Can General-Purpose Large Language Models Generalize to English-Thai Machine Translation?

Jirat Chiaranaipanich[1], Naiyarat Hanmatheekuna[2], Jitkapat Sawatphol[3], Krittamate Tiankanon[4], Jiramet Kinchagawat[4], Amrest Chinkamol[3,4], Parinthapat Pengpun[5], Piyalitt Ittichaiwong[4,6,7,*], Peerat Limkonchotiwat[3,*],

[1]Ruamrudee International School, [2]Chulalongkorn University, [3]Vidyasirimedhi Institute of Science and Technology, [4]PreceptorAI team, CARIVA Thailand, [5]Bangkok Christian International School, [6]Mahidol University, [7]King's College London, [*]Corresponding authors

piyalitt.itt@preceptorai.tech, peerat.l_s19@vistec.ac.th

Abstract

Large language models (LLMs) perform well on common tasks but struggle with generalization in low-resource and low-computation settings. We examine this limitation by testing various LLMs and specialized translation models on English-Thai machine translation and code-switching datasets. Our findings reveal that under more strict computational constraints, such as 4-bit quantization, LLMs fail to translate effectively. In contrast, specialized models, with comparable or lower computational requirements, consistently outperform LLMs. This underscores the importance of specialized models for maintaining performance under resource constraints.

1 Introduction

Large language models (LLMs) have shown remarkable capabilities in Neural Machine Translation (NMT) and code-switching (CS), attributed to their robustness and generalization (Vaswani et al., 2013; Naveed et al., 2024; Radford et al., 2019). Recent studies indicate that NMT and CS are largely solved for LLMs in high-resource languages (Zhang and Zong, 2020; Hamed et al., 2017; Zhou et al., 2020). However, our research reveals that this performance fails to generalize to low-resource and low-computation settings, which is critical for real-world settings where computational resources are constrained.

This paper explores the generalization of LLMs through two research questions: (i) How do general-purpose LLMs and specialized translation models generalize to low-resource language translation? (ii) How do real-life computational constraints affect performance metrics? To address these questions, we experiment with Llama-3 in various quantization settings. Additionally, we compare LLMs with specialized translation models like NLLB (Team et al., 2022) to evaluate performance and efficiency trade-offs.

2 Experimental Setup

Datasets. We evaluated two translation datasets: (i) a proprietary medical CS translation dataset[1], containing 63,982 English-Thai sentence pairs with retained English medical terms; and (ii) scb-mt-en-th-2020 (Lowphansirikul et al., 2021), a 1,001,752 sentence pair English-Thai translation dataset, from which we randomly selected 63,982 pairs to match the sample size of the CS dataset.

Models Our evaluation focused on three models pertinent to our research questions: Llama-3 8B (Meta, 2024), NLLB-600M, and NLLB-3.3B (Team et al., 2022). For the Llama-3 model, we assessed both the pre-trained and finetuned versions, with the latter quantized to 2, 3, 4, and 8 bits using GPTQ (Frantar et al., 2022). For the NLLB models, we evaluated both pre-trained and finetuned versions. All were finetuned for 3 epochs with a learning rate of 2e-4 on an A100 GPU.

Metrics We employed standard MT metrics for evaluation, such as BLEU3, METEOR, and CER. Additionally, we measured the CS boundary F1 score, which is the harmonic mean of precision and recall for correctly preserved English terms (Sterner and Teufel, 2023).

LLM-as-a-judge Evaluation. To analyze performance degradation, we used GPT4-o[2] as a judge with 3-shot prompting to identify failure modes in each predicted translation. GPT4-o received the source, target, and predicted sentences. The LLM judge assigned a multiple-choice label to each translation, categorizing them as "Forgot to translate," "Meaning changed," "Gibberish," or "Excellent," with a "Keywords not preserved" category for the CS translation task.

[1] https://cariva.co.th/

[2] snapshot gpt-4o-2024-05-13

3 Results

As illustrated in Table 1, NLLB-3.3B and NLLB-600M outperform Llama-3 8B on most metrics, despite using 2.35x and 10.81x less VRAM, respectively. This contrasts with prior studies indicating the superiority of general-purpose language models in specialized, low-resource tasks (Li et al., 2023; Nori et al., 2023; Naveed et al., 2024). Moreover, the average percentage difference between NLLB-3.3B and full-precision Llama-3 8B across BLEU and METEOR scores is $\sim$23.39% and $\sim$1.33% for the SCB and CS dataset, respectively. This minimal difference for the CS dataset suggests that NLLB's multilingual pre-training is not a significant advantage in translation-adjacent tasks.

Interestingly, Llama-3-8B excels in the METEOR metric for CS translation, which accounts for word stems and synonyms. This suggests Llama-3-8B produces relevant but imprecise translations, affecting metrics that require exact matches but not METEOR.

Dataset	Model Variant	BLEU3	METEOR	CER	CS-F1	Memory (GB)
CS	Llama-3-8b	0.421	0.615	6.606	0.330	31.48
	Llama-3-8b-8bit	0.421	**0.616**	6.622	0.332	9.87
	Llama-3-8b-4bit	0.392	0.591	6.833	0.320	7.13
	Llama-3-8b-3bit	0.214	0.410	8.437	0.280	5.42
	Llama-3-8b-2bit	0.001	0.013	4.565	0.002	4.48
	NLLB-3.3b	**0.443**	0.600	**0.419**	**0.398**	13.42
	NLLB-0.6b	0.410	0.576	0.438	0.394	2.91
SCB	Llama-3-8b	0.173	0.371	30.416	-	31.48
	Llama-3-8b-8bit	0.173	0.371	30.952	-	9.46
	Llama-3-8b-4bit	0.156	0.349	30.576	-	7.14
	Llama-3-8b-3bit	0.079	0.231	31.088	-	5.35
	Llama-3-8b-2bit	0.000	0.003	19.232	-	4.44
	NLLB-3.3b	**0.244**	**0.449**	0.585	-	13.31
	NLLB-0.6b	0.238	0.437	**0.574**	-	2.91

Table 1: Evaluation Results for LLMs and specialized translation models on CS and SCB datasets.

4 Analysis

Failure Analysis As shown in Figures 1a and 1b, we observed a divergence in failure modes between the two datasets. For the SCB dataset, errors initially rise in the "Meaning changed" category (from 16 to 4 bits) and then in the "Gibberish" category (from 4 to 2 bits). In the CS dataset, errors first increase in the "Meaning changed" category while decreasing in "Keywords not preserved" (from 16 to 4 bits), followed by an increase in "Gibberish" errors (from 4 to 2 bits). Notably, the best-performing models (NLLB-3.3B and NLLB-600M) exhibit the highest number of "Forgetting to preserve" errors. This suggests an alternative failure mode in CS translation, where top models first lose the ability to preserve medical keywords, then to translate accurately, and finally to translate at all. Importantly,

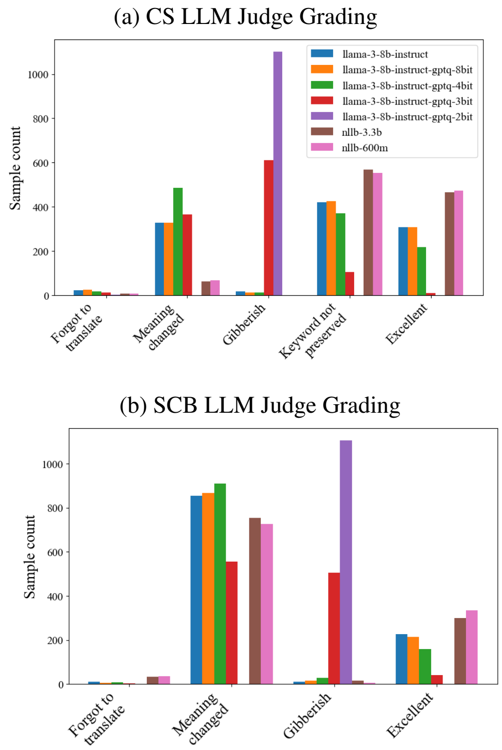

Figure 1: Llama-3 and NLLB failure analysis. Note that the legend is shared between Figures 1a and 1b.

despite higher errors in the "Forgetting to preserve" category, NLLB models perform better on the CS-F1 metric, Table 1, highlighting the importance of task-specific metrics.

Impact of Quantization Interestingly, CS results show greater resilience to quantization than SCB results. Across BLEU, CER, and METEOR metrics, CS translation results experience less degradation than SCB results when compared against the full-precision baseline. This may be due to the early loss of complex Thai vocabulary during quantization, while complex English vocabulary, rewarded in the CS task, is better preserved. The resilience of CS results suggests a novel approach for mitigating performance degradation in quantized multilingual models by leveraging CS outputs.

5 Conclusion

We study the performance of general-purpose and specialized language models on translation and translation-adjacent tasks. Our findings indicate that specialized translation models outperform general-purpose models, although the performance gap is smaller for CS translation. As models undergo increased quantization, the divergence in failure modes between SCB and CS datasets underscores the importance of task-specific metrics.

References

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2022. GPTQ: Accurate post-training quantization for generative pre-trained transformers.

Injy Hamed, Mohamed Elmahdy, and Slim Abdennadher. 2017. Building a first language model for code-switch Arabic-English. *Procedia Comput. Sci.*, 117:208–216.

Xianzhi Li, Samuel Chan, Xiaodan Zhu, Yulong Pei, Zhiqiang Ma, Xiaomo Liu, and Sameena Shah. 2023. Are chatgpt and gpt-4 general-purpose solvers for financial text analytics? a study on several typical tasks. *Preprint*, arXiv:2305.05862.

Lalita Lowphansirikul, Charin Polpanumas, Attapol T. Rutherford, and Sarana Nutanong. 2021. A large english–thai parallel corpus from the web and machine-generated text. *Language Resources and Evaluation*, 56(2):477–499.

Llama @ Meta. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2024. A comprehensive overview of large language models. *Preprint*, arXiv:2307.06435.

Harsha Nori, Yin Tat Lee, Sheng Zhang, Dean Carignan, Richard Edgar, Nicolo Fusi, Nicholas King, Jonathan Larson, Yuanzhi Li, Weishung Liu, Renqian Luo, Scott Mayer McKinney, Robert Osazuwa Ness, Hoifung Poon, Tao Qin, Naoto Usuyama, Chris White, and Eric Horvitz. 2023. Can generalist foundation models outcompete special-purpose tuning? case study in medicine. *Preprint*, arXiv:2311.16452.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Igor Sterner and Simone Teufel. 2023. TongueSwitcher: Fine-grained identification of German-English code-switching. In *Proceedings of the 6th Workshop on Computational Approaches to Linguistic Code-Switching*, pages 1–13, Singapore. Association for Computational Linguistics.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

Ashish Vaswani, Yinggong Zhao, Victoria Fossum, and David Chiang. 2013. Decoding with large-scale neural language models improves translation. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1387–1392.

Jiajun Zhang and Chengqing Zong. 2020. Neural machine translation: Challenges, progress and future. *CoRR*, abs/2004.05809.

Xuehao Zhou, Xiaohai Tian, Grandee Lee, Rohan Kumar Das, and Haizhou Li. 2020. End-to-end code-switching tts with cross-lingual language model. pages 7614–7618.

6 Appendix

6.1 Finetuning Prompts for Llama

Code-switching (CS) Prompt

```
You are a helpful code switching English to Thai
    language translation assistant. Translate
    the given English texts to Thai while
    preserving the medical keywords.
```

Machine translation (SCB) Prompt

```
You are a helpful English to Thai language
    translation assistant. Translate the given
    English texts to Thai.
```

6.2 LLM Judge Prompts

LLM Judge Code-switching Dataset Prompt

```
You will be given a user_text, model_answer, and
    system_translation trio. Your task is to
    provide a multiple choice answer, analyzing
    the cause of failure of the system's
    translation of the user's text when compared
    to the model_answer.Give your answer letter
    which can either be A, B, C, D, E.

Here are the choices.
A: The system_translation forgot to translate:
    missed translating a large part of the text
B: The system_translation translated wrongly:
    adds additional information or hallucinates;
    changes the meaning in some significant way
C: The system_translation is gibberish: it does
    not make sense and is just a jumble of words
    and characters
D: The system_translation forgot to preserve the
    CS keyword: although the text is translated
    ; the meaning is quite well preserved; the
    keywords are translated amd not preserved in
    the orignal language
E: The system_translation is excellent:
    preserves the keywords; has almost the
    meaning as the model answer; everything
    except for the keywords are translated

You MUST provide the answer letter. Do not
    provide anything else.
```

Table 2: Full Evaluation Result on the CS and SCB datasetes. "Memory(GB)" indicates the memory consumption for single-batch inference on an A100 GPU. "Runtime vs 16bit Llama" represents the inference time speedup compared to a 16bit Llama baseline.

Dataset	Model Variant	BLEU3	METEOR	CER	WER	chrF	CS-F1	Memory (GB)	Runtime vs 16bit Llama (%)
CS	Llama-3-8b	0.421	0.615	6.606	0.526	0.402	0.330	31.48	0
	Llama-3-8b-8bit	0.421	**0.616**	6.622	0.525	0.401	0.332	9.87	-17.08
	Llama-3-8b-4bit	0.392	0.591	6.833	0.559	0.386	0.320	7.13	-61.05
	Llama-3-8b-3bit	0.214	0.410	8.437	0.917	0.262	0.280	5.42	30.11
	Llama-3-8b-2bit	0.001	0.013	4.565	4.616	0.039	0.002	4.48	-15.17
	NLLB-3.3b	**0.443**	0.600	**0.419**	**0.460**	**0.571**	**0.398**	13.42	-85.66
	NLLB-0.6b	0.410	0.576	0.438	0.487	0.551	0.394	**2.91**	**-97.13**
SCB	Llama-3-8b	0.173	0.371	30.416	0.865	0.147	-	31.48	0
	Llama-3-8b-8bit	0.173	0.371	30.952	0.867	0.145	-	9.46	23.82
	Llama-3-8b-4bit	0.156	0.349	30.576	0.891	0.138	-	7.14	-55.41
	Llama-3-8b-3bit	0.079	0.231	31.088	1.142	0.105	-	5.35	-22.83
	Llama-3-8b-2bit	0.000	0.003	19.232	20.030	0.004	-	4.44	22.44
	NLLB-3.3b	**0.244**	**0.450**	**0.585**	0.729	**0.475**	-	13.31	-86.52
	NLLB-0.6b	0.238	0.437	0.574	**0.721**	0.461	-	**2.91**	**-97.05**

```
Here are examples with the best answer given
    plus reasoning.

EXAMPLE 1:
User Text: USER_TEXT_1
Model Answer: MODEL_ANSWER_1
System Translation: SYSTEM_TRANSLATION_1
Reasoning: REASONING_1

EXAMPLE 2:
User Text: USER_TEXT_2
Model Answer: MODEL_ANSWER_2
System Translation: SYSTEM_TRANSLATION_2
Reasoning: REASONING_2

EXAMPLE 3:
User Text: USER_TEXT_3
Model Answer: MODEL_ANSWER_3
System Translation: SYSTEM_TRANSLATION_3
Reasoning: REASONING_3

Below are the text, answer, and translation.
    Give a multiple choice response.

User Text: {user_text}
Model Answer: {model_answer}
System Translation: {system_translation}
```

LLM Judge Machine Translation Dataset (SCB) Prompt

```
You will be given a user_text, model_answer, and
     system_translation trio.
Your task is to provide a multiple choice answer,
     analyzing the cause of failure of the
    system's translation of the user's text when
     compared to the model_answer.
Give your answer letter which can either be A, B,
     C, D.

Here are the multiple choices.
A: The system_translation forgot to translate:
    missed translating a large part of the text
B: The system_translation translated wrongly:
    adds additional information or hallucinates;
     changes the meaning in some significant way
C: The system_translation is gibberish: it does
    not make sense and is just a jumble of words
     and characters
D: The system_translation is excellent: has
    almost the meaning as the model answer,
    everything is translated
You MUST provide the answer letter. Do not
    provide anything else other than the
    multiple choice answer letter.

Here are a few examples with the best multiple
    choice answer given plus reasoning.

EXAMPLE 1:
User Text: USER_TEXT_1
Model Answer: MODEL_ANSWER_1
System Translation: SYSTEM_TRANSLATION_1
Reasoning: REASONING_1

EXAMPLE 2:
User Text: USER_TEXT_2
Model Answer: MODEL_ANSWER_2
System Translation: SYSTEM_TRANSLATION_2
Reasoning: REASONING_2

EXAMPLE 3:
User Text: USER_TEXT_3
Model Answer: MODEL_ANSWER_3
System Translation: SYSTEM_TRANSLATION_3
Reasoning: REASONING_3

EXAMPLE 4:
User Text: USER_TEXT_4
Model Answer: MODEL_ANSWER_4
System Translation: SYSTEM_TRANSLATION_4
Reasoning: REASONING_4

Below are the user text and system translation
    pair. Give a multiple choice response.

User Text: {user_text}
Model Answer: {model_answer}
System Translation: {system_translation}
```