Optimizing Chinese-to-English Translation Using Large Language Models

Donghao HUANG[1,2], Zhaoxia WANG[1]
[1]School of Computing and Information Systems, Singapore Management University, Singapore
[2]Research and Development, Mastercard, Singapore
dh.huang.2023@smu.edu.sg; zxwang@smu.edu.sg

***Abstract*—The advent of Large Language Models (LLMs) has significantly advanced Chinese-to-English translation tasks. However, the translation process remains challenging due to the substantial differences in syntax, semantics, and morphology between these two languages, despite notable achievements. This paper presents a comprehensive study on Chinese-to-English translation, evaluating the performance of various LLMs. We explore a range of open-source models, from 3.5 billion to 72 billion parameters, and OpenAI's latest models, across zero-shot, few-shot, and fine-tuned learning paradigms. Our analysis assesses translation quality using the COMET metric, reliability with the Translation Completeness Ratio (TCR), and efficiency via Characters per Second (CPS). The results highlight substantial trade-offs between model size, translation accuracy, and processing speed. Larger models tend to produce higher-quality translations, whereas smaller models offer greater efficiency. Fine-tuning significantly improves the performance of open-source LLMs, surpassing few-shot learning in both translation quality and processing speed. Proprietary models like GPT-4o exhibit consistent high performance without significant gains from fine-tuning. We emphasize the potential of fine-tuning with techniques like LoRA/QLoRA to optimize the balance between translation accuracy and computational efficiency, offering valuable insights for deploying LLMs in real-world translation scenarios.**

***Index Terms*—Machine Translation, Large Language Models, Natural Language Processing, Chinese-English Translation, COMET Metric, Fine-tuning, Efficiency Analysis, Zero-shot Learning, Few-shot Learning**

I. Introduction

Machine Translation (MT) has been a pivotal area of research in Natural Language Processing (NLP), aiming to bridge communication gaps between speakers of different languages. The translation of Chinese to English poses particular challenges due to the significant differences in syntax, semantics, and morphology between these two languages. Traditional approaches, such as Rule-Based and Statistical Machine Translation (SMT), have relied on handcrafted rules or statistical patterns derived from parallel corpora [1]. However, these methods often struggle with complexities inherent in Chinese-English translation, particularly in handling idiomatic expressions, word order differences, and character-based text segmentation.

The development of Neural Machine Translation (NMT) marked a significant shift, enabling the automatic learning of translation patterns from data [2]. The introduction of the Transformer architecture [3] further advanced this field by leveraging self-attention mechanisms, leading to significant improvements in translation quality and efficiency. More recently, Large Language Models (LLMs), such as GPT-3 [4] and mBERT [5], have demonstrated substantial potential in MT tasks, including zero-shot and few-shot translation capabilities.

This paper explores the application of state-of-the-art LLMs to Chinese-to-English translation, evaluating open-source models (3.5B–72B parameters) and OpenAI's proprietary GPT-4o and GPT-4o-mini. This diverse selection highlights the capabilities of both open-source and cutting-edge proprietary models in machine translation.

Our work makes the following key contributions:

1) **Comprehensive Evaluation:** We benchmark various LLMs (3.5B–72B parameters), comparing proprietary models (e.g., GPT-4o) and open-source ones (e.g., Llama, Qwen) for Chinese-to-English translation tasks.
2) **Multi-metric Analysis:** We assess translation quality using COMET, reliability through the Translation Completeness Ratio (TCR), and efficiency using Characters per Second (CPS).
3) **Learning Paradigm Comparison:** We examine zero-shot, few-shot, and fine-tuned performance, showing that fine-tuning significantly improves both quality and speed, especially for open-source models.
4) **Fine-tuning Insights:** We provide a detailed analysis of the fine-tuning process, including performance improvements and optimal epochs for different models.
5) **Efficiency-Quality Trade-off:** Our study explores the balance between translation quality and processing speed, highlighting the benefits of techniques like LoRA and QLoRA for efficient LLM applications.
6) **Proprietary Model Observations:** We offer insights into proprietary models such as GPT-4o, which exhibit distinct behaviors in fine-tuning compared to open-source models.
7) **Data and Code Transparency**: To promote reproducibility, all data and code associated with this study are made publicly available on GitHub: https://github.com/inflaton-ai/translation.

II. Related Work

A. Evolution of Machine Translation

The field of Machine Translation (MT) has seen significant evolution, from Statistical Machine Translation (SMT) [1] to Neural Machine Translation (NMT) [2]. The introduction of the Transformer architecture [3] marked a pivotal moment, setting new benchmarks in translation quality and efficiency. This progress has been particularly impactful for challenging language pairs like Chinese-English, where linguistic differences pose substantial hurdles [6].

B. Large Language Models in Machine Translation

Recent years have witnessed the rise of Large Language Models (LLMs) in MT tasks. Models like BERT [5], GPT-3 [4], and T5 [7] have demonstrated impressive capabilities in zero-shot and few-shot translation settings [8]. For Chinese-to-English translation specifically, models such as mBERT and XLM-R have leveraged cross-lingual representations to enhance translation quality [9].

The application of LLMs to MT has opened new avenues for research:

- **Zero-shot and Few-shot Learning:** Studies have explored the ability of LLMs to perform translations with minimal or no task-specific training [4].
- **Prompting Techniques:** Research has investigated various prompting strategies to optimize LLM performance in translation tasks [10].
- **Fine-tuning Approaches:** Recent work has focused on efficient fine-tuning methods for adapting LLMs to specific translation tasks [11].

C. Efficient Fine-tuning Techniques

As LLMs grow in size, efficient fine-tuning becomes crucial. Two notable techniques have emerged:

- **Low-Rank Adaptation (LoRA):** Introduced by Hu et al. [12], LoRA significantly reduces the number of trainable parameters while maintaining model performance. It achieves this by adding low-rank decomposition matrices to the model's weight matrices.
- **Quantized LoRA (QLoRA):** Dettmers et al. [13] extended LoRA by incorporating 4-bit quantization, further reducing memory requirements. This enables fine-tuning of very large models (70B+ parameters) on consumer-grade hardware.

These techniques have made it feasible to adapt large models to specific tasks efficiently, opening up new possibilities for specialized MT applications.

D. Evaluation Metrics in Machine Translation and LLMs

The evaluation of MT systems has also evolved, moving beyond traditional metrics like BLEU [14].

- **COMET:** The Crosslingual Optimized Metric for Evaluation of Translation [15] utilizes contextual embeddings from pretrained language models to assess translation quality, demonstrating a strong correlation with human evaluations.
- **Efficiency Metrics:** Recent studies have increasingly emphasized the importance of efficiency metrics, such as inference speed, alongside traditional quality metrics when evaluating LLMs [16], [17].

These efforts aim to enhance MT performance, making high-quality translation more accessible and efficient.

III. Dataset

To evaluate the effectiveness of large language models in Chinese-to-English translation, we utilize the MAC (Manually Aligned Chinese-English) dataset, which is publicly available on GitHub[1]. The MAC dataset is a carefully curated collection of parallel Mandarin-English texts, designed to support a range of NLP tasks, including machine translation, text classification, and linguistic analysis.

The dataset comprises sentences from six Chinese novels and their corresponding English translations, spanning a diverse set of genres such as humor, martial arts, classics, war, romance, and science fiction. To ensure representativeness, sentences are sampled from various sections of the novels.

For our experiments, we preprocess the dataset by first concatenating the Development (Dev) and Test sets into a single TSV file. We then prune entries that lack either Chinese or English paragraphs to ensure completeness. Subsequently, the dataset is split into training (80%) and testing (20%) sets. This preprocessing results in a total of 5,661 entries, with 4,528 entries in the training set (mac-train.tsv) and 1,133 entries in the testing set (mac-test.tsv).

This preprocessing ensures that the dataset is of high quality and reliability, making it suitable for robust training and evaluation in our experiments. By leveraging a diverse and representative sample of texts, we aim to comprehensively assess the performance of large language models in the challenging task of Chinese-to-English translation.

IV. Methodology

A. Selection of Large Language Models (LLMs)

We selected a diverse set of state-of-the-art large language models (LLMs) for our experiments. This selection encompasses both open-source models with parameter counts ranging from 3.5 billion to 72 billion and the latest offerings from OpenAI, such as GPT-4o. For open-source models that do not natively support the Chinese language, we employed their fine-tuned Chinese versions. Table I presents an overview of all the models utilized in our study, including their names, sizes (where applicable), and corresponding HuggingFace model identifiers for the open-source models.

B. Few-shot Learning and Parameter-Efficient Fine-tuning

To enable the LLMs to perform translation tasks effectively, we employed few-shot prompting techniques. The system prompt used for translation is shown in Listing 1.

[1] https://github.com/bfsujason/mac

TABLE I: Overview of Large Language Models and Their Specifications

Company	Model Name	Size	HuggingFace Model ID
OpenAI	GPT-4o	N/A	N/A
	GPT-4o-mini	N/A	N/A
Meta	Llama-3.1-8B	8B	shenzhi-wang/Llama3.1-8B-Chinese-Chat
	Llama-3.1-70B	70B	shenzhi-wang/Llama3.1-70B-Chinese-Chat
Alibaba	Qwen2-7B	7B	Qwen/Qwen2-7B-Instruct
	Qwen2-72B	72B	Qwen/Qwen2-72B-Instruct
Shanghai AI Laboratory	InternLM2.5-7B	7B	internlm/internlm2_5-7b-chat
Mistral AI	Mistral-7B	7B	shenzhi-wang/Mistral-7B-v0.3-Chinese-Chat
Microsoft	Phi-3.5-mini	3.5B	microsoft/Phi-3.5-mini-instruct

Listing 1: System Prompt for LLM Translation

```
You are a helpful assistant that translates Chinese
to English.
```

The user prompt template for translation is provided in Listing 2.

Listing 2: User Prompt Template for LLM Translation

```
You will be given a Chinese sentence to translate.
If it is an incomplete sentence, or if you are
unsure about the meaning, simply copy the input text
 as your output. Do not output any additional
sentence such as explanation or reasoning.
{exemplars}
Chinese: {input}
English:
```

The prompt template is designed to be a versatile tool for text translation. In our study, the placeholders were populated with specific data:

- {exemplars}: For zero-shot translation, this field is left empty. For few-shot translation, exemplars are retrieved from the MAC training set and formatted as shown below:

  ```
  Example Translations:
  Chinese: 全仗着狐仙搭救。
  English: Because I was protected by a fox fairy.
  ... (additional examples omitted for brevity)
  ```

- {input}: Original Chinese text for translation.

To enhance translation performance, we fine-tuned all models using the training set for five epochs. The details of the fine-tuning process are as follows:

- **OpenAI's GPT-4o and GPT-4o-mini**: Fine-tuning was performed following OpenAI's official guidelines[2].
- **Llama-3.1-70B and Qwen2-72B variants**: Employed 4-bit quantization using LoRA (QLoRA) with the open-source library Llama Factory [18] on 4 Nvidia GPUs.
- **Other open-source models**: Used LoRA with the open-source library Llama Factory [18] on a single Nvidia GPU.

Utilizing these prompting techniques and fine-tuned adapters, we evaluated the performance of GPT-4o and GPT-4o-mini models via the OpenAI API, while the open-source models were evaluated on Nvidia L40 GPUs, , each equipped with 48GB of memory, using the open-source HuggingFace Transformers library [19].

C. Evaluation Metrics

To assess the performance of each model, we employed multiple evaluation metrics to capture various aspects of translation quality and efficiency.

[2]https://platform.openai.com/docs/guides/fine-tuning

1) COMET-22: We employed the Crosslingual Optimized Metric for Evaluation of Translation (COMET), a neural-based evaluation metric that utilizes contextual embeddings from pretrained language models to assess translation quality. COMET has demonstrated a strong correlation with human judgments, establishing itself as a reliable metric for translation evaluation. The latest version, COMET-22, has been shown to be particularly effective in evaluating Chinese-to-English translations, outperforming traditional metrics such as BLEU and TER in terms of alignment with human evaluations [15], [20]. By using COMET-22, we aim to measure the semantic and contextual accuracy of translations produced by different models more objectively.

2) Translation Completeness Ratio (TCR): During our evaluation, we observed that some outputs generated by the large language models (LLMs) contained untranslated segments with Chinese characters, resulting in incomplete translations. To quantify this issue, we introduced the Translation Completeness Ratio (TCR), which measures the proportion of translations that are fully rendered into the target language (English):

$$\text{TCR} = \frac{\text{Number of Complete Translations}}{\text{Total Number of Entries}} \tag{1}$$

The TCR provides a metric for assessing the reliability of each model in consistently producing complete translations. A higher TCR value indicates fewer instances of untranslated content, reflecting better performance in maintaining translation completeness.

3) Characters per Second (CPS): To evaluate the efficiency of the translation models in terms of processing speed, we introduced the Characters per Second (CPS) metric. This metric measures the number of Chinese characters translated per second, thereby reflecting the model's computational efficiency:

$$\text{CPS} = \frac{\text{Number of Chinese Characters in the Dataset}}{\text{Total Time Used for Translation (seconds)}} \tag{2}$$

The CPS metric is essential for understanding the practicality of each model in real-world applications where translation speed is a critical factor. A higher CPS value indicates faster translation speeds, which is particularly important for handling large volumes of text in time-sensitive scenarios.

V. Results and Discussion

Our study evaluated a diverse range of large language models for Chinese-to-English translation tasks, analyzing their performance across zero-shot, few-shot, and fine-tuned settings. We assessed these models using the COMET metric for translation quality, Translation Completeness Ratio (TCR) for reliability, and Characters per Second (CPS) for efficiency.

A. Zero-shot and Few-shot Performance

The performance of large language models in zero-shot and few-shot settings is summarized in Table II. This table presents the COMET scores, Translation Completeness Ratio

(TCR), and Characters per Second (CPS) across various few-shot settings.

The performance across zero-shot and few-shot settings reveals several notable trends:

- Zero-shot Performance:
  - The Qwen2-72B model demonstrated the highest performance, achieving a COMET score of 0.7324.
  - GPT-4o and GPT-4o-mini models followed closely, with COMET scores of 0.7258 and 0.7259, respectively.
  - Smaller models, such as InternLM2.5-7B and Qwen2-7B, also showed competitive results, with COMET scores of 0.7174 and 0.7170, respectively.
- Few-shot Learning:
  - The Qwen2-72B model achieved the highest overall COMET score of 0.7482 in the 50-shot learning setting, showing a consistent upward trend as more examples were provided.
  - The Llama3.1-70B model also showed significant improvement, reaching a score of 0.7390 with 50-shot learning, highlighting its capacity to learn effectively from additional examples.
  - The Qwen2-7B model demonstrated impressive gains, attaining its peak score of 0.7386 with 50-shot learning, surpassing both the GPT-4o and GPT-4o-mini models.
  - The GPT-4o and GPT-4o-mini models exhibited steady improvement across all few-shot settings, achieving their highest scores of 0.7362 and 0.7371, respectively, with 50-shot learning.
- Translation Completeness Ratio (TCR):
  - Most models maintained high TCR scores across different few-shot settings, often above 0.95, indicating consistent production of complete translations.
  - Larger models generally showed more stable TCR scores across different few-shot settings.
- Characters per Second (CPS):
  - Smaller models consistently demonstrated higher CPS rates, with Phi-3.5-mini and Qwen2-7B achieving over 27 CPS in zero-shot settings.
  - Larger models, while producing higher quality translations, showed significantly lower CPS rates. For instance, Qwen2-72B processed only 3.03 CPS in the zero-shot setting.
  - CPS rates generally decreased as the number of few-shot examples increased, likely due to the increased computational load of processing additional examples.

These results suggest that while there is a general trend of larger models yielding better performance in terms of translation quality (COMET scores), other factors such as model architecture and the quality of training data are also critical in determining overall effectiveness. The trade-off between translation quality and speed (CPS) is evident, with smaller models offering significantly higher processing speeds at the cost of somewhat lower quality. Few-shot learning generally improves performance across all models, demonstrating the adaptability of these models with limited training data.

B. Fine-tuning Results

Fine-tuning on a larger dataset led to notable improvements for most models. We observed important trends in the fine-tuning process over 5 epochs, as shown in Table III.

Based on the fine-tuning results presented in Table III, we can make the following observations:

- Performance Trends:
  - Initially, most models showed improved performance as the number of fine-tuning epochs increased.
  - However, some models began to show signs of performance plateau or slight decline within the 5 epochs, indicating the onset of overfitting.
- Model-Specific Observations:
  - The Qwen2-72B model achieved its peak performance after two epochs of fine-tuning, with a COMET score of 0.7564, representing a 3.28% improvement over its zero-shot performance. After this peak, the scores began to decrease slightly.
  - The Llama3.1-70B model showed improvement up to three epochs, reaching a score of 0.7495, a 4.51% increase from its zero-shot performance. Performance plateaued after this point.
  - Smaller models like Phi-3.5-mini showed a more extended improvement curve, with scores increasing up to the fifth epoch, reaching a peak COMET score of 0.7117, a 7.22% improvement over its zero-shot performance.
  - The Qwen2-7B model demonstrated significant improvement, peaking at 0.7474 after three epochs, a 4.24% increase from its zero-shot score.
- Epochs to Peak Performance:
  - Larger models tended to reach their peak performance earlier, often within 2-3 epochs. For example, Qwen2-72B peaked at 2 epochs, and Llama3.1-70B at 3 epochs.
  - Smaller models generally benefited from more epochs of fine-tuning within the 5-epoch limit. For instance, Phi-3.5-mini and InternLM2.5-7B showed their best performance at 5 and 4 epochs, respectively.
- Translation Completeness Ratio (TCR):
  - Most models showed improvement in TCR with fine-tuning, often reaching perfect or near-perfect scores (1.0000) after a few epochs.
  - Larger models like Llama3.1-70B achieved and maintained perfect TCR from the first epoch onwards.
- Characters per Second (CPS):
  - Most models showed a decrease in CPS (characters per second) following fine-tuning, suggesting a

TABLE II: Performance of Large Language Models Across Few-shot Settings

Model	Metric	0-shot	1-shot	5-shot	10-shot	50-shot
GPT-4o	COMET	0.7258	0.7269	0.7172	0.7282	**0.7362**
	TCR	**0.9771**	0.9612	0.8791	0.9365	0.9718
	CPS	17.64	**22.40**	22.40	15.06	13.31
GPT-4o-mini	COMET	0.7259	0.7270	0.7178	0.7270	**0.7371**
	TCR	**0.9806**	0.9603	0.8791	0.9303	0.9718
	CPS	16.92	17.21	22.40	**22.70**	20.91
Qwen2-72B	COMET	0.7324	0.7379	0.7413	0.7437	**0.7482**
	TCR	0.9673	0.9806	0.9859	0.9894	**0.9903**
	CPS	**3.03**	3.00	1.57	0.91	0.24
Llama3.1-70B	COMET	0.7171	0.7264	0.7337	0.7355	**0.7390**
	TCR	0.9515	0.9806	0.9806	0.9815	**0.9929**
	CPS	**3.44**	3.25	1.56	1.56	0.23
Llama3.1-8B	COMET	0.7006	0.7085	**0.7182**	0.7156	0.7013
	TCR	0.9859	**0.9894**	0.9779	0.9656	0.8959
	CPS	**25.08**	23.30	9.58	9.58	0.64
Qwen2-7B	COMET	0.7170	0.7266	0.7350	0.7340	**0.7386**
	TCR	0.9214	0.9241	0.9638	0.9762	**0.9850**
	CPS	**27.50**	25.61	11.78	6.73	1.54
InternLM2.5-7B	COMET	0.7174	0.7185	0.7235	**0.7256**	0.7117
	TCR	**0.9912**	0.9868	0.9876	0.9850	0.9594
	CPS	**22.03**	20.54	9.43	4.71	0.64
Mistral-7B	COMET	0.6896	0.6940	**0.7087**	0.7019	0.7087
	TCR	0.9841	0.9718	0.9850	0.9841	**0.9912**
	CPS	**22.38**	18.20	7.96	4.11	0.91
Phi-3.5-mini	COMET	0.6638	0.6667	0.6795	**0.6859**	0.6781
	TCR	0.9850	0.9620	0.9912	**0.9938**	0.9832
	CPS	**27.42**	22.33	9.20	3.93	0.45

trade-off between improved translation quality and processing speed. This reduction in speed could be attributed to the loading and inference of LoRA and QLoRA fine-tuned models.

- LoRA and QLoRA adapters enable fine-tuning by adding small low-rank matrices to the original model weights, allowing models to adapt to new tasks with minimal changes to the pre-existing parameters. While this method is efficient in terms of memory and computation during training, the dynamic combination of pre-trained and adapter weights during inference may lead to reduced CPS.
- The GPT-4o and GPT-4o-mini models were notable exceptions, displaying an increase in CPS after fine-tuning. This suggests that their architecture or fine-tuning process may optimize the loading and merging of weights more effectively, reducing computational overhead and thereby enhancing processing speed.

- OpenAI Models Exception:
  - Both GPT-4o and GPT-4o-mini showed no significant improvement in COMET scores during fine-tuning. Their scores remained essentially unchanged across all fine-tuning epochs.
  - For GPT-4o, the COMET score changed minimally from 0.7258 (zero-shot) to 0.7259 (after 5 epochs), a negligible 0.01% difference.
  - Similarly, GPT-4o-mini's score went from 0.7259 (zero-shot) to 0.7264 (after 5 epochs), a mere 0.07% increase.
  - However, both models showed notable improvements in CPS after fine-tuning, with GPT-4o-mini achieving its highest CPS of 32.26 at 5 epochs.

These results highlight the complex dynamics of fine-tuning large language models for translation tasks. While most models benefit from fine-tuning, the optimal number of epochs varies based on model size and architecture. The trade-off between translation quality, completeness, and speed becomes evident, with different models exhibiting unique patterns in these metrics across fine-tuning epochs.

C. Comparison of Metrics Across Settings

To better understand the impact of few-shot learning and fine-tuning on model performance, we compare the three key metrics (COMET, TCR, and CPS) for each model in three scenarios: baseline (0-shot), best few-shot, and best fine-tuned. Table IV presents this comparison.

From this comparison, we can observe several trends:

- COMET Scores:
  - All models show improvement in COMET scores from baseline to their best performance, whether achieved through few-shot learning or fine-tuning.
  - Larger models (Qwen2-72B, Llama3.1-70B) show the most significant improvements, with Qwen2-72B achieving the highest overall COMET score of 0.7564 after fine-tuning.
  - Smaller models also show notable improvements, with Phi-3.5-mini demonstrating the largest relative improvement from baseline (0.6638) to best fine-tuned (0.7117), a 7.22% increase.
  - GPT-4o and GPT-4o-mini show minimal improvements, with their best performances achieved in the few-shot setting rather than through fine-tuning.
- Translation Completeness Ratio (TCR):

TABLE III: Fine-tuning Results Across 0-5 Epochs

Model	Metric	0 epoch	1 epoch	2 epochs	3 epochs	4 epochs	5 epochs
GPT-4o	COMET	0.7258	0.7258	0.7260	**0.7263**	0.7257	0.7259
	TCR	**0.9771**	0.9788	0.9788	0.9779	0.9788	0.9788
	CPS	17.64	26.52	29.21	27.01	27.35	**26.29**
GPT-4o-mini	COMET	0.7259	0.7260	0.7259	0.7253	**0.7264**	0.7262
	TCR	**0.9806**	0.9779	0.9762	0.9788	0.9788	0.9797
	CPS	16.92	30.31	28.39	29.89	27.40	**32.26**
Qwen2-72B	COMET	0.7324	0.7552	**0.7564**	0.7475	0.7435	0.7365
	TCR	0.9673	0.9991	**1.0000**	0.9982	0.9903	0.9788
	CPS	**3.03**	1.78	1.74	1.73	1.67	1.62
Llama3.1-70B	COMET	0.7171	0.7386	0.7480	**0.7495**	0.7407	0.7361
	TCR	0.9515	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	CPS	**3.44**	1.76	1.76	1.68	1.66	1.66
Llama3.1-8B	COMET	0.7006	0.7235	0.7369	**0.7431**	0.7413	0.7384
	TCR	0.9859	**1.0000**	**1.0000**	**1.0000**	0.9991	**1.0000**
	CPS	**25.08**	21.99	22.50	21.84	22.12	21.82
Qwen2-7B	COMET	0.7170	0.7319	0.7378	**0.7474**	0.7445	0.7391
	TCR	0.9214	0.9938	0.9973	0.9646	0.9903	**0.9894**
	CPS	**27.50**	22.97	21.35	21.24	21.84	21.36
InternLM2.5-7B	COMET	0.7174	0.7199	0.7292	0.7345	**0.7397**	0.7362
	TCR	0.9912	0.9974	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	CPS	**22.03**	17.55	17.22	16.87	17.32	17.09
Mistral-7B	COMET	0.6896	0.7103	**0.7234**	0.7212	0.7183	0.7119
	TCR	0.9841	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	CPS	**22.38**	20.35	19.58	19.65	18.98	18.52
Phi-3.5-mini	COMET	0.6638	0.6944	0.7028	0.7053	0.7111	**0.7117**
	TCR	0.9850	0.9974	**1.0000**	**1.0000**	**1.0000**	0.9991
	CPS	**27.42**	15.84	16.30	16.42	15.89	15.89

TABLE IV: Comparison of Metrics Across Baseline, Best Few-shot, and Best Fine-tuned Settings

Model	Baseline (0-shot)			Best Few-shot				Best Fine-tuned			
	COMET	TCR	CPS	COMET	TCR	CPS	Shot	COMET	TCR	CPS	Epoch
GPT-4o	0.7258	0.9771	17.64	0.7362	0.9718	13.31	50	0.7263	0.9788	29.21	2
GPT-4o-mini	0.7259	0.9806	16.92	0.7371	0.9718	**20.91**	50	0.7264	0.9797	**32.26**	5
Qwen2-72B	**0.7324**	0.9673	3.03	**0.7482**	0.9903	0.24	50	**0.7564**	**1.0000**	1.78	2
Llama3.1-70B	0.7171	0.9515	3.44	0.7390	**0.9929**	0.23	50	0.7495	**1.0000**	1.76	3
Llama3.1-8B	0.7006	0.9859	25.08	0.7182	0.9779	9.58	5	0.7431	**1.0000**	22.50	3
Qwen2-7B	0.7170	0.9214	**27.50**	0.7386	0.9850	1.54	50	0.7474	0.9973	22.97	3
InternLM2.5-7B	0.7174	**0.9912**	22.03	0.7256	0.9850	4.71	10	0.7397	**1.0000**	17.55	4
Mistral-7B	0.6896	0.9841	22.38	0.7087	0.9912	0.91	5	0.7234	**1.0000**	19.65	2
Phi-3.5-mini	0.6638	0.9850	27.42	0.6859	0.9938	3.93	10	0.7117	**1.0000**	16.42	5

  - Most models achieve perfect or near-perfect TCR (1.0000) after fine-tuning, indicating highly reliable production of complete translations.
  - Baseline TCR scores are already high for most models, suggesting that incomplete translations are not a significant issue even in zero-shot settings.
  - The Qwen2-7B model shows the most significant improvement in TCR, from 0.9214 in the baseline to 0.9973 after fine-tuning.
- Characters per Second (CPS):
  - There's a notable trade-off between translation quality and speed, especially for larger models.
  - Smaller models (e.g., Phi-3.5-mini, Qwen2-7B) generally maintain higher CPS rates across all settings compared to larger models.
  - Larger models (Qwen2-72B, Llama3.1-70B) show significant decreases in CPS from baseline to best few-shot performance, but some recovery in the fine-tuned setting.
  - Interestingly, GPT-4o and GPT-4o-mini show improvements in CPS after fine-tuning, with GPT-4o-mini achieving the highest overall CPS of 32.26 in its best fine-tuned state.
- Overall Performance:
  - Qwen2-72B demonstrates the best overall performance, achieving the highest COMET score (0.7564) and perfect TCR (1.0000) after fine-tuning, albeit with lower CPS.
  - GPT-4o and GPT-4o-mini show strong baseline performance and maintain high CPS rates, but benefit less from few-shot learning and fine-tuning compared to other models.
  - Smaller models like Qwen2-7B and InternLM2.5-7B show impressive improvements with fine-tuning, approaching the performance of larger models while maintaining higher CPS rates.

This comparison highlights the complex interplay between model size, training approach, and performance across different metrics. While larger models generally achieve higher COMET scores, smaller models offer advantages in terms of processing speed. Notably, for all open-source Large Language Models (LLMs) in our study, fine-tuning not only leads to better COMET scores but also higher Characters per Second (CPS) rates compared to few-shot learning. This observation

strongly suggests that fine-tuning is the recommended approach for translation tasks using these models.

The benefits of fine-tuning are particularly significant when considering the flexibility it offers. With techniques like Low-Rank Adaptation (LoRA) or its quantized version (QLoRA), the same LLM can be fine-tuned for various tasks, including translation. Different adapters can then be used for inference, creating a highly versatile solution for LLM applications. This approach allows for task-specific optimization without the need to retrain the entire model, making it both resource-efficient and adaptable to diverse requirements.

VI. Conclusion and Future Work

Our study on large language models for Chinese-to-English translation provides key insights:

1) Larger models (e.g., 72B parameters) achieve the best COMET scores, but architecture and training data significantly influence performance, with some 7B models showing competitive results.
2) Fine-tuning improves both translation quality and efficiency for open-source models, outperforming few-shot learning and highlighting its importance for practical deployment.
3) Metrics like COMET and TCR effectively evaluate translation quality, with fine-tuned models achieving high scores.
4) Efficiency (CPS rates) varies greatly with model size: smaller models are faster, while larger models prioritize quality over speed.
5) Optimal fine-tuning epochs depend on model size, with larger models peaking earlier (2–3 epochs) than smaller ones (4–5 epochs).
6) OpenAI models (GPT-4o) show strong baseline performance with minimal gains from fine-tuning.

These findings reveal trade-offs between model size, quality, and speed, with fine-tuning and techniques like LoRA/QLoRA offering promising solutions for efficient applications.

Future work could focus on:

1) Multi-task learning approaches that combine translation with other NLP tasks.
2) Hybrid systems leveraging smaller models for initial translations and larger ones for refinement.
3) Extending analyses to other language pairs to assess the generalizability of the findings.

In conclusion, while large language models excel in translation tasks, optimizing performance and efficiency remains a key challenge. Advancements in architecture, training techniques, and hybrid systems hold the potential to further improve translation quality and accessibility, fostering better global communication.

References

[1] P. Koehn, *Statistical machine translation*. Cambridge University Press, 2009.

[2] I. Sutskever, "Sequence to sequence learning with neural networks," *arXiv preprint arXiv:1409.3215*, 2014.

[3] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[4] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 4171–4186.

[6] B. Zhang, D. Xiong, and J. Liu, "Improving chinese to english neural machine translation with linguistic tags," *The Prague Bulletin of Mathematical Linguistics*, vol. 108, pp. 233–244, 2017. [Online]. Available: https://ufal.mff.cuni.cz/pbml/108/art-zhang-xiong-liu.pdf

[7] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.

[8] Y. Liu, "Multilingual denoising pre-training for neural machine translation," *arXiv preprint arXiv:2001.08210*, 2020.

[9] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, 2020, pp. 8440–8451. [Online]. Available: https://aclanthology.org/2020.acl-main.747

[10] B. Zhang, B. Haddow, and A. Birch, "Prompting large language model for machine translation: A case study," in *International Conference on Machine Learning*. PMLR, 2023, pp. 41 092–41 110.

[11] X. Zhang, N. Rajabi, K. Duh, and P. Koehn, "Machine translation with large language models: Prompting, few-shot learning, and fine-tuning with qlora," in *Proceedings of the Eighth Conference on Machine Translation*, 2023, pp. 468–481.

[12] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[13] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," *arXiv preprint arXiv:2305.14314*, 2023.

[14] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[15] R. Rei, J. G. De Souza, D. Alves, C. Zerva, A. C. Farinha, T. Glushkova, A. Lavie, L. Coheur, and A. F. Martins, "Comet-22: Unbabel-ist 2022 submission for the metrics shared task," in *Proceedings of the Seventh Conference on Machine Translation (WMT)*, 2022, pp. 578–585.

[16] D. Huang, Z. Hu, and Z. Wang, "Performance analysis of llama 2 among other llms," in *2024 IEEE Conference on Artificial Intelligence (CAI)*. IEEE, 2024, pp. 1081–1085.

[17] D. Huang, X. Fu, X. Yin, H. Pen, and Z. Wang, "Automating maritime risk data collection and identification leveraging large language models," in *2024 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2024, pp. 433–439.

[18] Y. Zheng, R. Zhang, J. Zhang, Y. Ye, Z. Luo, Z. Feng, and Y. Ma, "Llamafactory: Unified efficient fine-tuning of 100+ language models," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*. Bangkok, Thailand: Association for Computational Linguistics, 2024. [Online]. Available: http://arxiv.org/abs/2403.13372

[19] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz *et al.*, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, 2020, pp. 38–45.

[20] M. Freitag, R. Rei, N. Mathur, C.-k. Lo, C. Stewart, E. Avramidis, T. Kocmi, G. Foster, A. Lavie, and A. F. Martins, "Results of wmt22 metrics shared task: Stop using bleu–neural metrics are better and more robust," in *Proceedings of the Seventh Conference on Machine Translation (WMT)*, 2022, pp. 46–68.