Instruction Position Matters in Sequence Generation with Large Language Models

Yijin Liu, Xianfeng Zeng, Chenze Shao, Fandong Meng* **and Jie Zhou**
Pattern Recognition Center, WeChat AI, Tencent Inc, China
{yijinliu, xianfzeng, chenzeshao, fandongmeng, withtomzhou}@tencent.com

Abstract

Large language models (LLMs) are capable of performing conditional sequence generation tasks, such as translation or summarization, through instruction fine-tuning. The fine-tuning data is generally a sequential concatenation of a specific task instruction, an input sentence, and the corresponding response. Considering the locality of self-attention modeling in LLMs, these models face the risk of *instruction forgetting* when generating responses for long input sentences. To mitigate this issue, we propose to enhance the instruction-following capability of LLMs by relocating the position of task instructions after the input sentences. Theoretical analysis suggests that our straightforward method can alter the model's learning focus, thereby emphasizing the training of instruction-following capabilities. Concurrently, experimental results demonstrate that our approach consistently outperforms traditional settings across various model scales (1B / 7B / 13B) and different sequence generation tasks (translation and summarization), without any additional data or annotation costs. Notably, our method significantly improves the zero-shot performance on conditional sequence generation, *e.g.,* up to 9.7 BLEU points on WMT zero-shot translation tasks. Further analysis reveals that our method can substantially enhance the model's instruction-following ability by 1x compared to the traditional approach.

1 Introduction

In recent years, there has been a rapid emergence of large language models (LLMs) like ChatGPT and GPT-4[1], which have shown promising performance in various traditional natural language processing tasks (Wang et al., 2023; Jiao et al., 2023b; Kasneci et al., 2023; Hill-Yardin et al., 2023; Šlapeta, 2023; Aydın and Karaarslan, 2023). Meanwhile, there is also a growing interest in open source medium-sized language models, such as the LLaMA model with 13 billion parameters (Touvron et al., 2023) and the BLOOMZ language model with 7.1 billion parameters (Muennighoff et al., 2022), to meet research and hardware deployment requirements.

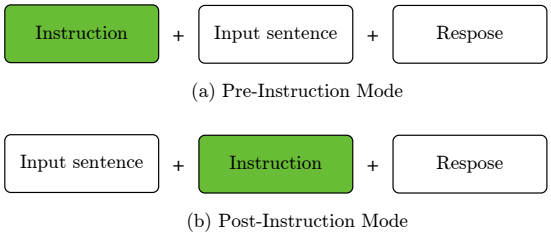


Figure 1: Example data in Pre-Instruction and Post-Instruction format. Different blocks represent textual data from different fields, while the '+' symbol signifies the concatenation operation for the textual data. For sequence generation tasks, the length of the input sentence is generally much larger than the length of the task instruction.

To align the outputs of language models with human intentions and unlock their full potential, InstructGPT (Ouyang et al., 2022) constructs a small amount of instruction-following data for fine-tuning LLMs and conducts reinforcement learning to align the model with human preferences. This approach to instruction following has gained widespread attention from both the academic and industrial communities (Brooks et al., 2023; Chung et al., 2022; Wei et al., 2022; Ahn et al., 2022; Wei et al., 2021; Aher et al., 2023).

Generally, the instruction-following data consists of three parts. Taking the machine translation task as an example, these parts include a specific task instruction (*e.g.,* "Please translate the following paragraph from English to French"), an input sentence (the English sentence to be translated), and the final response (the corresponding French translation). Since most large language models are

* Corresponding author.
[1] https://chat.openai.com/chat

Findings of the Association for Computational Linguistics ACL 2024, pages 11652–11663
August 11-16, 2024

based on the decoder-only structure of the Transformer (Vaswani et al., 2017; Radford et al., 2019; Brown et al., 2020), with a training objective of predicting the next token. Generally, these three parts of the instruction-following data are sequentially concatenated into a long nature sentence as the training data for language models. Given that the self-attention mechanism in Transformer decoders tends to focus more on nearby words *i.e.,* the locality of self-attention modeling (Beltagy et al., 2020; Kitaev et al., 2019; Voita et al., 2019), there is a considerable risk of instruction forgetting when predicting responses for long input sentences. For example, when performing long text summarization tasks, the input sentence may contain thousands of tokens. Consequently, the model may be at risk of forgetting the initial task instruction when predicting responses, leading to the generation of responses that do not fully comply with the user's intent. In this paper, we refer to this risk as the *instruction forgetting* issue.

To alleviate the above issue for LLMs during instruction fine-tuning, we first observe that the relative position of the input sentence and the task instruction is crucial. Therefore, we propose a simple and straightforward solution, namely, placing the task instruction at the end of the input sentence (referred to as 'Post-Ins'). In this way, when the model predicts the final response, it naturally attends to the nearest preceding sequence, which is just the task instruction indicating what content should be generated next. For comparison, we refer to the data format in existing studies where the task instruction is concatenated to the front of the input sentence as Pre-Instruction (abbreviated as 'Pre-Ins').

To verify whether Post-Ins improves the instruction-following ability of language models and alleviates the instruction forgetting issue on long sentences compared to Pre-Ins, we first analyze the conditional probability characteristics of the models under both data formats with the trinomial Bayes formula. Through appropriate assumptions and formula derivations, we draw the following conclusions: (1) Pre-Ins tends to model a reverse conditional probability (*e.g.,* reverse translation probability), emphasizing the coverage of the input sentence while insufficiently modeling the task instruction. (2) Post-Ins is more inclined to model a conditional probability about the task instruction (*e.g.,* predicting the task instruction given inputs and outputs), emphasizing the modeling of task instruction-following ability. In addition to the theoretical analysis, we conduct extensive experiments based on two widely used large language models, LLaMA and BLOOMZ, with various parameter sizes ranging from 1.7 billion to 13 billion. We select two common sequence generation tasks as specific downstream tasks, namely, machine translation and long text summarization. The experimental results show that Post-Ins consistently outperforms Pre-Ins across various settings without using any additional supervised data. Furthermore, due to the superior modeling ability of task instruction, Post-Ins exhibits stronger task instruction generalization capabilities, resulting in significant performance gains in zero-shot translation tasks (*e.g.,* up to a 9.7 BLEU score improvement). Further analysis reveals that our method can substantially enhance the model's instruction-following ability by 1x compared to the traditional approach, and about 4x improvements on specific instructions.

Our contribution can be summarized as follows: [2]

- We show that the position of task instruction is a key factor to conduct instruction fine-tuning with LLMs, and propose to relocate the task instruction after the input sequence (*i.e.,* Post-Ins), which could significantly enhance the instruction-following ability of LLMs by up to 4x improvements.

- Both our theoretical and experimental analyses demonstrate that Post-Ins pays more attentions on the model's instruction-following capabilities, yielding consistent performance improvements across two common sequence generation tasks. (*e.g.,* up to 9.7 BLEU and 3.5 ROUGE improvements on machine translation and text summarization respectively.)

2 Related Work

InstructGPT (Ouyang et al., 2022) is the first to unveil the immense potential of instruction learning. Remarkably, an InstructGPT model with 1.3 billion parameters can outperform the 175B GPT-3, despite having 100x fewer parameters. Then Stanford releases the Alpaca instruction-following dataset (Taori et al., 2023), which is constructed by the self-instruction data generation pipline (Wang

[2]Codes and data are at https://github.com/Adaxry/Post-Instruction

Instruction:
Translate the **following** sentences from [SRC] to [TGT].

Input:
At the start of this week the WADA had already raised the issue of tampering and demanded that Russia reply to the accusations within three weeks. Russia is now faced with the RUSADA again being excluded and in the worst case being banned from the 2020 Olympic Games in Tokyo.

Response:
Zu Beginn dieser Woche hatte bereits die Wada Manipulationen in den Raum gestellt und Russland aufgefordert, binnen drei Wochen auf die Vorwürfe zu antworten. Russland droht nun die erneute Sperre der Rusada und im schlimmsten Fall auch der Ausschluss von den Olympischen Spielen 2020 in Tokio.

(a) Pre-Ins

Input:
At the start of this week the WADA had already raised the issue of tampering and demanded that Russia reply to the accusations within three weeks. Russia is now faced with the RUSADA again being excluded and in the worst case being banned from the 2020 Olympic Games in Tokyo.

Instruction:
Translate the **above** sentences from [SRC] to [TGT].

Response:
Zu Beginn dieser Woche hatte bereits die Wada Manipulationen in den Raum gestellt und Russland aufgefordert, binnen drei Wochen auf die Vorwürfe zu antworten. Russland droht nun die erneute Sperre der Rusada und im schlimmsten Fall auch der Ausschluss von den Olympischen Spielen 2020 in Tokio.

(b) Post-Ins

Figure 2: An example of Pre-Ins formatted data for the machine translation tasks. '[SRC]' and '[TGT]' refer to the source and target language, which are respectively English and German in this example.

et al., 2022). In the field of machine translation, Parrot (Jiao et al., 2023a) builds contrastive and error-guided instructions to align the translation results of LLMs with humman preferences. Subsequently, Zeng et al. (2023) further extends the error-guided instructions with token-level Direct Preference Optimization (Rafailov et al., 2023). To better transfer the capabilities of sequence generation of LLMs, BayLing (Zhang et al., 2023b) proposes to conduct interactive translation task through instructing fine-tuning. Although the aforementioned methods have made considerable progress, we argue that the Pre-Ins data format ultilized in existing studies face the potential risk of instruction forgetting, what we aims to address in this paper.

3 Approach

3.1 Definition

The standard instruction-following data format consists of three components: a specific task instruction z, an input sentence x, and the corresponding response y. Taking the machine translation task as an example, z is a specific task instruction that directs the model to translate from the source language into the target language, while x and y are respectively the source input sentence and target translation. The z, x and y are sequentially concatenated into a long sequence, which is then fed into the LLMs for training in a teacher-forcing mode[3]. We provide a specific example in the Pre-ins format, as shown in Figure 2a.

Considering the nature of sequence generation tasks, the input part (z) often tends to be lengthy, such as translating an entire article or generating a summary of a paragraph. After applying the fine-grained tokenization, it can result in a long sequence of tokens for training. Generally, the mainstream LLMs are based the decoder-only architecture of Transformer, where the self-attention tends to pay more attention on nearby tokens. Therefore, in the case of long sequences as mentioned above, there is a significant risk that the model may forget the frontmost task instruction in the Pre-Ins data format, yiedling responses that do not follow the task instruction.

3.2 Preliminary Observations on Pre-Ins

To verify whether the Pre-Ins data format suffers from the above issue of instruction forgetting on long input sentences, we conduct preliminary experiments on the machine translation task (detailed experimental setups are at Section 4.1). We divide the training data into multiple groups based on the length range of the source text, ensuring that the total number of tokens in each group of training samples is approximately the same. Similarly, we also select corresponding test sets for different length ranges. Results on BLOOMZ are ploted in Table 1. We measure the degree of completion on a specific translation task instruction. Specifically, we ask the model to complete the task of "translating a sentence into a translation in a zero-shot direction," and then use the language identification tool lingua-py[4]to determine whether the model has correctly translated the current sentence into the target language. The zero-shot directions we chose are translation directions that the model has not seen during the supervised fine-tuning phase, specifi-

[3]Following existing studies (Taori et al., 2023; Jiao et al., 2023a), the cross-entropy loss is calculated merely on y, while x and z only participate in the forward encoding process.

[4]https://github.com/pemistahl/lingua-py

Instruction	len 1-30	len 31-60	len 61-90	len 91-120	len 121-150	len 150+
Pre-Instruction	11%	9%	8%	2%	2%	1%
Post-Instruction	33%	31%	27%	24%	20%	19%

Table 1: Accuracy of translation into the correct target language direction over different length invertals in our preliminary experiments.

cally Chinese-to-Japanese, Chinese-to-Ukrainian, English-to-Japanese, and English-to-Ukrainian. In addition, we concatenate the source and target sides of the above translation direction data to synthesize long sentence pairs. Since the above data has contextual relationships, sentence-level concatenation is coherent and meaningful. Under different sentence lengths, we observe the completion of the Pre-Instruction and Post-Instruction translation tasks, *i.e.,* the accuracy of correctly translating into the target language is shown in Table 1.

Based on the above experimental results, it can be seen that at different lengths, the task instruction completion rate of Post-Instruction is significantly higher than that of Pre-Instruction, and the performance decreases relatively gently with the increments in sentence length, indicating that Post-Instruction can alleviate the existing instruction-forgetting problem of Pre-Instruction. Furthermore, the instruction-forgetting problem of Pre-Instruction becomes more severe as the sentence lengthens. For instance, when the average sentence length exceeds 90, the accuracy drops from 8% to 2%. These observations indicate that the existing Pre-Ins data format has limited ability to follow the instructions, especially when the input sentence z is long. Pre-Ins exhibits a risk of instruction forgetting, resulting in outputs that are not faithful to the user's intent.

3.3 Post-Instruction

To address the above issue of instruction forgetting, we propose a simple and straightforward solution, namely, relocating the task instruction z after the input sentence x. As a result, the model can perceive the specific task instructions more closely when generating responses, regardless of the length of the input sentence. We refer this data format as Post-Instruction (Post-Ins), and provide a Post-Ins formatted example in Figure 2b.

Formally, the Post-Ins format of data encourages the LLMs to model the following conditional probability $p(y|x,z)$. Here, we can decompose the above formula using the trinomial Bayes' theorem as follows:

$$p(y|x,z) = \frac{p(y)\cdot p(x|y)\cdot p(z|y,x)}{p(x)\cdot p(z|x)} \tag{1}$$

where $p(inp|res)$ represents the probability of the input given the response.

Given that the task instruction z is not involved in the training loss, we can simply treat its predicted probability $p(z|x)$ as a constant, We and get the following form:

$$p(y|x,z) \approx \underbrace{p(y)}_{fluency} \cdot \underbrace{p(z|y,x)}_{instruct} \cdot \underbrace{p(x|y)/p(x)}_{irreducible} \tag{2}$$

where $p(y)$ denotes the modeling probability of the target response, which guarantees the fluency of the model in predicting the response. The irreducible item represents the ratio of conditional probability to unconditional probability. On the other hand, $p(z|y,x)$ represents the probability of the model determining which task instruction is currently being executed given the input x and response y. This can ensure that the model has a strong perception of the requirements of the task instruction.

3.4 Post-Instruction versus Pre-Instruction

As a comparison, we have also conducted a similar theoretical analysis for Pre-Ins, and ultimately obtain the following formula:

$$\begin{aligned} p(y|z,x) &= \frac{p(y)\cdot p(z|y)\cdot p(x|y,z)}{p(z)\cdot p(x|z)} \\ &= \underbrace{p(y)}_{fluency} \cdot \underbrace{p(x|y,z)}_{coverage} \cdot \underbrace{1/p(x/z)}_{irreducible} \end{aligned} \tag{3}$$

Similar to Post-Ins, Pre-Ins also includes a component responsible for modeling the fluency of the response, denoted as $p(y)$. The key difference lies in that the Pre-Ins emphasizes modeling the probability of the input given the instruction and response, namely, $p(x|y,z)$, which is similar to modeling the coverage in translation tasks (Tu et al., 2016). Such modeling approach may be suitable for a single task or a small number of tasks, as the model can memorize these few task instructions through supervised fine-tuning.

However, LLMs inherently have strong fundamental capabilities that can naturally be applied to various sequence generation tasks. When modeling multiple sequence generation tasks simultaneously (such as multiple translation directions), Pre-Ins may suffer from instruction forgetting and produce low-quality responses that do not follow instructions due to the lack of task instruction modeling. In contrast, Post-Ins, with its preference for directly modeling task instructions as shown in Equation (2), can easily handle various sequence generation tasks and has good transferability for task instructions. We experimentally verify the stronger instruction transferability of Post-Ins compared to Pre-Ins in zero-shot translation tasks in section 5.1. Furthermore, we analyze the modeling preferences of the two data formats from the perspective of attention distribution and observe that the observations are consistent with our above theoretical analysis in section 5.3.

4 Experiments and Evaluations

4.1 Datasets

Alpaca. The Alpaca dataset, released by Stanford (Taori et al., 2023), is widely used for the instruction-following tasks. The data format follows the aforementioned Pre-Ins format, consisting of three parts: instruction, input, and output. We adjust the positions of the instruction and input, yielding a Post-Ins formated Alpaca. We apply this Post-Ins formatted Alpaca dataset on Post-Ins experiments, while the other experiments are still conducted on the original Alpaca dataset.

WMT Datasets. We use the WMT development sets from 2017 to 2020 as high-quality translation training data, following existing settings (Jiao et al., 2023a; Zeng et al., 2023). For translation directions with multiple references, we duplicate the source side and then match them with the corresponding translations to form multiple translation sentence pairs. Finally, we obtain a collection of 51k sentence pairs for instruction fine-tuning. To facilitate comparison, we follow the settings of existing methods (Jiao et al., 2023a; Zeng et al., 2023) and fine-tune LLMs on data for three languages and four translation directions: Chinese-to-English, English-to-Chinese, German-to-English, and English-to-German. The test sets for these four directions in WMT-2022 are used to evaluate translation performance, while the remaining directions, such as French-to-German or Russian-to-English, are used to evaluate the zero-shot performance of the models. Furthermore, considering the similarity in data distribution over years in the WMT dataset (Barrault et al., 2020; Zeng et al., 2021), we also conduct evaluation and validation on another test set, namely the FLORES-200 benchmark.

Multidimensional Quality Metrics (MQM). The MQM dataset is based on the outputs of top systems from the WMT 2020 shared task, which provides error analysis of above translations annotated by professional translators. We follow the preprocessing scripts of existing studies and finally obtain a same sized training set with 99k examples (Jiao et al., 2023a; Zeng et al., 2023). In this paper, MQM is only used for translation task.

CNN/DailyMail. The popular CNN/DailyMail Dataset (See et al., 2017) is a collection of English-language news articles, comprising slightly more than 300k unique articles authored by journalists from CNN and the Daily Mail. The average sentence length of the source text of these data is approximately 665 words, or about a thousand tokens, which served as a widely used benchmark for long text summarization (Tang et al., 2023; Zhang et al., 2023a; Lin et al., 2023). We follow the pre-processing and post-processing scripts of existing studies (Qi et al., 2020). We use the CNN/DailyMail dataset only for the text summarization task and conduct the evaluation on the standard test set with 11,490 samples.

4.2 Evaluation

Inference Settings. For all tasks, we set the batch size to 1 during inference to avoid the effect of padding side (*e.g,* BLOOMZ applies left-padding mode, while LLaMA uses right-padding mode when batching the input data). As for the decoding strategies, we apply the beam search for all tasks, and set beam size to 4 for machine translation. While for the text summarization task, we have to decrease the beam size to 2, as encoding the long input sentences will consume a large portion of GPU memory.

Metrics For the machine translation task, we use SacreBLEU[5] to calculate the BLEU scores. Given the limitations of N-gram-based metrics to measure semantic similarity, we also calculate the popular neural-based metric, namely COMET22[6]. We use

[5] https://github.com/mjpost/sacrebleu
[6] https://github.com/Unbabel/COMET

Systems	#Params	Instruction	SacreBLEU				COMET22			
			De ⟺ En		Zh ⟺ En		De ⟺ En		Zh ⟺ En	
WMT22 Winners										
WMT22 Winners	N/A	N/A	33.70	38.40	33.50	54.30	85.46	88.09	81.12	87.84
BLOOMZ-based										
Parrot (Jiao et al., 2023a)	7.1B	Pre-Ins	24.96	20.56	22.72	34.58	78.09	73.62	79.00	83.54
TIM (Zeng et al., 2023)	7.1B	Pre-Ins	24.31	**20.63**	23.42	37.20	77.65	74.16	79.50	84.89
BLOOMZ	1.7B	Pre-Ins	21.01	15.51	20.31	33.35	72.63	61.63	77.44	82.56
	1.7B	Post-Ins	20.99	16.68	20.15	34.02	73.76	63.64	77.38	82.97
	3.0B	Pre-Ins	23.29	17.02	22.20	35.02	75.42	66.96	78.85	83.33
	3.0B	Post-Ins	23.70	18.24	22.21	35.62	76.12	68.64	78.70	83.77
	7.1B	Pre-Ins	24.37	19.77	22.98	36.64	78.45	73.77	79.54	84.72
	7.1B	Post-Ins	**25.59**	20.45	**23.69**	**37.68**	**78.86**	**74.19**	**79.71**	**84.92**
LLaMA-based										
Parrot (Jiao et al., 2023a)	7.0B	Pre-Ins	27.38	26.14	20.23	30.33	82.47	81.67	75.90	80.34
BayLing (2023b) *	7.0B	Pre-Ins	28.16	25.66	20.31	38.19	83.19	82.18	77.48	84.43
TIM (Zeng et al., 2023)	7.0B	Pre-Ins	27.91	25.02	19.33	30.07	82.80	**82.56**	75.46	80.03
BayLing (2023b) *	13.0B	Pre-Ins	27.34	25.62	20.12	37.92	83.02	82.69	77.72	84.62
TIM (Zeng et al., 2023)	13.0B	Pre-Ins	29.03	26.71	20.27	32.14	83.48	83.31	76.64	81.30
LLaMA	7.0B	Pre-Ins	29.98	25.23	17.68	23.83	82.63	81.27	72.90	75.70
	7.0B	Post-Ins	**30.41**	**26.50**	**21.69**	**30.50**	**83.62**	82.32	**76.60**	**80.66**
	13.0B	Pre-Ins	30.92	28.51	21.95	32.55	84.03	83.14	77.02	81.16
	13.0B	Post-Ins	**31.25**	**28.70**	**22.37**	**33.04**	**84.19**	**83.65**	**77.33**	**82.16**

Table 2: SacreBLEU and COMET22 score(%) of different models with varying instruction modes on the WMT-2022 test sets. 'De', 'En' and 'Zh' are the language code of 'German', 'English' and "Chinese", respectively. The **bolded** scores correspond to the best performance under the same or comparable settings for models with more than 7B parameters. Results marked with '*' indicate that they are not directly comparable with other results because of the use of additional supervised data.

the paired bootstrap resampling methods (Koehn, 2004) to compute the statistical significance of translation results. For the text summarization task, we report the F1 scores of ROUGE-1, ROUGE-2, and ROUGE-L following existing studies (Tang et al., 2023; Qi et al., 2020).

5 Main Results and Analysis

In this section, we first list the detailed experimental results of both the machine translation and text summarization tasks in Section 5.1 and Section 5.2. Subsequently, we show the analysis of the distributions of self-attention in Section 5.3, and human evaluation results in Appendix A.

5.1 Results of Machine Translation

Supervised Translation. Table 2 presents experimental results on WMT22. Our proposed method, Post-Ins, consistently outperforms Pre-Ins in most of translation directions over different model sizes of BLOOMZ (from 1.7B to 7.1B). Specifically, LLaMA-7B achieves a remarkable increase of +6.67 BLEU and +4.96 in COMET22 in En⇒Zh translation. Table 3 showcases the performance of our method on the Flores-200 test set. Our Post-Ins outperforms Pre-Ins in 13 out of 16 settings, with maximum improvements reaching +7.20 BLEU and +6.16 COMET22 score in En⇒Zh.

Zero-Shot Translation. Furthermore, we observe significant improvements in zero-shot translation in the post-ins mode. Table 4 reports the results of different instruction modes in the WMT22 zero-shot test set. In terms of BLOOMZ, there is an impressive increase of +8.8 in De⇒Fr translation, with an average improvement of +1.4 BLEU. For LLaMA-7B, an average improvement of +2.1 BLEU is achieved, and LLaMA-13B exhibits an average improvement of +1.6 BLEU. Notably, LLaMA-13B showcases the highest improvement of +9.7 BLEU in De⇒Fr translation. Overall, the consistent improvements of Post-Ins over Pre-Ins indicate that Post-Ins exhibit stronger instruction generalization capabilities, being able to generate responses effectively even for task instructions it has never encountered during fine-tuning.

Systems	#Params	Instruction	SacreBLEU				COMET22			
			De ⟺ En		Zh ⟺ En		De ⟺ En		Zh ⟺ En	
TIM (Zeng et al., 2023)	7.0B	Pre-Ins	39.15	29.31	**22.30**	28.43	88.19	85.05	83.32	80.55
LLaMA	7.0B	Pre-Ins	38.86	29.51	18.10	21.69	88.05	84.57	80.69	75.07
	7.0B	Post-Ins	**41.12**	**31.27**	**21.80**	**28.89**	**88.63**	**85.53**	**83.57**	**81.23**
	13.0B	Pre-Ins	41.78	33.62	22.21	30.74	88.91	86.36	84.26	82.50
	13.0B	Post-Ins	**42.23**	**34.12**	**22.62**	**31.37**	**89.02**	**86.75**	**84.38**	**82.83**

Table 3: SacreBLEU and COMET22 score(%) of different models with varying instruction modes on the FLORES-200 test sets. The **bolded** scores correspond to the best performance under the same or comparable settings.

Systems	#Para.	Ins.	SacreBLEU												
			Cs ⇔ En		De ⇔ Fr		Ja ⇔ En		Uk ⇔ En		Ru ⇔ En		Liv ⇔ En		Average
BLOOMZ	7.0B	Pre-Ins	6.0	4.3	15.6	23.0	11.0	2.5	**11.0**	1.9	21.7	5.8	3.0	3.5	9.1
	7.0B	Post-Ins	**8.6**	**4.5**	**24.4**	**23.9**	11.0	**2.6**	10.2	**2.0**	**22.6**	**6.1**	**5.9**	**4.2**	**10.5**
LLaMA	7.0B	Pre-Ins	36.8	13.7	3.0	3.4	12.2	4.8	33.9	4.6	34.8	16.8	5.9	2.6	14.3
	7.0B	Post-Ins	**36.8**	**17.4**	**3.2**	**8.8**	**12.8**	**7.3**	**34.6**	**11.7**	**35.2**	**18.9**	**6.0**	**3.3**	**16.4**
	13.0B	Pre-Ins	39.5	19.7	4.9	27.5	**13.9**	3.4	36.8	17.2	37.6	21.1	5.5	2.9	19.1
	13.0B	Post-Ins	**39.7**	**20.2**	**14.6**	**30.0**	13.6	**6.0**	**37.4**	**17.6**	**38.1**	**22.3**	**5.6**	**3.0**	**20.7**

Table 4: SacreBLEU score(%) of different models with varying instruction modes on the WMT-2022 zero-shot test sets. The **bolded** scores correspond to the best performance under the same or comparable settings. 'Para.' is short for 'Parameters' and 'Ins.' stands for the data format for the instruction-following data. 'CS', 'Uk', 'Ja', 'Ru' and 'Liv' are the language code for 'Czech', 'Ukrainian', 'Japanese', 'Russian' and 'Livonian', respectively.

#Params	Instruction	RG-1	RG-2	RG-L
	BLOOMZ-based			
3.0B	Pre-Ins	35.41	16.33	25.81
3.0B	Post-Ins	**38.90**	**17.84**	**27.67**
7.0B	Pre-Ins	37.54	17.04	26.90
7.0B	Post-Ins	**38.61**	**17.64**	**27.49**
	LLaMA-based			
7.0B	Pre-Ins	37.55	17.17	26.30
7.0B	Post-Ins	**38.11**	**17.66**	**26.88**

Table 5: F1 scores of ROUGE-1 / ROUGE-2 / ROUGE-L on the test set of CNN/DailyMail. 'RG' is an abbreviation for 'ROUGE'. The **bolded** scores correspond to the best performance under the same settings.

5.2 Results of Text Summarization

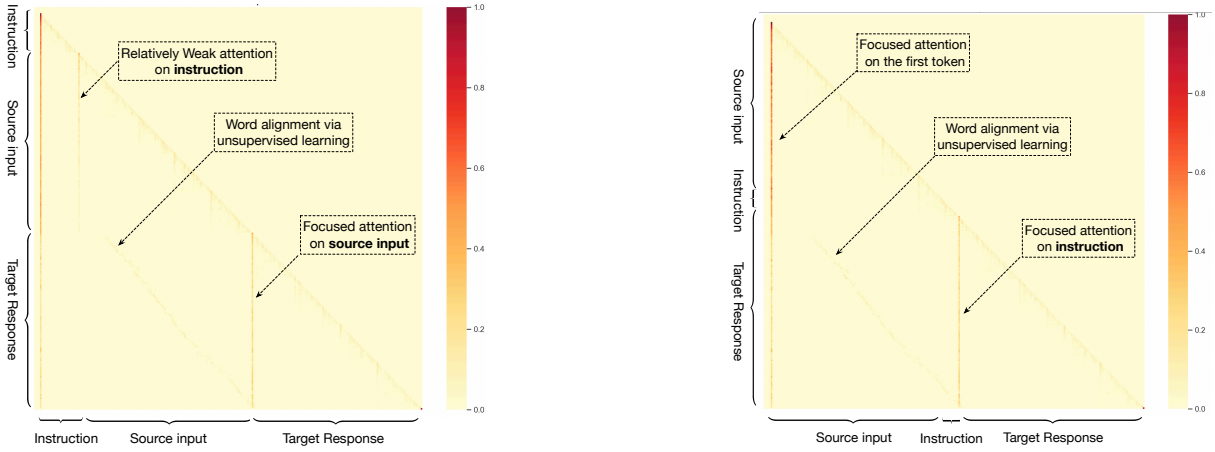
To further validate whether Post-Ins can effectively alleviate the issue of instruction forgetting, we perform experiments on tasks such as the long text summarization task, where the average length of input tokens is over 1,000. Table 5 presents the experimental results for the text summarization task on CNN / DailyMail, where we report the F1 scores of ROUGE-1, ROUGE-2, and ROUGE-L. It is evident that all models achieved significant performance improvements of up to +3.49 in BLOOMZ-3B when utilizing the Post-Ins approach. The superior performance on the text summarization task demonstrates the effectiveness of Post-Ins on handling long inputs.

5.3 Distributions of Self-attention

Given that the distribution of self-attention can explain the behavior of the Transformer model to some extent (Hao et al., 2021; Mahmood et al., 2021; Dai et al., 2021; Braşoveanu and Andonie, 2020), we plot the heatmap of self-attention for models trained with Pre-Ins and Post-Ins in Figure 3. We take BLOOMZ-7.1B as the base model and conduct forward propagation on the training samples of machine translation to obtain the attention scores. To mitigate the impact of fluctuations of multi-head attention and various layers, we average the scores of all heads over different layers to obtain the final score. We reach the following observations:

- A greater concentration of attention scores is observed at the beginning of sentences and along the diagonal positions of the attention matrix, which is consistent with existing conclusions (Liu et al., 2023).

- As shown in the lower right part of Figure 3b, Post-Ins pays more attentions on the task instruction when generating the response, while Pre-Ins mainly foucses on the source input and pays weak attentions on instruction as shown in the upper left part of Figure 3a.



(a) Attention heatmap of Pre-Ins.

(b) Attention heatmap of Post-Ins.

Figure 3: The visualization analysis of self-attention for the instruction fine-tuned BLOOMZ-7.1B model, where thicker lines indicate higher attention for the corresponding positions, while thinner lines indicate lower attention.

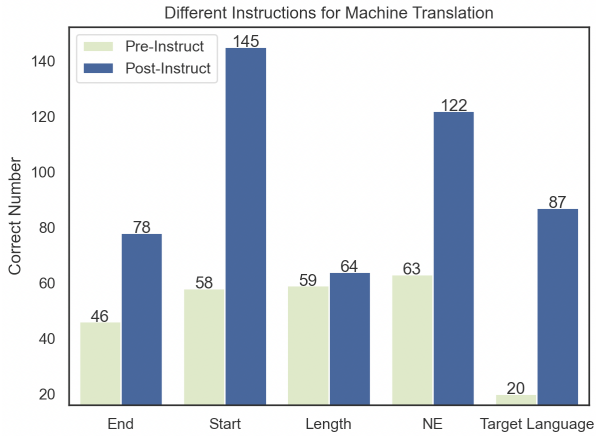


Figure 4: Number of different translation instructions that are correctly executed by models.

- After instruction fine-tuning on the machine translation data, models learn latent word alignment information on both data formats. That is, models tend to allocate more attention to aligned parts of the source when generating responses word by word, which is similar to the conclusions of the traditional encoder-decoder structure in the field of machine translation (Bahdanau et al., 2014; Lample et al., 2017).

In summary, through the visualization of the self-attention heatmap, we observe that Post-Ins naturally tends to emphasize the task instruction, which is relatively overlooked in Pre-Ins. This finding is consistent with the theoretical analysis and conclusions presented earlier in Section 3.4.

5.4 Instruction-Following Evaluation

Inspired by IFEval (Zhou et al., 2023), We construct a dataset to evaluate the model's ability to follow instructions. Specifically, we constructed fine-grained translation instruction compliance data based on the WMT-2022 Chinese-to-English and English-to-Chinese test sets. This involves five task instructions: (1) Please translate the following sentence into Chinese (or English), requiring the prefix of the translation to be [Prefix_Placeholder]; (2) Please translate the following sentence into Chinese (or English), requiring the suffix of the translation to be [Suffix_Placeholder]; (3) Please translate the following sentence into Chinese (or English), requiring the length of the translation to be greater than or equal to [Len_Min_Placeholder] and less than [Len_Max_Placeholder]; (4) Please translate the following sentence into Chinese (or English), requiring the translation to contain [NE_Placeholder]; (5) Please translate the ollowing sentence into [Language_Placeholder]. The xxx_Placeholder in the above instructions will be replaced with the corresponding content according to the golden references and the corresponding instruction requirements, with each type of task instruction containing approximately 300 test examples on average. For the last task instruction, [Language_Placeholder], we used the target languages of two zero-shot translation directions, namely Japanese and Ukrainian.

In terms of evaluation, for the first four task instructions, we can evaluate through simple rule matching. As for the last task requiring a switch of

target language, we judge whether the task instruction is correctly followed and executed by using the language identification tool lingua-py. Each type of instruction has 300 instances. We conduct experiments based on LLaMA-7B, and list results in the Figure 4. We observe that Post-Ins consistently outperforms Pre-Ins in terms of instruction-following ability, especially for switching target languages, with about 4x improvement. Overall, Post-Ins improves instruction-following ability by about 1x over the traditional Pre-Ins.

6 Conclusion

This paper highlights the importance of task instruction positioning in the instruction fine-tuning process of LLMs for conditional sequence generation tasks. We propose a simple yet effective method, Post-Ins, which relocates the task instruction after the input sequence to enhance the instruction-following ability of LLMs. Our theoretical analysis and experimental results demonstrate that Post-Ins effectively shifts the model's learning focus, leading to improved performance across various model scales and different tasks, such as machine translation and long text summarization. Notably, our method significantly boosts zero-shot performance without additional data or annotation costs.

Limitation

The advantages of Post-Ins are likely to be more pronounced with longer text data (such as paragraph data), and we plan to explore its performance on extended text in future work. The effectiveness of Post-Ins on larger scale models also warrants further investigation.

References

Gati V Aher, Rosa I Arriaga, and Adam Tauman Kalai. 2023. Using large language models to simulate multiple humans and replicate human subject studies. In *International Conference on Machine Learning*, pages 337–371. PMLR.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. 2022. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*.

Ömer Aydın and Enis Karaarslan. 2023. Is chatgpt leading generative ai? what is beyond expectations? *What is beyond expectations*.

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

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Adrian M. P. Braşoveanu and Răzvan Andonie. 2020. Visualizing transformers for nlp: A brief survey. In *2020 24th International Conference Information Visualisation (IV)*, pages 270–279.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. 2023. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *NeurIPS 2020*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2021. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2021. Self-attention attribution: Interpreting information interactions inside transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12963–12971.

Elisa L Hill-Yardin, Mark R Hutchinson, Robin Laycock, and Sarah J Spencer. 2023. A chat (gpt) about the future of scientific publishing. *Brain Behav Immun*, 110:152–154.

Wenxiang Jiao, Jen tse Huang, Wenxuan Wang, Xing Wang, Shuming Shi, and Zhaopeng Tu. 2023a. Parrot: Translating during chat using large language models. In *ArXiv*.

Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Xing Wang, and Zhaopeng Tu. 2023b. Is chatgpt a good translator? a preliminary study. *arXiv preprint arXiv:2301.08745*.

Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. 2023. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103:102274.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2019. Reformer: The efficient transformer. In *International Conference on Learning Representations*.

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

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*.

Zhenghao Lin, Yeyun Gong, Yelong Shen, Tong Wu, Zhihao Fan, Chen Lin, Nan Duan, and Weizhu Chen. 2023. Text generation with diffusion language models: A pre-training approach with continuous paragraph denoise. In *International Conference on Machine Learning*, pages 21051–21064. PMLR.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.

Kaleel Mahmood, Rigel Mahmood, and Marten Van Dijk. 2021. On the robustness of vision transformers to adversarial examples. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7838–7847.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. ProphetNet: Predicting future n-gram for sequence-to-SequencePre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2401–2410, Online. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Jan Šlapeta. 2023. Are chatgpt and other pretrained language models good parasitologists? *Trends in Parasitology*.

Moming Tang, Chengyu Wang, Jianing Wang, Cen Chen, Ming Gao, and Weining Qian. 2023. Parasum: Contrastive paraphrasing for low-resource extractive text summarization. In *International Conference on Knowledge Science, Engineering and Management*, pages 106–119. Springer.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS 2017*, pages 5998–6008.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. 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.

Jiaan Wang, Yunlong Liang, Fandong Meng, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023. Is chatgpt a good nlg evaluator? a preliminary study. *arXiv preprint arXiv:2303.04048*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Jiali Zeng, Fandong Meng, Yongjing Yin, and Jie Zhou. 2023. Tim: Teaching lm to translate with comparison. In *ArXiv*.

Xianfeng Zeng, Yijin Liu, Ernan Li, Qiu Ran, Fandong Meng, Peng Li, Jinan Xu, and Jie Zhou. 2021. WeChat neural machine translation systems for WMT21. In *Proceedings of the Sixth Conference on Machine Translation*, pages 243–254, Online. Association for Computational Linguistics.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023a. Diffusum: Generation enhanced extractive summarization with diffusion. *arXiv preprint arXiv:2305.01735*.

Shaolei Zhang, Qingkai Fang, Zhuocheng Zhang, Zhengrui Ma, Yan Zhou, Langlin Huang, Mengyu Bu, Shangtong Gui, Yunji Chen, Xilin Chen, and Yang Feng. 2023b. Bayling: Bridging cross-lingual alignment and instruction following through interactive translation for large language models. *arXiv preprint arXiv:2306.10968*.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

A Human Analysis

We employ two linguistics professionals to evaluate the translation task from English to Chinese. Specifically, the annotators are requested to judge whether the model's output faithfully follow to the translation instructions and the source input. If translation hallucinations occurred, *i.e.,* the response contains content that is not present in the source sentence, or if the translation task is not effectively executed, the label was marked as '0'; otherwise, it is marked as '1'. The manual annotation results on 1,000 samples show that the hallucination rate for Pre-Ins is 4.8%, while 1.7% for Post-Ins. We also provide several examples in Figure 5. In summary, Post-Ins can enhance the model's instruction-following capability and effectively reduce the proportion of prediction hallucinations.

Source Input	Golden References	Response of Post-Ins	Response of Pre-Ins
Connect your eReader to a power source by doing one of the following:	执行以下操作之一，将您的电子阅读器连接到电源：	执行以下操作之一：将电子阅读器连接到电源上	1. 将eReader连接至电源，打开电源开关，将eReader放置在电源附近。\n2. 将电源线的一端连接eReader，另一端连接电源，eReader会自动检测到电源。将电源线的另一端插入eReader的电源插口。
-Maintains flexibility of movement in all working and pet dogs regardless of age, size and level of exercise.	-保持所有工作犬和宠物犬的运动灵活性，无论其年龄、体型和运动水平如何。	- 无论年龄大小、体型大小、运动水平高低，都能保持灵活的动作。	-能够根据不同年龄、不同大小、不同运动水平的工作和宠物狗进行运动量调整。\n-运动量足够大，保证所有年龄、大小和运动水平工作及宠物狗的灵活性。
Pictures come in a sturdy carded envelope in the box with the frame.	照片装在一个结实的信封中，信封与相框装在一个盒子中。	照片附在画框内，在盒内还有一张卡片。	在盒子上方有张照片，照片中有一幅画框，画框中有一张照片。\n照片的底色为白色，背景为黑色。照片中有一名男子

Figure 5: Case studies on Chinese-to-English translation tasks, where the highlighted red texts indicate the model deviates from the translation instruction, generating content not present in the source text.