Ladder: A Model-Agnostic Framework Boosting LLM-based Machine Translation to the Next Level

Anonymous ACL submission

Abstract

General-purpose Large Language Models (LLMs) like GPT-4 have achieved remarkable advancements in machine translation (MT) by leveraging extensive web content. On the other hand, translation-specific LLMs are built by pre-training on domain-specific monolingual corpora and fine-tuning with human-annotated translation data. Despite the superior performance, these methods either demand an unprecedented scale of computing and data or substantial human editing and annotation efforts. In this paper, we develop **Ladder**, a novel model-agnostic and cost-effective tool to refine the performance of general LLMs for MT. Ladder is trained on pseudo-refinement triplets which can be easily obtained from existing LLMs without additional human cost. During training, we propose a hierarchical fine-tuning strategy with an easy-to-hard schema, improving Ladder's refining performance progressively. The trained Ladder can be seamlessly integrated with any general-purpose LLMs to boost their translation performance. By utilizing Gemma-2B/7B as the backbone, Ladder-2B can elevate raw translations to the level of top-tier open-source models (e.g., refining BigTranslate-13B with +6.91 BLEU and +3.52 COMET for XX→En), and Ladder-7B can further enhance model performance to be on par with the state-of-the-art GPT-4. Extensive ablation and analysis corroborate the effectiveness of Ladder in diverse settings. Data and code will be released.

1 Introduction

General-purpose Large Language Models (LLMs) like GPT-4 (Achiam et al., 2023) have exhibited strong translation abilities (Hendy et al., 2023; Zhu et al., 2023; Jiao et al., 2023b), but achieving this performance requires enormous model scale, infrastructure, and deployment costs. On the other hand, translation-specific LLMs like ALMA (Xu et al., 2023a) and Aya 23 (Aryabumi et al., 2024) have reached top-tier levels through continued pretraining on large monolingual corpora (e.g., 20B tokens from Common Crawl (Su'arez et al., 2019)) and fine-tuning on high-quality translation data (e.g., 10.5M translation examples from Aya Dataset (Singh et al., 2024)), which is also time-consuming and costly. These observations raise a question: *can we enhance the MT performance of existing LLMs in a model-agnostic manner, achieving results comparable to translation-specific LLMs or even GPT-4, without incurring the significant costs associated with human annotations or extensive training?*

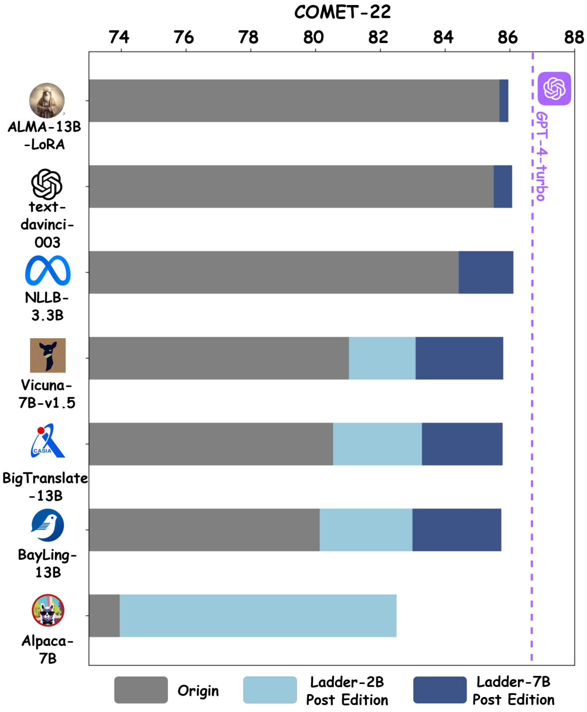

Figure 1: The average translation quality improvements across 8 translation directions on WMT22 test set (Zh↔En, De↔En, En↔Ru, En↔Cs) using Ladder-2B or 7B. The metric scores are calculated by COMET-22 (*wmt22-comet-da*) (Rei et al., 2020).

There are two potential approaches to achieving this goal. The first is the prompt-based method, which involves developing effective prompting strategies to better stimulate LLMs' translation capabilities, such as using in-context translation examples, as outlined in works (Agrawal et al., 2023; Garcia et al., 2023; Peng et al., 2023; Chen et al., 2023; Feng et al., 2024). However, Zhang et al. (2023a) indicate that prompting methods overly rely on the language model, often under-translate the input and generate hallucinations. Additionally, Moslem et al. (2023) demonstrate that the same prompting strategy can lead to different performance across different models. Furthermore, most of these prompting strategies like agent debating or self-correction (Liang et al., 2023; Feng et al., 2024) cannot be applied to some popular neural machine translation models like NLLB (Costa-jussà et al., 2022). These limitations make the learning-free method non-model-agnostic and unstable.

Another line of work employs learning-based paradigms by fine-tuning LLMs to adapt Quality Estimation (QE, Specia et al., 2010) and Automatic Post-Editing (APE, Simard et al., 2007) tasks to refine raw translations. QE involves automatically predicting translation quality, typically using Multidimensional Quality Metrics (MQM) datasets (Freitag et al., 2021), where human experts annotate error spans and assign quality scores. APE aims to address systematic errors of a black-box MT system and tailor the output to the lexicon and style required in a specific application domain. APE datasets are manually collected from real-world post-editing triplets like QT21 (Specia et al., 2017). Built on these well-defined tasks and annotated datasets, prior works (Zeng et al., 2023; Xu et al., 2023b; Alves et al., 2024) have shown the promising utility and generalization of the learning-based method. Xu et al. (2023b) trained PaLM2 (Anil et al., 2023) on MQM datasets to refine translations, and Alves et al. (2024) trained TowerInstruct on 637k translation examples, integrating APE datasets, outperforming all open models and GPT-3.5-turbo on APE tasks. However, these works heavily rely on human-annotated evaluation data and lack extensive validation in model-agnostic and multilingual scenarios. Additionally, the overall refinement in translation quality, particularly for translation-specific models, remains limited.

In this paper, we introduce **Ladder**, a model-agnostic and cost-effective tool for multilingual translation refinement. Instead of directly fine-tuning a translation-target LLM, we train an LLM to refine translations using refinement datasets without human evaluation or post-edits, employing an instruction-following refinement task (Section 2.1). We notice that the *reference* in existing parallel corpus can serve as a natural refined translation. By sampling a translation for the source sentence from an existing LLM as the *intermediate translation*, we create a pseudo-refinement translation triplet [*source, intermediate translation, reference*], allowing us to construct training data without extra labor costs. During training, we split the training triplets into three hierarchies (*Easy*, *Medium*, *Hard*) based on their COMET (Rei et al., 2020) scores and propose a hierarchical fine-tuning strategy to improve Ladder's refining performance step by step. Comprehensive experiments demonstrate that effectiveness of our Ladder across various LLMs on multiple translation tasks.

2 Ladder

2.1 Problem Formulation and Overview

Previous works (Zhang et al., 2023b; Xu et al., 2023a) adapt LLMs to translation tasks by fine-tuning on a parallel corpus [*source*, *reference*] using direct translation ($\mathcal{P}_D$) as shown in Figure 3. In contrast, we define our task as a refinement-target translation ($\mathcal{P}_R$) as shown in Figure 3, teaching the pre-trained base model to refine the existing translation of LLMs to the reference, rather than translating directly to the reference. Specifically, we introduce the concept of *intermediate translation*, which denotes the translation sampled from existing LLMs. Then we add the intermediate translation to the pair [*source*, *reference*] to form a pseudo-refinement triplet [*source*, *intermediate translation*, *reference*], taking the reference as the pseudo-refined translation. The concept of translation refinement rather than direct translation is a key distinction of our work compared to previous translation-specific LLM approaches.

Ladder models are created in two steps: 1) Sampling; and 2) Hierarchical Fine-tuning (HFT). First, given an existing LLM $\mathcal{M}_S$ and a parallel corpus $\mathcal{C}$, we use $\mathcal{M}_S$ to generate intermediate translations $i \sim \mathcal{M}_S(s, \mathcal{P}_D)$ for each source sentence s in the pair $(s, r) \in \mathcal{C}$, where r is the reference. We then combine i with (s, r) to create pseudo-refinement triplets (s, i, r), forming our training triplets $\mathcal{T}$. Second, we apply a hierarchical fine-tuning method with an easy-to-hard schema to fine-tune the base model on our instruction-following refinement task with triplet training data to obtain

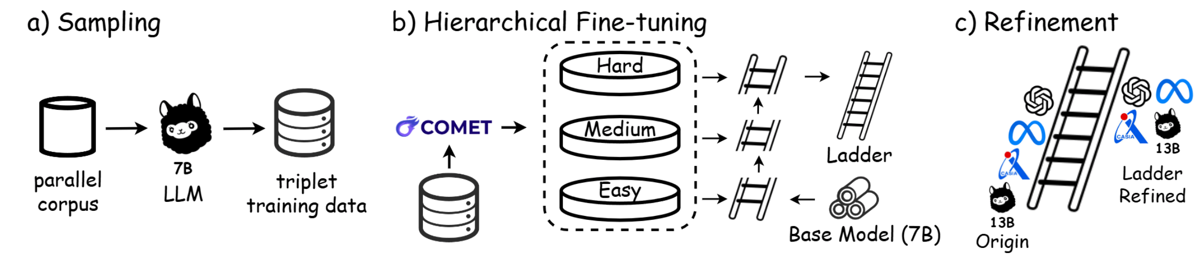

Figure 2: Obtain Ladder in two steps: a) Sample from an LLM using the parallel corpus to create pseudo-refinement triplet training data. b) Use a hierarchical fine-tuning method with an easy-to-hard schema to tune the pre-trained base model and obtain Ladder. Ladder can refine models with significantly higher parameter counts than the sampling LLM and base model. It can also enhance original translations from various sources to the next level.

Ladder $\mathcal{L}_a$. When applying $\mathcal{L}_a$ to refine the target LLM $\mathcal{M}_T$, $\mathcal{M}_T$ first generates the translation $i_{test} \sim \mathcal{M}_T(s_{test}, \mathcal{P}_D)$. $\mathcal{L}_a$ then refines i_{test} into the final translation $y_{final} \sim \mathcal{L}_a(s_{test}, i_{test}, \mathcal{P}_R)$. Figure 2 shows the pipeline.

2.2 Pseudo-refinement Triplet Construction

Our pseudo-refinement triplet [*source*, *intermediate translation*, *reference*] is similar in format to APE triplet [*source*, *translation with errors*, *post-edits*]. However, the APE annotation procedure involves significant human costs for evaluation, error marking, and post-editing, focusing on word- or phrase-level corrections rather than overall translation quality improvement (Specia et al., 2017). In contrast, our work uses reference r as the supervised label, focusing on overall quality. Given the sampling LLM $\mathcal{M}_S$ with parameters θ_S, parallel corpus $\mathcal{C}$ and prompt $\mathcal{P}_D$, the intermediate translation i for each pair $(s, r) \in \mathcal{C}$ can be generated auto-regressively as $i_t \sim p_{\theta_S}(i_t \mid s, \mathcal{P}_D, i_{<t})$. Naturally, the quality of i is inferior to r, so we treat r as the refined translation and construct our pseudo-refinement triplet training data $(s, i, r) \in \mathcal{T}$ without additional human costs.

2.3 Hierarchical Fine-tuning

Before fine-tuning, we use COMET (Rei et al., 2020) to categorize the pseudo-refinement triplet training data $\mathcal{T}$ into three levels: *Easy*, *Medium*, and *Hard* and propose a hierarchical fine-tuning (HFT) strategy to achieve better refinement performance by learning from *Easy* to *Hard* examples. *Easy* translations differ significantly from the reference, offering the most room for refinement. *Hard* translations are nearly perfect, with minimal differences, making them the hardest to refine. *Medium* translations fall between these two poles. Translations with COMET scores below μ are classified as *Easy*, scores between μ and ν as *Medium*, and scores above ν as *Hard*. We set thresholds μ and ν to 0.75 and 0.85, respectively, and analyze the effects of HFT and its robustness against these thresholds in Section 3.3.

a) Direct Translation

```
###You are a good [source language]-[target language] translator.
###Translate this from [source language] to [target language]:
[source language]: [source sentence]
[target language]:
```

b) Refinement-target Translation

```
###You are a good [source language]-[target language] translator.
###Now I have a translation pair including '[source language]
Source' and 'Intermediate Translation'. Modify the 'Intermediate
Translation' to the Final [target language] translation:
[source language] Source: [source sentence]
Intermediate Translation: [intermediate translation]
Final [target language] Translation:
```

Figure 3: Prompts used: [*source language*] and [*target language*] represent the full names of the languages. [*source sentence*] is the sentence to be translated. [*intermediate translation*] is the sampled translation. For Direction Translation, we follow Xu et al. (2023a).

We fine-tune the pre-trained base model using instruction tuning (IT), aiming to obtain the model $\mathcal{L}_a(\theta)$ on pseudo-refinement triplet training data $\mathcal{T} = \{s^{(k)}, i^{(k)}, r^{(k)}\}_{k=1}^{N}$ by minimizing the following objective:

$$\mathcal{L}(\boldsymbol{\theta}; \mathcal{T}) = -\mathbb{E}_{(\boldsymbol{s},\boldsymbol{i},\boldsymbol{r})\sim\mathcal{T}} \left[\log \mathcal{L}_a(\boldsymbol{r} \mid \boldsymbol{s}, \boldsymbol{i}, \mathcal{P}_R; \theta)\right] \quad (1)$$

We start with *Easy* examples to help the base model capture detectable differences, then progressively fine-tune with the next level of examples, building on the previous stage.

2.4 Translation Refinement

When using Ladder $\mathcal{L}_a$ with parameters $\theta_{\mathcal{L}_a}$ for refinement, given any target LLM $\mathcal{M}_T$ capable of translation, we first utilize $\mathcal{M}_T$ to generate the intermediate translation $i_{test} \sim \mathcal{M}_T(s_{test}, \mathcal{P}_D)$.

Models	Zh-En		De-En		Ru-En		Cs-En		Avg.	
	BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET
Open										
Alpaca-7B	11.80	73.36	24.52	81.37	30.49	80.68	27.31	77.99	23.53	78.35
BigTranslate-13B	14.32	74.63	23.17	81.04	28.05	78.38	34.49	81.99	25.01	79.01
BayLing-13B	20.12	77.72	27.36	83.03	33.95	82.07	33.87	81.64	28.83	81.12
Vicuna-7B-v1.5	19.99	78.97	28.96	83.38	35.06	82.54	34.56	81.71	29.64	81.65
NLLB-3.3B	21.07	76.93	29.55	83.43	40.08	83.95	49.06	85.92	34.94	82.56
ALMA-7B-LoRA	24.00	80.18	29.98	84.16	38.43	84.80	43.96	86.00	34.09	83.79
ALMA-13B-LoRA	25.48	80.21	31.26	84.56	40.26	85.27	45.36	86.47	35.59	84.13
Closed										
text-davinci-003	25.00	81.62	30.88	84.79	38.47	84.80	44.52	86.16	34.72	84.34
GPT-4	23.80	82.46	32.46	85.35	40.98	85.87	46.77	87.26	36.00	85.24
Ladder-2B Refinement										
Alpaca-7B	22.73	78.98	28.53	83.34	36.05	83.34	37.08	83.08	31.10	82.19
	(+10.93)	(+5.62)	(+4.01)	(+1.97)	(+5.56)	(+2.66)	(+9.77)	(+5.09)	(+7.57)	(+3.84)
BigTranslate-13B	22.58	79.28	28.48	83.45	36.31	83.22	40.32	84.15	31.92	82.53
	(+8.26)	(+4.65)	(+5.31)	(+2.41)	(+8.26)	(+4.84)	(+5.83)	(+2.16)	(+6.91)	(+3.52)
BayLing-13B	23.84	79.55	29.05	83.64	36.92	83.69	38.85	83.59	32.17	82.61
	(+3.72)	(+1.83)	(+1.69)	(+0.61)	(+2.97)	(+1.62)	(+4.98)	(+1.95)	(+3.34)	(+1.49)
Vicuna-7B-v1.5	24.11	80.05	29.85	83.76	37.72	83.85	38.81	83.60	32.62	82.82
	(+4.12)	(+1.08)	(+0.89)	(+0.38)	(+2.66)	(+1.31)	(+4.25)	(+1.89)	(+2.98)	(+1.17)
NLLB-3.3B	23.97	79.34	29.83	83.89	39.02	84.27	45.10	85.30	34.48	83.20
	(+2.90)	(+2.41)	(+0.28)	(+0.46)	(-1.06)	(+0.32)	(-3.96)	(-0.62)	(-0.46)	(+0.64)
Ladder-7B Refinement										
BigTranslate-13B	26.49	81.08	31.13	84.58	39.22	85.25	45.87	86.43	35.68	84.34
	(+12.17)	(+6.45)	(+7.96)	(+3.54)	(+11.17)	(+6.87)	(+11.38)	(+4.44)	(+10.67)	(+4.83)
NLLB-3.3B	26.91	81.25	32.37	84.88	41.97	85.65	50.11	87.09	37.84	84.72
	(+5.84)	(+4.32)	(+2.82)	(+1.45)	(+1.89)	(+1.70)	(+1.05)	(+1.17)	(+2.90)	(+2.16)
ALMA-7B-LoRA	26.91	81.39	31.61	84.65	39.42	85.33	46.15	86.63	36.02	84.50
	(+2.91)	(+1.21)	(+1.63)	(+0.49)	(+0.99)	(+0.53)	(+2.19)	(+0.63)	(+1.93)	(+0.71)
ALMA-13B-LoRA	27.19	81.23	31.71	84.68	40.00	85.43	46.45	86.59	36.34	84.48
	(+1.71)	(+1.02)	(+0.45)	(+0.12)	(-0.26)	(+0.16)	(+1.09)	(+0.12)	(+0.75)	(+0.36)
text-davinci-003	27.10	81.67	31.61	84.67	39.51	85.52	46.71	86.73	36.23	84.65
	(+2.10)	(+0.05)	(+0.73)	(-0.12)	(+1.04)	(+0.72)	(+2.19)	(+0.57)	(+1.52)	(+0.31)
GPT-4	27.20	81.86	32.71	85.08	42.17	85.80	49.83	87.25	37.73	85.24
	(+3.40)	(-0.60)	(+0.25)	(-0.27)	(+1.19)	(-0.07)	(+3.06)	(-0.01)	(+1.98)	(-0.24)

Table 1: Performance of Ladder on WMT22 XX→En test set. The original translation using $\mathcal{P}_D$ prompt are at the top. The middle shows the Ladder-2B refined scores, and the bottom shows the Ladder-7B refined scores. Blue boxes indicate improved Ladder-refined scores, while Red boxes indicate decreased scores.

Ladder then refines i_{test} into the final translation y_{final} in an auto-regressive manner: $y_{final_t} \sim p_{\theta_{\mathcal{L}_a}}(y_{final_t} \mid s_{test}, i_{test}, \mathcal{P}_R, y_{final_{<t}})$. Notably, Ladder is model-agnostic, meaning $\mathcal{M}_T$ can be a translation model like ALMA (Xu et al., 2023a), or a general LLM like Alpaca (Taori et al., 2023).

3 Experiments

3.1 Experimental Setup

Datasets. For training, we choose Vicuna-7B-v1.5 (Chiang et al., 2023) as the sampling model. Vicuna-7B-v1.5, fine-tuned from LLaMA2 (Touvron et al., 2023), possesses a certain level of translation ability (see Tables 1 and 2). For parallel corpus, we collect test datasets from WMT'17 to WMT'20, along with Flores-200 (Costa-jussà et al., 2022), covering 8 translation directions (En ⇔ XX) and 5 languages: English (En), German (De), Czech (Cs), Chinese (Zh), and Russian (Ru). The trained Ladder is evaluated on the same translation directions using data from WMT22 [1]. Detailed statistics are in Table 5.

We evaluate Ladder under two scenarios. 1) We examine the effectiveness of Ladder to refine both translation-specific LLMs, such as BigTranslate (Yang et al., 2023), BayLing (Zhang et al., 2023b), NLLB (Costa-jussà et al., 2022), ALMA (Xu et al., 2023a), and general LLMs, such as Alpaca (Taori et al., 2023), Vicuna (Chiang et al., 2023), GPT-3.5-text-davinci-003 [2] (Ouyang et al., 2022), GPT-4 [3] (Achiam et al., 2023). 2) We compare Ladder to SoTA translation refinement or APE methods, i.e., LLMRefine (Xu et al., 2023b) and TowerInstruct (Alves et al., 2024). Details are in Appendix B.

Metrics. Following Xu et al. (2023a) and Alves et al. (2024), we use the lexical metric BLEU (Post,

[1] https://github.com/wmt-conference

[2] GPT-3.5 results are sourced from Xu et al. (2023a).

[3] GPT-4 results are sourced from Xu et al. (2024).

Models	En-Zh		En-De		En-Ru		En-Cs		Avg.	
	BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET
Open										
Alpaca-7B	7.85	51.79	18.22	78.22	14.10	74.87	13.13	73.51	13.33	69.60
Vicuna-7B-v1.5	31.42	82.68	22.65	80.82	19.60	81.07	16.37	77.25	22.51	80.46
BayLing-13B	37.93	84.63	25.62	82.70	12.77	71.01	16.43	78.22	23.19	79.14
BigTranslate-13B	29.89	81.83	22.99	80.54	19.52	81.56	22.68	84.50	23.77	82.11
NLLB-3.3B	32.53	81.57	33.97	86.24	30.11	87.51	36.30	89.90	33.23	86.31
ALMA-7B-LoRA	36.26	85.16	29.43	85.41	26.49	87.05	29.28	89.01	30.37	86.66
ALMA-13B-LoRA	39.87	85.96	31.49	85.62	29.03	87.53	32.47	89.79	33.22	87.23
Closed										
text-davinci-003	38.34	85.76	31.85	85.61	27.55	86.74	31.28	88.57	32.26	86.67
GPT-4	42.78	87.19	34.49	87.29	28.67	88.70	33.66	90.81	34.90	88.50
Ladder-2B Refinement										
Alpaca-7B	34.66	83.56	24.81	81.55	21.51	83.71	20.62	82.57	25.40	82.85
	(+26.81)	(+31.77)	(+6.59)	(+3.33)	(+7.41)	(+8.84)	(+7.49)	(+9.06)	(+12.07)	(+13.25)
Vicuna-7B-v1.5	36.47	84.62	25.73	81.86	22.59	83.84	21.51	83.19	26.58	83.38
	(+5.05)	(+1.94)	(+3.08)	(+1.04)	(+2.99)	(+2.77)	(+5.14)	(+5.94)	(+4.07)	(+2.92)
BayLing-13B	38.54	85.03	26.71	82.32	21.67	83.22	21.74	82.93	27.17	83.38
	(+0.61)	(+0.40)	(+1.09)	(-0.38)	(+8.90)	(+12.21)	(+5.31)	(+4.71)	(+3.98)	(+4.24)
BigTranslate-13B	37.65	84.74	26.82	82.62	23.04	84.03	24.39	84.82	27.98	84.05
	(+7.76)	(+2.91)	(+3.83)	(+2.08)	(+3.52)	(+2.47)	(+1.71)	(+0.32)	(+4.21)	(+1.94)
NLLB-3.3B	39.06	84.79	29.97	83.59	25.03	85.19	28.34	86.06	30.60	84.91
	(+6.53)	(+3.22)	(-3.97)	(-2.65)	(-5.08)	(-2.32)	(-7.96)	(-3.84)	(-2.63)	(-1.40)
Ladder-7B Refinement										
BigTranslate-13B	42.10	86.56	32.00	85.92	28.11	87.38	30.49	89.00	33.18	87.22
	(+12.21)	(+4.73)	(+9.01)	(+5.38)	(+8.59)	(+5.82)	(+7.81)	(+4.50)	(+9.41)	(+5.11)
NLLB-3.3B	43.40	86.65	33.33	86.34	29.55	87.71	33.74	89.37	35.01	87.52
	(+10.87)	(+5.08)	(-0.64)	(+0.10)	(-0.56)	(+0.20)	(-2.56)	(-0.53)	(+1.78)	(+1.21)
ALMA-7B-LoRA	42.17	86.73	32.33	86.20	28.58	87.65	30.90	89.30	33.50	87.47
	(+5.91)	(+1.57)	(+2.90)	(+0.79)	(+2.09)	(+0.60)	(+1.62)	(+0.29)	(+3.13)	(+0.81)
ALMA-13B-LoRA	42.72	86.83	32.54	85.93	29.04	87.65	31.70	89.43	34.00	87.46
	(+2.85)	(+0.87)	(+1.05)	(+0.31)	(+0.01)	(+0.12)	(-0.77)	(-0.36)	(+0.79)	(+0.24)
text-davinci-003	43.62	86.75	32.90	86.12	28.58	87.92	32.57	89.25	34.42	87.51
	(+5.28)	(+0.99)	(+1.05)	(+0.51)	(+1.03)	(+1.18)	(+1.29)	(+0.68)	(+2.16)	(+0.84)
GPT-4	44.35	87.02	33.81	86.55	29.32	88.15	32.65	89.69	35.03	87.85
	(+1.57)	(-0.17)	(-0.68)	(-0.74)	(+0.65)	(-0.55)	(-1.01)	(-1.12)	(+0.13)	(-0.65)

Table 2: Results of Ladder on WMT22 En→XX test set. Ladder-2B refines LLMs with higher parameter counts than itself. Ladder-7B refines all translators except for GPT-4. The color and marker are the same in Table 1.

Models	COMET			
	Zh-En	En-Zh	De-En	En-De
Palm2	74.70	-	-	81.80
+LLMRefine	75.90	-	-	82.30
BigTranslate-13B	74.63	81.83	81.04	80.54
+TowerInstruct-7B	76.17	85.62	82.03	84.89
+TowerInstruct-13B	77.92	85.91	82.26	85.86
+Ladder-2B	79.28	84.74	83.45	82.62
+Ladder-7B	**81.08**	**86.56**	**84.58**	**85.92**

Table 3: Comparison with baselines on WMT22 test set. Palm2 and LLMRefine results are from Xu et al. (2023b). **Bold font** and underline indicate the best and second best performance, respectively.

2018) and the reference-based metric COMET-22 (Rei et al., 2020) as the main metrics to evaluate the translation quality. We further employ the reference-free QE model COMETKiwi (Rei et al., 2022) to evaluate the overall translation quality.

Backbones. Ladder uses Gemma-2B and Gemma-7B[4] as the backbones, which are further fine-tuned using LoRA (Hu et al., 2021) with a rank of 16. We update 0.9% of the parameters for the 2B model and 0.6% for the 7B model.[5]

3.2 Main Results

Refinement Performance over LLMs. Table 1 and 2 show that Ladder can significantly improve the overall translation quality for all 8 translation directions across most translation-specific and general-purpose LLMs. Specifically, Ladder-2B improves Alpaca-7B by +12.07 BLEU and +13.25 COMET for En→XX on average, and refines BigTranslate-13B by +6.91 BLEU and +3.52 COMET for XX→En. As for Ladder-7B, it shows improvement over all open-source models on average. Notably, it even enhances 7 out of 8

[4] They utilize a vocabulary size of 256k tokens, ensuring effective applicability in multilingual scenarios.

[5] The training details are presented in Appendix C.

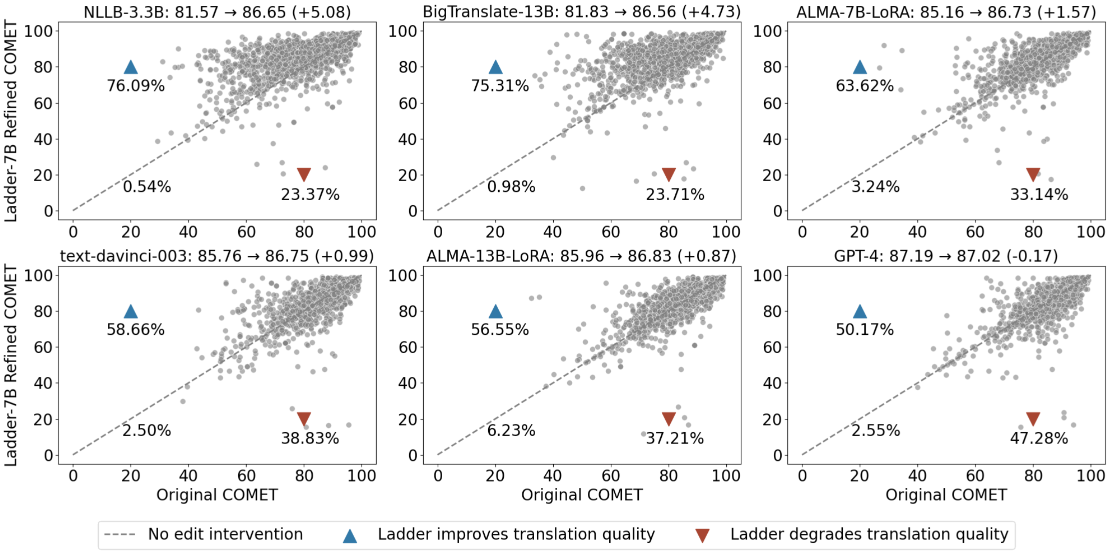

Figure 4: Comparison of original translation quality (x-axis) with Ladder-7B refined quality (y-axis). Each dot is a WMT22 En-Zh translation. The percentages represent the proportion of each part, attached next to the markers.

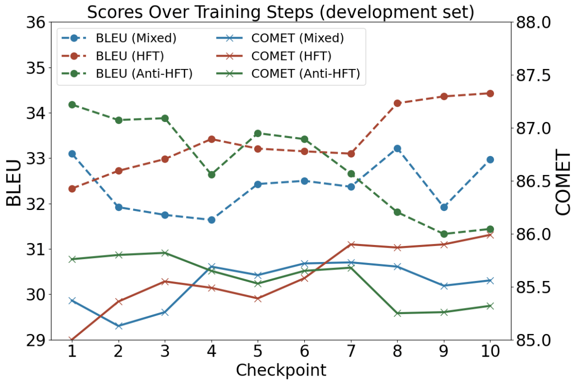

Figure 5: Trends in BLEU and COMET during training. HFT represents our hierarchical fine-tuning from *Easy* to *Hard* examples, while Mixed denotes using mixed data shuffling without hierarchical fine-tuning. Anti-HFT refers to reversing the HFT process.

translations for GPT-3.5-text-davinci-003 and improves +1.05 BLEU score for GPT-4 on average. We also find that while Ladder-2B shows inferior performance on the strong NLLB-3.3B, our Ladder-7B exhibits significant translation refinements on average. This aligns with our intuitions that different base models might exhibit varying levels of refinement performance across different LLMs, see detailed analysis in Figure 4.

Comparison with SoTA Baselines. We compare Ladder with two SoTA baselines on four translation directions from WMT22, as reported in Table 3. We can notice that Ladder-7B significantly outperforms all baselines in all four directions. Meanwhile, Ladder-2B exhibits performance on par with the best-performing TowerInstruct-13B baseline. We report the performance of LLMRefine on Palm2 as it is not available for BigTranslate, which is far inferior to Ladder.

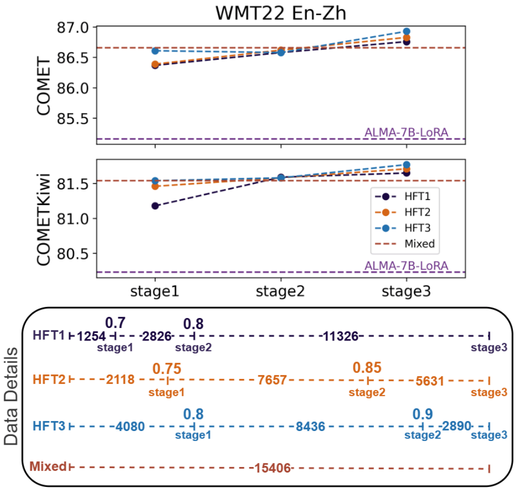

Figure 6: Robustness against threshold μ and ν. HFT1: $(\mu,\nu) = (0.7, 0.8)$, HFT2: $(\mu,\nu) = (0.75, 0.85)$, and HFT3: $(\mu,\nu) = (0.8, 0.9)$. Mixed denotes mixed training. ALMA-7B-LoRA is the model to refine.

3.3 Ablation and Analysis

Analysis of HFT. As depicted in Figure 5 [6], our HFT exhibits stable improvements and the best performance regarding BLEU and COMET in all ten checkpoints, while the traditional mixed training strategy fluctuates with inferior performance. We also conduct another "Anti-HFT" experiment by

[6] We examine the effectiveness of HFT with the Gemma-7B on the development set (see Appendix A), automatically saving 10 checkpoints to calculate metric scores.

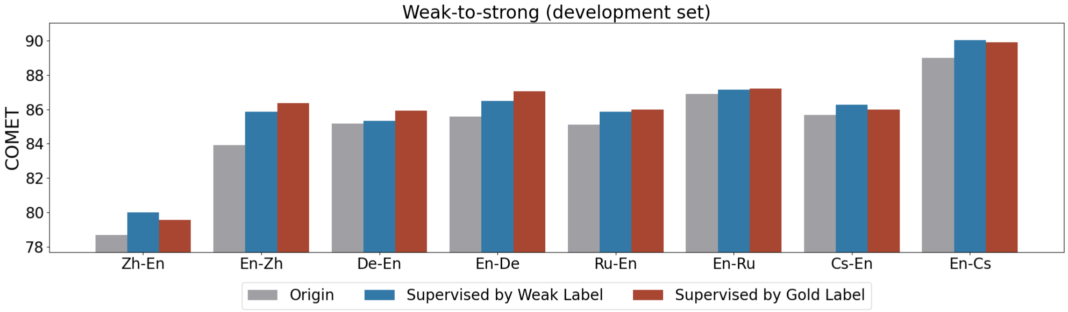

Figure 7: Weak-to-strong potential. We fine-tune Gemma-7B using different *references* as the labels to refine the development set. Origin denotes ALMA-7B-LoRA translation. Blue represents using ALMA-7B-LoRA as the *weak reference* to fine-tune Ladder. Red represents using the gold label as the *reference*.

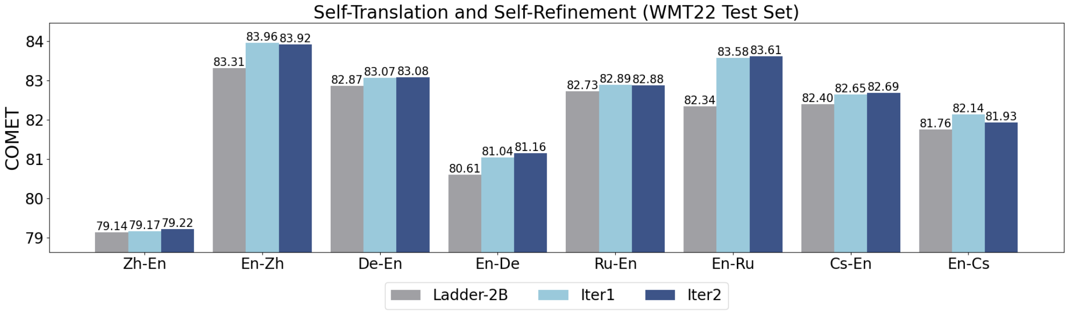

Figure 8: Self-translation and Self-refinement. Ladder-2B represents performing direct translation with prompt $\mathcal{P}_D$, demonstrating translation capabilities comparable to 7B and 13B LLM-based translators. Iter1 denotes Ladder-2B refining its original translation. Iter2 denotes Ladder-2B refining the translation from Iter1.

Ladder Pipeline			**WMT22 En-Zh**		
Sampling Model	**Base Model**	**Refine Model**	**BLEU**	**COMET**	**COMETKiwi**
Gemma-2B-it	Gemma-2B	Gemma-2B-it	35.46	84.41	79.55
		Gemma-7B-it	35.86	84.60	79.58
Vicuna-7B-v1.5	LLaMA-2-7B	Vicuna-7B-v1.5	34.31	84.12	79.41
		Vicuna-13B-v1.5	36.19	84.74	79.86
Baseline					
Gemma-2B-it			21.07	78.67	73.70
Gemma-7B-it			30.55	81.50	76.98
Vicuna-7B-v1.5			31.42	82.68	77.86
Vicuna-13B-v1.5			35.14	83.38	78.67

Table 4: Ablation of different sampling and backbones. Evaluate Gemma and LLaMA suite models on En-Zh.

reverting the order of the corpus employed during HFT, i.e., the Ladder is trained following a hard-to-easy schema. Results in Figure 5 shows that "Anti-HFT" initially achieves its best performance and then gradually declines.

We further scrutinize the model performance during HFT to verify its effectiveness. We report two metrics, the average improvement Δ and its standard deviation σ of the above three strategies during the training process, while larger Δ and smaller σ indicate better and more stable refinement improvements. The results are in Figure 9.

We notice that HFT results in a gradual increase of Δ and a decrease of σ. However, "Anti-HFT" shows the opposite trend, and the mixed training fluctuates in both Δ and σ. The increasing σ in "Anti-HFT" suggests that learning on *Easy* triplets might affect the stability of refinements. These results align with our hypothesis that refining *Hard* samples requires fewer adjustments, while *Easy* samples, which exhibit substantial deviations from the reference, demand more corrections and can cause significant fluctuations if utilized for fine-tuning in the final stage. See samples in Table 6 and 7 for intuitive understandings. Our findings suggest that the way triplet data is partitioned and ordered for HFT can impact model performance for instruction-following refinement, while more robust fine-tuning strategies are of high necessity in future work.

We also investigate the sensitivity of the threshold μ and ν used for splitting hierarchies and conduct HFT with three different thresholds on En-Zh training set, as shown in Figure 6. The results indicate that HFT consistently outperforms mixed training, with similar performance across different thresholds.

Refinements Degrade as the Original LLM Becomes Stronger. We analyze the quality score changes between the original translations and the Ladder-refined versions as shown in Figure 4. We observe that Ladder consistently improves a higher proportion of translations than it degrades, even for GPT-4. The trend in the proportion of improved translations aligns with the average score improvement trend. Specifically, as the model's translation capability increases, the proportion of improvements decreases, and the average improvement score also decreases. Our findings suggest that stronger translations have fewer and more complex errors that are harder to refine, consistent with our assumption in Section 2.3.

Ablation Study of Different Sampling and Backbones. As shown in Table 4, Ladder trained using different sampling and backbones consistently improves translation quality across instruction-tuning models of various sizes, demonstrating the effectiveness of our instruction-following refinement strategy. Notably, Gemma-2B (Vicuna-7B) with Ladder even surpasses Gemma-7B (Vicuna-13B), highlighting the potential to enhance the capabilities of smaller models to next level.

Instruction-following Refinement Enables Weak-to-Strong Generalization. Typically, the capabilities after fine-tuning are upper-bounded by the supervised label, i.e., the *reference* in our task. Here, we explore using ALMA-7B-LoRA sampled translation as the *weak reference* and Vicuna-7B sampled translation as the *intermediate translation* to create pseudo-refinement training triplets [*source*, *intermediate translation*, *weak reference*]. Figure 7 and 10 show that Ladder trained under this weak supervision can refine translations from the weak label annotator ALMA-7B-LoRA, surpassing it in both BLEU and COMET scores. Remarkably, it even outperforms gold label supervision in three translation directions. This demonstrates the potential of our instruction-following refinement method to exceed the current limits of supervision.

Ladder Can Act as a Good Translator and Execute Self-refinement. We evaluate the translation capability of Ladder and explore its self-refinement potential. Figure 8 shows that Ladder-2B can also execute the direct translation task and can improve its own initial translations across 8 translation directions, with increased COMET scores. However, the refinement effect becomes less pronounced with each iteration. More metrics are in Appendix D.

4 Related Work

Automatic Post-Edition and Refinement APE aims to cope with systematic errors of an MT system and adapt the output to the lexicon/style requested in a specific application domain. Correia and Martins (2019) proposed a BERT-based method for APE using transfer learning. Other studies (Negri et al., 2018; Vu and Haffari, 2018; Chatterjee, 2019; Shterionov et al., 2020; Voita et al., 2019; Góis et al., 2020; Chollampatt et al., 2020; do Carmo et al., 2020) investigated dataset construction, model architectures, and context integration to improve post-edited translations.

With the development of LLMs, learning-based approaches have trained LLMs for refining translations to improve the overall translation segment quality (Xu et al., 2023b; Alves et al., 2024; Koneru et al., 2023). Recent works (Chen et al., 2023; Raunak et al., 2023; Feng et al., 2024) have also explored using powerful LLMs, such as ChatGPT, to refine translations through prompting strategies like in-context learning and self-correction.

LLMs for Machine Translation LLM-based machine translation falls into two main categories. The first focuses on strategies like prompt design, in-context example selection, and evaluation in various contexts such as low-resource, document-level, and multilingual translation (Vilar et al., 2022; Zhang et al., 2023a; Peng et al., 2023; Wang et al., 2023; Liang et al., 2023; He et al., 2024a). The second category focuses on training translation-specific LLMs. Prior studies (Zeng et al., 2023; Jiao et al., 2023a; Kudugunta et al., 2024; Zan et al., 2024; Li et al., 2024; Guo et al., 2024; He et al., 2024b; Wu et al., 2024; Xu et al., 2024) have explored aspects such as dataset construction, training paradigms, and exploring different contexts to achieve better translation performance.

5 Conclusion

In this paper, we introduce Ladder, a model-agnostic and cost-effective tool for multilingual translation refinement that bridges the gap between off-the-shelf models and top-tier translation models. We sample translations from existing models to create pseudo-refinement training triplets without human annotations, which makes training cost-efficient. The proposed hierarchical fine-tuning strategy improves Ladder's refining performance step by step, following an easy-to-hard schema. Our exploration of training paradigms demonstrates good performance in effectiveness and robustness, as well as promising results in weak-to-strong generalization and self-refinement, providing valuable insights to the MT area.

Limitations

Although Ladder has shown promising results in bridging the gap between the translation performance of different models, it has some limitations. We have validated Ladder's support for sentence-level translations, but document-level support still needs exploration. Expanding Ladder's usage to support more languages, especially low-resource languages, is also crucial for future work. Additionally, deploying this approach to larger models (e.g., 70B) or smaller models (e.g., less than 1B) is worth exploring in future research. Leveraging the principles of Ladder to explore instruction-following refinement in more generation tasks is also an interesting direction for future work.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Sweta Agrawal, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. 2023. In-context examples selection for machine translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8857–8873, Toronto, Canada. Association for Computational Linguistics.

Duarte M Alves, José Pombal, Nuno M Guerreiro, Pedro H Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, et al. 2024. Tower: An open multilingual large language model for translation-related tasks. *arXiv preprint arXiv:2402.17733*.

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*.

Viraat Aryabumi, John Dang, Dwarak Talupuru, Saurabh Dash, David Cairuz, Hangyu Lin, Bharat Venkitesh, Madeline Smith, Kelly Marchisio, Sebastian Ruder, et al. 2024. Aya 23: Open weight releases to further multilingual progress. *arXiv preprint arXiv:2405.15032*.

Rajen Chatterjee. 2019. Automatic post-editing for machine translation. *arXiv preprint arXiv:1910.08592*.

Pinzhen Chen, Zhicheng Guo, Barry Haddow, and Kenneth Heafield. 2023. Iterative translation refinement with large language models. *arXiv preprint arXiv:2306.03856*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Shamil Chollampatt, Raymond Susanto, Liling Tan, and Ewa Szymanska. 2020. Can automatic post-editing improve nmt? In *Proceedings of EMNLP*.

Gonçalo M. Correia and André F. T. Martins. 2019. A Simple and Effective Approach to Automatic Post-Editing with Transfer Learning. In *Proceedings of ACL*.

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*.

Félix do Carmo, D. Shterionov, Joss Moorkens, Joachim Wagner, Murhaf Hossari, Eric Paquin, Dag Schmidtke, Declan Groves, and Andy Way. 2020. A review of the state-of-the-art in automatic post-editing. *Machine Translation*, 35:101 – 143.

Zhaopeng Feng, Yan Zhang, Hao Li, Wenqiang Liu, Jun Lang, Yang Feng, Jian Wu, and Zuozhu Liu. 2024. Improving llm-based machine translation with systematic self-correction. *Preprint*, arXiv:2402.16379.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Xavier Garcia, Yamini Bansal, Colin Cherry, George Foster, Maxim Krikun, Melvin Johnson, and Orhan Firat. 2023. The unreasonable effectiveness of few-shot learning for machine translation. In *Proceedings of the 40th International Conference on Machine Learning*, pages 10867–10878. PMLR.

António Góis, Kyunghyun Cho, and André Martins. 2020. Learning non-monotonic automatic post-editing of translations from human orderings. *arXiv preprint arXiv:2004.14120*.

Jiaxin Guo, Hao Yang, Zongyao Li, Daimeng Wei, Hengchao Shang, and Xiaoyu Chen. 2024. A novel paradigm boosting translation capabilities of large language models. *arXiv preprint arXiv:2403.11430*.

Zhiwei He, Tian Liang, Wenxiang Jiao, Zhuosheng Zhang, Yujiu Yang, Rui Wang, Zhaopeng Tu, Shuming Shi, and Xing Wang. 2024a. Exploring human-like translation strategy with large language models. *Transactions of the Association for Computational Linguistics*, 12:229–246.

Zhiwei He, Xing Wang, Wenxiang Jiao, Zhuosheng Zhang, Rui Wang, Shuming Shi, and Zhaopeng Tu. 2024b. Improving machine translation with human feedback: An exploration of quality estimation as a reward model. *arXiv preprint arXiv:2401.12873*.

Amr Hendy, Mohamed Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Hassan Awadalla. 2023. How good are gpt models at machine translation? a comprehensive evaluation. *arXiv preprint arXiv:2302.09210*.

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*.

Wenxiang Jiao, Jen-tse Huang, Wenxuan Wang, Zhiwei He, Tian Liang, Xing Wang, Shuming Shi, and Zhaopeng Tu. 2023a. ParroT: Translating during chat using large language models tuned with human translation and feedback. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15009–15020, Singapore. Association for Computational Linguistics.

Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Xing Wang, and Zhaopeng Tu. 2023b. Is chatgpt a good translator? a preliminary study. *arXiv preprint arXiv:2301.08745*, 1(10).

Sai Koneru, Miriam Exel, Matthias Huck, and Jan Niehues. 2023. Contextual refinement of translations: Large language models for sentence and document-level post-editing. *arXiv preprint arXiv:2310.14855*.

Sneha Kudugunta, Isaac Caswell, Biao Zhang, Xavier Garcia, Derrick Xin, Aditya Kusupati, Romi Stella, Ankur Bapna, and Orhan Firat. 2024. Madlad-400: A multilingual and document-level large audited dataset. *Advances in Neural Information Processing Systems*, 36.

Jiahuan Li, Hao Zhou, Shujian Huang, Shanbo Cheng, and Jiajun Chen. 2024. Eliciting the translation ability of large language models via multilingual finetuning with translation instructions. *Transactions of the Association for Computational Linguistics*, 12:576–592.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. 2023. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*.

Yasmin Moslem, Rejwanul Haque, John D. Kelleher, and Andy Way. 2023. Adaptive machine translation with large language models. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 227–237.

Matteo Negri, Marco Turchi, Rajen Chatterjee, and Nicola Bertoldi. 2018. Escape: a large-scale synthetic corpus for automatic post-editing. *arXiv preprint arXiv:1803.07274*.

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.

Keqin Peng, Liang Ding, Qihuang Zhong, Li Shen, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. 2023. Towards making the most of ChatGPT for machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5622–5633, Singapore. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 3505–3506, New York, NY, USA. Association for Computing Machinery.

Vikas Raunak, Amr Sharaf, Hany Hassan Awadallah, and Arul Menezes. 2023. Leveraging gpt-4 for automatic translation post-editing. *arXiv preprint arXiv:2305.14878*.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. Comet: A neural framework for mt evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702.

Ricardo Rei, Marcos Treviso, Nuno M Guerreiro, Chrysoula Zerva, Ana C Farinha, Christine Maroti, José GC De Souza, Taisiya Glushkova, Duarte Alves, Luísa Coheur, et al. 2022. Cometkiwi: Ist-unbabel 2022 submission for the quality estimation shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 634–645.

Dimitar Shterionov, Félix do Carmo, Joss Moorkens, Murhaf Hossari, Joachim Wagner, Eric Paquin, Dag Schmidtke, Declan Groves, and Andy Way. 2020. A roadmap to neural automatic post-editing: an empirical approach. *Machine Translation*, 34(2–3):67–96.

Michel Simard, Nicola Ueffing, Pierre Isabelle, and Roland Kuhn. 2007. Rule-based translation with statistical phrase-based post-editing. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 203–206, Prague, Czech Republic. Association for Computational Linguistics.

Shivalika Singh, Freddie Vargus, Daniel Dsouza, Börje F Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividas Mataciunas, Laura OMahony, et al. 2024. Aya dataset: An open-access collection for multilingual instruction tuning. *arXiv preprint arXiv:2402.06619*.

Lucia Specia, Kim Harris, Frédéric Blain, Aljoscha Burchardt, Viviven Macketanz, Inguna Skadin, Matteo Negri, and Marco Turchi. 2017. Translation quality and productivity: A study on rich morphology languages. In *Proceedings of Machine Translation Summit XVI: Research Track*, pages 55–71, Nagoya Japan.

Lucia Specia, Dhwaj Raj, and Marco Turchi. 2010. Machine translation evaluation versus quality estimation. *Machine Translation*, 24(1):39–50.

Pedro Javier Ortiz Su'arez, Benoit Sagot, and Laurent Romary. 2019. Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures. Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019, pages 9 – 16, Mannheim. Leibniz-Institut für Deutsche Sprache.

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

Hugo Touvron, 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*.

David Vilar, Markus Freitag, Colin Cherry, Jiaming Luo, Viresh Ratnakar, and George Foster. 2022. Prompting palm for translation: Assessing strategies and performance. *arXiv preprint arXiv:2211.09102*.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. Context-aware monolingual repair for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 877–886, Hong Kong, China. Association for Computational Linguistics.

Thuy-Trang Vu and Gholamreza Haffari. 2018. Automatic post-editing of machine translation: A neural programmer-interpreter approach. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3048–3053, Brussels, Belgium. Association for Computational Linguistics.

Longyue Wang, Chenyang Lyu, Tianbo Ji, Zhirui Zhang, Dian Yu, Shuming Shi, and Zhaopeng Tu. 2023. Document-level machine translation with large language models. *arXiv preprint arXiv:2304.02210*.

Minghao Wu, Thuy-Trang Vu, Lizhen Qu, George Foster, and Gholamreza Haffari. 2024. Adapting large language models for document-level machine translation. *arXiv preprint arXiv:2401.06468*.

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

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation. *Preprint*, arXiv:2401.08417.

Wenda Xu, Daniel Deutsch, Mara Finkelstein, Juraj Juraska, Biao Zhang, Zhongtao Liu, William Yang Wang, Lei Li, and Markus Freitag. 2023b. Pinpoint, not criticize: Refining large language models via fine-grained actionable feedback. *arXiv preprint arXiv:2311.09336*.

Wen Yang, Chong Li, Jiajun Zhang, and Chengqing Zong. 2023. Bigtranslate: Augmenting large language models with multilingual translation capability over 100 languages. *arXiv preprint arXiv:2305.18098*.

Changtong Zan, Liang Ding, Li Shen, Yibing Zhen, Weifeng Liu, and Dacheng Tao. 2024. Building accurate translation-tailored llms with language aware instruction tuning. *arXiv preprint arXiv:2403.14399*.

Jiali Zeng, Fandong Meng, Yongjing Yin, and Jie Zhou. 2023. Tim: Teaching large language models to translate with comparison. *arXiv preprint arXiv:2307.04408*.

Biao Zhang, Barry Haddow, and Alexandra Birch. 2023a. Prompting large language model for machine translation: A case study. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23.

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

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Lingpeng Kong, Jiajun Chen, Lei Li, and Shujian Huang. 2023. Multilingual machine translation with large language models: Empirical results and analysis. *arXiv preprint arXiv:2304.04675*.

A Dataset Statistics

Table 5 presents statistic details of the data we used. For the development set, we randomly sampled 100 examples from the development parallel data and used ALMA-7B-LoRA to generate intermediate translations, totaling 800 development triplets.

B Baseline Models

Translation Models

- BigTranslate (Yang et al., 2023) extends LLaMA to over 100 translation directions.
- BayLing (Zhang et al., 2023b) is an instruction-following large language model equipped with advanced language alignment.
- NLLB (Costa-jussà et al., 2022) is a translation model with encoder-decoder architecture.
- ALMA (Xu et al., 2023a) is a many-to-many LLM-based translation model. It represents the top level of open-source translators.

Non-translation Models

- Alpaca (Taori et al., 2023) is a LLaMA Model fine-tuned on 52K instruction-following data.
- Vicuna-v1.5 (Chiang et al., 2023) is fine-tuned from LLaMA2 with supervised instruction fine-tuning. The training data is around 125K conversations collected from ShareGPT [7].
- text-davinci-003 is a GPT-3.5 model with 175B parameters (Ouyang et al., 2022).
- GPT-4 (Achiam et al., 2023) is the latest and the most powerful version of GPT-series. We use OpenAI API gpt-4-1106-preview.

SoTA APE Models

- LLMRefine (Xu et al., 2023b) is a PaLM2 (Bison) fine-tuned LLM to refine LLM's output with fine-grained actionable feedback iteratively.
- TowerInstruct (Alves et al., 2024) is an effective translation post editor. It is fine-tuned on high-quality parallel translation data totaling 637k examples. The APE-related tasks include MQM evaluation data (WMT20 to WMT22) annotated with multidimensional quality metrics (Freitag et al., 2021), accounting for 20.9%. Translation data with post-edits from QT21 (Specia et al., 2017) and Ape-Quest [8] are used for general translation and automatic post-editing, making up 3.1% and 3.3% of the data, respectively. TowerInstruct outperforms open models and GPT-3.5-turbo on APE.

C Training Details

We fine-tune our model using LoRA with a rank of 16 and a learning rate of 1e-4. All models are fine-tuned for 1 epoch with a batch size of 16, imposing a maximum text length of 512. We adopt deepspeed (Rasley et al., 2020) to accelerate our training.

D Self-translation and Self-refinement

For Section 3.3, we supplement the BLEU and COMETKiwi of Ladder-2B (see Figure 11 and 12) and all metrics of Ladder-7B (see Figure 13, 14 and 15).

[7] https://sharegpt.com

[8] https://apequest.wordpress.com/

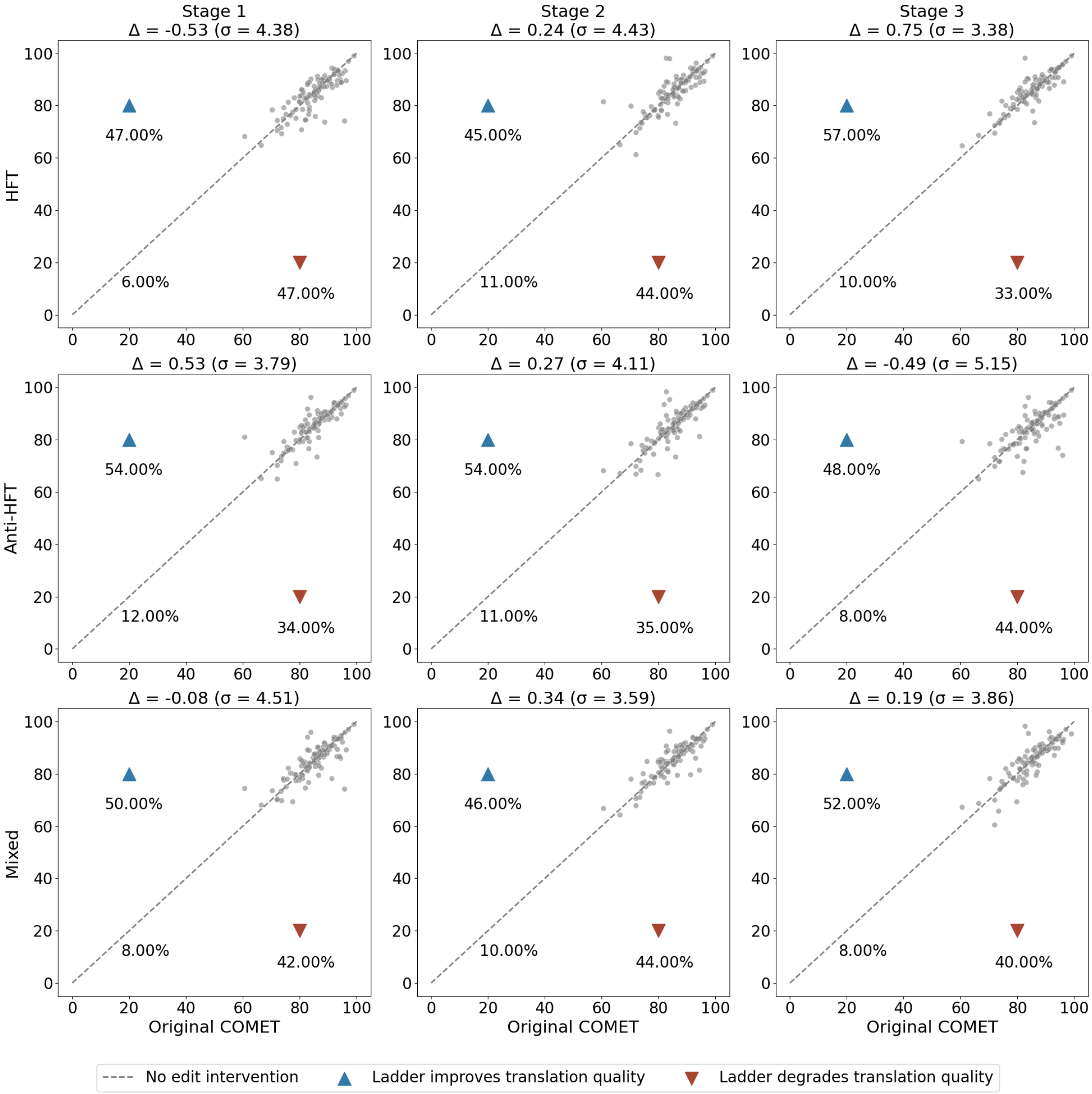

Figure 9: Comparison of original translation quality (x-axis) with refined quality (y-axis) in different fine-tuning stages. Each dot is a WMT22 De-En translation in our development set. We select the checkpoint at 2, 6, and 10 from Figure 5 (which we refer to as Stage 1, Stage 2 and Stage 3 here). Δ denotes the average improvement. σ refers to the standard deviation of Δ. The percentages represent the proportion of each part, attached next to the markers.

Language	Parallel Data			
	Train	Development	Test (from English)	Test (to English)
Chinese (Zh)	15406	1002	2037	1875
German (De	14211	1002	2037	1984
Russia (Ru)	15000	1002	2037	2016
Czech (Cs)	12076	1002	2037	1448

Table 5: The statistics for the parallel data we used.

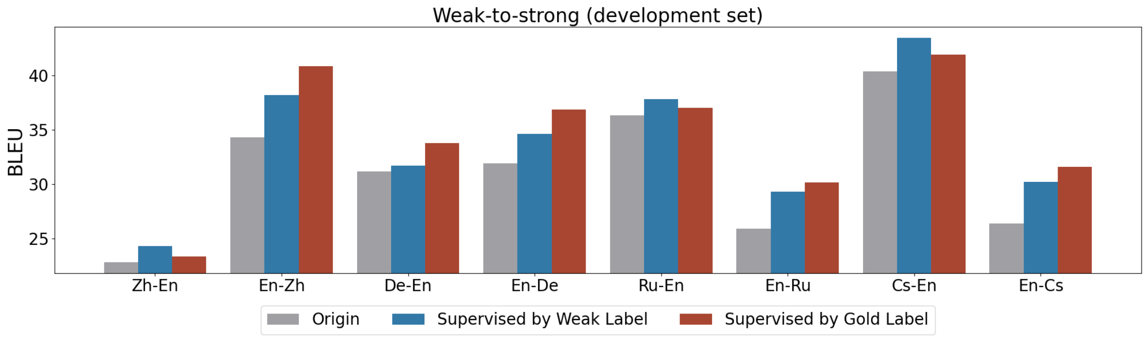

Figure 10: Weak-to-strong BLEU scores. We fine-tune Gemma-7B using different *references* as the label to refine the development set. Origin denotes ALMA-7B-LoRA translation. Blue represents using ALMA-7B-LoRA as *references*. Red represents using the gold as *references*.

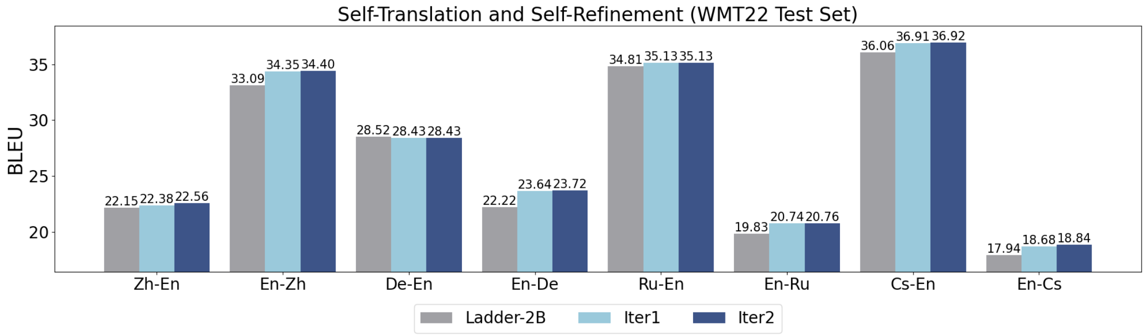

Figure 11: BLEU scores for Self-translation and Self-refinement. Iter1 denotes Ladder-2B refines its original translation. Iter2 denotes Ladder-2B refines the Ladder-2B edited translation in Iter1.

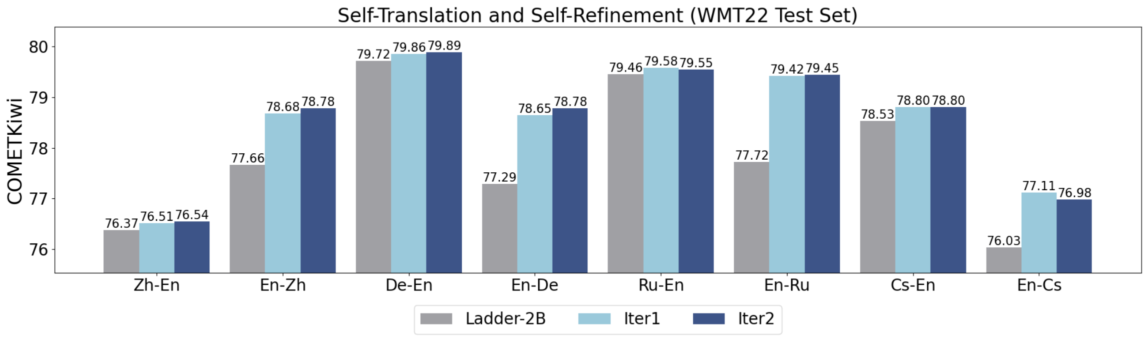

Figure 12: COMETKiwi scores for Self-translation and Self-refinement. Iter1 denotes Ladder-2B refines its original translation. Iter2 denotes Ladder-2B refines the Ladder-2B edited translation in Iter1.

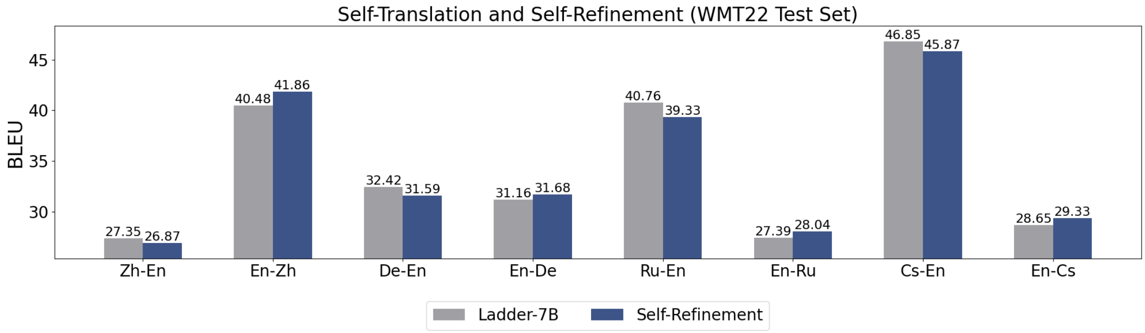

Figure 13: BLEU scores for Self-translation and Self-refinement with Ladder-7B. Self-Refinement denotes Ladder-7B refines its original translation.

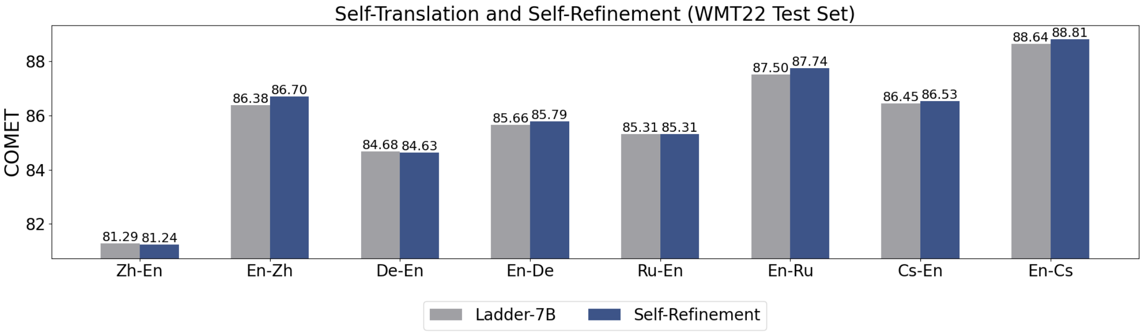

Figure 14: COMET scores for Self-translation and Self-refinement with Ladder-7B. Self-Refinement denotes Ladder-7B refines its original translation.

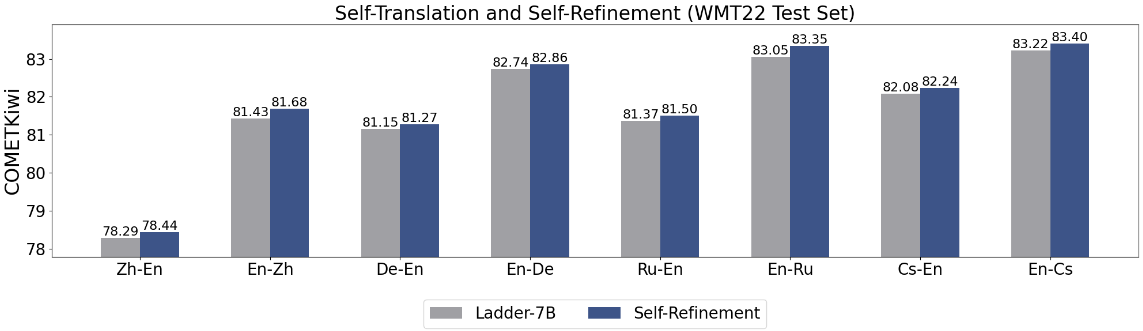

Figure 15: COMETKiwi scores for Self-translation and Self-refinement with Ladder-7B. Self-Refinement denotes Ladder-7B post-edits its original translation.

Anti-HFT Case		COMET
German Source	So jedenfalls macht die grandiose F1-Saison wesentlich weniger Spaß als es mit einem vernünftigen Sender möglich wäre.	-
English Reference	At any rate, it really makes the grand F1 season considerably less fun as would be the case with a reasonable broadcaster.	-
Intermediate Translation	So, in any case, the grandiose F1 season is much less fun than it would be with a reasonable broadcaster.	-
Anti-HFT Stage1 (Hard)	So, at any rate, the grandiose F1 season is much less fun than it would be with a reasonable broadcaster.	87.55
Anti-HFT Stage2 (Hard+Medium)	So, at least, the grandiose F1 season is much less fun than it would be with a reasonable broadcaster.	83.32
Anti-HFT Stage3 (Hard+Medium+Easy)	So the great F1 season is much less fun than it would be with a decent broadcaster.	81.57
HFT Cases		**COMET**
German Source	Es ist schade, dass wir den Flow nicht mitnehmen konnten.	-
English Reference	It is a shame that we were not able to get into the flow.	-
Intermediate Translation	It is a shame that we couldn't take the flow with us.	-
HFT Stage1 (Easy)	It's a shame we couldn't keep the momentum going.	79.54
HFT Stage2 (Easy+Medium)	It's a shame that we couldn't take the flow with us.	81.18
HFT Stage3 (Easy+Medium+Hard)	It's a shame that we couldn't keep the flow going.	84.10

Table 6: Case study. Stage corresponds to Figure 9.

	COMET:69.73
Chinese Source	但八年前濒临倒闭，不得不接受救助从那时开始便放弃了那样的追求。
Intermediate Translation	But eight years ago, it was on the verge of bankruptcy and had to accept help. From that time on, I gave up such pursuits.
English Reference	It has retreated from them since it nearly collapsed eight years ago and had to be bailed out.
	COMET:83.37
English Source	Representatives of junior doctors have called on their union to authorise fresh industrial action in their dispute about a new contract.
Intermediate Translation	低级医生代表呼吁他们的工会授权新的工业行动，因为他们对新合同的争议仍未得到解决。
Chinese Reference	初级医生代表号召联盟批准其针对新合同纠纷采取新的劳工行动。
	COMET:91.84
German Source	Ich hätte mich gefreut, wenn Mesut Özil weiter für Deutschland gespielt hätte.
Intermediate Translation	I would have been delighted if Mesut Özil had continued to play for Germany.
English Reference	I would be happy if Mesut Özil continued to play for Germany.

Table 7: Cases of triples with different COMET scores.