Enhancement of LLMs-based Translation by Agentic Machine Translation in Thai Language

Artid Boonrerng, Nouh Elmitwally, Edlira Vakaj, and Shadi Basurra

Birmingham City University, Birmingham B4 7XG, United Kingdom
artid.boonrerng@mail.bcu.ac.uk

Abstract. This experiment has a focus on the enhancement capability of large language models (LLMs) for translation in low-resource languages, specifically Thai. Moreover, the experiment of translation by selected different LLMs, such as GPT-3.5 Turbo, Claude 3.5 Sonnet, SeaLLMs, and Typhoon, translating from English to Thai in general text, found that translation by specific languages pre-trained LLMs such as Typhoon has more accuracy than multilingual pre-trained LLMs such as GPT-3.5 Turbo. Furthermore, attempt to implement Agentic Machine Translate to enhance translation, which is a process that uses 2 LLMs; first assign as translator and second assign as reflector. This experiment has 2 methods, first using the same LLMs as translation and reflection, and second methods using different LLMs. Additionally, the results before and after translation by reflection were assessed by BERTScore, COMET, METEOR, and BLEU. By using different LLMs, the quality of translation increases with Typhoon as translator and Claude as reflector. As a result, the efficiency of translation by agentic flow depends on the generated reflection prompt and pre-trained language. Although this experiment was not to confirm that agentic machine translation can enhance LLMs-based translation accuracy, it showed there is a challenge in leveraging LLMs-based translation instead of machine translation to improve translation in low-resource languages.

Keywords: Large Language Models · Translation Quality · Agentic Machine Translation.

1 Introduction

This experiment is one part of the main research "Challenges of Leveraging Generative AI and Large Language Models in Thai Language Medical Data to Support Health Informatic System" which develops generative AI and large language models (LLMs) to support health informatic systems and focuses on Thai language datasets.

A major problem in developing LLMs in healthcare with non-English language is a lack of high-quality training medical datasets. The requirement for enormous quantities of training data, combined with appropriate annotations for supervised learning, remains a fundamental problem with methods based on deep learning. There are certain Natural Language Processing (NLP) tasks that are able to be used with languages other than English, such as sentence boundary detection, parsing, or sequence segmentation.

Nonetheless, in the case of Thai which do not have white spaces between words, that causes word boundary issues or helps identify each word. On the one hand, Kedtiwerasak et al. [11] try to use the TextRank algorithm, which has more accuracy, to extract keywords in many languages, including English and Spanish, with Thai, but the algorithm is not satisfied due to the non-word boundary such as Chinese and Thai. Thus, the complex structure of Thai languages makes it a major challenge to implement generative AI and LLMs in Thai.

This experiment has a focus on the enhancement capability of LLMs for translation in low-resource languages, specifically Thai, by implementing agentic machine translation to LLMs-based translation.

2 Related Works

2.1 Large Language Models

In recent years, the leveraging of generative AI has grown across a wide range of fields, including the arts, mathematics, and healthcare. The main objective of LLMs is to respond to conversations based on the given input with human-like text. Moreover, LLMs are pre-trained by a large quantity of text data and structurally designed to analyze the relationship between the sequence of predicted words and data chunks using probability and dependability relative to the real world [22]. Additionally, the structure of LLMs is based on deep learning architectures, primarily the transformer model demonstrated by Vaswani et al. [23]. The transformer model has revolutionized NLP and LLMs by enabling more efficient parallel processing and improved handling of long-range dependencies in text. A LLM transformer architectures as shown in Fig. 1.

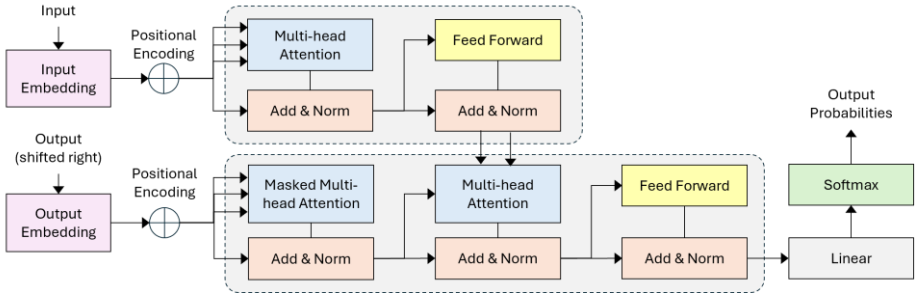

Fig. 1: Transformer-Based LLM Architectures

A transformer-based LLM architecture leverages the Transformer framework, which processes input sequences in parallel rather than sequentially, allowing for faster computation and better long-range dependency modeling. Moreover, the architecture consists of multiple layers of encoders and decoders. Each layer contains multi-head self-attention and feed-forward networks, with residual connections and layer normalization to enhance stability. Additionally, the model calculates attention scores between all

tokens in the input sequence, enabling it to focus on relevant parts of the sequence for each token. This is followed by a feed-forward neural network applied to each token and using positional encoding to maintain information about the order of tokens. Thus, transformer-based LLM are fine-tuned on specific tasks, enabling capabilities like translation, summarization, and text generation.

2.2 LLMs-based Translation

Neural Machine Translation (NMT), which developed from Machine Translation (MT), has made significant progress in the field of translation in recent years [21]. Additionally, MT is a fundamental task in natural NLP that aims to translate natural language sentences from one language to another using computers based on translation rules and linguistic knowledge. However, as natural languages are inherently complex and language translation rules are difficult to cover all, more researchers try to apply LLMs to improve translation quality focusing on accuracy, fluency, and target language style [10].

First, Moslem et al. [13] demonstrate several methods to improve translation quality by leveraging the flexibility of LLMs. Additionally, their present techniques, such as zero-shot, few-shot, and fuzzy-match-based translation, which adaptively modify translation results depending on the previous example and context. As a result, LLMs provide feedback, enhance translation consistency and accuracy, and modify translation in real-time.

On the one hand, Zhu et al. [27] show that the translation capabilities of LLMS have an impressive quality. However, translation capabilities vary significantly across different languages. For example, even though GPT-4[1] performs highly in Indo-European languages, while low-resource languages such as languages from the Afro-Asiatic and Austronesian performed low translation quality. Moreover, researchers suggested training the LLMs with high-quality data and quantity of training data, which is effective for the performance of translation quality.

In conclusion, the important advantage of LLMs-based translation over traditional machine translation is the entire context of a sentence or paragraph understanding, which allows for more accurate and natural translations, especially in complex or unclear situations. Despite that there are numerous papers and studies that focus on the use of LLMs in human-like translation, there are still gaps for enhancing the translation quality, especially in handling low-resource languages.

2.3 Agentic Machine Translation

In mid-2024, Andrew Ng, founder of DeepLearning.AI, posted on LinkedIn, "I think AI agentic machine translation has huge potential for improving over traditional neural machine translation." [14]. The main idea is to use LLMs to perform like AI agents on iterative prompts with reflection for enhanced results. Furthermore, the agentic machine translation uses a LLMs to reflect the first translation text, then passes through the

[1] GPT-4 Turbo and GPT-4, https://platform.openai.com/docs/models/gpt-4-turbo-and-gpt-4

reflection with constructive suggestions, the original text (text in the source language), and the first translation text (text in the target language) to create prompts to ask LLMs to enhance the translation, a workflow as shown in Fig. 2. Moreover, Shavit et al. [20] define the degree of agenticness as "the degree to which a system can adaptably achieve complex goals in complex environments with limited direct supervision." According to the definition, adaptable reflections or suggestions from agentic LLMs will improve the quality of translation by LLMs.

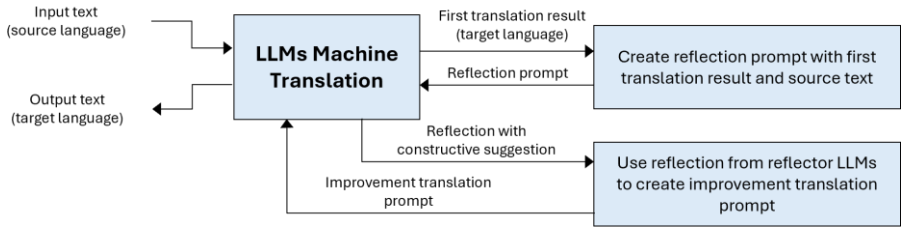

Fig. 2: A workflow of Agentic Machine Translation

2.4 Translation Quality Assessment

The rapid improvement of machine translation technology has driven efficient methods for assessing translation quality. Quality assessment (QA) in MT is important to guarantee that translated texts cover the required standards of accuracy, fluency, and usability. In this experiment, we chose 4 metrics to evaluate the translation result, which are classified by calculation mechanism, such as BLEU, COMET, METEOR, and BERTScore.

Table 1: comparison of BLEU, COMET, METEOR, and BERTScore in terms of precision, recall, and fluency

Metrics	Precision	Recall	Fluency
BLEU [16]	Strong focus on n-gram precision	Limited recall	Although not intended to assess fluency, translations with higher BLEU scores often correlate with more fluent translations that encourage short, incomplete translations.
COMET [19]	Can incorporate precision (depends on the model)	Incorporates recall through semantic similarity	Designed to incorporate human considerations, METEOR can evaluate fluency effectively.
METEOR [1]	Uses precision at the word and synonym level	Recall is given more importance (penalizes missing words more than BLEU)	Scores incline to correlate well with fluency and human considerations as synonym matching and stemming.
BERTScore [25]	Precision is based on word embedding similarity	High recall due to embedding-level matching	Considers fluency indirectly by focusing on semantic similarity, which often aligns with fluency.

Table 1 shows A Comparative Evaluation of BLEU, COMET, METEOR, and BERTScore in Precision, Recall, and Fluency. First, BLEU mainly focuses on

precision, measuring n-gram overlaps between the output and reference. On the other hand, BLUE has a recall problem because it lacks an account for paraphrasing or synonyms. Second, COMET leverages pre-trained models to evaluate both precision and recall through semantic similarity, offering preferable correlation with human consideration, particularly in fluency. Next, METEOR prioritizes recall by matching stems, synonyms, and paraphrases, which allows it to capture a wider range of translations better than BLEU. Lastly, BERTScore balances both precision and recall by using contextualized embeddings to measure semantic similarity between the hypothesis and reference, making it effective at capturing paraphrases and making it more aligned with human evaluation than traditional n-gram-based metrics like BLEU. Overall, COMET and BERTScore offer a more extensive assessment than BLEU or METEOR.

2.5 Technical Background

N-grams [12] are continuous sequences of words or tokens of length N from a given text. In addition, N-grams are commonly used in NLP tasks like machine translation and text generation evaluation metrics, such as BLEU. An n-gram is a sequence of N consecutive words in a text. For a sentence with words W_1, W_2, …, W_T, the n-grams are defined as:

$$\text{N-gram } (N) = \{(W_i, W_{i+1}, \ldots, W_{i+N-1}) \mid i = 1, \ldots, T\text{-}N\text{+}1)\} \quad (1)$$

Where W_i represents the i-th word in the sequence. N is the length of the n-gram. T is the total number of words in the text. In BLEU, n-gram precision is used to evaluate the overlap between machine-generated and reference translations. The precision for n-grams of length N is calculated as:

$$\text{Precision}_N = \frac{\sum_{n-gram \in Hypothesis} \min(count_{Hypothesis}(n-gram), count_{Reference}(n-gram))}{\sum_{n-gram \in Hypothesis} count_{Hypothesis}(n-gram)} \quad (2)$$

Where *Hypothesis* is the machine-generated translation, *Reference* is the reference translation, and *count* is the number of times an *n-gram* appears.

METEOR Score [1] is a combination of the F-score, which is computed from precision and recall with the chunk penalty. Additionally, METEOR creates an alignment between candidate and reference translation by mapping words in the candidate to words in the reference based on exact matches, stemmed matches, synonym matches, and paraphrase matches. The METEOR score is calculated as follows:

$$P = \frac{m}{W_t}; \quad R=\frac{m}{W_r}; \quad F_{mean}=(\frac{10PR}{R+9P}); \quad p=0.5(\frac{c}{u_m})^3; \quad \text{METEOR} = F_{mean}(1-p) \quad (3)$$

Where P and R are the unigram precision and recall based on the alignment, m is the number of mapped unigrams in the system translation, and W_t and W_r are the number of unigrams in the candidate translation and the reference translation, c is the number of chunks, and u_m is the number of unigrams that have been mapped, the F-mean computes a harmonic mean of precision and recall. Finally, the METEOR score ranges from 0 to 1, with 1 indicating a perfect match to the reference translation.

3 Methodology

3.1 Large Language Models Selection

For our study, we aim to examine the performance of machine translation after implementing the agentic workflow. We selected from the difference of the pre-trained dataset language.

1. Pre-trained LLMs with Multilingual: We selected 2 LLMs. Our first LLMs is GPT-3.5 Turbo[2] by OpenAI[3]. In natural language processing, GPT-3.5 Turbo has shown interesting capabilities in tasks such as text completion, summarization, and translation [2]. Moreover, few-shot learning abilities have been important because they can adapt to new tasks with minimal examples [24]. Furthermore, GPT-3.5 Turbo performs highly in natural language processing, enhanced contextual reasoning, and the potential to process complex queries [7]. For the second model, we used Claude[4] which developed by Anthropic[5], designed for various tasks including natural language processing, analysis, and generation. It represents capabilities such as coding, creative writing, and complex problem-solving [8]. In addition, Notable Claude capability is to maintain context over long conversations and following detailed instructions.

2. Pre-trained LLMs with Southeast Asia language: SeaLLMs [15] represent a significant development in language technology for Southeast Asian languages. While LLMs have made great progress in recent years, their development has mainly focused on high-resource languages, particularly English. SeaLLMs aims to address this gap by providing a multilinguistic model for Southeast Asian languages. The development of SeaLLMs builds upon previous work in multilingual language models, such as mBERT [6] and XLM-R [4]. However, these models often attempt to apply with low-resource languages in Southeast Asia. Furthermore, SeaLLMs develops transfer learning from larger models trained on English and other high-resource languages, similar to the approach used in ULMFiT [9]. This allows the model to leverage general language understanding while fine-tuning Southeast Asian languages. Additionally, the architecture of SeaLLMs may be based on recent developments in transformer models, such as GPT-3 [2] or PaLM [3].

3. Pre-trained LLMs with Thai language: Typhoon [17] is a series of LLMs specifically designed for the Thai language. The team developed ThaiExam, a benchmark based on high school and investment professional examinations in Thailand, which includes XNLI [5], XCOPA [18], and M3Exam [26]. Three ONET[6] levels are used to create the Thai subset of M3Exam: low (grade 6), medium (grade 9), and high (grade 12). Moreover, Typhoon was fine-tuned to follow Thai instructions and evaluated on Thai instruction datasets, translation tasks, summarization tasks, and question-answering tasks. Furthermore, the results of experiments across various Thai benchmarks

[2] GPT-3.5 Turbo, https://platform.openai.com/docs/models/gpt-3-5-turbo
[3] OpenAI, https://openai.com/
[4] Meet Claude \ Anthropic, https://www.anthropic.com/claude
[5] Home \ Anthropic, https://www.anthropic.com/
[6] O-NET: Ordinary National Educational Test, https://www.niets.or.th/en/catalog/view/2211

show Typhoon performs better than all open-source Thai language models. Additionally, Typhoon's performance is comparable to GPT-3.5 in Thai, despite having only 7 billion parameters. In addition, Typhoon is 2.62 times more efficient in tokenizing Thai text compared to GPT-3.5.

3.2 The LLMs-based Translation with Agentic flow

In this research, the LLMs-based Translation with Agentic Flow was designed with implementing an agentic machine translation workflow to translate the source text twice; first without reflection and second with reflection and suggestions from reflector LLMs. In both translations, they used the same source text in English and used 5 references in Thai with different styles and fluency, which translate from the original text by Google Translate and humans to compare with the translation results; see Fig. 3, then assessed the second translation result with the same translation quality assessment tools and method as the first translation result.

Source text:
"I love eating spicy Thai food at Thai restaurants in Chinatown."

Reference texts:
"ฉันชอบทานอาหารรสเผ็ดที่ร้านอาหารไทยในไชน่าทาวน์" **translate by Google translate**
"ฉันชอบทานอาหารไทยที่มีรสจัดที่ร้านอาหารไทยในไชน่าทาวน์" **translate by human**
"ฉันชื่นชอบการทานอาหารไทยที่มีรสจัดที่ร้านอาหารไทยในย่านไชน่าทาวน์" **translate by human**
"ฉันรักการทานอาหารไทยที่มีรสจัดที่ร้านอาหารไทยในไชน่าทาวน์" **translate by human**
"ฉันชอบทานอาหารไทยรสแซ่บที่ร้านอาหารไทยในย่านไชน่าทาวน์" **translate by human**

Fig. 3: The Source text and reference texts

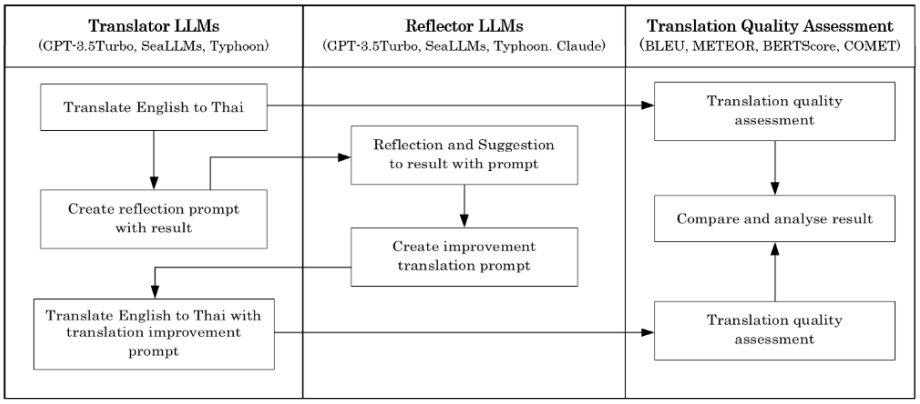

Fig. 4: Implemented LLMs-based Translation with Agentic flow

The workflow started by assigning the first LLMs as translators to be expert linguists specializing in translating from English to Thai by system prompt, then assessing the result from the first translation using selected translation quality assessment tools such

as BLEU, METEOR, BERTScore, and COMET. Furthermore, we used PyThaiNLP[7], a Python package for text processing and linguistic analysis similar to NLTK[8] with a focus on Thai, to segment the Thai sentence into tokens. Next, assign another LLMs as reflectors to be expert linguists specializing and create prompts from the template, as shown in Table 2. The prompt intends to improve the first translation result by focusing on accuracy, fluency, style, and terminology and writing the specific, helpful, and constructive suggestion. Next, we used the reflection and suggestion output from the reflector LLMs to create a prompt to improve the first translation by the first LLMs, then assess the second translation result with the same translation quality assessment tools and method as the first translation result. Finally, comparing before and after implementing agentic flow translation scores by using different translator LLMs, reflector LLMs, and assessment tools, a workflow as shown in Fig. 4.

Table 2: The reflection prompt template

Reflection prompt
Your task is to carefully read a source text and a translation from {SOURCE_LANG} to {TARGET_LANG}, and then give constructive criticism and helpful suggestions to improve the translation. The source text and initial translation, delimited by XML tags are as follows: <SOURCE_TEXT>{SOURCE_TEXT} </SOURCE_TEXT><TRANSLATION>{first_translate_text}</TRANSLATION> When writing suggestions, pay attention to whether there are ways to improve the translation's (i) accuracy (by correcting errors of addition, mistranslation, omission, or untranslated text), (ii) fluency (by applying {TARGET_LANG} grammar, spelling and punctuation rules, and ensuring there are no unnecessary repetitions), (iii) style (by ensuring the translations reflect the style of the source text and takes into account any cultural context), (iv) terminology (by ensuring terminology use is consistent and reflects the source text domain; and by only ensuring you use equivalent idioms {TARGET_LANG}). Write a list of specific, helpful and constructive suggestions for improving the translation. Each suggestion should address one specific part of the translation. Output only the suggestions and nothing else.

4 Results and Discussion

In this part of the experiment, we translate the original sentence by LLMs-based translation with agentic flow 30 times. Table 3 shows examples of translation results from translator LLMs before and after using the suggestions from reflector LLMs. As a result, the reflector LLMs suggest using natural words to improve the fluency of translation and to better convey the meaning of words. Then, we assess before and after translation results with translation quality assessment tools. Lastly, we compare the average translation quality scores before and after implementing agentic flow as shown in Table 4. Moreover, the result shows that it has effects of reflection and suggestion from reflector LLMs that make the translation quality score decrease, such as the reflector suggesting replacing an incorrect word with a correctly translated word. Mostly decreased score from Typhoon, which received high assessment scores from the first translation. Furthermore, the reflection and suggestion from Claude demonstrate more details to enhance translation by accuracy, fluency, style, and terminology than other LLMs that related to remarkable ability to maintain context over long conversations and following detailed instruction.

[7] PYthaiNLP - we build thai nlp, https://pythainlp.org/
[8] NLTK :: Natural Language Toolkit, https://www.nltk.org/

Table 3: Example of before and after translation result and suggestion from LLMs

Translator: GPT-3.5 Turbo, **Reflector:** GPT-3.5 Turbo **1st translation:** “ฉันรักการกินอาหารไทยที่เผ็ดที่ร้านอาหารไทยในชานทาวน์” **2nd translation:** “ฉันชอบกินอาหารไทยเผ็ดที่ร้านอาหารไทยที่ชานทาวน์” **Reflection and suggestion:** 1. Use the term "ชอบ" instead of "รัก" for expressing like or enjoy when referring to the act of eating in this context. 2. Add the word "ที่" before "ชานทาวน์" to specify the location of Chinatown. 3. Consider using a more natural word order to improve the fluency of the translation, such as "ฉันชอบกินอาหารไทยเผ็ดที่ร้านอาหารไทยที่ ชานทาวน์".
Translator: SeaLLMs, **Reflector:** Claude **1st translation:** “ฉันชอบทานอาหารไทยที่รสแซ่บมากๆ ที่ร้านอาหารไทยในชุมชนจีนตอง” **2nd translation:** “ฉันชอบทานอาหารไทยเผ็ดๆที่ร้านอาหารไทยตามร้านอาหารไทยในไชน่าทาวน์” **Reflection and suggestion:** 1. Replace "แซ่บ" with "เผ็ด" to more accurately reflect "spicy" in the source text. 2. Consider adding "ในไชน่าทาวน์" instead of "ในชุมชนจีนตอง" to better represent "Chinatown" and improve fluency. 3. Adjust word order to "ฉันชอบทานอาหารไทยเผ็ดๆ" to improve fluency and match Thai sentence structure. 4. Add "ตามร้านอาหารไทย" to accurately convey "at Thai restaurants" and maintain the source text's meaning. 5. Consider using "เยอะแยะ" instead of "มากๆ" to better reflect the casual tone of "love" in the source text.
Translator: Typhoon, **Reflector:** Claude **1st translation:** “ฉันรักการทานอาหารไทยรสจัดที่ร้านอาหารไทยในเยาวราช” **2nd translation:** “ฉันชอบการกินอาหารไทยรสจัดร้านอาหารไทยในย่านไชนาทาวน์” **Reflection and suggestion:** 1. Consider replacing "การทาน" with "การกิน" for a more natural and colloquial Thai expression. 2. Add the word "ชอบ" after "ฉัน" to better convey the sense of "love" in the original text. 3. Replace "รสเผ็ด" with "รสจัด" to more accurately reflect the English "spicy" in this context. 4. Consider using the more specific term "ย่านไชน่าทาวน์" instead of just "ไชน่าทาวน์" to better convey the idea of "in Chinatown." 5. Add a space before the full stop at the end of the sentence for improved readability. 6. Consider rephrasing to "ร้านอาหารไทยในย่านไชน่าทาวน์" to avoid repetition of "ไทย" in close proximity.

Table 4: Average translation quality score before and after implement an agentic flow (n=30)

		Translation LLMs					
		GPT-35. Turbo		**SeaLLMs**		**Typhoon**	
	Reflector LLMS	**before**	**after**	**before**	**after**	**before**	**after**
BLEU	**GPT-3.5 Turbo**	0.397	0.325	0.173	0.231	0.440	0.286
	SeaLLMs	-	-	0.209	0.174	-	-
	Typhoon	-	-	-	-	0.369	0.195
	Claude	-	-	0.240	0.299	0.528	0.371
METEOR	**GPT-3.5 Turbo**	0.534	0.451	0.504	0.498	0.560	0.454
	SeaLLMs	-	-	0.516	0.477	-	-
	Typhoon	-	-	-	-	0.528	0.425
	Claude	-	-	0.528	0.530	0.593	0.544
BERTScore	**GPT-3.5 Turbo**	0.584	0.593	0.579	0.577	0.574	0.575
	SeaLLMs	-	-	0.596	0.555	-	-
	Typhoon	-	-	-	-	0.574	0.573
	Claude	-	-	0.570	0.563	0.576	0.573
COMET	**GPT-3.5 Turbo**	0.872	0.842	0.796	0.786	0.766	0.520
	SeaLLMs	-	-	0.766	0.755	-	-
	Typhoon	-	-	-	-	0.884	0.800
	Claude	-	-	0.830	0.785	0.854	0.876

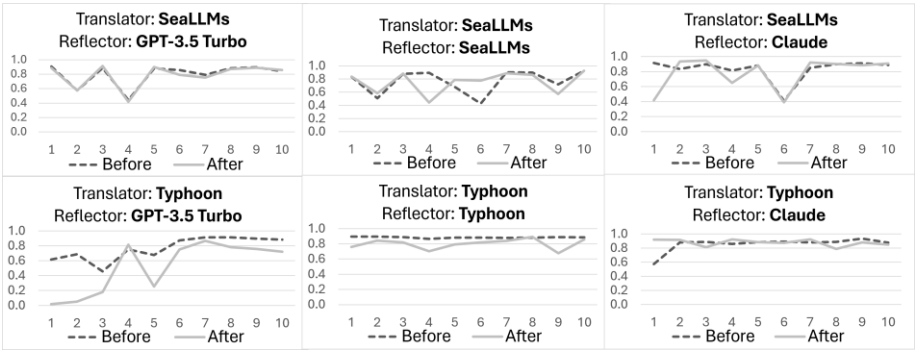

Fig. 5: The COMET score comparing before and after implementing an agentic flow of SeaLLMs and Typhoon with different reflector LLMs

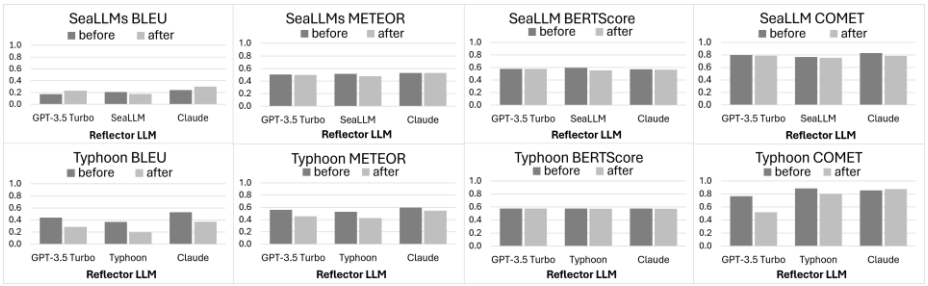

Fig. 6: Average translation quality score of SeaLLMs and Typhoon before and after implementing an agentic flow

The graph in Fig. 5 shows the high-quality translation ability of SeaLLMs and Typhoon in the COMET score, which is designed to assess the text generated from LLMs by predicting the next words. Furthermore, almost the score after was lower than the score before and after implementing agentic flow. As a result, this experiment does not confirm that agentic machine translation can enhance LLM-based translation accuracy. Moreover, the score after using agentic machine translation each time is almost lower than before using agentic flow. However, the translation quality score is increasing with Typhoon as translator LLM and Claude as reflector LLM, as demonstrated in Fig. 6. On the other hand, the lowest score is using GPT-3.5 Turbo as reflector LLM when comparing with SeaLLMs and Typhoon, which pre-train with specific languages such as Southeast Asian or Thai. However, GPT-3.5 Turbo is still lower than Claude, which pre-trains with multilingual language.

5 Conclusion and Future Work

Although this experiment was not to confirm that the agentic machine translation can enhance LLMs-based translation accuracy. Because there are other translation factors, such as tokenizer, which is used for word separation. However, this experiment showed

the ability of LLMs which can suggest enhancing translation by accuracy, fluency, style, and terminology. Moreover, the specific languages pre-trained LLMs have more efficiency in translation than multilingual pre-trained LLMs, and we can use LLMs to enhance translation by specific languages pre-trained LLMs, such as Typhoon. Nonetheless, this experiment showed there is a challenge in leveraging LLMs-based translation instead of machine translation to improve translation in low-resource languages.

References

1. Banerjee, S., Lavie, A.: Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In: Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization. pp. 65–72 (2005)
2. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Nee-lakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
3. Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., et al.: Palm: Scaling language modeling with pathways. Journal of Machine Learning Research **24**(240), 1–113 (2023)
4. Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V.: Unsupervised cross-lingual representation learning at scale (2020), https://arxiv.org/abs/1911.02116
5. Conneau, A., Lample, G., Rinott, R., Williams, A., Bowman, S.R., Schwenk, H., Stoyanov, V.: Xnli: Evaluating cross-lingual sentence representations. arXiv preprint arXiv:1809.05053 (2018)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
7. Emmanuel, C.: Gpt-3.5 and gpt-4 comparison: Exploring the developments in ai-language models (2023), https://medium.com/@chudeemmanuel3/gpt-3-5-and-gpt-4-comparison-47d837de2226, accessed: 2024-07-01
8. Grammarly: Claude ai 101: What it is and how it works (2024), https://www.grammarly.com/blog/what-is-claude-ai, accessed: 2024-07-01
9. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification (2018), https://arxiv.org/abs/1801.06146
10. IBM: What are large language models (llms)? (2024), https://www.ibm.com/topics/large-language-models, accessed: 2024-07-01
11. Kedtiwerasak, R., Adsawinnawanawa, E., Jirakunkanok, P., Kongkachandra, R.: Thai keyword extraction using textrank algorithm. In: 2019 14th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP). pp. 1–6. IEEE (2019)
12. Majumder, P., Mitra, M., Chaudhuri, B.: N-gram: a language independent approach to ir and nlp. In: International conference on universal knowledge and language. vol. 2 (2002)
13. Moslem, Y., Haque, R., Kelleher, J.D., Way, A.: Adaptive machine translation with large language models. arXiv preprint arXiv:2301.13294 (2023)
14. Ng, A.: Translation agent (2024), https://www.linkedin.com/posts/andrewyng_github-andrewyngtranslation-agent-activity-7206347897938866176-5tDJ, accessed: 2024-07-04
15. Nguyen, X.P., Zhang, W., Li, X., Aljunied, M., Tan, Q., Cheng, L., Chen, G., Deng, Y., Yang, S., Liu, C., et al.: Seallms–large language models for southeast asia. arXiv preprint arXiv:2312.00738 (2023)

16. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the Association for Computational Linguistics. pp. 311–318 (2002)
17. Pipatanakul, K., Jirabovonvisut, P., Manakul, P., Sripaisarnmongkol, S., Patomwong, R., Chokchainant, P., Tharnpipitchai, K.: Typhoon: Thai large language models. arXiv preprint arXiv:2312.13951 (2023)
18. Ponti, E.M., Glavaš, G., Majewska, O., Liu, Q., Vulić, I., Korhonen, A.: Xcopa: A multilingual dataset for causal commonsense reasoning. arXiv preprint arXiv:2005.00333 (2020)
19. Rei, R., Stewart, C., Farinha, A.C., Lavie, A.: Comet: A neural framework for mt evaluation. arXiv preprint arXiv:2009.09025 (2020)
20. Shavit, Y., Agarwal, S., Brundage, M., Adler, S., O'Keefe, C., Campbell, R., Lee, T., Mishkin, P., Eloundou, T., Hickey, A., et al.: Practices for governing agentic ai systems. Research Paper, OpenAI, December (2023)
21. Tang, G., Yousuf, O., Jin, Z.: Improving bertscore for machine translation evaluation through contrastive learning. IEEE Access **12**, 77739–77749 (2024). https://doi.org/10.1109/ACCESS.2024.3406993
22. UK Parliament: Communications and digital committee (2024), https://publications.parliament.uk/pa/ld5804/ldselect/ldcomm/54/5402.htm, accessed: 2024-06-30
23. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need (2023), https://arxiv.org/abs/1706.03762
24. Wei, J., Bosma, M., Zhao, V.Y., Guu, K., Yu, A.W., Lester, B., Du, N., Dai, A.M., Le, Q.V.: Finetuned language models are zero-shot learners. arXiv preprint arXiv:2109.01652 (2021)
25. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: Bertscore: Evaluating text generation with bert. arXiv preprint arXiv:1904.09675 (2019)
26. Zhang, W., Aljunied, S.M., Gao, C., Chia, Y.K., Bing, L.: M3exam: A multilingual, multimodal, multilevel benchmark for examining large language models (2023), https://arxiv.org/abs/2306.05179
27. Zhu, W., Liu, H., Dong, Q., Xu, J., Huang, S., Kong, L., Chen, J., Li, L.: Multilingual machine translation with large language models: Empirical results and analysis. arXiv preprint arXiv:2304.04675 (2023)