 9: Dependency relationships learned by T2T syntactical heads in the finetuned models. × means the head is pruned away with automatic pruning at $\lambda = 35$, while ✓ means the head remains after pruning.

[10]We used the parsing results by the Stanford parser (Manning et al., 2014) as the ground truth label in our experiments.

Head	DeEn	FrEn	RoEn	EnDe	EnFr	EnRo
0:7	**0.88**	0.81	**0.86**	**0.85**	0.77	0.72
1:5	0.81	0.76	0.82	0.77	0.73	0.77
2:5	**0.88**	**0.84**	0.85	0.81	0.75	0.68
2:7	0.81	0.73	0.84	0.77	**0.80**	**0.87**

Table 10: Accuracy of C-alignment heads.

S2S "4:2" head for the obj-f relationship in Table 7), but not all translation directions share all syntactical heads. For a more direct overview of how different translation directions share syntactical heads, Table 8 gives the accuracy of different S2S heads for the obj-f relationship, which clearly shows that the "2:2" head mainly learns obj-f for EnDe, EnFr, and EnRo, while the "4:2" head mainly learns obj-f for DeEn, FrEn, and RoEn. Meanwhile, the S2S "2:2" head acquires nearly 0 obj-f accuracy for the DeEn direction, although this head is the most accurate obj-f head for EnDe, EnFr, and EnRo.

Cross-attention C-alignment Heads. We find that some cross-attention heads in the shallow layers learn word alignment for the current target word, i.e., the maximum weight is assigned to the source word aligned to the current target word. If a head achieves more than 80% accuracy for aligning the current target word, we call such a head a "C-alignment" head. When we calculate the alignment accuracy[11], we only consider situations when the current target word is a content word[12], as function words generally do not have clear alignments between different languages. By attending to the contextualized representation of the source word aligned to the current target word, C-alignment heads can help to retrieve the full context of the current target word (both the left-side and right-side context), in contrast to target-to-target self-attention, which can only attend to the left-side context of the current target word. Table 10 gives the accuracy of C-alignment heads in the finetuned models (R-finetune), showing that C-alignment heads generally learn the current word alignment for all translation directions, while the highest-accuracy C-alignment head for different directions may differ.

Head	DeEn	FrEn	RoEn	EnDe	EnFr	EnRo
2:0	0.70	0.67	0.66	0.62	0.72	0.65
3:0	0.72	0.72	0.69	0.65	0.68	0.56
3:7	0.73	0.72	0.72	0.74	**0.77**	0.77
4:0	0.78	0.80	0.75	0.76	0.76	0.71
4:2	0.78	0.81	0.78	0.79	**0.77**	**0.78**
4:5	0.66	0.70	0.66	0.64	0.62	0.60
4:6	0.67	0.71	0.68	0.64	0.70	0.59
4:7	**0.82**	**0.83**	**0.81**	**0.81**	0.75	0.75
5:4	0.69	0.74	0.70	0.65	0.72	0.63
5:5	0.63	0.70	0.67	0.67	0.66	0.59
5:6	0.66	0.71	0.66	0.68	0.61	0.65

Table 11: Accuracy of N-alignment heads.

Cross-attention N-alignment Heads. We further find that some cross-attention heads in the deep layers learn word alignment for the next target word, i.e., the maximum weight is assigned to the source word aligned to the next target word. As the next word is unknown at the current decoding step, N-alignment heads are rather learning to predict the next target word than just aligning the next target word. Therefore, the N-alignment accuracies are generally lower than the C-alignment accuracies. We identify "N-alignment" heads as heads that achieve more than 70% accuracy for aligning the next word. When we calculate the N-alignment accuracy, we again only consider situations when the next target word is a content word, as before for the C-alignment accuracy. Figure 1 shows that C-alignment heads occur in shallow layers and N-alignment heads occur in deep layers, which indicates that the Transformer decoder appears to first use C-alignment heads to obtain the context of the current target word, and then, based on the context of the current target word, predicts which word to generate next. Table 11 gives the accuracy of N-alignment heads in the finetuned models (R-finetune), which shows that different language pairs generally shared N-alignment heads, which is surprising since the purpose of N-alignment heads is learning word reordering and different language pairs should have different reordering rules. Table 11 also shows that the highest-accuracy N-alignment heads are most likely from the 4th layer, i.e., the N-alignment accuracy first increases and then decreases as the layer number increases.

3.3 Head Behavior Analysis

We provide a further analysis of head behavior by comparing head accuracies in different multilingual and bilingual models. Table 12 gives the highest

[11] We use human-annotated word alignments (Zhang and van Genabith, 2021) as the ground truth label for computing word alignment accuracies.

[12] For each language, we judge whether a word is a function word or a content word using a list of stopwords from NLTK, https://www.nltk.org/

			DeEn	FrEn	RoEn	EnDe	EnFr	EnRo
S2S	`obj-f`	R-bi	**0.58**	0.40$^{\times}$	0.46	0.57	**0.55**	0.53
		R-multi	0.42	0.43$^{\times}$	0.45	0.60	0.49	0.46$^{\times}$
		R-finetune	0.46 △	**0.44**$^{\times}$ △	**0.48** △	**0.65** △	0.51 △	**0.54** △
	`obj-b`	R-bi	**0.46**	**0.41**$^{\times}$	**0.60**	**0.69**	0.51	**0.61**
		R-multi	0.41	0.37$^{\times}$	0.53	0.57	0.55	0.46
		R-finetune	0.44 △	0.38$^{\times}$ △	0.57 △	0.59 △	**0.56** △	0.50 △
T2T	`obj-f`	R-bi	**0.84**	0.76	0.69	**0.74**	0.72	0.53
		R-multi	0.79	0.76	0.68	0.63	**0.73**	0.52
		R-finetune	0.81 △	**0.78** △	**0.70** △	0.65 △	0.72 ▽	**0.55** △
	`obj-b`	R-bi	–	–	–	0.37$^{\times}$	–	–
		R-multi	–	–	–	**0.38**$^{\times}$	–	–
		R-finetune	–	–	–	0.37$^{\times}$ ▽	–	–
T2S	C-a	R-bi	**0.89**	**0.88**	**0.89**	**0.90**	**0.82**	**0.87**
		R-multi	0.88	0.83	0.85	0.84	0.79$^{\times}$	0.86
		R-finetune	0.88	0.84 △	0.86 △	0.85 △	0.80 △	**0.87** △
	N-a	R-bi	0.80	**0.83**	0.79	0.81	**0.79**	**0.78**
		R-multi	**0.83**	**0.83**	**0.82**	**0.83**	**0.79**	**0.78**
		R-finetune	0.82 ▽	**0.83**	0.81 ▽	0.81 ▽	0.77 ▽	**0.78**

Table 12: Highest accuracy for syntactical (`obj`), C-alignment (C-a), and N-alignment (N-a) heads in different models. $^{\times}$ means the accuracy is not high enough to be identified as function head. △ (▽) means finetuning increased (decreased) the head accuracy (i.e., R-finetune $>$ ($<$) R-multi).

accuracy of different types of function heads in the multilingual and bilingual models. As shown in Table 12, finetuning a multilingual model for a specific translation direction tended to increase accuracies of function heads (e.g., the syntactical `obj` heads and C-alignment heads) for the given translation direction, which is unsurprising. However, Table 12 indeed shows two interesting head behaviors in multilingual and bilingual models.

Cooperative Behavior of N-alignment Heads. First, we find the highest N-alignment accuracy tended to decrease instead of increasing during finetuning (the average accuracy of N-alignment heads also decreased). The fact that finetuning decreased N-alignment accuracies is surprising, considering N-alignment heads are crucial for predicting the next target word. We hypothesize that this is because N-alignment heads work in a cooperative way. Since there are multiple different gold translations (reorderings) in the target language for one source sentence, the Transformer uses different heads to learn different options for predicting (aligning) the next target word. Thus, the accuracy of a single N-alignment head is less important. Table 13 gives the accuracy of at least one head from the 4th layer (the most important N-alignment layer) correctly aligning the next target word. The results show that when we consider the whole layer, the next word alignment accuracy is fairly high and the layer accuracy generally increased during finetuning. The fact that the accuracy of any individual N-alignment head nevertheless tended to decrease during finetuning while the overall N-alignment layer accuracy tended to increase during finetuning indicates that N-alignment heads work in a cooperative way to collect different options for word reordering.

	DeEn	FrEn	RoEn	EnDe	EnFr	EnRo
R-b	**96.8**	**97.2**	94.7	**96.7**	**95.9**	94.3
R-m	96.7	96.9	95.7	96.3	95.3	94.3
R-f	**96.8**	96.8	**96.0**	96.4	95.4	**95.2**

Table 13: Layer accuracy (4th layer) for N-alignment. R-b: R-bi; R-m: R-multi; R-f: R-finetune.

Multilingual Interference for Head Accuracy. Second, we find that, although finetuning generally improved the accuracies of function heads, the finetuned models (R-finetune) still tended to have lower accuracy than the bilingual baseline models (R-bi), especially the C-alignment accuracy and N-alignment layer accuracy for our high-resource language pairs De↔En and Fr↔En, as shown in Tables 12 and 13. This is surprising considering that R-finetune achieved higher translation qualities compared to R-bi, and suggests that language interference tends to cause an accuracy decrease for function heads and can be an important disadvantage of multilingual models compared to bilingual models. Regarding the reason why R-finetune generally had lower head accuracies but higher translation quality compared to R-bi for De↔En and Fr↔En tasks, it could be that the multilingual pretraining helps the model to learn better representations (word embeddings) for less frequent words via the shared vocabulary.

4 Conclusion

This paper analyzes attention head functions and behaviors in multilingual Transformer translation models. We find that different language pairs, in spite of having different syntax and word orders, tend to share the same heads for the same functions, such as syntax heads and reordering heads. However, the different characteristics of different language pairs clearly cause interference in function heads and affect head accuracies, which can be an important disadvantage of multilingual models compared to bilingual models. Additionally, we reveal an interesting behavior of the Transformer cross-attention: the deep-layer cross-attention heads work in a cooperative way to learn different options for word reordering, which can be caused by the nature of translation tasks having multiple different gold translations (reorderings) in the target language for one source sentence.

Limitations

Our study focuses on models trained for particular source to target language pairs. It covers six translation directions with limited typological diversity in the considered languages, due to the need for ground truth word alignments. In future work, multilingual models covering many more languages with more linguistic diversity can be investigated following our methodology.

Acknowledgements

The authors acknowledge the financial support by the German Federal Ministry for Education and Research (BMBF) within the project "KI-Servicezentrum Berlin Brandenburg" 01IS22092. Hongfei Xu is supported by the National Natural Science Foundation of China (Grant No. 62306284) and the Natural Science Foundation of Henan Province (Grant No. 232300421386).

References

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

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

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Mike Schuster, Noam Shazeer, Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. 2018. The best of both worlds: Combining recent advances in neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–86, Melbourne, Australia. Association for Computational Linguistics.

Yun Chen, Yang Liu, Guanhua Chen, Xin Jiang, and Qun Liu. 2020. Accurate word alignment induction from neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 566–576, Online. Association for Computational Linguistics.

Ting-Rui Chiang, Yi-Pei Chen, Yi-Ting Yeh, and Graham Neubig. 2022. Breaking down multilingual machine translation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2766–2780, Dublin, Ireland. Association for Computational Linguistics.

Javier Ferrando and Marta R. Costa-jussà. 2021. Attention weights in transformer NMT fail aligning words between sequences but largely explain model predictions. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 434–443, Punta Cana, Dominican Republic. 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.

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.

Zae Myung Kim, Laurent Besacier, Vassilina Nikoulina, and Didier Schwab. 2021. Do multilingual neural machine translation models contain language pair specific attention heads? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2832–2841, Online. Association for Computational Linguistics.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075, Online. Association for Computational Linguistics.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133.

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.

Zehui Lin, Liwei Wu, Mingxuan Wang, and Lei Li. 2021. Learning language specific sub-network for multilingual machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 293–305, Online. Association for Computational Linguistics.

Christos Louizos, Max Welling, and Diederik P Kingma. 2017. Learning sparse neural networks through l_0 regularization. *arXiv preprint arXiv:1712.01312*.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Telmo Pires, Robin Schmidt, Yi-Hsiu Liao, and Stephan Peitz. 2023. Learning language-specific layers for multilingual machine translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14767–14783, Toronto, Canada. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Uri Shaham, Maha Elbayad, Vedanuj Goswami, Omer Levy, and Shruti Bhosale. 2023. Causes and cures for interference in multilingual translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15849–15863, Toronto, Canada. Association for Computational Linguistics.

Gongbo Tang, Philipp Rönchen, Rico Sennrich, and Joakim Nivre. 2021. Revisiting negation in neural machine translation. *Transactions of the Association for Computational Linguistics*, 9:740–755.

Gongbo Tang, Rico Sennrich, and Joakim Nivre. 2018. An analysis of attention mechanisms: The case of word sense disambiguation in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 26–35, 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.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019a. The bottom-up evolution of representations in the transformer: A study with machine translation and language modeling objectives. 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 4396–4406, Hong Kong, China. 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, Long Zhou, Jiajun Zhang, Feifei Zhai, Jingfang Xu, and Chengqing Zong. 2019. A compact and language-sensitive multilingual translation method. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1213–1223, Florence, Italy. Association for Computational Linguistics.

Zirui Wang, Zachary C. Lipton, and Yulia Tsvetkov. 2020. On negative interference in multilingual models: Findings and a meta-learning treatment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4438–4450, Online. Association for Computational Linguistics.

Wanying Xie, Yang Feng, Shuhao Gu, and Dong Yu. 2021. Importance-based neuron allocation for multilingual neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for*

Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 5725–5737, Online. Association for Computational Linguistics.

Haoran Xu, Jean Maillard, and Vedanuj Goswami. 2023. Language-aware multilingual machine translation with self-supervised learning. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 526–539, Dubrovnik, Croatia. Association for Computational Linguistics.

Hongfei Xu, Qiuhui Liu, Josef van Genabith, and Deyi Xiong. 2021a. Modeling task-aware MIMO cardinality for efficient multilingual neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 361–367, Online. Association for Computational Linguistics.

Hongfei Xu, Josef van Genabith, Qiuhui Liu, and Deyi Xiong. 2021b. Probing word translations in the transformer and trading decoder for encoder layers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–85, Online. Association for Computational Linguistics.

Fei Yuan, Yinquan Lu, Wenhao Zhu, Lingpeng Kong, Lei Li, Yu Qiao, and Jingjing Xu. 2023. Lego-MT: Learning detachable models for massively multilingual machine translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11518–11533, Toronto, Canada. Association for Computational Linguistics.

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

Jingyi Zhang and Josef van Genabith. 2021. A bidirectional transformer based alignment model for unsupervised word alignment. 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 283–292, Online. Association for Computational Linguistics.

Changfeng Zhu, Heng Yu, Shanbo Cheng, and Weihua Luo. 2020. Language-aware interlingua for multilingual neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1650–1655, Online. Association for Computational Linguistics.

Yaoming Zhu, Jiangtao Feng, Chengqi Zhao, Mingxuan Wang, and Lei Li. 2021. Counter-interference adapter for multilingual machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2812–2823, Punta Cana, Dominican Republic. Association for Computational Linguistics.