

NJTSC: Advancing Low-Resource Spelling Correction through Neighboring-Language Joint Training and In-Context Learning

Weixiong Zheng¹, YiWei Li², Zhanbo Wang¹, Peijian Zeng^{1†}, Aimin Yang^{1,†}

¹ School of Computer Science and Technology, Guangdong University of Technology

² School of Information Engineering and Artificial Intelligence, Lanzhou University of Finance and Economics

zengpeijian1@mails.gdut.edu.cn

Abstract

Spelling correction plays a crucial role in text preprocessing and language digitalization for low-resource languages, serving as a fundamental task for improving text quality and promoting digital inclusion. Addressing spelling errors in Southeast Asian languages such as Indonesian and Vietnamese is particularly challenging due to limited corpora, high annotation costs, and unique linguistic characteristics. This paper addresses the IMLIP 2025 Shared Task by proposing a Neighboring Languages Joint Training Framework for Spelling Correction (**NJTSC**). Our team, **Jiujiuwo**, introduces a neighboring languages joint training strategy that exploits typological similarities through large language model (LLM)-based translation and direct data incorporation from similar languages. Furthermore, we employ in-context learning (ICL) with structured prompt templates to enrich model inputs and utilize Low-Rank Adaptation (LoRA) for efficient fine-tuning. Experimental results show that NJTSC achieves competitive performance with fewer than 3 billion parameters, demonstrating the effectiveness of our approach for low-resource spelling correction.

Keywords: Spelling Correction; Low-Resource Languages; Cross-Lingual Learning; Parameter-Efficient Fine-Tuning; In-Context Learning

1. Introduction

Spelling correction is a fundamental task in natural language processing (NLP) that helps improve text quality, reduce input noise, and enhance the performance of downstream applications such as word segmentation, named entity recognition (NER), and optical character recognition (OCR)[1, 2]. Moreover, it plays a critical role in content security: malicious content often evades moderation by intentionally introducing spelling errors[3], and effective correction techniques can normalize these variants, significantly improving the accuracy and robustness of sensitive keyword detection systems.

Existing approaches to spelling correction have evolved significantly. For mainstream languages, some researchers focus on developing detection models based on rules, statistics, or neural networks[4, 5], while others leverage pre-trained language models to achieve more accurate

[†]Peijian Zeng and Aimin Yang are co-corresponding authors.

spelling correction by capturing contextual information[6, 7]. Additionally, with the development of large language models (LLMs), some researchers have explored utilizing prompt learning and the powerful capabilities of LLMs[8, 9]. However, for Indonesian and Vietnamese, both Southeast Asian languages, research on spelling correction remains extremely limited. Although NLP technology in this region is generally undergoing rapid development, driven primarily by the proliferation of AI-powered language processing tools and the sustained growth in demand for digital content in local languages, spelling correction as a fundamental task still faces significant challenges. To address these challenges, IMLIP2025 proposed a spelling correction shared task covering Indonesian and Vietnamese, which includes two subtasks: error detection and error correction, aiming to promote a comprehensive and detailed evaluation of models’ capabilities in handling spelling correction at different levels. This paper aims to address the data scarcity and cross-language generalization challenges in this task.

In this study, we address the challenge of spelling correction for low-resource Southeast Asian languages by proposing a Neighboring Languages Joint Training Framework for Spelling Correction (NJTSC). Our method first leverages the powerful translation capabilities of LLMs (e.g., the Qwen3-Max API) to augment training data, such as translating Vietnamese training sets into Indonesian to further expand the dataset. Additionally, we propose a neighboring languages joint training approach to augment training data. Specifically, when addressing spelling correction tasks for a target language, we incorporate training data from typologically similar Southeast Asian languages. Although these languages are not identical, they exhibit certain grammatical and structural similarities that enable effective data augmentation, thereby enhancing cross-lingual knowledge transfer and model generalization. Furthermore, demonstration examples selected through the in-context learning (ICL) method further enrich the model’s inputs. We incorporate the contextual information into a structured prompt template to guide the model’s analysis. To improve the efficiency and adaptability of our model, we employ Low-Rank Adaptation (LoRA), which reduces the number of trainable parameters, thus lowering computational demands. The main contributions of this paper are:

- (1) We propose a neighboring language joint training approach that addresses data scarcity by leveraging LLM-based translation and incorporating data from typologically similar languages, exploiting their shared grammatical and structural patterns to enhance knowledge transfer.
- (2) We employ in-context learning to select contextually relevant demonstration examples, enriching the model’s inputs with additional contextual information to improve spelling error detection and correction accuracy.
- (3) Experimental results demonstrate that despite using models with fewer than 3 billion parameters and limited training samples, NJTSC achieves competitive performance on the IMLIP2025 spelling correction shared task, validating the effectiveness of our parameter-efficient multilingual approach.

2. Related Work

Early research on automatic text correction has largely focused on high-resource languages. UMRSpell[10] was originally developed for Chinese spelling correction, targeting character-level errors (e.g., missing, redundant, or incorrect characters) via a Detection–Transmission Self-Attention mechanism coupled with multi-task learning. ReLM[11] reframes spelling correction as end-to-end sentence rewriting to improve generalization compared to methods based on classification, and was primarily evaluated on languages with standardized tokenization, such as

English. Multi-Head Grammatical Error Correction(MH-GEC)[12], initially designed for Grammatical Error Correction (GEC), uses a multi-head sequence labeling framework that decomposes correction into fine-grained operations, including insertion, deletion, replacement, and transformation, where subtasks like “transformation” are tailored for grammatical phenomena such as tense and voice. BertChecker[13] exemplifies word-level correction using pre-trained models, achieving strong results on English benchmarks (e.g., JFLEG, BEA-4660) by predicting correct words from averaged subword embeddings.

Although these methods work well in their original settings, they suffer from limited error coverage, poor language adaptability, and strong dependence on textual conventions like spacing and tokenization. They struggle to generalize to languages with divergent writing systems and scarce annotations. Spelling correction approaches specifically designed for low-resource Southeast Asian languages, such as Indonesian and Vietnamese, remain extremely limited, highlighting the need for more robust and transferable solutions.

3. Problem Formulation

This shared task focuses on spelling correction for low-resource Southeast Asian languages, comprising two closely related subtasks: spelling error detection and spelling error correction.

The task involves three primary types of spelling errors, as illustrated in Table 1: (1) **Word Replacement** errors, where an incorrect word is substituted (e.g., “monstera” → “monster”); (2) **Redundant Space** errors, where inappropriate spaces split a word into multiple tokens (e.g., “bea siswa” → “beasiswa”); and (3) **Missing Space** errors, where missing spaces merge separate words (e.g., “cinderamata” → “cinder amata”).

Table 1: Examples of three spelling error types.

Error Type	Example
Word Replacement	Incorrect: Ada banyak jenis monstera . Correct: Ada banyak jenis monster .
Redundant Space	Incorrect: Randi mendapat bea siswa ke Tokyo. Correct: Randi mendapat beasiswa ke Tokyo.
Missing Space	Incorrect: Semua cinderamata tersimpan dalam kotak kaca. Correct: Semua cinder amata tersimpan dalam kotak kaca.

We propose an end-to-end unified task formulation that addresses both subtasks within a single framework. Unlike traditional approaches that treat detection and correction as separate tasks, our method formulates the problem as a sequence-to-sequence generation task. Given an input sentence S with potential spelling errors, the model predicts both error positions and their corrections in a structured output format:

$$O = f(S), \quad (1)$$

where O represents the structured output containing error positions and corrections, and f is the prediction function. The output O is formatted as follows:

```
{
  "sentence": "Randi mendapat bea siswa ke Tokyo .",
  "correction": [[[2, 3], "beasiswa"]]
}
```

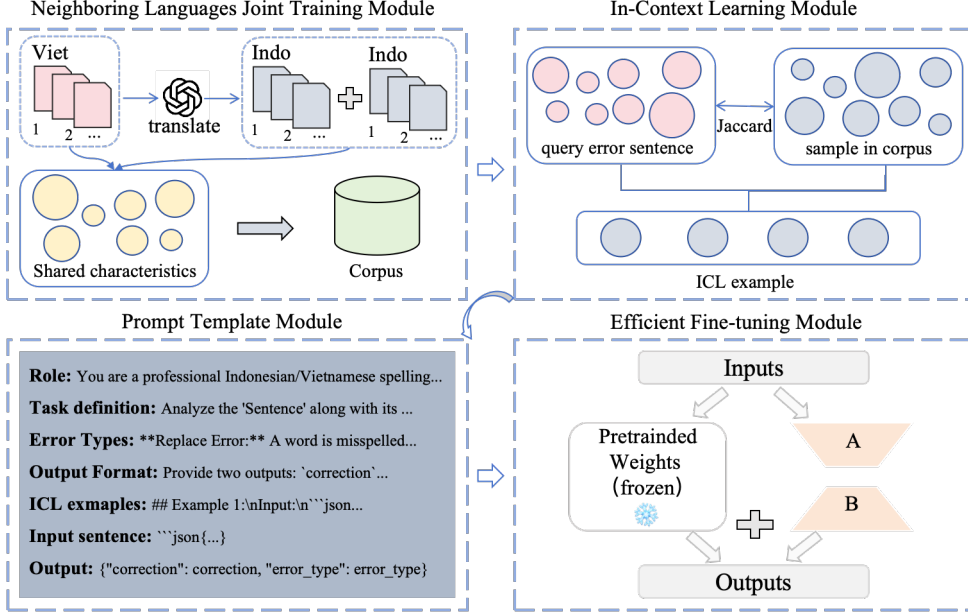


Figure 1: The overall architecture of our work.

4. The Proposed Approach

Our approach utilizes a dual-level data augmentation strategy to exploit typological similarities between Indonesian and Vietnamese. First, we leverage LLM translation capabilities to perform cross-lingual augmentation, translating training samples bidirectionally between the two languages to expand the available data. Second, we directly incorporate the original training sets of both languages into a unified training process, enabling the model to learn from their shared grammatical and structural patterns. Additionally, demonstration examples identified by the ICL method enrich the model’s inputs. We integrate the multilingual training data and contextual information into a structured prompt template that guides the model’s spelling correction process. This template includes task descriptions, error type definitions, demonstration examples, and output format specifications. The model predicts both error positions and corrections in a unified JSON format, facilitating simultaneous detection and correction.

To enhance efficiency, we employ LoRA to achieve parameter-efficient fine-tuning. By formulating the task as sequence-to-sequence generation, our framework enables end-to-end error detection and correction in one pass. The overall framework is depicted in Figure 1.

4.1. Neighboring Languages Joint Training

Addressing data scarcity in low-resource languages presents a fundamental challenge in spelling correction. Traditional monolingual approaches often struggle with limited training samples, leading to poor model generalization and inadequate coverage of diverse error patterns. To overcome this limitation, we propose a neighboring languages joint training strategy that ex-

exploits typological similarities between Indonesian and Vietnamese. Our approach operates at two complementary levels to maximize data efficiency and enable cross-lingual knowledge transfer.

At the first level, we leverage ChatGPT[14] to perform cross-lingual translation. Specifically, we translate the 5,000 Vietnamese training sentences into Indonesian. Crucially, we instruct the LLM to preserve spelling error patterns during translation, maintaining all three error types in the target language. This process effectively doubles the available Indonesian training data to 10,000 sentences.

At the second level, we combine the original and translated data from both languages into a unified multilingual corpus containing 15,000 instances (10,000 Indonesian + 5,000 Vietnamese). Training a single model on this corpus enables the learning of shared error patterns across languages. Both languages exhibit similar error distributions, with word replacement being the most frequent error type. By jointly training on both languages, the model learns cross-lingual representations that facilitate knowledge transfer and improve generalization. This dual-level strategy further expands the training dataset, enhances robustness through diverse error patterns, and enables the model to exploit shared linguistic knowledge between typologically similar languages.

4.2. In-Context Learning

In addressing spelling correction tasks, providing relevant demonstration examples can greatly improve how well large language models identify and correct errors. In this study, we use a method called the Jaccard similarity measure to find example sentences from the training corpus that are similar to the sentence we’re processing. To mitigate the risk of underestimating similarity due to surface-level spelling errors, we compute Jaccard similarity on the gold-standard corrected versions of sentences during retrieval. Specifically, similarity is computed at the word level using the official pre-segmented tokens provided in the dataset. These similar sentences, along with their corrections, are then added to the model’s input to provide additional contextual guidance.

The Jaccard similarity is a common way to compare text, and it works by comparing the tokens in two sentences. Specifically, it looks at how many tokens the two sentences share compared to the total number of tokens in both sentences combined. We calculate this similarity between the tokens in our current input sentence and those in sentences from the training set. The sentences that score the highest are considered the most similar and are used to provide demonstration examples for our model. The application of this approach can be formalized as follows:

$$J(S_i, S_j) = \frac{|T(S_i) \cap T(S_j)|}{|T(S_i) \cup T(S_j)|}, \quad (2)$$

where $J(S_i, S_j)$ represents the Jaccard similarity between sentence S_i and S_j , while $T(S_x)$ denotes the set of tokens extracted from sentence S_x . This measure helps in selecting sentences that share significant tokens with the current input sentence S_i , thereby providing relevant demonstration examples that guide the model’s correction process.

We select the top k sentences with the highest Jaccard similarity scores from the training corpus. These k sentences, along with their corresponding error annotations and corrections, are then incorporated into the structured prompt template. By enriching the input with contextually relevant examples, the model can better understand the error patterns and correction strategies applicable to the current sentence, thereby improving the quality and accuracy of the generated corrections.

```

# Role: You are a professional Indonesian spelling and grammar checker.

# Task: Analyze the 'Sentence' along with its provided 'Word2Position' mapping to determine whether it violates the "Incorrect Sentence" rules.
- The 'input_sentence' refers to a segmented sentence, where each word, number, punctuation mark, and bracket has already been split by a tokenizer and is separated by single spaces.
- Certain patterns such as " word", "( word)", "word )", "word .", "word .", and "word %" (e.g., punctuation or symbols surrounded by proper spaces) are treated as correct.
- Words, numbers, and individual punctuation marks (like commas, periods, parentheses, etc.) being separated into their own components are also correct; do not treat them as issues.
- Base your evaluation entirely on the provided input without seeking external context or evidence.)

# Evaluation Criteria:
1. Analyze the 'input_sentence' sequentially from left to right for the following types of errors.
2. Record each identified error as a separate entry in 'detected_errors'.
3. If there are multiple instances of the same error, list them individually instead of merging them.

## Error Types of "Incorrect Sentence":
1. **Word Replacement:** A word is misspelled, including punctuation-related errors.
2. **Redundant Space:** A word is improperly split into two words due to excessive spacing.
3. **Missing Space:** Two words are incorrectly combined into one due to a lack of proper spacing.

# Output Expectations:
- Provide two outputs: 'correction' and 'error_type' based on the analysis of the input provided information.
1. 'correction': A list of adjustments required to correct the errors. Represent each correction in the format '[[word_positions], "corrected_word"]':
  - 'word_positions': Index or indices of the problematic word(s).
  - 'corrected_word': The corrected form of the word(s) or phrase.
2. 'error_type': A corresponding list of detected error types. You must include one type for each correction.

# Few-Shot Examples:
## Example 1:
Input:
'''json
{
  "Sentence": "Tidak bisa menahan kaget, saya spontan berlari.",
  "Word2Position": [{"Tidak", 0}, {"bisa", 1}, {"menahan", 2}, {"kaget", 3}, {"saya", 4}, {"spontan", 5}, {"berlari", 6}, {".", 7}, {"8}]
}
'''
Output:
'''json
{
  "correction": [[{"7", "berkata"}],
  "error_type": ["Spelling error"]
}
'''
## Example 2:
...

Your turn:
# Input:
(input_sss)

# Output JSON:
'''json
{
  "correction": correction, "error_type": error_type
}
'''

```

Figure 2: The template of prompt.

4.3. Prompt Template Construction

As shown in Figure 2, we employ a structured prompt template to optimize large language model performance on spelling correction. The template explicitly defines the model’s role, task constraints, error taxonomy, and output format—ensuring that corrections are grounded strictly in the provided input and adhere to precise syntactic expectations. This design minimizes ambiguity and hallucination, particularly for nuanced cases such as punctuation-aware spacing and morphological variants. By providing comprehensive, step-by-step instructions and few-shot demonstrations, the design promotes consistency in error detection and correction, while supporting robust training and evaluation across low-resource languages.

4.4. Efficient Fine-tuning

In order to efficiently fine-tune our model for spelling correction in low-resource languages, we utilize Low-Rank Adaptation (LoRA)[15]. LoRA is a technique designed to reduce the number of trainable parameters in large language models, making the fine-tuning process more efficient and less resource-intensive.

LoRA decomposes the weight updates into low-rank matrices, significantly reducing the computational burden. Specifically, given a weight matrix $W_0 \in \mathbb{R}^{d \times k}$, LoRA approximates the update as the product of two lower-rank matrices $A \in \mathbb{R}^{d \times r}$ and $B \in \mathbb{R}^{r \times k}$, where $r \ll \min(d, k)$. The LoRA update can be formulated as:

$$W = W_0 + \Delta W, \quad (3)$$

$$\Delta W = A \times B, \quad (4)$$

where ΔW is the low-rank update applied to the original weight matrix W_0 , and W represents the updated weight matrix post-LoRA. This approach allows for a substantial reduction in the number of parameters, which leads to faster training times and lower memory usage, making it feasible to fine-tune even llms efficiently for tasks such as spelling correction.

5. Experiments and Analysis

5.1. Datasets and Experimental Setup

5.1.1. Datasets

The training dataset comprises 10,000 Indonesian sentences and 10,000 Vietnamese sentences, with an average sentence length of 25.07 words for Indonesian and 43.61 words for Vietnamese. Each sentence contains at least one spelling error. For our experiments, we partition each language dataset into 5,000 sentences for training and 5,000 sentences for validation. To address data scarcity and enable our neighboring language joint training approach, we leverage LLM translation capabilities to perform cross-lingual data augmentation. Specifically, we translate the 5,000 Vietnamese training sentences into Indonesian, thereby creating an additional augmented training set for Indonesian. The official test set contains 50,000 sentences for each language, which is used for final evaluation in the shared task.

5.1.2. Settings

We conduct experiments on an A100 (40GB) GPU using the PyTorch framework¹. For fine-tuning open-source LLMs, we utilize the LLaMA-Factory framework², which supports LoRA-based parameter-efficient fine-tuning. The base model used in our experiments is Qwen3-1.7B[16]. During fine-tuning, we set the batch size to 8, train for 3 epochs, and use a learning rate of 1×10^{-4} .

5.1.3. Metrics

Performance is assessed with three standard metrics: precision (P), recall (R), and F1. Metrics are computed by set-level overlap between system outputs and gold annotations. Let $\mathcal{G} = \{g_1, g_2, \dots, g_n\}$ denote the reference error set and $\mathcal{E} = \{e_1, e_2, \dots, e_n\}$ the predicted error set for a sentence with n errors. The metrics are:

$$P = \frac{\sum_{i=1}^n |e_i \cap g_i|}{\sum_{i=1}^n |e_i|}, \quad (5)$$

$$R = \frac{\sum_{i=1}^n |e_i \cap g_i|}{\sum_{i=1}^n |g_i|}, \quad (6)$$

$$F1 = \frac{2PR}{P + R}. \quad (7)$$

Matching differs by subtask. In error detection, each e_i specifies only the character-level span, and matching requires exact span alignment. In error correction, each e_i includes both the span and replacement text, and a match requires exact agreement in both.

¹<https://github.com/pytorch/pytorch>

²<https://github.com/hiyouga/LLaMA-Factory>

5.2. Overall Comparison Results

5.2.1. Validation Set Performance

Table 2 presents our performance on the validation sets. Our model achieves 88.09% F1 on Indonesian detection and 75.92% on correction (82.01% overall), while Vietnamese obtains 81.63% and 68.28% (74.95% overall). Indonesian consistently outperforms Vietnamese by 6-8 percentage points, which can be attributed to data imbalance: we translated 5,000 Vietnamese sentences into Indonesian but not vice versa, resulting in 10,000 training instances for Indonesian versus only 5,000 for Vietnamese. Detection substantially outperforms correction on both languages (gap 12%), which aligns with expectations as correction requires both accurate localization and exact generation. Given that our approach trains on only 15,000 instances using a single end-to-end model, these results demonstrate the effectiveness of our neighboring languages joint training strategy.

Table 2: Validation set performance on Indonesian and Vietnamese spelling correction.

Language	Error Detection			Error Correction			Overall
	P	R	F1	P	R	F1	
Indonesian	88.37	87.74	88.09	69.66	83.42	75.92	82.01
Vietnamese	82.15	81.12	81.63	62.43	75.33	68.28	74.95

5.2.2. Test Set Performance

Table 3 shows our performance on the official test set (50,000 sentences per language). On Indonesian, we achieve 84.58% F1 on detection and 62.18% on correction (73.38% overall). On Vietnamese, we obtain 61.43% on detection and 25.66% on correction (43.54% overall). The detection results follow patterns consistent with validation: Indonesian outperforms Vietnamese, and detection remains substantially better than correction. However, Vietnamese correction performance (25.66%) appears anomalously low. Analysis revealed this stems from an answer extraction issue in post-processing rather than model failure—outputs were not correctly parsed into the required format due to a post-processing parsing error in the JSON output extraction step. Despite this technical issue, our approach demonstrates competitive performance in the shared task. Our system achieved 3rd place in error detection for both languages and 4th place in Indonesian correction among six participating teams. These rankings are notable given our training constraints: only 5,000 instances per language (plus 5,000 translated instances) with a single 1.7B-parameter model. The competitive detection performance validates that neighboring languages joint training effectively leverages typological similarities for cross-lingual knowledge transfer with limited training data.

Table 3: Final Test set performance on Indonesian and Vietnamese spelling correction.

Language	Error Detection			Error Correction			Overall
	P	R	F1	P	R	F1	
Indonesian	84.11	85.06	84.58	61.84	62.53	62.18	73.38
Vietnamese	63.03	59.91	61.43	26.32	25.02	25.66	43.54

5.3. Ablation Study

Table 4 presents a step-by-step ablation study on the Indonesian validation set to assess how each component contributes to the NJTSC framework. The zero-shot base model performs poorly, achieving only 14.64% detection F1 and 2.33% correction F1. This suggests that off-the-shelf LLMs, despite their general language competence, struggle to follow the strict output format required for spelling correction and are unaware of the idiosyncratic noise patterns in Indonesian user-generated text. Supervised fine-tuning on a 5K Indonesian dataset dramatically improves performance, raising the overall F1 from 14.64% to 83.13%. This jump underscores that task-specific alignment is not just helpful but essential in this low-resource setting.

Beyond this baseline, each added module targets a different limitation. In-context learning with structured templates yields an additional 2.97-point gain. We find that these demonstrations help the model avoid over-correcting legitimate morphological variants, such as informal affixations or dialectal spellings, by providing clear examples of what constitutes a true error. Translation augmentation, implemented via synthetic error-correction pairs generated using the Qwen3-Max API, adds another 1.65 points. This expansion from 5K to 10K training instances appears particularly beneficial for low-frequency words, where data scarcity previously led to brittle corrections. Finally, joint training with 5K Vietnamese samples (Row 5) provides a modest but consistent improvement of 0.42 F1. While small, this gain aligns with our hypothesis that certain typographical errors, especially those arising from shared script characteristics across Southeast Asian languages, exhibit cross-lingual regularities that the model can exploit. Altogether, the full NJTSC system reaches 88.09% detection F1 and 75.92% correction F1, suggesting that each component addresses a non-overlapping challenge in the pipeline.

Table 4: Ablation study on Indonesian validation set.

Model	Training Data	Detection			Correction			Overall
		P	R	F1	P	R	F1	
Base Model	–	9.23	35.31	14.64	1.37	7.93	2.33	8.48
+ Fine-tuning	Indo (5K)	80.86	85.74	83.23	63.09	80.42	70.71	76.97
+ ICL	Indo (5K)	86.32	86.10	86.21	66.98	81.81	73.66	79.94
+ Translation Aug.	Indo (10K)	88.36	87.18	87.77	69.27	82.71	75.47	81.59
+ Joint Training	Indo+Viet (15K)	88.37	87.74	88.09	69.66	83.42	75.92	82.01

6. Conclusion and Future Work

In this paper, we introduced NJTSC, a neighboring languages joint training framework that leverages cross-lingual data augmentation and in-context learning to enhance spelling correction for low-resource Southeast Asian languages. The results from the IMLIP2025 spelling correction shared task showcase the competitive nature of our approach. Despite utilizing a model with only 1.7 billion parameters and training on merely 15,000 instances, our method achieved 3rd place in error detection for both Indonesian and Vietnamese, and 4th place in Indonesian error correction. This emphasizes the efficiency of our framework in handling data scarcity and computational constraints while maintaining competitive performance.

Our current study was limited to translating Vietnamese into Indonesian due to constraints in training resources, resulting in data imbalance between the two languages. In future work, we aim to explore more balanced multilingual training strategies, such as bidirectional translation augmentation. Additionally, investigating larger language models and more sophisticated

demonstration selection mechanisms could enhance the model’s correction capabilities, particularly for the more challenging error correction subtask.

References

- [1] Q.-D. Nguyen, D.-A. Le, and I. Zelinka, “Ocr error correction for unconstrained vietnamese handwritten text,” in *Proceedings of the 10th International Symposium on Information and Communication Technology*, ser. SoICT ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 132–138. [Online]. Available: <https://doi.org/10.1145/3368926.3369686>
- [2] V. H. Nguyen, H. T. Nguyen, and V. Snasel, “Named entity recognition in vietnamese tweets,” in *Computational Social Networks*, M. T. Thai, N. P. Nguyen, and H. Shen, Eds. Cham: Springer International Publishing, 2015, pp. 205–215.
- [3] W. Wang, J.T. Huang, W. Wu, J. Zhang, Y. Huang, S. Li, P. He, and M. R. Lyu, “Mttm: Metamorphic testing for textual content moderation software,” in *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, May 2023, pp. 2387–2399.
- [4] Y. Yanfi, R. Setiawan, H. Soeparno, and W. Budiharto, “Specil: Spell error corpus for the indonesian language,” *IEEE Access*, vol. 11, pp. 93 227–93 237, 2023.
- [5] Y. Yanfi, H. Soeparno, R. Setiawan, and W. Budiharto, “Multi-head attention based bidirectional lstm for spelling error detection in the indonesian language,” *IEEE Access*, vol. 12, pp. 188 560–188 571, 2024.
- [6] T. H. Ngo, H. D. Tran, T. Huynh, and K. Hoang, “A combination of bert and transformer for vietnamese spelling correction,” in *Intelligent Information and Database Systems*, N. T. Nguyen, T. K. Tran, U. Tukayev, T.-P. Hong, B. Trawiński, and E. Szczerbicki, Eds. Cham: Springer International Publishing, 2022, pp. 545–558.
- [7] D. N. Tien, T. T. T. Minh, L. L. Vu, and T. D. Minh, “Vietnamese spelling error detection and correction using bert and n-gram language model,” in *Intelligent Systems and Networks*, N. L. Anh, S.-J. Koh, T. D. L. Nguyen, J. Lloret, and T. T. Nguyen, Eds. Singapore: Springer Nature Singapore, 2022, pp. 427–436.
- [8] B. Yao, G. Chen, R. Zou, Y. Lu, J. Li, S. Zhang, Y. Sang, S. Liu, J. Hendler, and D. Wang, “More samples or more prompts? exploring effective few-shot in-context learning for LLMs with in-context sampling,” in *Findings of the Association for Computational Linguistics: NAACL 2024*, K. Duh, H. Gomez, and S. Bethard, Eds. Mexico City, Mexico: Association for Computational Linguistics, Jun. 2024, pp. 1772–1790. [Online]. Available: <https://aclanthology.org/2024.findings-naacl.115/>
- [9] R. Agarwal, A. Singh, L. Zhang, B. Bohnet, L. Rosias, S. Chan, B. Zhang, A. Anand, Z. Abbas, A. Nova, J. D. Co-Reyes, E. Chu, F. Behbahani, A. Faust, and H. Larochelle, “Many-shot in-context learning,” in *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, Eds., vol. 37. Curran Associates, Inc., 2024, pp. 76 930–76 966. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2024/file/8cb564df771e9eacbf9d72bd46a24a9-Paper-Conference.pdf
- [10] Z. He, Y. Zhu, L. Wang, and L. Xu, “UMRSpell: Unifying the detection and correction parts of pre-trained models towards Chinese missing, redundant, and spelling correction,” in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 10 238–10 250. [Online]. Available: <https://aclanthology.org/2023.acl-long.570/>
- [11] L. Liu, H. Wu, and H. Zhao, “Chinese spelling correction as rephrasing language model,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 17, pp. 18 662–18 670, Mar. 2024. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/29829>
- [12] K. Al-Sabahi, K. Yang, W. Liu, G. Jiang, X. Li, and M. Yang, “Multi-head sequence tagging model for grammatical error correction,” *Engineering Applications of Artificial Intelligence*, vol. 133, p. 108314, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S095219762400472X>
- [13] S. M. Jayanthi, D. Pruthi, and G. Neubig, “NeuSpell: A neural spelling correction toolkit,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Q. Liu and D. Schlangen, Eds. Online: Association for Computational Linguistics, Oct. 2020, pp. 158–164. [Online]. Available: <https://aclanthology.org/2020.emnlp-demos.21/>
- [14] OpenAI, “Introducing chatgpt [sic],” 2022, accessed: 08/08/2023. [Online]. Available: <https://openai.com/blog/chatgpt>
- [15] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “LoRA: Low-rank adaptation of large language models,” in *International Conference on Learning Representations*, 2022. [Online]. Available: <https://openreview.net/forum?id=nZeVKeeFYf9>
- [16] Q. Team, “Qwen3 technical report,” 2025. [Online]. Available: <https://arxiv.org/abs/2505.09388>