IOL Research Machine Translation Systems for WMT24 General Machine Translation Shared Task

Wenbo Zhang, Qiaobo Deng, Zeyu Yan, and **Hongbao Mao**
Transn IOL Research, Wuhan, China

Abstract

This paper illustrates the submission system of the IOL Research team for the WMT24 General Machine Translation shared task. We submitted translations for all translation directions in the general machine translation task. According to the official track categorization, our system qualifies as an open system due to the utilization of open-source resources in developing our machine translation model. With the growing prevalence of large language models (LLMs) as a conventional approach for managing diverse NLP tasks, we have developed our machine translation system by leveraging the capabilities of LLMs. Overall, we first performed continued pretraining using the open-source LLMs with tens of billions of parameters to enhance the model's multilingual capabilities. Subsequently, we employed open-source Large Language Models, equipped with hundreds of billions of parameters, to generate synthetic data. This data was then blended with a modest quantity of additional open-source data for precise supervised fine-tuning. In the final stage, we also used ensemble learning to improve translation quality. Based on the official automated evaluation metrics, our system excelled by securing the top position in 8 out of the total 11 translation directions, spanning both open and constrained system categories.

1 Introduction

In the current year's WMT General Machine Translation shared task, our team, IOL Research, took part in all 11 translation tasks, which involved translating text between various language pairs such as Czech to Ukrainian (cs->uk), Japanese to Chinese (ja->zh), English to Chinese (en->zh), English to Czech (en->cs), English to German (en->de), English to Hindi (en->hi), English to Icelandic (en->is), English to Japanese (en->ja), English to Russian (en->ru), English to Spanish (en->es), and English to Ukrainian (en->uk). One notable difference in this year's task compared to previous years is that participants were required to translate paragraph-level texts, with one paragraph equating to one line. This change has significantly increased the length of the text to be translated. While traditional neural machine translation systems (Vaswani et al., 2017) based on encoder-decoder structures may struggle with processing long texts due to the lack of enough document parallel data. However, the large language models (LLMs) do not necessitate a large amount of lengthy text data for fine-tuning, making them more effective in handling long texts. As a result, we meticulously trained an LLM with 20 billion parameters to successfully address all translation tasks in the competition.

Our main strategy is to explore using LLMs to build machine translation systems. This includes fine-tuning the translation task on foundational LLMs and leveraging advanced open-source instruction-tuned LLMs to generate high-quality translation data for further enhancement. Before supervised fine-tuning, we also performed continued pretraining, which has been proven to be very beneficial for translation tasks (Xu et al., 2023), because many open-source LLMs such as LLaMA (Touvron et al., 2023) are usually pretrained on English monolingual data, lacking the necessary knowledge of other languages required for translation tasks. Moreover, we experimented with ensemble learning, a technique known to be effective for neural machine translation models. We discovered that it provided some degree of assistance for machine translation tasks based on LLMs. In the end, our billion-parameter machine translation system achieved comparable performance to hundred billion parameter LLMs in high-resource languages and even outperformed them in certain low-resource languages.

The subsequent paper is designed as follows. We introduce the data source and processing strategy in Section 2; Section 3 describes the details of our training procedure; Section 4 presents the experi-

Proceedings of the Ninth Conference on Machine Translation, pages 147–154
November 15-16, 2024 ©2024 Association for Computational Linguistics

mental settings and results.

2 System Overview

2.1 Model Architecture

We selected the Qwen1.5 model (Bai et al., 2023) as our foundational model because of its outstanding performance and considerable multilingual capabilities. Specifically, we utilized the Qwen1.5-14B[1] as our starting point, which has 40 layers and 14 billion parameters. To enhance the model's capacity within our hardware constraints, we concatenated the first 32 layers with the last 32 layers, resulting in duplication of the middle 24 layers, following the approach used in SOLAR (Kim et al., 2023). This fusion led to a scaled-up model with 64 layers and 21 billion parameters. Since this approach alters the structure of the pretrained model, continual pretraining becomes a necessary step to recover its performance.

2.2 Continual Pretraining

Continual pretraining is an effective method to enhance the knowledge embedded within LLMs. This method has been extensively utilized to adapt LLMs from English to various other languages, as well as to augment the domain-specific knowledge inherent in these models. In the context of using LLMs for translation tasks, it has been substantiated that the continuous pretraining of LLMs with multilingual monolingual data, encompassing languages involved in all the translation directions, is crucial (Xu et al., 2023). This year's WMT24 general machine translation task includes 11 translation directions, involving 10 distinct languages. Therefore, our continued pretraining is carried out on monolingual data in these 10 languages.

We sampled the required multilingual monolingual data from the mC4 (Raffel et al., 2019) and OSCAR (Jansen et al., 2022) datasets, then proceeded to refine the chosen data. For refinement processes, we employed fastText (Joulin et al., 2017) for language identification, the minLSH algorithm for document deduplication, and KenLM (Heafield, 2011) tool for filtering the documents with high perplexity. Many studies (Lin et al., 2020; Yang et al., 2021) have shown that integrating bilingual data with monolingual data in the pretraining stage can help the model achieve better cross-lingual proficiency. Therefore, we also incorporated a portion of the CC-Aligned parallel data (El-Kishky et al., 2019) into our continuous pretraining stage. This data includes language pairs such as English-Czech, English-Ukrainian, English-Japanese, English-Chinese, English-German, English-Hindi, English-Icelandic, English-Russian, and English-Spanish. Specifically, we randomly swapped the order of the two articles in the bilingual document, and then merged them into a new document as the pretraining document. The distribution of the number of documents in all languages in the pretraining dataset is shown in Table 1.

Language	Rate(%)
en	21.99
ja	15.02
de	12.48
cs	11.60
es	10.35
zh	9.32
uk	7.98
ru	7.2
hi	3.53
is	0.47

Table 1: The distribution of the number of documents in all languages in the pretraining dataset.

2.3 Supervised Fine-tuning

Through supervised Fine-tuning, we can unlock the capabilities of LLMs using only a minimal amount of aligned data. Many fine-tuning LLMs experiences (Zhou et al., 2024; Xia et al., 2024) have demonstrated that the quality and diversity of fine-tuning data are far more important than its quantity. In the context of translation tasks, high-quality parallel data is the ideal fine-tuning data for LLMs. However, obtaining such high-quality parallel data is challenging. Usually, we need to invest significant effort and undergo numerous steps to clean publicly available parallel data, aiming to achieve high-quality data. However, this process does not always guarantee the quality of filtered data due to its inherent complexity. On the other hand, start-of-the-art machine translation systems have shown competitive performance comparable to human translators. Consequently, we opted to employ LLMs to generate parallel data as the supervised fine-tuning data.

We used the c4ai-command-r-plus[2] and

[1] https://huggingface.co/Qwen/Qwen1.5-14B

[2] https://huggingface.co/CohereForAI/c4ai-command-r-

Qwen1.5-110B-Chat[3], these two instruction fine-tuned models, to generate synthetic parallel data for all languages, with the exception of Icelandic. Specifically, when the task requires generating Chinese content, our go-to model is the Qwen1.5-110B-Chat. However, for English content generation, we make a random selection between the Qwen1.5-110B-Chat and c4ai-command-r-plus models. For all other scenarios, we consistently utilize the c4ai-command-r-plus model. The selection of models in different languages is based on our evaluation of these two models in translation tasks. Please refer to Table 3 for specific comparison. Considering the lack of proficiency of both c4ai-command-r-plus and Qwen1.5-110B-Chat in generating Icelandic content, we adopted an alternative strategy. We leveraged our supervised fine-tuning model, which has been fine-tuned on synthetic data of all other languages, to produce the synthetic data for Icelandic. Therefore, our model only utilized Icelandic monolingual data for pre-training, and the Icelandic bilingual synthesis data was generated by unsupervised method.

We have tried two synthetic data generation methods commonly used in traditional neural machine translation systems, forward translation (Kim and Rush, 2016) and back translation (Sennrich et al., 2016). Forward translation refers to using the established translation model to translate real source language sentences into target language sentences, and then combining the translated target language sentences with the real source language sentences to form synthetic parallel sentence pairs. Back translation refers to translating real target language sentences back into the source language using another established reverse translation model, and then combining the real target language sentences with the translated source language sentences to form synthetic parallel sentence pairs. In the process of generating back translation data based on real target language data, we found that the real target language data has many problems such as incoherence, fluency deficits, and even grammatical errors. To address these problems, we utilized automatic post-editing technology. This approach involves taking the translated source language sentences and the real target language sentences as inputs, and subsequently producing superior quality target language sentences. These improved sentences are then used to replace the real target language sentences in the back translation synthetic data. Lastly, we also utilized LLMs to filter all the generated synthetic data, including both forward and back translation data, to ensure higher quality fine-tuning data. All the prompts we use to generate synthetic data are shown in the table 2. For each language pair, after filtering, we retained around 100,000 FT and BT sentence pairs respectively.

In addition to synthetic data, we also incorporated document parallel data from News Commentary v18.1[4], which assists the model in translating long text, and instruction fine-tuning data TowerBlocks-v0.2 (Alves et al., 2024) to help the model follow more diverse instructions. The News Commentary v18.1 data we used includes sections ja-zh, en-zh, en-de, en-hi, en-ja, en-ru, en-es, en-cs, cs-ru, cs-de, cs-es, cs-hi, cs-ja, cs-zh, and ja-ru. We also excluded the data from TowerBlocks-v0.2 that includes FLoRes (Goyal et al., 2021), and the NTREX-128 (Federmann et al., 2022) sections, as we used these two datasets as our test sets to verify the performance of the model.

2.4 Ensemble Learning

The ensemble learning approach has demonstrated significant efficacy in a wide range of machine learning tasks. In machine translation tasks, ensemble learning completes the generation of the entire translation by using multiple different machine translation models to autoregressively vote for the probability distribution of the next word. However, for LLMs, this method implies a huge memory occupancy and computational resource consumption, so we use transductive ensemble learning (Wang et al., 2020) to replace this way of generating with multiple models simultaneously. Transductive ensemble learning first utilizes multiple different translation models to generate translations for the same test set separately, then aggregates all translations as fine-tuning data. The final translation is generated by one translation model after fine-tuning on this data. Ensemble learning conventionally entails training diverse models via different random initializations. However, this approach proves inefficient in our context, as we are mandated to employ the identical pre-trained model for supervised fine-tuning. Therefore, we used dif-

plus

[3] https://huggingface.co/Qwen/Qwen1.5-110B-Chat

[4] https://data.statmt.org/news-commentary/v18.1/

Task	Prompt
Forward and back translation	Translate the following text from SRC_LANG to TGT_LANG. SRC_CONTENT
Automatic post-editing	Given a source SRC_LANG sentence and its TGT_LANG translation, please modify and correct the TGT_LANG translation to get a more accurate and fluent TGT_LANG translation. Source (SRC_LANG): SRC_CONTENT Translation (TGT_LANG): TGT_CONTENT Corrected translation (TGT_LANG):
Synthetic data filtering	Source (SRC_LANG): SRC_CONTENT Translation (TGT_LANG): TGT_CONTENT Please check if the above translation is an accurate and fluent translation of its source text? Please only answer "yes" or "no"

Table 2: All the prompts we use to generate synthetic data.

ferent fine-tuning data to train multiple models for ensemble learning. Different fine-tuning data is obtained by randomly sampling synthesized data from different parts.

3 Experiments

3.1 Experiment Settings

For continual pretraining phase, we trained the scaled-up model with 21 billion parameters on 8 NVIDIA H800 GPUs. For the optimization process, we employed the Adam optimizer (Kingma and Ba, 2014), with $\beta 1 = 0.9, \beta 2 = 0.99$. We adopted a learning rate scheduling strategy that remained constant after warmup phase, setting the number of warmup steps to 200, the maximum learning rate at 0.00001 and weight decay to 0.1. The batch size was set to 3.14 million tokens, the length of each sequence was set to 4096, and a total of 56 billion tokens have been trained.

For supervised fine-tuning phase, we fine-tuned the continual pretrained model on 16 NVIDIA H800 GPUs. We leveraged the Adam optimizer for the optimization process, setting $\beta 1 = 0.9, \beta 2 = 0.99$. We employed a cosine learning rate scheduling strategy, with a warmup ratio of 0.01, a peak learning rate at 0.000007, and a weight decay of 0.1. Configuring the batch size to 480 sentences, we trained the model for a single epoch encompassing approximately 1.5 million sentences.

When conducting transductive ensemble learning, we increased the batch size to 800 sentences, adopted a fixed learning rate, and reduced the learning rate to 0.000001. Similarly, we only fine-tune for one epoch on the ensemble data.

3.2 Results

The FLoRes (Goyal et al., 2021) and NTREX-128 (Federmann et al., 2022) test sets were utilized as our evaluation benchmarks. The performance of the machine translation system was assessed using SacreBLEUpost-2018-call and COMET (Rei et al., 2022)[5] metrics. We uesed vLLM (Kwon et al., 2023) to infer all LLMs. We chose c4ai-command-r-plus and Qwen1.5-110B-Chat as our baselines for comparison, and all results were obtained through zero-shot evaluation.

Test results on the FLoRes test set for all translation directions are shown in Table 3. We used greedy decoding and beam search with beam size = 5 to generate translations for our model, and provided the ensemble effect on this test set. It is clear that, just like traditional neural machine translation models, beam search performs better than greedy decoding in terms of BLUE and COMET scores across all translation directions. Ensemble learning has a steady improvement on BLEU scores, but the overall change in COMET scores is not significant. Compared with the two baseline systems CMD-R-P and Qwen1.5-L, our model achieved equivalent or better performance in the seven directions of cs→uk, en→zh, en→de, en→hi, en→is, en→uk, and en→cs. The performance outcomes presented in Table 4 are based on evaluations conducted using the NTREX-128 test set. These results mirror those observed in the FLoRes test set, indicating a consistent performance trend across both datasets.

[5] https://huggingface.co/Unbabel/wmt22-comet-da

		CMD-R-P	Qwen1.5-L	our model greedy decoding	our model beam search	our model ensemble learning
cs→uk	BLEU	24.1	20.5	23.9	24.4	24.6
	COMET	90.47	87.96	90.18	90.41	90.47
ja→zh	BLEU	31.6	34.1	34.8	35.3	35.0
	COMET	87.91	88.10	87.99	88.11	87.98
en→zh	BLEU	39.9	44.0	46.9	47.5	47.6
	COMET	88.71	89.08	89.22	89.28	89.26
en→de	BLEU	41.1	33.9	40.5	41.1	41.6
	COMET	88.84	87.37	88.60	88.73	88.84
en→hi	BLEU	27.3	19.9	27.6	28.5	28.7
	COMET	80.47	75.01	79.99	80.75	80.67
en→is	BLEU	12.1	9.8	19.8	20.5	20.7
	COMET	71.41	63.82	82.77	83.66	84.02
en→ja	BLEU	49.8	42.2	49.4	50.1	50.4
	COMET	91.70	89.88	91.50	91.59	91.61
en→ru	BLEU	32.4	27.6	31.3	31.9	32.4
	COMET	90.70	87.98	90.09	90.32	90.28
en→es	BLEU	30.4	27.1	29.4	29.4	29.5
	COMET	87.29	86.64	87.01	87.06	86.98
en→uk	BLEU	30.4	24.6	31.2	32.0	32.2
	COMET	90.88	88.19	90.56	90.83	90.92
en→cs	BLEU	32.7	26.6	32.8	34.3	34.4
	COMET	92.09	90.04	91.78	92.15	92.13

Table 3: Test results on the FLoRes test set for all translation directions. CMD-R-P represents c4ai-command-r-plus, and Qwen1.5-L represents Qwen1.5-110B-Chat.

		CMD-R-P	Qwen1.5-L	our model greedy decoding	our model beam search
cs→uk	BLEU	20.9	16.8	20.4	20.8
	COMET	88.26	84.57	87.80	88.00
ja→zh	BLEU	25.6	28.7	28.7	29.0
	COMET	84.42	84.84	84.83	84.85
en→zh	BLEU	31.7	36.6	39.0	39.5
	COMET	85.60	86.41	86.76	86.83
en→de	BLEU	33.9	27.1	33.2	33.9
	COMET	87.05	84.64	86.63	86.78
en→hi	BLEU	22.2	16.8	23.3	24.1
	COMET	78.07	72.23	77.96	78.58
en→is	BLEU	14.8	11.2	23.4	24.1
	COMET	70.14	62.32	82.75	83.67
en→ja	BLEU	41.3	35.0	41.3	42.4
	COMET	89.51	87.40	89.37	89.45
en→ru	BLEU	29.9	23.8	30.4	31.5
	COMET	88.13	84.47	87.47	87.88
en→es	BLEU	42.5	38.2	42.4	42.7
	COMET	87.06	85.82	86.69	86.85
en→uk	BLEU	26.2	20.5	26.3	26.9
	COMET	88.86	85.42	88.51	88.73
en→cs	BLEU	29.0	22.6	29.1	30.4
	COMET	89.90	87.06	89.73	90.19

Table 4: Test results on the NTREX-128 test set for all translation directions. CMD-R-P represents c4ai-command-r-plus, and Qwen1.5-L represents Qwen1.5-110B-Chat.

	cs→uk	ja→zh	en→zh	en→de	en→hi	en→is	en→ja	en→ru	en→es	en→uk	en→cs
FT	90.32	87.79	89.21	88.55	80.48	83.37	91.55	90.11	86.94	90.64	91.93
BT	90.43	88.18	89.24	88.77	81.45	84.12	91.64	90.43	87.20	90.99	92.38
MIX	90.32	88.12	89.26	88.63	80.76	84.08	91.50	90.19	87.06	90.63	91.93

Table 5: COMET scores of models fine-tuned on different data on the Flores test set. FT is fine-tuned on forward translation data. BT is fine-tuned on back translation data. MIX is fine-tuned on both forward and back translation data.

3.3 Forward Translation vs Back Translation

To determine the effectiveness of forward translation versus back translation, we separately finetuned the continual pretrained model using forward translation data, back translation data, and a combination of both. For each approach, we randomly chose 80,000 data samples per language translation direction. For the combined dataset, we selected 40,000 samples from both the forward translation and back translation pools. The results are presented in Table 5, all of which were generated using beam search. We can see that the back translation yields better performance, whereas mixed data does not result in significant improvement. Due to time constraints, we used mixed data in the WMT24 competition, this conclusion will guide us to further improve our model in the future.

4 Conclusion

In this paper, we present IOL Research's contributions to the WMT24 General Translation shared task, covering all translation aspects. Our approach utilizes LLMs to develop an effective translation system. Experimental results demonstrate that our model, which contains 21 billion parameters, achieves competitive results comparable to models with 100 billion parameters. According to the official automatic evaluation metrics (Kocmi et al., 2024), our system achieved 8 first places in 11 translation directions spanning both open and constrained system categories, including Czech to Ukrainian, English to German, English to Spanish, English to Hindi, English to Russian, English to Ukrainian, English to Chinese, and Japanese to Chinese.

References

Duarte M. Alves, José Pombal, Nuno M. Guerreiro, Pedro H. Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, Pierre Colombo, José G. C. de Souza, and André F. T. Martins. 2024. Tower: An open multilingual large language model for translation-related tasks.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2019. Ccaligned: A massive collection of cross-lingual web-document pairs. *arXiv preprint arXiv:1911.06154*.

Christian Federmann, Tom Kocmi, and Ying Xin. 2022. Ntrex-128–news test references for mt evaluation of 128 languages. In *Proceedings of the First Workshop on Scaling Up Multilingual Evaluation*, pages 21–24.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2021. The flores-101 evaluation benchmark for low-resource and multilingual machine translation.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197.

Tim Jansen, Yangling Tong, Victoria Zevallos, and Pedro Ortiz Suarez. 2022. Perplexed by Quality: A Perplexity-based Method for Adult and Harmful Content Detection in Multilingual Heterogeneous Web Data. *arXiv e-prints*, page arXiv:2212.10440.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics.

Dahyun Kim, Chanjun Park, Sanghoon Kim, Wonsung Lee, Wonho Song, Yunsu Kim, Hyeonwoo Kim, Yungi Kim, Hyeonju Lee, Jihoo Kim, et al. 2023. Solar 10.7 b: Scaling large language models with simple yet effective depth up-scaling. *arXiv preprint arXiv:2312.15166*.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. *arXiv preprint arXiv:1606.07947*.

DiederikP. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv: Learning,arXiv: Learning*.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Marzena Karpinska, Philipp Koehn, Benjamin Marie, Christof Monz, Kenton Murray, Masaaki Nagata, Martin Popel, Maja Popović, Mariya Shmatova, Steinþór Steingrímsson, and Vilém Zouhar. 2024. Findings of the WMT24 general machine translation shared task:

the LLM era is here but mt is not solved yet. In *Proceedings of the Ninth Conference on Machine Translation*, USA. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Zehui Lin, Xiao Pan, Mingxuan Wang, Xipeng Qiu, Jiangtao Feng, Hao Zhou, and Lei Li. 2020. Pre-training multilingual neural machine translation by leveraging alignment information. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2649–2663, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*.

Ricardo Rei, José GC De Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André FT Martins. 2022. Comet-22: Unbabel-ist 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timoth'ee Lacroix, Baptiste Rozi'ere, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

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.

Yiren Wang, Lijun Wu, Yingce Xia, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2020. Transductive ensemble learning for neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6291–6298.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*.

Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. 2023. A paradigm shift in machine translation: Boosting translation performance of large language models. *arXiv preprint arXiv:2309.11674*.

Ziqing Yang, Wentao Ma, Yiming Cui, Jiani Ye, Wanxiang Che, and Shijin Wang. 2021. Bilingual alignment pre-training for zero-shot cross-lingual transfer. *arXiv preprint arXiv:2106.01732*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.