Grammatical Error Correction of Thai L2-Learner Text with Self-Instruct Data Augmentation

Anonymous ACL submission

Abstract

Grammatical error correction (GEC) for Thai has become increasingly important as the number of Thai language learners has grown in recent years; however, research on Thai GEC has not kept pace with this demand. A key open question is whether data augmentation techniques that have proven effective for English can be successfully applied to low-resource languages like Thai, which lack large-scale annotated corpora. In this work, we present the CTFL-Corpus, the first annotated dataset for Thai grammatical error correction, and propose a self-instruction–based data augmentation framework to address data scarcity. Unlike corpus-dependent augmentation methods, our self-instruct approach enables the generation of diverse error patterns without relying on a limited parallel corpus, thereby improving coverage of learner error types. Our experiments show that round-trip translation, commonly used in English GEC systems, performs poorly in the Thai setting, while the proposed self-instruction framework yields substantial performance gains. We further find that supervised encoder–decoder architectures consistently outperform proprietary zero-shot LLM under low-resource Thai GEC setting.

1 Introduction

The demand for learning Thai as a second language is increasing, as indicated by the widespread presence of Thai language teaching institutes and courses offered by universities, private organizations, and international institutions. However, mastering Thai presents varying challenges for second-language learners due to its tonal nature, script structure, and grammatical rules. Learners often struggle with various types of errors, including grammatical errors (e.g., incorrect word order), word choice issues, and spelling errors. For instance, a single tonal mark can drastically alter the meaning of a word, as seen in ไม่ใช่เวลา /mâj t͡ɕʰâj wēːlāː/ meaning "not the time," can be miswritten as ไม่ใช้เวลา /mâj t͡ɕʰáj wēːlāː/ meaning "does not take time." This is where the Grammatical Error Correction (GEC) system becomes crucial, offering learners essential support in identifying and correcting such mistakes.

Recent work in GEC has primarily advanced through pre-trained transformer models for English (Omelianchuk et al., 2020; Rothe et al., 2021; Xue et al., 2021) and other languages (Xue et al., 2021; Palma Gomez et al., 2023; Náplava and Straka, 2019), but Thai remains under-studied. To the best of our knowledge, there are no established Thai GEC corpora or benchmarks comparable to those available for English. To address this gap, we build the CTFL-GEC corpus and introduce a Self-Instruct-inspired synthetic data pipeline that generates diverse, grammar-focused Thai error–correction pairs. We evaluate both encoder–decoder and decoder-only model families under realistic low-resource conditions and compare augmentation strategies (self-instruct vs. round-trip translation) using CHRF as our primary metric.

We create the CTFL-GEC Corpus to address the complete lack of annotated resources for Thai Grammatical Error Correction (GEC). As the first and largest GEC corpus for Thai, it contains 4,047 sentences written by L2 learners, segmented from 370 writing samples. Native Thai linguistics students manually annotated the data using a minimal-edit policy to support efficient training of sequence-to-sequence models (Lichtarge et al., 2019; Kementchedjhieva and Søgaard, 2023). We categorized each error into one of four types. The corpus includes 8,583 error tags categorized into four types —Grammar, Spelling, Punctuation, and Fluency—based on the UA-GEC framework (Syvokon and Nahorna, 2022). This resource provides an important foundation for Thai GEC research, enabling both model development and standardized

evaluation in this low-resource language context.

Given the limited size of the manually annotated CTFL-GEC corpus, we investigate whether synthetic data augmentation can effectively improve Thai GEC under low-resource conditions. We compare two augmentation strategies: round-trip translation (RTT), which has been widely used in prior GEC work (Lichtarge et al., 2019; Kementchedjhieva and Søgaard, 2023), and a Self-Instruct-inspired generation framework that produces grammar-focused Thai correction pairs (Wang et al., 2023). Our experiments show that synthetic data quality is critical in Thai GEC: while instruction-generated examples substantially improve encoder–decoder performance, RTT-based augmentation severely degrades model performance by introducing noisy and semantically distorted corrections.

Our contribution can be summarized as follows:

- We introduce CTFL-GEC, the first manually annotated Thai grammatical error correction corpus for L2 learner text, including both corrected and error-tagged versions for research and benchmarking.
- We investigate synthetic data augmentation for low-resource Thai GEC and show that our proposed instruction-generated synthetic pairs substantially improve performance, while round-trip translation severely degrades model quality in the Thai setting.
- We provide a comparative evaluation of encoder–decoder and decoder-only approaches for Thai GEC and analyze their correction behavior across multiple error categories.

2 Related Work

GEC has progressed from rule-based and SMT approaches to neural encoder–decoder and Transformer-based models (e.g., (Chollampatt and Ng, 2017; Vaswani et al., 2017; Rothe et al., 2021)), with multilingual variants such as mT5 effective for low-resource languages (Xue et al., 2021). Decoder-only LLMs show strong zero-shot capabilities, but encoder–decoder models typically perform better when fine-tuned for GEC (Wang et al., 2022; Fu et al., 2023). Recent Thai-focused models include OpenThaiGPT and Typhoon (Viriyayudhakorn et al., 2024; Pipatanakul et al., 2023).

The focus of GEC corpora and studies has largely been on English, which benefits from abundant annotated corpora and benchmarks such as the CoNLL Shared Tasks (Ng et al., 2013, 2014) and the BEA Workshop datasets (Bryant et al., 2019). These resources have enabled the development of state-of-the-art models and evaluations (Bryant et al., 2019; Grundkiewicz et al., 2019) while most other languages have much smaller datasets (Tajiri et al., 2012; Daudaravicius et al., 2016); minimal-edit annotation is standard for instructional GEC corpora, whereas fluency edits prioritize stylistic rewriting (Yannakoudakis et al., 2011; Napoles et al., 2017). Research on Thai L2 learners is limited; existing studies report frequent grammatical omissions, word-choice errors, and L1 interference (e.g., (Lu, 2012; Bannatham, 2012; Saengawoot, 2020)), and Thai's tonal orthography and script amplify orthographic and segmentation challenges.

Synthetic data generation or data augmentation has become an essential step for improving GEC systems in low-resource settings. Common strategies include rule-based error synthesis (Choe et al., 2019; Awasthi et al., 2019), round-trip translation (RTT) using NMT to inject realistic grammatical noise, and model-based generation (e.g., Self-Instruct prompts (Park et al., 2024; Wang et al., 2024)) to produce synthetic correction pairs; these methods boost robustness but require careful quality control since RTT can alter semantics and poor MT or unconstrained model generation can degrade training data quality (Choe et al., 2019; Lichtarge et al., 2019; Wang et al., 2023; Park et al., 2024). However, the effectiveness of these augmentation strategies remains underexplored for Thai and other low-resource Southeast Asian languages.

3 Thai GEC Corpus Creation

The raw data for the Thai GEC Corpus is drawn from the CTFL Corpus, which consists of 605 writing pieces by Thai L2 learners from the Center for Thai as a Foreign Language, Faculty of Arts, Chulalongkorn University. A total of 370 writing pieces were segmented into individual sentences using an automatic sentence tokenizer to facilitate further processing. To ensure the accuracy and quality of the data, the corpus was manually corrected and annotated by native Thai students with a background in linguistics. The details of the error types, the data annotation process, and the descriptive statistics of the data set will be explained in the following subsections. We classify errors in

the Thai GEC corpus into four primary categories: Grammar, Spelling, Punctuation, and Fluency. This categorization, inspired by the UA-GEC framework (Syvokon and Nahorna, 2022), allows systematic annotation for Thai GEC.

We adopt a minimal edit approach for error correction, ensuring that changes to the text were as limited as possible while maintaining clarity and accuracy. The corrected errors were tagged using the UA-GEC format, which employs in-text notation in the structure: *{error=>edit:::Tag}*. In this format, "error" represents the text item before correction, "edit" indicates the corrected form, and "Tag" specifies the corresponding error category. Table 7 in the appendix presents annotated example sentences for each high-level error category, demonstrating the practical application of this method.

We enroll two annotators who are native Thai speakers with a background in linguistics at the undergraduate level. The annotation process is divided into three main stages. First, a data overview is conducted, during which annotators review the data set and remove unnecessary texts. Second, annotators proceed with error correction, addressing grammatical, spelling, punctuation, and fluency errors. Third, corrected data are annotated with error tags to classify each identified issue.

During the annotation process, each sentence is independently annotated by two annotators before adjudication. We measure agreement at the sentence level by comparing whether the two annotators produce identical correction and tagging decisions prior to adjudication. The annotators achieve an overall agreement rate of 92%. However, agreement is substantially lower for the distinction between Grammar and Fluency categories, with agreement decreasing to 51.1% for sentences involving these edits. This ambiguity primarily arises because omission or insertion of function words in Thai may reflect either a grammatical violation or a stylistic fluency preference. An example of this ambiguity is shown in Table 1. Final adjudication is performed by the researchers to ensure consistency across the corpus.

The CTFL-GEC corpus is available in two versions. The first version, CTFL-GEC (untagged), consists of original sentences written by Thai L2 learners. This dataset contains 4,219 sentences derived from 370 writing pieces, with an average sentence length of 12 words. This dataset is cleaned and split into training, development, and test subsets at the document level to prevent sentences from the same writing piece from appearing across multiple splits. This setup reduces topic and stylistic leakage between training and evaluation data. The cleaned dataset contains 4,047 rows, with 26.14% of sentences requiring no edits. The training set comprises 90% of the data (3,642 sentences), and the development and test sets each account for approximately 5% (202 and 203 sentences, respectively). The proportion of unedited sentences is balanced across all subsets.

The second version, CTFL-GEC (tagged), includes error annotations and comprises 3,975 annotated sentences. This difference in size is due to the two-stage annotation process, in which correction and tagging are performed separately. This version contains a total of 8,583 error tags, distributed in four categories. Table 2 shows the distribution of errors in each category. Note that the tagged version was not used for model training or evaluation; it is provided for linguistic analysis and corpus documentation purposes.

Edit Stage	Example
Original	ฉัน ชอบ เดินป่า ไป จุดชมวิว 1SG like hike go viewpoint I like hike go viewpoint "I like hiking to the viewpoint."
Corrected	ฉัน ชอบ เดินป่า และ ไป จุดชมวิว 1SG like hike and go viewpoint I like hike and go viewpoint "I like to hike and go to the viewpoint."
Annotation A	ฉัน ชอบ เดินป่า {=>และ:::Grammar} ไป จุดชมวิว 1SG like hike {=>and:::Grammar} go viewpoint I like hike {=>and:::Grammar} go viewpoint
Annotation B	ฉัน ชอบ เดินป่า {=>และ:::Fluency} ไป จุดชมวิว 1SG like hike {=>and:::Fluency} go viewpoint I like hike {=>and:::Fluency} go viewpoint

Table 1: Example of Annotation Disagreement Between Grammar and Fluency Categories. Annotation A labels the omission of “and” as a grammatical error, while Annotation B considers it a fluency issue, reflecting subjective interpretation in error categorization.

Error Type	#Error Tags	Percentage (%)
Grammar	816	9.5
Spelling	1,087	12.7
Fluency	3,106	36.2
Punctuation	3,574	41.6

Table 2: Distribution of Error Types in the CTFL-GEC (Tagged) Corpus

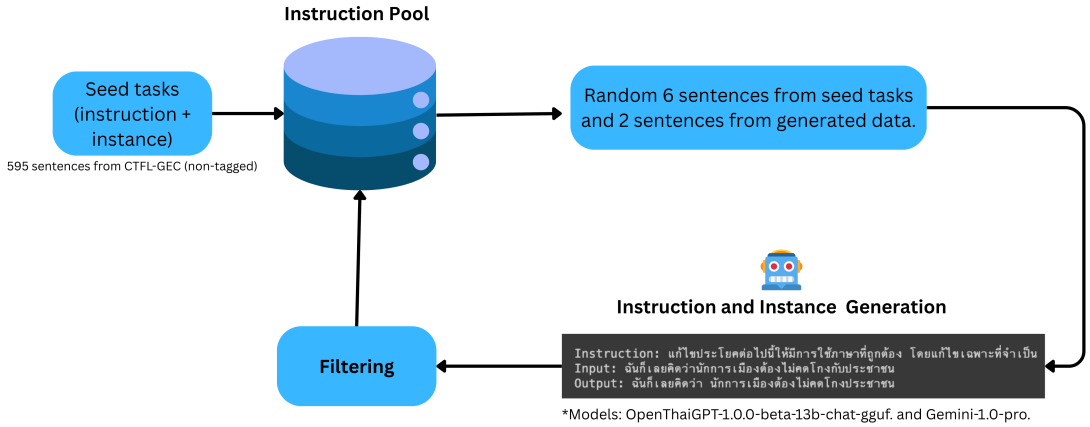

Figure 1: Overview of Self-Instruct dataset generation for Thai GEC. A small set of CTFL-GEC examples is used to prompt a language model to generate grammar-focused correction pairs.

4 Our Approach

We adopt an instruction-based, Self-Instruct-style pipeline to generate synthetic Thai correction pairs from a small curated seed of human-annotated examples and bootstrap these pairs with LLMs. We then fine-tune a text-to-text encoder–decoder model (mT5) on mixtures of gold and synthetic pairs. Encoder–decoder models are a natural fit for this setup because they explicitly model the conditional mapping from an erroneous source to a corrected target, support token-level supervision and copy/edit behaviors through encoder–decoder attention, and align directly with T5-style pretraining (text-to-text and span corruption). These properties make mT5 well-suited to learn minimal, source-faithful edits produced by Self-Instruct generation while preserving contextual and syntactic structure.

To generate high-quality synthetic data, we propose a Thai-specific instruction-based generation pipeline inspired by the Self-Instruct method (Wang et al., 2023). We generate instruction data specifically for Thai grammar correction, focusing on pairs of input and output sentences. To initiate the process, we created a set of seed instructions using 595 sentences from the CTFL-GEC (untagged) dataset, concentrating on grammar and fluency errors. Using this seed set as examples, we prompted a large language model (LLM) to generate more instruction data. We iteratively expand the data by bootstrapping (Figure 1). Each iteration included eight examples, six from the CTFL dataset and two from previously self-instructed data, to ensure variety and maintain the model's ability to produce fresh and diverse instructions. Table 9 in the appendix illustrates the exact prompt format used.

5 Experimental Setup

We compare encoder-decoder mT5 variants to decoder-only LLM baselines under matched data regimes. As baselines we include zero-shot decoder-only models (GPT-3.5-Turbo, Gemini, OpenThaiGPT, Typhoon) and QLoRA-finetuned open models. For augmentation baselines we evaluate round-trip translation (RTT). We use SCB-MT-EN-TH-2020 machine translation model trained a large-scale English-Thai machine translation (Lowphansirikul et al., 2021) to generate RTT pairs.

All encoder–decoder models (`mT5-small`, `mT5-base`, and `mT5-large`) were fine-tuned using the AdamW optimizer with a learning rate of 2×10^{-5} and a weight decay of 0.01. Training was conducted with a batch size of 16 for 30 epochs without early stopping. During inference, we used greedy decoding with a maximum sequence length of 128 tokens. All experiments were conducted on a single NVIDIA A100 GPU (40GB VRAM) via Google Colab. Hyperparameters were kept consistent across model sizes to facilitate direct comparison.

For decoder-only models (`OpenThaiGPT-13B` and `Typhoon-7B`), we employed QLoRA-based parameter-efficient fine-tuning with 4-bit `nf4` quantization and double quantization enabled. We applied LoRA adapters to all linear layers with

rank $r = 8$, scaling factor $\alpha = 32$, and dropout rate of 0.1. Models were trained using the `paged_adamw_8bit` optimizer with a learning rate of 2×10^{-4}, weight decay of 0.001, and an effective batch size of 4 achieved through gradient accumulation. Training was capped at one epoch with a maximum of 101 update steps and 10 warmup steps. During inference, we used greedy decoding with a maximum sequence length of 2,048 tokens. Gradient checkpointing and DeepSpeed Stage 2 were enabled to reduce memory usage, and all experiments were conducted on a single NVIDIA A100 GPU (40GB VRAM).

To study scale effects we report experiments with three main data conditions: (1) gold-only (the manually corrected CTFL-GEC corpus), (2) gold + RTT (CTFL plus 7K RTT pairs), and (3) gold + SI at increasing sizes (roughly +5K, +10K and +14K synthetic pairs). Models are trained/fine-tuned under the same regimes for fair comparison; SI data is applied incrementally to measure marginal gains. Evaluation uses CHRF as the primary metric, alongside F0.5, BLEU, and GLEU for comparability with prior GEC work. We prioritize CHRF because Thai does not employ explicit whitespace segmentation between words and the automatic word segmentation does not perform well on ungrammatical text, making token-level evaluation sensitive to segmentation inconsistencies and hard to interpret. We report F0.5 additionally because it remains a widely used metric in grammatical error correction benchmarks.

6 Results and Discussion

Among the LLMs, Gemini-1.0-Pro achieves the highest performance, with a CHRF score of 0.81 and 0.43 F0.5 (Table 4). However, fine-tuning decoder-only models, including OpenThaiGPT-1.0-13B and Typhoon-7B, does not lead to significant improvements over their zero-shot performance. These findings suggest that the evaluated small opensource LLMs are less effective under the current low-resource training setup when fine-tuned with a limited amount of supervised data.

mT5-base and mT5-large show the strongest gains when fine-tuned with Self-Instruct synthetic data, each achieving a CHRF score of 0.87. In contrast, mT5-small shows no improvement, maintaining a CHRF score of 0.85 at all augmentation levels (Figure 2). This suggests that the benefits of instruction-based synthetic data are more pronounced for larger models, which can better leverage the additional training signal. All fine-tuned mT5 variants outperform the best zero-shot decoder-only model (Gemini-1.0-Pro). Fine-tuned open-source decoder-only models showed little gain over zero-shot and often produced low-precision, high-recall outputs (overcorrections or hallucinations). This suggests the importance of model architecture and training strategy in Thai GEC, with encoder-decoder models better suited to learn from synthetic data in our experiments.

Model	P	R	F1	F0.5	BLEU	GLEU	CHRF
Encoder-Decoder Models (mT5-small)							
MT5-small-FT	0.36	0.27	0.31	0.34	0.61	0.61	0.85
MT5-small-RTT	0.07	0.53	0.12	0.08	0.00	0.01	0.07
MT5-small-SI	0.37	0.29	0.32	0.35	0.62	0.62	0.85
MT5-small-SI-150%	0.37	0.28	0.32	0.35	0.62	0.65	0.85
MT5-small-SI-200%	0.38	0.28	0.32	0.35	0.63	0.63	0.85
MT5-small-SI-300%	0.38	0.29	0.33	0.36	0.62	0.62	0.85
Encoder-Decoder Models (mT5-base)							
MT5-base-FT	0.41	0.38	0.39	0.40	0.64	0.64	0.86
MT5-base-RTT	0.09	0.64	0.16	0.11	0.00	0.01	0.06
MT5-base-SI	0.44	0.41	0.43	0.44	0.66	0.66	**0.87**
MT5-base-SI-150%	0.42	0.40	0.41	0.42	0.65	0.65	**0.87**
MT5-base-SI-200%	0.45	0.43	0.44	0.44	0.66	0.66	**0.87**
MT5-base-SI-300%	0.42	0.39	0.40	0.41	0.64	0.64	0.86
Encoder-Decoder Models (mT5-large)							
MT5-large-FT	0.26	0.37	0.30	0.28	0.54	0.53	0.82
MT5-large-SI	0.46	0.44	0.45	0.45	**0.69**	**0.68**	**0.87**
MT5-large-SI-150%	0.44	0.45	0.45	0.45	0.68	**0.68**	**0.87**
MT5-large-SI-200%	**0.47**	0.47	**0.47**	**0.47**	**0.69**	**0.68**	**0.87**
MT5-large-SI-300%	0.40	0.44	0.42	0.41	0.67	0.66	0.86
Decoder-Only Models							
Gemini-1.0-Pro-ZS	0.41	0.54	**0.47**	0.43	0.62	0.62	0.81
GPT-3.5-Turbo-ZS	0.20	0.27	0.23	0.21	0.49	0.48	0.74
OpenThaiGPT-1.0-13B-ZS	0.12	0.40	0.19	0.14	0.23	0.26	0.52
Typhoon-7B-ZS	0.11	0.11	0.11	0.11	0.15	0.30	0.48
OpenThaiGPT-1.0-13B-FT	0.06	**0.77**	0.12	0.08	0.08	0.09	0.51
Typhoon-7B-FT	0.07	**0.77**	0.13	0.08	0.10	0.11	0.52
OpenThaiGPT-1.0-13B-SI	0.06	0.73	0.11	0.07	0.05	0.11	0.54
Typhoon-7B-SI	0.06	0.75	0.12	0.08	0.09	0.02	0.48

Table 3: Evaluation results of decoder-only and encoder-decoder models on Thai GEC using multiple metrics. Suffixes—ZS = zero-shot; FT = fine-tuned on CTFL-GEC; SI = fine-tuned; RTT = round-trip translation.

Beyond CHRF, the `mT5-large` model trained on the CTFL-GEC dataset augmented with 9,686 instruction-generated sentence pairs achieves the highest scores across most evaluation metrics (Table 3). These results highlight that instruction-based synthetic data are most effective when combined with larger capacity models.

Although self-instruct augmentation improves model performance, scaling the synthetic dataset beyond a moderate size yields diminishing returns. As shown in Figure 3, mT5-base and mT5-large both reach their maximum CHRF scores of 0.87 with 100–200% Self-Instruct data (4,843 to 9,686 rows). Increasing the size of the dataset further to 300% (14,529 rows) does not improve performance and in some cases leads to a slight decline. For example, the CHRF of mT5-large decreased

Model	CHRF
Zero-shot (ZS)	
Gemini-1.0-Pro-ZS	**0.81**
GPT-3.5-Turbo-ZS	0.74
OpenThaiGPT-1.0-13B-ZS	0.52
Typhoon-7B-ZS	0.48
Fine-tuning (FT)	
OpenThaiGPT-1.0-13B-FT	0.51
Typhoon-7B-FT	0.52
Self-Instruct (SI)	
OpenThaiGPT-1.0-13B-SI	0.54
Typhoon-7B-SI	0.48

Table 4: Fine-tuning and synthetic data yield limited improvements over zero-shot decoder-only baselines.

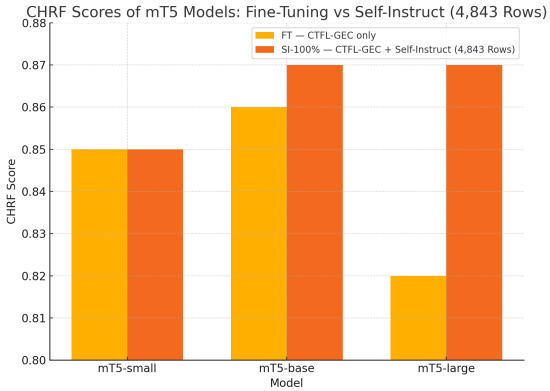

Figure 2: CHRF comparison of mT5 fine-tuning with and without Self-Instruct augmentation.

from 0.87 to 0.86, and its F1 score decreased from 0.47 to 0.42. These results suggest that excessive augmentation may introduce noise or reduce the signal-to-noise ratio in training.

We also compared self-instruct augmentation with the round-trip translation approach, but the results suggest limited value of round-trip translation data for model fine-tuning. Models trained on round-trip translation data achieved an average CHRF score of only 0.06, indicating substantial performance degradation. This suggests that low-quality synthetic pairs, such as those generated via round-trip translation, negatively impact model learning and reduce the effectiveness of a manually curated, small dataset. These findings reinforce the importance of high-quality synthetic data in GEC tasks and caution against using poorly aligned translation-based augmentation techniques. Table 3 in the appendix provides a comprehensive view of the performance of the model in all experimental configurations.

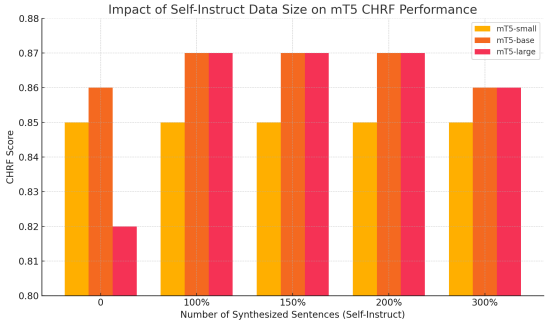

Figure 3: Impact of Self-Instruct data size on CHRF performance of mT5 models.

7 Error Analysis

In addition to quantitative evaluation, this section presents a qualitative analysis of errors and examines key factors that influence performance. For this analysis, we selected models that represent each core experimental approach. Specifically, we chose mT5-large-FT and mT5-large-SI to contrast the effects of Self-Instruct augmentation against standard fine-tuning. Gemini-1.0-Pro-ZS was selected as the strongest zero-shot decoder-only baseline, while OpenThaiGPT-1.0-13B-SI represents a fine-tuned open-source decoder-only model. These selections provide a balanced view of how architecture and training strategy influence correction quality in Thai GEC.

To quantify model performance across distinct linguistic dimensions, we report the recall scores in Table 5 which categorized errors into five specific dimensions: *Grammar*, *Fluency*, *Spelling*, *Punctuation*, and *No-Edit* stability. This metric measures the proportion of errors successfully corrected by the model relative to the total number of errors annotated in the test set for each category.

Model	No Edit	G	F	S	P
mT5-large-FT	0.70	0.06	0.06	0.11	0.18
mT5-large-SI	**0.83**	**0.42**	**0.21**	**0.31**	**0.40**
Gemini-1.0-Pro	0.50	0.13	0.13	0.22	0.32
OpenThaiGPT-13B	0.15	0.00	0.00	0.02	0.01
OpenThaiGPT-13B-SI	0.02	0.00	0.00	0.00	0.01

Table 5: Recall scores by error type: No Edit (stability), Grammar (G), Fluency (F), Spelling (S), Punctuation (P).

The breakdown of recall scores reveals a clear hierarchy in task difficulty. Across all models, Fluency errors remained the most challenging to correct. Even the best-performing model (mT5-large-SI) achieved a recall of only 0.21 for such cases, compared to 0.42 for Grammar. This sug-

gests that while models can more effectively learn rigid syntactic rules, they struggle with the subtle semantic shifts required for stylistic fluency in Thai. Crucially, the comparison highlights that Self-Instruct (SI) augmentation is the key driver of performance, boosting Grammar recall by 0.36 absolute points (from 0.06 to 0.42) compared to standard fine-tuning.

The models vary in their correction behavior (Table 6). For example, the original input lacks a linking verb and the annotated correction introduced a copula to improve clarity. Both variants mT5-large fail to correct the error, but preserve the meaning of the sentence without distortion. Gemini-1.0-Pro offers an acceptable and fluent alternative phrasing. In contrast, OpenThaiGPT-1.0-13B hallucinates content not related to the original input, and the SI-augmented version generated a factually incorrect and off-topic sentence, demonstrating instability in decoder-only models when fine-tuned for Thai GEC.

7.1 Why is RTT inferior to Self-Instruct for Thai GEC?

To better understand the role of synthetic data, we compare the two augmentation methods: Round-Trip Translation (RTT) and Self-Instruct (SI).RTT-based model outputs frequently exhibited repetitive and incoherent text, such as repeated words or phrases that bore no semantic relation to the input sentence. For instance, given the input "I spent ten-year repay debt," the model collapsed into hallucination, outputting "China China China" (see Figure 5). In many cases, the output was completely unintelligible (such as nonsensical stream of the Thai repetition mark) and failed to preserve the meaning or structure of the original input. This pattern suggests that RTT-generated data for Thai may introduce unnatural noise that hinders rather than helps model learning.

RTT-generated data introduces syntactic and discourse-level variation that disrupts the consistency of the learner-error distribution, making it less suitable for Thai GEC. Unlike English, Thai grammar focuses more on particles, word order, discourse markers, and contextual fluency rather than overt inflectional morphology. As a result, back-translation often produces sentences that are semantically valid but pragmatically unnatural, overly formal, or stylistically inconsistent with authentic learner errors. For example, RTT unnecessarily altered the neutral command "Don't buy this item" into the emphatic warning "Don't you dare buy this thing" (see Figure 4), a shift that confuses the model's objective. This failure stems from RTT's tendency to produce structurally and lexically distorted sentences. Many back-translated outputs were overly formal, literal, or semantically misaligned with the original input. These inconsistencies reduce the coherence of the dataset and make it difficult for models to generalize effectively. When RTT modifies these elements, the resulting sentences no longer resemble authentic learner errors, thereby weakening the model's ability to learn relevant correction patterns.

8 Conclusion

We present the first Thai Grammatical Error Correction (GEC) corpus for L2 learners, which includes both human-annotated corrections and a synthetic dataset. This resource provides a valuable foundation for future research in Thai GEC, enabling the development and evaluation of models trained on authentic and artificially generated data. Our experiments demonstrate that the quality of synthetic augmentation data plays a critical role in low-resource Thai GEC, with instruction-generated data substantially improving model performance while RTT-based augmentation introduces harmful noise.

Our findings indicate that encoder-decoder models, particularly those fine-tuned with self-instruct data, outperform decoder-only architectures in Thai GEC tasks. The superior performance of mT5 models highlights the advantages of bidirectional encoding in capturing grammatical structures more effectively. In contrast, decoder-only models, including fine-tuned open-source models, struggle to generalize well, often introducing overcorrections or hallucinations. Although commercial models like Gemini-1.0-Pro perform decently well in a zero-shot setting, their tendency to make unnecessary modifications limits their reliability for precise grammatical corrections.

Future research should focus on improving decoder-only architectures for Thai GEC by exploring domain-specific pre-training, alternative fine-tuning strategies, or reinforcement learning techniques to improve precision. Furthermore, further investigation of data augmentation strategies is needed to balance dataset size and model stability, particularly in low-resource settings. Our study underscores the critical role of high-quality training

Model	Example	Observation
Original Sentence	หนังสือ นี้ สำหรับ สุขภาพ book DEM for health book this for health *(This book for health.)*	Learner fluency error; lacks copula and natural structure.
Target Correction (CTFL-GEC)	หนังสือ นี้ เป็น หนังสือ เกี่ยวกับ สุขภาพ book DEM be book about health book DEM be book about health *(This book is about health.)*	Adds linking verb and restructures sentence for clarity.
mT5-large-FT / SI	หนังสือ นี้ สำหรับ สุขภาพ book DEM for health book this for health *(This book for health.)*	No correction; avoids hallucination but fails to improve fluency.
Gemini-1.0-Pro	หนังสือ เล่ม นี้ เกี่ยวกับ สุขภาพ book CLF DEM about health book classifier this about health *(This book is about health.)*	Fluent and accurate rephrasing; maintains input meaning.
OpenThaiGPT-1.0-13B	หนังสือ เล่ม นี้ เป็น ประโยชน์ book CLF DEM be useful book classifier this be useful *(This book is useful.)*	Mild hallucination; semantically different but coherent.
OpenThaiGPT-1.0-13B-SI	ฮังการี สำหรับ สุขภาพ สาธารณะ นี้ เป็น ตัวเลือก ที่ ดี Hungary for health public DEM be option REL good Hungary for health public this be option that good *(Hungary is a good option for public health.)*	*Severe hallucination; introduces unrelated and incorrect content.*

Table 6: Examples of Hallucination and Correction Behaviors Across Thai GEC Models

data and model architecture in the advancement of Thai GEC and provides a benchmark for future work in this area.

9 Limitations

Our proposed study establishes strong baselines for Thai GEC, which utilizes synthetic data generation to address the low-resource context. The proposed dataset only presents the first step for developing a Thai GEC system. As the current systems rely on supervised learning, we hope that expanding the dataset would lead to better performance of the future Thai GEC system. Moreover, our proposed method and the experimental setup also depends on a strong LLM performance. The future work might include incorporating more advanced LLMs into the self-instruct framework or the zero-shot performance evaluation.

References

Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. 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 4260–4270, Hong Kong, China. Association for Computational Linguistics.

Panitan Bannatham. 2012. Chinese students and some errors in writing the thai language. Research report, Suan Sunandha Rajabhat University.

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.

Yo Joong Choe, Jiyeon Ham, Kyubyong Park, and Yeoil Yoon. 2019. A neural grammatical error correction system built on better pre-training and sequential transfer learning. *Preprint*, arXiv:1907.01256.

Shamil Chollampatt and Hwee Tou Ng. 2017. Connecting the dots: Towards human-level grammatical error correction. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 327–333, Copenhagen, Denmark. Association for Computational Linguistics.

Approach	Error Sentence (Input)	Corrected Sentence (Target)
SI	แต่ว่า เขา มา ทำไม กัน นะ but-CONJ 3SG come why EMPH Q but-conjunction he come why emphatic particle question particle “But, why did he come?”	แต่ เขา มา ทำไม กัน นะ but 3SG come why EMPH Q but he come why emphatic particle question particle “But, why did he come?”
SI	เด็กๆ ชื่นชอบ การเล่น เกมส์ child-RED like NOM-play game child-reduplication like nominalizer-play game “Children likes playing games.”	เด็กๆ ชื่นชอบ การเล่น เกม child-RED PUNCT like NOM-play game child-reduplication punctuation like nominalizer-play game “Children likes playing games.”
RTT	อย่าซื้อ รายการ นี้ NEG.IMP-buy item DEM do-not-buy item this “Do not buy this item.”	อย่าซื้อ ของ อันนี้ เชียว NEG.IMP-buy thing CLF-DEM EMPH do-not-buy thing classifier-this emphatic particle “Don’t you dare buy this thing!”
RTT	ตัวผลิตภัณฑ์ เอง ก็ เยี่ยม มาก CLF-product self also excellent very classifer-product self also excellent very “The product itself is very excellent.”	ตัวสินค้า เอง มัน เยี่ยม มาก CLF-product self 3SG excellent very classifer-product self it excellent very “The product itself is very excellent.”

Figure 4: Examples from Self-Instruct (SI) and Round-Trip Translation (RTT) Datasets.

Model	Output
Original Sentence	ผม ใช้ เวลา 10ปี (เพื่อ) ชำระ หนี้ 1SG use time ten-year (for) repay debt “I spent 10 years to pay off debt.”
mT5-base-SI	ผม ใช้ เวลา 10 ปี เพื่อ ชำระ หนี้ 1SG use time ten year for repay debt “I spent 10 years to pay off debt.”
mT5-base-RTT	จีน จีน จีน China China China

Figure 5: Example of model output using SI and RTT augmentation

Vidas Daudaravicius, Rafael E. Banchs, Elena Volodina, and Courtney Napoles. 2016. A report on the automatic evaluation of scientific writing shared task. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 53–62, San Diego, CA. Association for Computational Linguistics.

Zihao Fu, Wai Lam, Qian Yu, Anthony Man-Cho So, Shengding Hu, Zhiyuan Liu, and Nigel Collier. 2023. Decoder-only or encoder-decoder? interpreting language model as a regularized encoder-decoder. *Preprint*, arXiv:2304.04052.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. Neural grammatical error correction systems with unsupervised pre-training on synthetic data. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 252–263, Florence, Italy. Association for Computational Linguistics.

Yova Kementchedjhieva and Anders Søgaard. 2023. Grammatical error correction through round-trip machine translation. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2208–2215, Dubrovnik, Croatia. Association for Computational Linguistics.

Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. Corpora generation for grammatical error correction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3291–3301, Minneapolis, Minnesota. Association for Computational Linguistics.

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.

Ye Lu. 2012. Error analysis of using interrogative sentences of chinese students learning thai as a foreign language. Master’s thesis, Chiang Mai University, Chiang Mai, Thailand.

Jakub Náplava and Milan Straka. 2019. Grammatical error correction in low-resource scenarios. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 346–356, Hong Kong, China. Association for Computational Linguistics.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. JFLEG: A fluency corpus and benchmark for grammatical error correction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 229–234, Valencia, Spain. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The CoNLL-2013 shared task on grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared*

Task, pages 1–12, Sofia, Bulgaria. Association for Computational Linguistics.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. GECToR – grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA → Online. Association for Computational Linguistics.

Frank Palma Gomez, Alla Rozovskaya, and Dan Roth. 2023. A low-resource approach to the grammatical error correction of Ukrainian. In *Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP)*, pages 114–120, Dubrovnik, Croatia. Association for Computational Linguistics.

Jeiyoon Park, Chanjun Park, and Heuiseok Lim. 2024. Chatlang-8: An llm-based synthetic data generation framework for grammatical error correction. *Preprint*, arXiv:2406.03202.

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

Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. A simple recipe for multilingual grammatical error correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 702–707, Online. Association for Computational Linguistics.

Siwarin Saengawoot. 2020. An analysing of the errors in writing thai by thai majoring chinese students to study in yunnan province china. *Journal of MCU Social Science Review*, 6(2):133–144.

Oleksiy Syvokon and Olena Nahorna. 2022. Ua-gec: Grammatical error correction and fluency corpus for the ukrainian language. *Preprint*, arXiv:2103.16997.

Toshikazu Tajiri, Mamoru Komachi, and Yuji Matsumoto. 2012. Tense and aspect error correction for ESL learners using global context. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 198–202, Jeju Island, Korea. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Preprint*, arXiv:1706.03762.

Kobkrit Viriyayudhakorn, Sumeth Yuenyong, Thaweewat Rugsujarit, Jillaphat Jaroenkantasima, Norapat Buppodom, Koravich Sangkaew, Peerawat Rojratchadakorn, Surapon Nonesung, Chanon Utupon, Sadhis Wongprayoon, Nucharee Thongthungwong, Chawakorn Phiantham, Patteera Triamamornwooth, Nattarika Juntarapaoraya, Kriangkrai Saetan, and Pitikorn Khlaisamniang. 2024. Openthaigpt 1.0.0. Accessed: 2024-05-10.

Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. 2022. What language model architecture and pretraining objective work best for zero-shot generalization? *Preprint*, arXiv:2204.05832.

Yixuan Wang, Baoxin Wang, Yijun Liu, Qingfu Zhu, Dayong Wu, and Wanxiang Che. 2024. Improving grammatical error correction via contextual data augmentation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10898–10910, Bangkok, Thailand. Association for Computational Linguistics.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. *Preprint*, arXiv:2212.10560.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. *Preprint*, arXiv:2010.11934.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, Oregon, USA. Association for Computational Linguistics.

Appendix

Error Type	Definition	Example
Grammar	Violations of syntactic rules.	ขอบคุณ {=>ที่:::Grammar} เป็น ครู ของ เรา thank {=>COMP:::Grammar} be teacher POSS 1PL thank {=>for:::Grammar} be teacher of us "Thank you for being our teacher."
Spelling	Incorrect word formation due to letter placement errors.	มี ร้านอาหาร {เอยะ=>เยอะ:::Spelling} มาก อยู่ บน ถนน EXIST restaurant {*many=>many:::Spelling} very be.LOC on road have restaurant {*many=>many:::Spelling} very be on road "There are many restaurants on the street."
Punctuation	Misuse or omission of punctuation marks and improper spacing.	ฉัน มา กรุงเทพ เมื่อ ประมาณ {1ปี=>1 ปี:::Punctuation} ก่อน 1SG come Bangkok when about {*1year=>1 year:::Punctuation} before I come Bangkok when about {*1year=>1 year:::Punctuation} before "I came to Bangkok one year ago."
Fluency	Incorrect vocabulary usage.	เรา เป็น มนุษย์ บางครั้ง {ต้อง=>จึง:::Fluency} ทำ ผิด 1PL be human sometimes {must=>so:::Fluency} do wrong we are human sometimes {must=>so:::Fluency} do wrong "We are human, so we make mistake sometimes."

Table 7: Illustrative Examples of Thai GEC Error Types with Definitions and Annotated Sentences, Accompanied by Leipzig Glossing for Translation

Edit Type	Example
Original Sentence	ถ้าปล่อยให้ติดเชื้อทำให้เสียชีวิต 100 % If infected, they die 100%
Minimal Edit	ถ้าปล่อยให้ติดเชื้อจะทำให้เสียชีวิต 100% If infected, they will die 100%
Fluency Edit	หากปล่อยให้ติดเชื้อจะมีโอกาสเสียชีวิตสูง If infected, the chance of death is high.

Table 8: Examples of Minimal and Fluency Edits

```
Create an instruction data as in the following
examples:

Instruction: Correct the following sentence for
grammar. Only make necessary changes.
Input: (original sentence 1)
Output: (corrected sentence 1)

Instruction: (same task instruction)
Input: (original sentence 2)
Output: (corrected sentence 2)

Instruction: (same task instruction)
Input: (original sentence 3)
Output: (corrected sentence 3)

Instruction: (same task instruction)
Input: (original sentence 4)
Output: (corrected sentence 4)

Instruction: (same task instruction)
Input: (original sentence 5)
Output: (corrected sentence 5)

Instruction: (same task instruction)
Input: (original sentence 6)
Output: (corrected sentence 6)

Instruction: (same task instruction)
Input: (original sentence 7)
Output: (corrected sentence 7)

Instruction: (same task instruction)
Input: (original sentence 8)
Output: (corrected sentence 8)
```

Table 9: Prompt template used for Self-Instruct generation