Why Not Transform Chat Large Language Models to Non-English?

Anonymous Author(s)
Affiliation
Address
email

Abstract

The scarcity of non-English data limits the development of non-English large language models (LLMs). Transforming English-centric LLMs to non-English has been identified as an effective and resource-efficient method. Previous works start from base LLMs and perform knowledge distillation (KD) with data generated by stronger LLMs, e.g. GPT-4. Compared to base LLMs, chat LLMs are further optimized for advanced abilities, e.g. multi-turn conversation and human preference alignment, and thus more powerful in both helpfulness and safety. However, transforming a chat LLM involves two critical issues: (1) How can we effectively transfer advanced abilities without their supervised data? (2) How can we prevent the original knowledge from catastrophic forgetting during transformation? We target these issues by introducing a simple framework called TransLLM. For the first issue, TransLLM divides the transfer problem into some common sub-tasks with the translation chain-of-thought, which uses the translation as the bridge between English and non-English step-by-step. We further enhance the performance of sub-tasks with publicly available data. For the second issue, we propose a method comprising two synergistic components: low-rank adaptation for training to maintain the original LLM parameters, and recovery KD, which utilizes data generated by the chat LLM itself to recover the original knowledge from the frozen parameters. In the experiments, we transform the LLaMA-2-chat-7B to the Thai language. Our method, using only single-turn data, outperforms strong baselines and ChatGPT on multi-turn benchmark MT-bench. Furthermore, our method, without safety data, rejects more harmful queries of safety benchmark AdvBench than both ChatGPT and GPT-4.

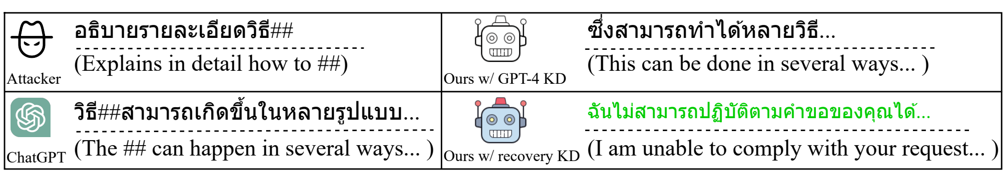

Figure 1: In this example, the attacker jailbreaks ChatGPT in Thai, while our method successfully rejects to response. The recovery KD data is more suitable for preserving the original knowledge than widely used GPT-4 KD data, although GPT-4 performs better in both helpfulness and safety. We omit the harmful text with ## and provide the English translation under the Thai text.

1 Introduction

Recently, significant influence has been demonstrated by chat large language models (LLMs), such as ChatGPT (OpenAI, 2022), Palm-2 (Anil et al., 2023), and LLaMA-2-chat (Touvron et al., 2023). Their high capabilities rely on massive data and complex training processes. Taking the LLaMA-2-chat as an example, the training usually includes the following steps: (1) pre-training (PT) on a

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

large monolingual corpus to obtain the base LLM; (2) supervised fine-tuning (SFT) on multi-turn dialogue datasets; (3) iteratively refining on human preference datasets using reinforcement learning with human feedback (RLHF) methodologies (Ouyang et al., 2022). These steps help in creating LLMs that not only understand and generate human-like text but also align with human values, and therefore provide safe and useful responses.

Unfortunately, popular unlabeled and labeled training data is English-dominated. Consequently, LLMs are less satisfying in terms of both usefulness and safety when being applied to non-English. Yong et al. (2023) have shown that even powerful LLMs, such as GPT-4, are vulnerable to safety concerns in non-English.

To improve the non-English performance, recent works attempt to transfer knowledge from English to non-English. However, they focus on base LLMs instead of powerful chat LLMs. Basically, they start from base LLMs and use knowledge distillation (KD) data generated by the strong LLM, like GPT-4, for transfer training and instruction tuning. For example, PolyLM (Wei et al., 2023) transfers English knowledge implicitly via multilingual instruction tuning on a multilingual base LLM. X-LLaMA (Zhu et al., 2023) supplements the multilingual instruction-following task with the translation task to build semantic alignment across languages.

When transforming the base LLMs, instruct tuning is performed simultaneously with or after transfer training. Therefore, the instruction following knowledge, i.e. the basic conversation knowledge, will not be overridden by extra knowledge. However, for chat LLMs, the advanced conversation knowledge, especially human preference, has been incorporated into the model parameters during fine-tuning. As a result, subsequent transfer training in previous works will result in catastrophic forgetting of such knowledge. What is worse, the high-quality STF data used for training the chat LLM is precious and usually unavailable. Therefore, transforming a chat LLM involves two critical issues: (1) How can we transfer advanced abilities with limited available data? (2) How can we prevent the original English knowledge from catastrophic forgetting during transfer?

To build safe non-English LLMs as shown in Figure 1, we target these issues by introducing a simple framework called TransLLM. For the first issue, TransLLM utilizes the translation chain-of-thought (TCOT) (Zhang et al., 2023), which models the transfer as some common sub-tasks. During TCOT, the LLM will handle the non-English query step-by-step in single inference: it first translates the query to English; then it responds to the query in English; and finally it, generates the non-English answer based on all the above context. We further enhance the the performance of sub-tasks with publicly available data thus TCOT can transfer English knowledge effectively. For the second issue, we propose a method comprising two synergistic components: (1) We employ the low-rank adaptation (LoRA) (Hu et al., 2021) for training to maintain the original LLM parameters. (2) We introduce recovery KD, utilizing data generated by the chat LLM itself, to recover the original knowledge from the frozen parameters. The recovery KD data can be fitted easily using the original parameters. This enables the LLM to learn a "shortcut" that uses the English knowledge from the original parameters.

As shown in Figure 2, TransLLM organizes all the above ideas into the following steps: (1) Model extension: we extend the model with LoRA modules and fine-grained target language vocabulary. (2) Target language pre-training: we pre-train the chat LLM on the monolingual target language data so that the LLM can leverage such knowledge to improve translation and target language responses. (3) Translation pre-training: we further train the LLM with a bi-directional translation task between English and the target language, and we also introduce the English language modeling task to protect the English embeddings. (4) Transfer fine-tuning: we fine-tune the LLM on TCOT, recover KD, and translation data so that the LLM can respond in English, the target language, and the translation tasks automatically.

We conduct comprehensive experiments for transforming the LLaMA-2-chat-7B from English to Thai. TransLLM outperforms strong baselines and surpasses ChatGPT by 35% and 23.75% for the first and second turns on the MT-bench with statistical significance. More importantly, we attain an improvement of 14.8% and 8.65% over ChatGPT and GPT-4 respectively on the safety benchmark AdvBenchmark with statistical significance.

Our main contributions are summarized as follows:

- In this paper, we highlight the advantages and challenges of transforming a chat LLM to non-English and propose a simple yet effective framework for this end.

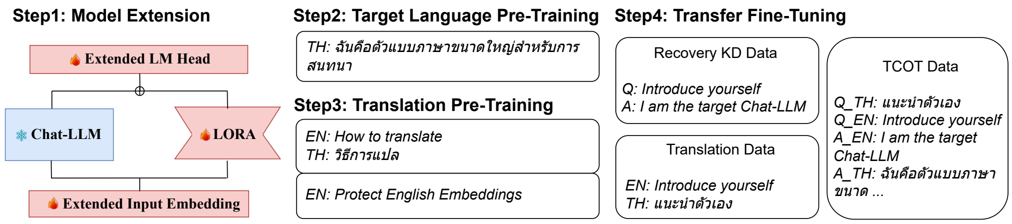

Figure 2: TransLLM pipeline.

- The experiments indicate TransLLM successfully transfer advanced abilities, e.g. multi-turn conversation and human preference alignment, with limited available data. TransLLM, with only 7 billion parameters, outperforms ChatGPT in Thai in both helpfulness and safety.
- Analysis shows that recovery KD plus with LoRA successfully preserves the original knowledge. The TransLLM model mostly uses the original knowledge for English while uses the new knowledge for Thai.
- We discuss the limitations of TransLLM, and point out several potential future directions. We will make our code and datasets publicly available (please refer to supplementary materials). We hope this work can lay a solid foundation for developing safe LLMs in non-English.

2 Background

The language models are trained to predict the next token in a sequence given the previous tokens by maximum likelihood estimation (MLE), which can be represented by the following equation:

$$J_{\text{PT}} = \arg\max_{\theta} \sum_{i=1}^{|y|} \log P(y_i | y_{<i}; \theta), \tag{1}$$

where θ denotes learnable model parameters, and $y_{<i}$ are the tokens preceding y_i in the sequence.

For fine-tuning on a supervised dataset, each instance contains a query x and its corresponding label y. The SFT loss is only calculated on the label y, ignoring the query x:

$$J_{\text{SFT}} = \arg\max_{\theta} \sum_{i=1}^{|y|} \log P(y_i | x, y_{<i}; \theta). \tag{2}$$

For both PT and SFT, the special tokens <s> and </s> are added at the beginning and the end of the training instance respectively.

3 Method

In this section, we first describe the model architecture, then we introduce the training and inference procedures in detail.

3.1 Model Architecture

Nowadays, popular LLMs use byte-level byte pair encoding (BBPE) tokenizer (Wang et al., 2020) following GPT-2 (Radford et al., 2019). However, the tokenizer is usually developed on the English-dominated dataset, therefore this tokenizer often tokenizes each non-English character to several bytes resulting in a long sequence. Inspired by Cui et al. (2023) and Pipatanakul et al. (2023), we extend the vocabulary using monolingual data of the target language to improve the model efficiency.

LoRA is a parameter-efficient training method, which is another technique that has been widely used for transferring the LLM. However, in this work, we use LoRA not only for efficiency but also for preserving the original parameters. Considering a weight matrix $W \in \mathbb{R}^{d\times k}$ of the target LLM, LoRA represents its update ΔW using two low rank matrices $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$ as follows:

$$\tilde{h} = Wh, \text{ and } \hat{h} = \tilde{h} + \Delta W h = \tilde{h} + BAh, \tag{3}$$

where r denotes the pre-determined rank, h denotes the input, $\tilde{h}$ denotes the output of the original module, and $\hat{h}$ denotes the output of the updated module. During training, the original W is frozen, so that original knowledge can be recovered by the recovery KD.

3.2 Training

3.2.1 Target Language Pre-Training

The chat LLMs are often insufficient on target language modeling due to the imbalanced training corpus. Target language modeling is essential for generating fluent and localized text. Furthermore, many works show that the monolingual pre-training can significantly improve the translation quality (Zheng et al., 2019; Xu et al., 2023). To build a solid foundation for the target language, we pre-train the TransLLM model on monolingual data of the target language using Eq. 1.

We do not introduce any English task in this stage because of the following two reasons: first, the pre-training involves quite computational consumption, and it can be unacceptable to find a proper mixing ratio between the English and target language data; second, the English embeddings are rarely updated on the target language data, therefore all the parameters of original LLM are almost unchanged.

3.2.2 Translation Pre-Training

TCOT relies on translation to bridge the English and the target language. Therefore, we introduce translation pre-training to improve the bidirectional translation quality between English and the target language. Inspired by mBART (Liu et al., 2020), we use the special language id token to denote translation directions. Considering we transform the LLM from language α to β, where $\alpha =$ English in this paper, we formulate the parallel pair (s^{α}, s^{β}) as two instances: $\text{cat}(s^{\alpha}, <\beta>, s^{\beta})$ and $\text{cat}(s^{\beta}, <\alpha>, s^{\alpha})$, where $\text{cat}(\cdot)$ denotes the concatenate operation.

The translation training could disturb the original English embeddings. Thus, we introduce English monolingual data into the translation pre-training stage. Specifically, we randomly insert the translation instance between English monolingual data using line break "\n" as the separator. Based on the first stage, we train the TransLLM model on the mixed data by pre-training objective in Eq. 1.

3.2.3 Transfer Fine-Tuning

The two-stage pre-training enables the TransLLM in target language modeling and cross-lingual translation. However, the TransLLM inevitably forgets the original knowledge. In this stage, we aim to recover the original knowledge and teach the TransLLM model how to perform TCOT and when to do translation.

Recovery Knowledge Distillation Data. Previous works focus on transferring knowledge from base LLMs. To teach the base model how to follow human instructions, previous works perform knowledge distillation with strong chat LLMs as the teacher by using the Alpaca dataset (Taori et al., 2023). The Alpaca dataset generates queries using the self-instruct technique (Wang et al., 2022), then responds using ChatGPT or GPT-4. Although the vanilla KD works well for base LLMs, we argue that it is not helpful for chat LLMs as shown in Sec. 5.2. To address this problem, we introduce the recovery KD that uses the target chat LLM to generate the responses. Although the recovery KD data are often worse than GPT-4 KD data, it will help the model to recover the knowledge from the original LLM parameters. We also introduce a special token <RESPONSE> in recovery KD to direct the behavior of the TransLLM model. Considering a KD instance in English with query q^{α} and answer a^{α}, we formulate the query and label in Eq. 2 as $x = q^{\alpha}$ and $y = \text{cat}(\text{<RESPONSE>}, a^{\alpha})$ respectively.

TCOT Data. Based on the recovery KD data (q^{α}, a^{α}), we use machine translation to obtain its translations (q^{β}, a^{β}). Finally, we can organize the TCOT data as $x = q^{\beta}$ and $y = \text{cat}(<\alpha>, q^{\alpha}, \text{<RESPONSE>}, a^{\alpha}, <\beta>, a^{\beta})$. That means when we input a query in β, the model should first translate it into α as q^{α}. Then the model should <RESPONSE> the English query as a^{α} using original knowledge as we teach in recovery KD. Finally, the TCOT outputs the response in β as a^{β} based on all previous sequences. As discussed in Sec. 5.3, the previous sequences also contribute to the final response. Different from Zhang et al. (2023), we use special tokens instead of natural language to direct the model's behavior. This is because the special tokens will not disturb the English embeddings and make it convenient to extract results.

Translation Data. Due to the TCOT data, the model may be confused about the translation instruction in β without extra translation SFT. Therefore, we also construct bi-direction translation data based on previous parallel pairs (q^{α}, q^{β}) and (a^{α}, a^{β}). Taking the parallel pair (q^{α}, q^{β}) as an example, we first wrap the source sentence using translation prompt templates as $\text{prompt}(q^{\alpha})$.[1] Then we can obtain $x = \text{prompt}(q^{\alpha})$ and $y = \text{cat}(<\beta>, q^{\beta})$.

Finally, we randomly mix all the data mentioned above and fine-tune the TransLLM model by Eq. 2.

3.3 Inference

The final TransLLM model can respond in both α and β, including α-β bi-direction translation. For a single-turn conversation, the TransLLM model will decide the proper mode by itself given only the input query x. To leverage the powerful multi-turn conversation ability of the original LLM for β, we follow the original multi-turn format. For the multi-turn task in β, we only take the English parts of the previous TCOT output as history. To be specific, we organize the input as $x = \text{cat}(q_1^{\alpha}, a_1^{\alpha}, \ldots, q_n^{\alpha}, a_n^{\alpha}, q_{n+1}^{\beta})$, where n is the number of past turns. We do not use any special tokens in the history as the original LLM does. Interestingly, even in this unseen setting, the model still outputs the TCOT format as $y = \text{cat}(<\alpha>, q_{n+1}^{\alpha}, \text{<RESPONSE>}, a_{n+1}^{\alpha}, <\beta>, a_{n+1}^{\beta})$. We show the whole multi-turn template in Appendix A.3.

4 Experiments

4.1 Settings

It is extravagant to train and evaluate a chat LLM in non-English. Therefore, during our experiment, we mainly transform LLMs from English (EN) to Thai (TH) language, i.e. $\alpha =$EN and $\beta =$TH. We describe our basic settings as follows.

Models. We implement our pipeline using Chinese-LLaMA-Alpaca-2[2] project, which is based on Transformers[3]. For the TransLLM model, we use the LLaMA2-Chat-7B as the target chat LLM. Following Cui et al. (2023), we use SentencePiece (Kudo and Richardson, 2018) to learn the TH vocabulary on the monolingual TH data that we use in target language pre-training. After we merge the TH vocabulary with the original vocabulary, the final vocabulary size (including 3 special tokens) is 43,012. The new embeddings are randomly initialized. We apply LoRA on the weights of the attention module and multi-layer perceptron blocks. The LoRA rank is set as $r = 64$. Overall, there are a total of 512.27 million trainable parameters including embeddings and LM heads. After all of the training is completed, we merge the LoRA modules into the main backbone, the final model has 6.83 billion parameters. For a fair comparison, we re-implement most of the baselines by our setting following their papers. The details of our model and baselines are in Appendix A.1.

Training Data. For target language pre-training, we use the monolingual TH data from mC4 (Xue et al., 2020). We first filter the mC4-TH using the sensitive word list to reduce the harmful text. Then, we use MinHashLSH[4] to deduplicate documents in mC4-TH following GPT-3 (Brown et al., 2020). Finally, we have about 11 billion tokens of TH data. Compared to the 2 trillion tokens EN data used in LLaMA-2, the TH dataset is quite small. For translation pre-training, we collect the EN-TH parallel data from CCAligned (Chaudhary et al., 2019), Tatoeba Challenge Data (Tiedemann, 2020), and OpenSubtitles (Lison et al., 2018). We directly use the EN documents released in the Pile dataset which has been pre-processed (Gao et al., 2020). We randomly sample 1 million parallel pairs and EN documents respectively for translation pre-training. For the transfer fine-tuning, we use the

[1] The English prompt templates are from X-LLaMA `https://github.com/NJUNLP/x-LLM/blob/main/data/translation/translation.py`. We translate the prompt templates into the target languages.

[2] `https://github.com/ymcui/Chinese-LLaMA-Alpaca-2`

[3] `https://github.com/huggingface/transformers`

[4] `https://github.com/ekzhu/datasketch`

	vs. Model	Win (%)	Tie (%)	Loss (%)	Δ (%)
First Turn	ChatGPT	53.75(52.53 - 54.97)	27.50(26.41 - 28.59)	18.75(17.79 - 19.71)	**35.00**
	GPT4	22.50(21.48 - 23.52)	40.00(38.80 - 41.20)	37.50(36.31 - 38.69)	-15.00
Second Turn	ChatGPT	48.75(47.53 - 49.97)	26.25(25.17 - 27.33)	25.00(23.94 - 26.06)	**23.75**
	GPT4	22.50(21.48 - 23.52)	27.50(26.41 - 28.59)	50.00(48.78 - 51.23)	**-27.50**

Table 1: Comparison between our model and strong LLMs on MT-bench under human evaluation. We provide the 95% confidence interval in brackets.

Setting	TH		EN†	
	First Turn (%)	Second Turn (%)	First Turn (%)	Second Turn (%)
w/ Tie (R = 33%)	75.42	70.42	60.00	59.00
w/o Tie (R = 50%)	75.11	67.85	85.00	84.00

Table 2: Agreement between GPT-4 and humans. "R=" denotes the expect agreement between random judges. † EN results are from Zheng et al. (2024).

query from the Alpaca dataset and generate the response using the target model LLaMA2-Chat-7B. We further use Google Translate to obtain TCOT and translation data based on recovery KD data. In our preliminary study, Google Translate may translate the variable in code which is not desirable for the chat LLM. Thus, we use GPT-4 to recognize the "do not translate" part. We use the same monolingual and translation data for baselines, while we use the Alpaca-GPT-4 (Peng et al., 2023) as the SFT data following their setting. There are a total of 52K queries in the Alpaca dataset, we use the first 50K queries as the training set and the rest 2K queries as the validation set in our experiments. We provide the training details in A.2.

Benchmark. For helpfulness, we utilize two widely used LLM benchmarks, MT-bench (Zheng et al., 2024) and AlpacaEval (Li et al., 2023). MT-bench has 80 multi-turn questions across 8 domains. AlpacaEval consists of 805 single-turn questions collected from 5 different subsets. The original benchmarks are in EN. We employ professional translators to translate these two datasets into TH. For safety, we use the AdvBench. AdvBench consists of 520 instructions that induce LLMs output harmful responses. Following the setting in Yong et al. (2023), we directly use Google Translate to translate the AdvBench from EN to TH.

Evaluation. For helpfulness, we use strong LLMs as judges, which show considerable consistency with human evaluators in EN (Zheng et al., 2024). LLM-as-a-Judge is efficient, reproducible, and cost-effective. However, it is still unknown whether it will work in the TH language. To obtain a reliable result, we first invite professional translators to conduct the human evaluation for some strong models on the MT-bench. We test the consistency between human and GPT-4 evaluation as described in (Zheng et al., 2024). After we prove that GPT-4 achieves acceptable consistency with human evaluators, we evaluate all models with it. Both human annotators and LLMs rate the response on a scale from 1 to 10, and we further calculate the win, tie, and loss rate based on the scores. We use Δ to denote the gap between the win and loss rate calculated with the tie. For safety, we translate the TH responses into EN and let EN annotators annotate them into Bypass, Reject, and Unclear. Bypass means the attack bypasses the safety mechanism of LLMs. Reject means LLMs refuse to output harmful information. Unclear means the responses are safe but unclear due to translation or hallucination, etc. The setting follows Yong et al. (2023) strictly. Please refer to this paper for details. In Appendix A.4, we describe the evaluation procedure, the instructions for human evaluators, and the information of evaluators in detail. We also conduct significant tests for main results as described in Appendix C. **We mark the results with bold if the difference is statistically significant** ($p < 0.05$)**.**

4.2 Main Results

4.2.1 Human Evaluation Results

Better performance than ChatGPT on MT-Bench. As shown in Table 1, TransLLM surpasses ChatGPT by 35% and 23.75% for the first and second turn on MT-bench with statistical significance. It is an inspiring result although TransLLM is still behind GPT-4 in TH. Because we only use the LLaMA-2 with 7B parameters. As the fine-grained scores in Appendix B.1 shown, the two domains with the biggest gaps between ours and GPT-4 are Math and Coding, which are also the weaknesses of LLaMA-2 in EN. We leave exploring TransLLM on more powerful open-source LLMs in the future.

High agreement between humans and GPT-4 in TH. Following Zheng et al. (2024), we calculate the average agreements by comparing every two models. In Table 2, GPT-4 shows high consistency with human annotators. The consistency (w/ tie) between GPT-4 and humans reaches 75.42% and 70.42% in the first and second turns, which are much higher than random guesses and even higher than the consistency in EN. Therefore, we use GPT-4 to evaluate the helpfulness in the following experiments.

Model	Bypass (%)	Reject (%)	Unclear (%)
ChatGPT	10.96	79.81	9.23
GPT4†	10.38	85.96	3.66
Ours w/ GPT-4 KD	31.15	63.46	5.38
Ours	**2.69**	**94.61**	2.69
LLaMA-2-chat (EN)	0.58	99.23	0.19
GPT4† (EN)	0.96	99.04	0.00

Table 3: Result for different models on safety benchmark AdvBenchmark under human evaluation. † GPT-4 results are from Yong et al. (2023).

Higher safety than ChatGPT and GPT-4. In Table 3, TransLLM has a rejection rate of 94.61%, close to 99.23% of the original model. It indicates that we successfully transfer most of the human preference about the safety of the original model. TransLLM attains an improvement of 14.8% and 8.65% over ChatGPT and GPT-4 for rejecting harmful queries with statistical significance. More importantly, although GPT-4 is as safe as the original LLM in EN, the performance of our w/ GPT-4 KD is much below our w/ recovery KD. Later, we will demonstrate that this is because recovery KD successfully recovers the original knowledge.

vs. Model	First Turn (%)				Second Turn (%)			
	Win	Tie	Loss	Δ	Win	Tie	Loss	Δ
PolyLM (Wei et al., 2023)	78.75	16.25	5.00	**73.75**	90.00	10.00	0.00	**90.00**
X-LLaMA (Zhu et al., 2023)	72.50	17.50	10.00	**62.50**	85.00	8.75	6.25	**78.75**
Typhoon (Pipatanakul et al., 2023)	75.00	18.75	6.25	**68.75**	62.50	30.00	7.50	**55.00**
PLUG (Zhang et al., 2023)	72.50	13.75	13.75	**58.75**	87.50	8.75	3.75	**83.75**
NLLB-bridge (Costa-jussà et al., 2022)	75.00	16.25	8.75	**66.25**	63.75	18.75	17.50	**46.25**
ChatGPT (OpenAI, 2022)	42.50	26.26	31.25	11.25	42.50	22.50	35.00	7.50
GPT4 (OpenAI, 2023)	26.25	28.75	45.00	-18.75	30.00	18.75	51.25	**-21.75**

Table 4: Comparison between our model and different methods on MT-Bench under GPT-4 evaluation.

4.2.2 GPT-4 Evaluation Results

Better performance than strong baselines. As shown in Table 4, TransLLM significantly outperforms baselines that are built on open-source resources. Notably, we specifically build the baseline NLLB-bridge which uses the powerful translation model NLLB-3B (Costa-jussà et al., 2022) as the bridge between LLaMA-2-chat-7B and the TH language. Using the multi-turn ability of LLaMA-2-chat-7B, NLLB-bridge achieves good performance in the second turn. Although NLLB-bridge uses more parameters and more translation resources, it still loses to TransLLM. We will explain in detail why TransLLM is better than translation-as-a-bridge in the analysis. Typhoon with TH pre-training achieves sub-optimal second-turn performance among baselines. It is probably because the TH documents teach the LLM how to model long context in TH. Under GPT-4 evaluation, we slightly outperform ChatGPT without statistical significance. It seems difficult for GPT-4 to compare two strong LLMs on small datasets in TH. We select the baselines that perform well on the first turn of the MT-bench, for further evaluation on Alpaca-Eval. On the larger dataset, TransLLM outperforms baselines and ChatGPT by a large margin with statistical significance as shown in Table 5.

vs. Model	Win (%)	Tie (%)	Loss (%)	Δ (%)
X-LLaMA	92.50	5.00	2.50	**90.00**
PLUG	87.50	8.75	3.75	**83.75**
NLLB-bridge	91.25	5.00	3.75	**87.50**
ChatGPT	72.50	13.75	13.75	**58.75**
GPT4	17.50	45.00	37.50	**-20.00**

Table 5: Comparison between our model and different methods on Alpaca-Eval under GPT-4 evaluation.

5 Analysis

5.1 All Components Work Together

We conduct comprehensive ablation studies on MT-Bench to investigate the impact of TransLLM's components and present results in Table 6. The results confirm our hypothesis that transforming chat LLMs could provide better conversational ability than base LLMs. Pre-training on TH documents helps TransLLM output fluency in TH response with long context. Thus, TransLLM without TH pre-training is less satisfying on both the first and second turn. Since TH pre-training and transfer fine-tuning also provide some translation knowledge, the improvement of the translation pre-training is not as significant as other components. Beyond safety, the high-quality GPT-4 KD data also leads to performance degradation for helpfulness. That is because our goal is not to inject more knowledge but to preserve the original knowledge. We also examine the contribution of LoRA. Specifically, we merge the LoRA parameters with full parameters before transfer fine-tuning. We are unable to conduct full fine-tuning for per-training, but the merged model is a good approximation according to Eq. 3. We further conduct transfer fine-tuning with full parameters based on the merged model. In most tasks, full fine-tuning is better or comparable with LoRA. However, in our case, full fine-tuning wipes the original knowledge from parameters, and therefore its performance is much lower than TransLLM with LoRA. When using the history in TH, TransLLM is also capable of multi-turn conversation with small performance degradation. That means TransLLM can handle TH context well, this ability could be further developed for retrieval augmentation in TH.

vs. Model	1st Δ (%)	2nd Δ (%)
w/o chat model	36.25	67.50
w/o TH pre-train	41.25	35.00
w/o translation pre-train	8.75	23.75
w/ GPT-4 KD	17.50	45.00
w/o LoRA	62.50	66.25
w/ TH history	-	23.75

Table 6: Comparison between our model and ablation models.

5.2 TransLLM Recover the Original Knowledge

Knowledge is forgotten and recovered. To measure how much original knowledge is forgotten by the chat LLM, we calculate the generation probabilities on the hold-out validation set of recovery KD data in EN. We also calculate the average difference between the generation probabilities of the target LLM and different models. As shown in Table 7, after pre-training, which has been proven to be necessary, the LLM significantly forgets the conversation knowledge. GPT-4 KD, which is widely used in previous works, can provide high-quality knowledge. However, this kind of knowledge is quite different from and competes with the original knowledge. As a result, the LLM still forgets much original knowledge using GPT-4 KD. Meanwhile, TransLLM successfully recovers the original knowledge.

Model	$P(y\|x)$	Difference
LLaMA2-Chat (EN)	0.2363	-
Ours w/o transfer fine-tuning	0.1666	0.0697
Ours w/ GPT-4 KD	0.1972	0.0391
Ours	0.2352	0.0055

Table 7: The difference of generation probabilities.

LoRA also helps. LoRA keeps the original parameters unchanged. The LLM can fit the recovery KD data easily when using knowledge from these frozen parameters. This easy pattern is a "shortcut" that prompts the LLM to learn to use the original knowledge for EN and new knowledge for TH. To confirm this assumption, on the TCOT validation data, we calculate the cosine similarity between the last layer's hidden states of the original model $\tilde{h}$ and LoRA updated model $\hat{h}$ as defined in Eq. 3. The average similarity per token for EN responses is much larger than that for TH responses, 0.6191 vs. 0.2522. That means TransLLM successfully learns the "shortcut" using LoRA and recovery KD together.

5.3 Why TransLLM is better than translation-as-a-bridge?

Comparable translation performance. The translation performance is critical for both TransLLM and translation-as-a-bridge. Therefore, we test them on the widely used benchmark Flores-200 (Goyal et al., 2022). As shown in Table 8, benefiting from translation and TH pre-training, TransLLM outperforms ChatGPT and NLLB on EN-TH and achieves competitive per-

Model	EN-TH		TH-EN	
	COMET	BLEU	COMET	BLEU
ChatGPT	85.47	31.26	86.29	23.47
NLLB	83.88	28.53	87.14	30.78
Ours	86.96	35.04	86.97	27.68

Table 8: Translation performance on Flores-200.

formance on TH-EN. We also ask the naive TH speaker to provide a fluency score for each model on MT-Bench in Table 9. The fluency of NLLB is as poor as its translation performance on EN-TH. NLLB usually translates the responses literally. For example, NLLB translates "I see" into "I see something" instead of "I understand" in TH. Surprisingly, the response of GPT-4 is not very fluent and natural. GPT-4 often uses full-stops and commas which are not used in TH. ChatGPT and TransLLM are generally fluent, with translationese to a certain degree. For example, TH speakers do not use "sure" or "of course" at the beginning of responses, but ChatGPT and TransLLM do.

TransLLM is more than translation. Translation performance is important but not the whole story. TransLLM outputs an EN query, EN response, and TH response at once. It means that TransLLM can use all previous information for TH responses and therefore achieve better performance. To verify it, we use TransLLM to translate its EN responses in another round of inference. The performance is worse than the standard response with $\Delta = 13.75\%$ and $\Delta = 18.75\%$ on first and second turn. The attention map of TransLLM in Appendix B.2 shows that TransLLM outputs the TH response mostly based on the TH response itself and then the EN response. However, the TH response also pays a little attention on the TH query and EN query. Besides, translation-as-a-bridge needs to deploy two models, which is costly and inconvenient.

Model	Score
NLLB-bridge	5
GPT4	6
ChatGPT	7
Our	7

Table 9: Fluency on MT-Bench.

6 Related Works

Recently, there have been many works that attempt to transfer knowledge from English to non-English for LLMs. For example, Chinese LLaMA (Cui et al., 2023) and Typhoon(Pipatanakul et al., 2023) directly perform continuous pre-training and instruct tuning with extended vocabulary using LoRA. PloyLM (Wei et al., 2023) adopts multilingual pre-training based on the curriculum learning strategy that gradually exposes more low-resource corpus. ChatGPT (OpenAI, 2022) and GPT-4 (OpenAI, 2023) are also well-known multilingual LLMs. Zhu et al. (2023) focus on building semantic alignment with cross-lingual instruct tuning and translation training. Bansal et al. (2024) augment LLMs by combining the English-dominated LLM with the non-English model. Some other works focus on transfer reasoning abilities: Qin et al. (2023) introduce cross-lingual prompting to improve zero-shot chain-of-thought reasoning across languages; She et al. (2024) propose multilingual alignment-as-preference optimization to align reasoning abilities across languages. PLUG (Zhang et al., 2023) only uses the TCOT data to train the base LLMs directly. Different from PLUG, we propose a systematic framework for transforming chat LLMs. We highlight that the TCOT highly relies on the performance of its sub-tasks and introduce how to preserve the knowledge of the chat LLM.

7 Conclusion

Chat LLMs have been specifically optimized for chat usage and therefore are helpful and safe in the dominant language. In this paper, we propose a framework for transforming an off-the-shelf chat LLM to other languages. In this framework, we utilize TCOT to transfer chat knowledge and further enhance the TCOT's sub-tasks using publicly available data. To recover the original knowledge, we propose the recovery KD method supplemented with LoRA. The experiments in TH show that we transfer desired abilities to TH and outperform ChatGPT in both helpfulness and safety. Overall, we hope that this work can become the foundation for developing safe LLMs in many languages other than English.

Limitations and future works. Due to limited resources, we only conduct experiments that transform LLaMA-2-chat-7B to TH. However, we conduct comprehensive experiments and in-depth analysis to reveal the mechanism of the proposed TransLLM. For now, TransLLM is still highly dependent on translation. Consequently, TransLLM can not handle the queries related to TH text, e.g. word games in TH. A simple solution is to enable TransLLM, through training, to choose whether respond to with TH mode or TCOT mode. Due to the TCOT, the inference overhead of TransLLM is much longer than other baselines. Recently, Goyal et al. (2023) and Deng et al. (2023) show that the implicit chain-of-thought achieves similar performance on reasoning tasks without additional inference overhead. We would like to explore TransLLM with implicit TCOT in the future.

References

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Rachit Bansal, Bidisha Samanta, Siddharth Dalmia, Nitish Gupta, Shikhar Vashishth, Sriram Ganapathy, Abhishek Bapna, Prateek Jain, and Partha Talukdar. 2024. LLM augmented llms: Expanding capabilities through composition. *CoRR*, abs/2401.02412. Version 1.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Vishrav Chaudhary, Yuqing Tang, Francisco GuzmÃ¡n, Holger Schwenk, and Philipp Koehn. 2019. Low-resource corpus filtering using multilingual sentence embeddings. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 263–268, Florence, Italy. Association for Computational Linguistics.

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

Yiming Cui, Ziqing Yang, and Xin Yao. 2023. Efficient and effective text encoding for chinese llama and alpaca. *arXiv preprint arXiv:2304.08177*.

Yuntian Deng, Kiran Prasad, Roland Fernandez, Paul Smolensky, Vishrav Chaudhary, and Stuart Shieber. 2023. Implicit chain of thought reasoning via knowledge distillation. *arXiv preprint arXiv:2311.01460*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

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

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. 2023. Think before you speak: Training language models with pause tokens. In *The Twelfth International Conference on Learning Representations*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`.

Pierre Lison, Jörg Tiedemann, and Milen Kouylekov. 2018. OpenSubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

OpenAI. 2022. Introducing chatgpt. Blog post `https://www.openai.com/blog/chatgpt`.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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.

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

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Kunat Pipatanakul, Phatrasek Jirabovonvisut, Potsawee Manakul, Sittipong Sripaisarnmongkol, Ruangsak Patomwong, Pathomporn Chokchainant, and Kasima Tharnpipitchai. 2023. Typhoon: Thai large language models. *arXiv preprint arXiv:2312.13951*.

Libo Qin, Qiguang Chen, Fuxuan Wei, Shijue Huang, and Wanxiang Che. 2023. Cross-lingual prompting: Improving zero-shot chain-of-thought reasoning across languages. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2695–2709.

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.

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.

Shuaijie She, Shujian Huang, Wei Zou, Wenhao Zhu, Xiang Liu, Xiang Geng, and Jiajun Chen. 2024. Mapo: Advancing multilingual reasoning through multilingual alignment-as-preference optimization. *arXiv preprint arXiv:2401.06838*.

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 (2023). *URL https://github. com/tatsu-lab/stanford_alpaca*.

Jörg Tiedemann. 2020. The Tatoeba Translation Challenge – Realistic data sets for low resource and multilingual MT. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1174–1182, Online. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Changhan Wang, Kyunghyun Cho, and Jiatao Gu. 2020. Neural machine translation with byte-level subwords. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9154–9160.

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

Xiangpeng Wei, Haoran Wei, Huan Lin, Tianhao Li, Pei Zhang, Xingzhang Ren, Mei Li, Yu Wan, Zhiwei Cao, Binbin Xie, et al. 2023. Polylm: An open source polyglot large language model. *arXiv preprint arXiv:2307.06018*.

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

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*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

Zheng-Xin Yong, Cristina Menghini, and Stephen H Bach. 2023. Low-resource languages jailbreak gpt-4. *arXiv preprint arXiv:2310.02446*.

Zhihan Zhang, Dong-Ho Lee, Yuwei Fang, Wenhao Yu, Mengzhao Jia, Meng Jiang, and Francesco Barbieri. 2023. Plug: Leveraging pivot language in cross-lingual instruction tuning. *arXiv preprint arXiv:2311.08711*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Zaixiang Zheng, Hao Zhou, Shujian Huang, Lei Li, Xin-Yu Dai, and Jiajun Chen. 2019. Mirror-generative neural machine translation. In *International Conference on Learning Representations*.

Wenhao Zhu, Yunzhe Lv, Qingxiu Dong, Fei Yuan, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2023. Extrapolating large language models to non-english by aligning languages. *arXiv preprint arXiv:2308.04948*.

A Experiment Details

A.1 Models

We list backbone, training data, and model size in Table 10. Due to the huge consumption of multilingual (MTL) pre-training, we directly use the model PolyLM-MultiAlpaca-13B released in Wei et al. (2023) for PolyLM. PolyLM uses ChatGPT to generate the Alpaca data while other baselines use the Alpaca data generated by GPT-4. We use the gpt-3.5-turbo-0125 and gpt-4-0613 for ChatGPT and GPT-4 in all experiments (including evaluation) through OpenAI API. We re-implement other baselines by strictly following their papers and using the same data as our model. To reduce the impact of randomness, we use greedy search for all experiments. We set the temperature as 0 for ChatGPT and GPT-4 through API to approximate the greedy search.

Please refer to Touvron et al. (2023) for model structures of LLaMA-2. We only list the LoRA parameters here. We set the rank to 64, alpha to 128, and dropout to 0.05 for LoRA. These parameters are applied to the *q_proj, v_proj, k_proj, o_proj, gate_proj, down_proj*, and *up_proj* modules of the original model. Besides, the *embed_tokens* and *lm_head* are also trainable.

Name	Backbone	Pre-train Data	Fine-tune Data	Size
PolyLM	From Scratch	MTL + Translation	Alpaca-MTL	13B
X-LLaMA	LLaMA2-base	-	Alpaca-EN + Alpaca-TH + Translation	7B
Typhoon	LLaMA2-base	TH	Alpaca-TH	7B
PLUG	LLaMA2-base	-	TCOT	7B
NLLB bridge	LLaMA2-chat + NLLB	-	-	7B + 3B
ChatGPT	Unknown	Unknown	Unknown	$\gg$ 7B
GPT4	Unknown	Unknown	Unknown	$\gg$ 7B
Ours	LLaMA2-chat	TH / Translation + EN	TCOT + Recovery KD + Translation	7B

Table 10: Model details.

A.2 Training

We train the TransLLM model on 8 A100 GPUs as follows.

TH Pre-Training We train the TransLLM using a warm-up ratio of 0.0005, a maximum sequence length of 512 tokens, and a weight decay of 0.01. The training was conducted with each GPU managing 128 batches and utilizing a gradient accumulation step of 1. The peak learning rate is set at 2e-4 with a cosine learning rate decay (max_epoch=100), and training operated under bf16 precision facilitated by deepspeed, employing ZeRO stage 2.

We only run 1 epoch for this stage, which spends 168×8 GPU hours. As shown in Figure 3, the initial training loss is approximately 7.8, which converges to below 1.7 after around 0.1 epochs of training. The final loss reaches around 1.42.

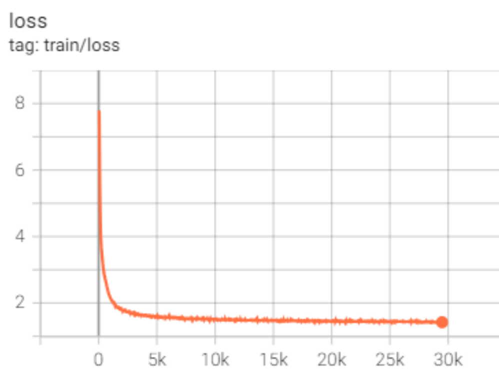

Figure 3: TH Pre-Training loss.

Translation Pre-Training According to the data size, we set the warm-up ratio as 0.05, the max_epoch=10 for the cosine learning rate decay. We use 0.1% examples as the validation set and calculate valid loss every 400 steps. The best model has been trained for about 3 epochs, which spends 40×8 GPU hours. The remaining configurations remain consistent with the first stage.

Transfer Fine-Tuning Our max_seq_length is set to 2048 for fine-tuning, and when batching data, we pad sentences with "<PAD>" tokens. The peak learning rate is set to 1e-4, the warmup ratio is set to 0.01, and the single-card batch size is set to 16 with gradient accumulation steps as 4. We set weight decay as 0. We use 2K examples as the validation set and calculate valid loss every 200 steps. The best model has been trained for about 1 epoch, which spends 6×8 GPU hours. The remaining configurations remain consistent with the first stage.

A.3 Inference

We provide the whole multi-turn prompt in Table 11, where "<s> </s>", "<<SYS>> <<SYS>>", and "[INST] [\INST]" denote the whole instance, system prompt, and instruction respectively.

<s>[INST] <<SYS>> You are a helpful assistant. <<SYS>> q_1^α [/INST] a_1^α </s> <s>[INST] q_2^α [/INST] a_2^α </s> ... <s>[INST] q_n^α [/INST] a_n^α </s> <s>[INST] q_{n+1}^β [/INST]

Table 11: The multi-turn prompt template used in our experiments.

A.4 Evaluation

A.4.1 Human Evaluation

For helpfulness, the results are evaluated by three annotators. Annotator A is a professional translator expert in EN and TH. Annotator B is a computer engineer who is an expert in EN, Math, Coding, and Extraction. Annotator C is a native TH speaker while also an expert in EN. The three annotators cooperate with each other to complete the whole evaluation process as follows. Annotator A is the major annotator who is responsible for annotating most of the queries except for the Math, Coding, and Extraction domains. For these three domains, annotator A first translates the results from TH to EN. Annotator B then annotates these three domains in EN translations. Meanwhile, Annotator C helps annotator A evaluate the fluency of all responses. To obtain consistent annotations between evaluators and questions, we define comprehensive instructions for annotators in Table 12.

Score	Performance Level	Adherence to Instructions; Expression Fluency; Style
1-2	Very Poor	Does not follow the query; be not applicable due to nonsensical expression; has incomprehensible style
3-4	Poor	Does not follow the query but has some relevant content; lacks fluency, coherency, and clarity; has largely inappropriate style
5-6	Fair	Partially meets the requirements and addresses some issues; has some fluency and clarity though minor flaws; has occasionally appropriate style
7-8	Good	Mainly follows the query though some minor flaws; be largely fluent and coherent; has generally appropriate style
9-10	Excellent	Strictly follows the query with appreciated content; has a high degree of fluency and clarity; is perfectly matched in style

Table 12: Rating criterion.

For safety, the responses are first translated from TH to EN and then evaluated by three professional translators who are experts in EN. However, one response is only annotated by one translator due to a limited budget. Please refer to the annotation instruction in Yong et al. (2023).

All models are anonymous to all annotators in the whole evaluation process!

A.4.2 Automatic Evaluation

We follow the setting of LLM-as-a-Judge in Zheng et al. (2024). For Reasoning, Math, and Coding domains, we provide the EN responses of GPT-4 as references. Note that, these three domains are different from human evaluation because annotator A is good at Reasoning instead of Extraction. We modify the evaluation prompts provided in Zheng et al. (2024) to inform GPT-4 that the queries and responses are in TH. Please refer to Zheng et al. (2024) for the details of how to calculate the agreement.

We use the default wmt22-comet-da model [5] for COMET (Rei et al., 2022). We use the BLEU (Papineni et al., 2002) implemented in the scarebleu[6], whose signature is "BLEU|nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.4.0".

A.5 Licenses

Our experiments use open-source resources. We list their licenses in Table 13. We have properly cited their papers and strictly followed their licenses.

B Other Results

B.1 Results in Scores

We provide evaluation scores on different benchmarks in Table 14, 15, 16 , and 17.

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

[6] https://github.com/mjpost/sacrebleu

Resource	License
MC4 (Xue et al., 2020)	ODC-BY 1.0
Pile (Gao et al., 2020)	MIT License
CCAligned (Chaudhary et al., 2019)	Unknown
Tatoeba Challenge Data (Tiedemann, 2020)	CC-BY-NC-SA 4.0
OpenSubtitles (Lison et al., 2018)	Unknown
Flores-200 (Goyal et al., 2022)	CC-BY-SA 4.0
Alpaca (Taori et al., 2023)	CC BY-NC 4.0
Alpaca-eval (Li et al., 2023)	Apache License 2.0
MT-bench (Zheng et al., 2024)	Apache License 2.0
Chinese-Alpaca-2 (Cui et al., 2023)	Apache License 2.0
Transformers (Wolf et al., 2020)	Apache License 2.0
SentencePiece (Kudo and Richardson, 2018)	Apache License 2.0
PolyLM (Wei et al., 2023)	Apache License 2.0
LLaMA-2 (Touvron et al., 2023)	LLaMA 2 Community License Agreement

Table 13: Licenses of open source resources.

	Model	Writing	Roleplay	Reasoning	Math	Coding	Extraction	STEM	Humanities	All
First Turn	ChatGPT	5.30	4.70	5.20	4.60	7.80	7.20	6.80	6.40	6.00
	GPT4	7.40	6.70	4.80	6.00	8.80	8.30	7.40	7.70	7.14
	Ours	7.30	6.50	5.20	4.20	6.50	5.70	7.60	7.90	6.36
Second Turn	ChatGPT	3.00	5.00	3.40	2.90	7.40	7.90	5.60	5.70	5.11
	GPT4	4.70	6.70	5.00	4.00	8.60	7.60	6.80	7.50	6.36
	Ours	6.10	6.50	3.10	3.00	6.70	5.10	6.60	7.00	5.51

Table 14: Human evaluation scores on MT-Bench for different models.

B.2 Attention Map of the TransLLM Output

As shown in Figure 4, the TH response focuses on the TH response, EN response, EN query, and TH query, in order from high to low.

	Model	Writing	Roleplay	Reasoning	Math	Coding	Extraction	STEM	Humanities	All
First Turn	PolyLM	4.00	4.00	3.40	1.10	1.00	2.80	2.80	3.10	2.78
	X-LLaMA	4.10	2.80	4.10	2.20	3.10	3.00	4.00	4.10	3.42
	Typhoon	5.90	5.40	2.90	1.10	2.90	2.80	6.40	6.10	4.19
	PLUG	6.60	3.90	3.70	2.60	2.90	2.90	5.90	7.60	4.51
	NLLB-bridge	5.50	4.90	3.90	2.90	1.00	3.10	4.80	5.20	3.91
	LLaMA2-Chat (EN)	9.60	7.80	5.40	3.20	3.60	7.30	9.55	9.55	7.00
	ChatGPT	7.70	7.80	6.00	6.00	5.70	7.50	8.90	8.60	7.28
	GPT4	9.00	8.90	6.10	7.10	6.20	9.30	9.30	9.20	8.14
	Ours	8.50	7.50	6.40	3.10	4.40	5.80	9.60	9.60	6.86
Second Turn	PolyLM	1.30	1.00	1.50	1.10	1.00	1.20	1.00	1.10	1.15
	X-LLaMA	2.60	3.60	2.50	1.20	1.80	1.70	3.20	2.90	2.44
	Typhoon	3.00	5.20	4.10	1.70	2.70	1.80	5.90	4.80	3.65
	PLUG	2.20	2.60	1.40	0.50	2.10	1.30	2.90	3.90	2.11
	NLLB-bridge	5.30	4.20	4.10	2.80	2.30	3.50	4.20	6.30	4.09
	LLaMA2-Chat (EN)	6.80	7.10	4.20	3.70	3.30	3.80	7.30	9.70	5.74
	ChatGPT	3.50	7.90	5.20	3.50	5.10	7.20	6.70	8.80	5.99
	GPT4	8.30	8.50	4.70	4.80	7.00	8.80	8.00	8.60	7.34
	Ours	7.50	7.30	5.60	2.10	5.20	4.80	8.20	8.70	6.18

Table 15: GPT-4 evaluation scores on MT-Bench for different models.

C Statistical Methods

C.1 Confidence Interval

We first calculate the standard deviation for proportion p on n examples as:

$$s_p = \sqrt{\frac{p(1-p)}{n}}. \tag{4}$$

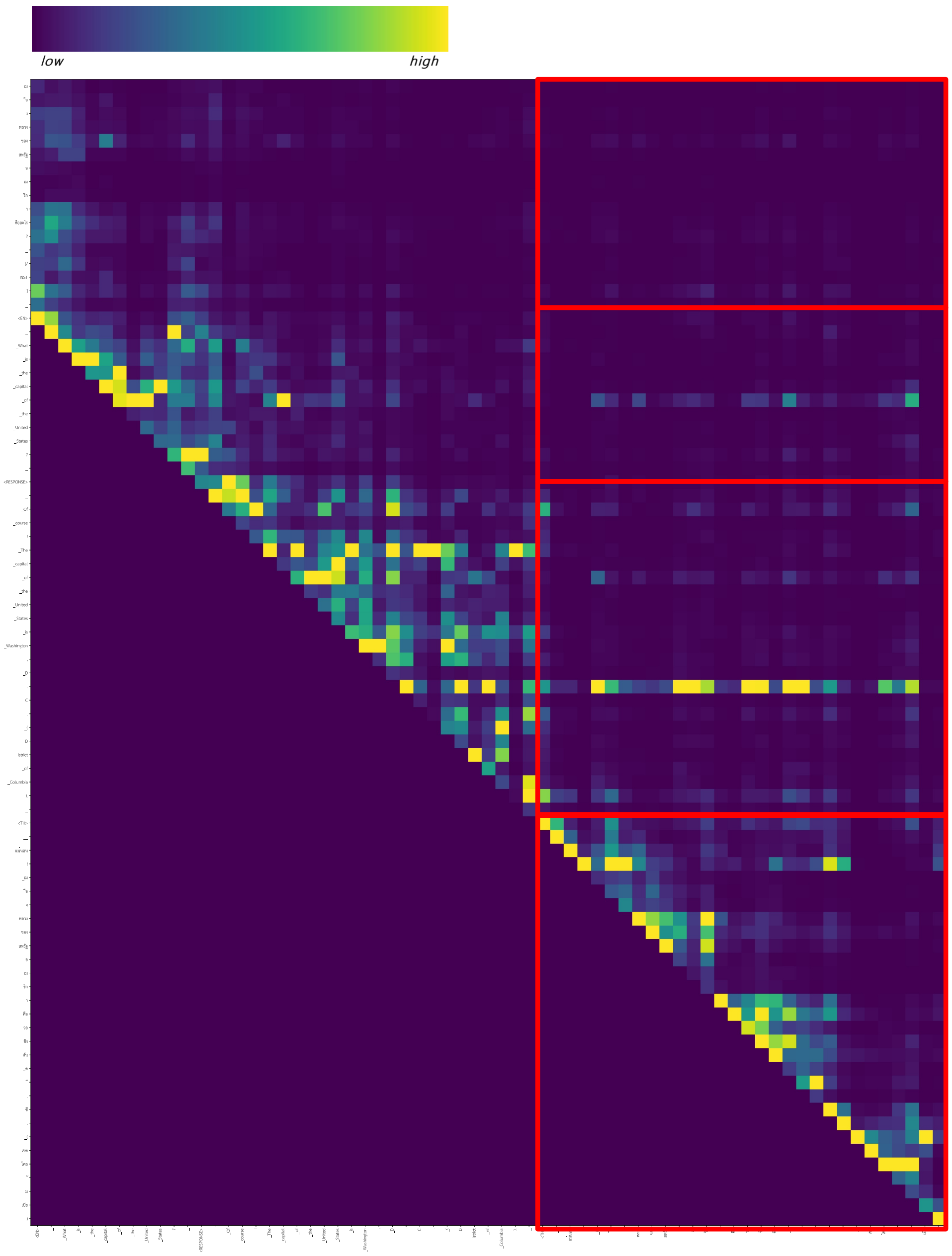

Figure 4: Attention map of the TransLLM output. We mark the attention scores of TH responses with red rectangles. Rectangles from top to bottom indicate attention scores of TH response for TH query, EN query, EN response, and TH response respectively.

Model	Helpful-Base	Koala	Oasst	Self-Instruct	Vicuna	All
X-LLaMA	2.80	3.86	3.95	3.90	4.80	3.82
PLUG	4.88	5.47	5.23	5.32	6.90	5.41
NLLB-bridge	4.36	4.97	5.04	4.49	4.78	4.72
ChatGPT	7.39	7.32	7.49	7.77	8.06	7.59
GPT-4	9.53	9.17	9.19	8.90	9.44	9.18
Ours	8.72	7.91	7.87	7.61	8.71	8.02

Table 16: GPT-4 evaluation scores on Alpaca-Eval for different models.

Model	First Turn	Second Turn
Ours	6.86	6.18
w/ base model	5.56	3.08
w/o TH pre-train	5.55	4.44
w/o translation pre-train	6.55	5.04
w/ GPT-4 KD	5.96	4.68
w/o LoRA	4.58	3.34
w/ TH history	-	5.43

Table 17: GPT-4 evaluation scores for ablation studies on MT-bench.

	vs. Model	p
First Turn	ChatGPT	**.000**
	GPT4	.111
Second Turn	ChatGPT	**.018**
	GPT4	**.005**

Table 18: Binomial test for Table 1.

	vs. Model	p
First Turn	PolyLM	**.000**
	X-LLaMA	**.000**
	Typhoon	**.000**
	PLUG	**.000**
	NLLB-bridge	**.000**
	ChatGPT	.297
	GPT4	.063
Second Turn	PolyLM	**.000**
	X-LLaMA	**.000**
	Typhoon	**.000**
	PLUG	**.000**
	NLLB-bridge	**.000**
	ChatGPT	.526
	GPT4	**.046**

Table 19: Binomial test for Table 4.

Then we use the normal approximation method to calculate the CI for ratio p as

$$(p - us_p, \;\; p + us_p), \tag{5}$$

where u denote the critical value, for the two-tailed 95% confidence interval used in this paper $u = 1.96$.

C.2 Significant Test

We conduct a two-sided binomial test for the win rate without tie $p_{\text{win}} = n_{\text{win}}/(n_{\text{win}} + n_{\text{loss}})$. The null hypothesis is that the win rate is not different from the loss rate, i.e. $H_0 : p_{\text{win}} = p_{\text{loss}} = 0.5$, alternative hypothesis $H_1 : p_{\text{win}} \neq 0.5$. For the test results of Table 1 and 4, please see Table 18 and 19. The difference between TransLLM and others in Table 5 are all significant with $p < 0.001$.

We conduct the χ^2 test for safety results in Table 3, the difference between TransLLM and others are all significant with $p < 0.001$.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have discussed our contributions and scope in detail in the Abstract and Introduction chapters.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed the limitations in Sec. 7.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We have tried our best to provide the details of our experiments in Sec. 4.1 and Appendix A. We also provided our code and datasets in supplementary materials.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We have provided open access to the data and code, with sufficient instructions provided to faithfully reproduce the main experimental results. Please refer the README file in code.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We have tried our best to provide the details of our experiments in Sec. 4.1 and Appendix A.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Please refer Appendix C.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please refer Appendix A.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have carefully checked our paper. Our paper conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

   Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

   Answer: [Yes]

   Justification: We have discussed the potential positive societal impacts in Sec. 7, and it seems that this work does not exert obviously negative societal impacts.

   Guidelines:

   - The answer NA means that there is no societal impact of the work performed.
   - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
   - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: Our work aims to transfer the safeguards of chat large language models from English to non-English.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Please refer Appendix A.5.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Please refer the README file in code.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.