VNU Journal of Science: Com. Science & Com. Eng., Vol. 42, No. 3 (2026) 87–99

VNU Journal of Science:
Computer Science and Communication Engineering

Journal homepage: http://www.jcsce.vnu.edu.vn/index.php/jcsce

Original Article

A Two-Stage Vietnamese Spelling Correction Pipeline Combining Underthesea and BARTpho

Hao T. N. Huynh[1], Long S. T. Nguyen[2], Nam H. Nguyen[2], Hoang M. Nguyen[2], Tho T. Quan[2*]

[1]*Digital Transformation Center, Department of Science and Technology of Tay Ninh Province, Vietnam*
[2]*URA Research Group, Ho Chi Minh City University of Technology (HCMUT), Vietnam*

Received 19th February 2026
Revised 07th March 2026; Accepted 26th March 2026

Abstract: Vietnamese spelling correction is challenging due to the language's rich diacritic system, syllable-based tokenization, and the frequent presence of strict entities in administrative and legal texts. While sequence-to-sequence models achieve strong correction accuracy, they are prone to over-correction and unintended rewriting under domain shift, which limits their reliability in high-stakes applications. In this paper, we propose a deployment-oriented two-stage Vietnamese spelling correction pipeline. The first stage performs text normalization and conservative error detection using Underthesea, combined with entity masking to preserve rigid identifiers and formatting. The second stage applies context-aware correction with a BARTpho sequence-to-sequence model, followed by detector-guided post-processing and iterative masked refinement to control unnecessary edits. To support realistic evaluation, we construct a hybrid dataset that mixes synthetic spelling noise with real-world errors collected from administrative documents. Experiments against strong multilingual and Vietnamese-specific baselines show that the proposed pipeline achieves high correction accuracy while significantly reducing over-correction. Beyond standard end-to-end metrics, we introduce detection-oriented analyses that explicitly quantify correction behavior at flagged positions, providing clearer evidence of practical safety for real-world deployment.

Keywords: Vietnamese spelling correction, controlled text editing, sequence-to-sequence models, Underthesea, BARTpho, administrative text

1. Introduction

Spelling errors are common in Vietnamese text and can severely impact downstream *Natural Language Processing* (NLP) systems. Unlike English and many other languages with relatively simple orthography, Vietnamese uses the Latin alphabet together with a rich diacritic system to encode tones and distinguish vowel qualities. These diacritics are essential for meaning: for instance, "ma", "má", "mà", "mả", and "mã" share the same base characters but convey differ-

*Corresponding author.
E-mail address: qttho@hcmut.edu.vn

https://doi.org/10.25073/2588-1086/vnucsce.7020

ent semantics. Consequently, even a small mistake in placing or omitting a tone mark can cause substantial meaning shifts.

In practice, such errors frequently appear in digital text due to input-method limitations, typing mistakes, and artifacts introduced during copying and manual editing. Spelling noise directly degrades many NLP applications, including information retrieval, text mining, document classification, and machine translation, and it is particularly problematic for administrative and legal document processing. In high-stakes domains such as public administration, healthcare, and finance, textual accuracy is mandatory: uncorrected errors may lead to misinterpretation, legal inconsistencies, and reduced trust in automated workflows.

Vietnamese spelling correction has traditionally relied on rule- and dictionary-based methods, which detect errors via lexicon lookup and simple morphological constraints. These approaches are fast and stable, making them attractive for real deployments, but they struggle with context-dependent mistakes and real-word errors, and they often over-flag domain-specific symbols and abbreviations (e.g., "15/2023/ND-CP"). Recent advances in Transformer-based models have renewed interest in neural correction. *Sequence-to-sequence* (seq2seq) generators such as BART [1] and T5 [2] achieve strong performance on text editing tasks, and for Vietnamese, BARTpho [3] provides a natural "noisy-to-clean" backbone. However, applying seq2seq models directly remains challenging in practice: general pre-training and end-to-end fine-tuning do not guarantee robust handling of Vietnamese-specific noise (diacritic placement, Unicode normalization, syllable-level spacing), and neural generators may over-correct under domain shift, unintentionally rewriting or corrupting strict entities such as identifiers, dates, and legal references [4].

These observations motivate a deployment-oriented design that combines conservative error detection, contextual correction, and explicit preservation of strict entities. Accordingly, we propose a controlled two-stage pipeline for Vietnamese spelling correction. In the first stage, the input is normalized and preprocessed with Underthesea[1], using rule-based and dictionary-matching signals to identify potentially erroneous positions, while masking important entities to protect administrative/legal structure. In the second stage, the processed text is corrected by a BARTpho model fine-tuned on "noisy–clean" sentence pairs to perform context-aware edits. A detector-guided post-processing step then restores masked entities and controls edits through masking-based refinement, improving reliability in high-stakes settings.

To mitigate the scarcity of labeled real-world Vietnamese spelling errors, we construct a hybrid training dataset that mixes controlled synthetic noise (diacritic perturbations and keyboard typos) with naturally occurring errors collected from real administrative documents, following prior directions for low-resource correction settings [5, 6]. Beyond standard end-to-end metrics, we adopt a deployment-driven evaluation that explicitly quantifies detection conservativeness (e.g., over-flagging) and correction behavior at flagged positions (over-correction vs. missed errors), providing clearer evidence of practical safety.

The main contributions of this work are summarized as follows.

- We propose an end-to-end, controlled two-stage pipeline for Vietnamese spelling correction, combining Underthesea-based preprocessing and conservative error detection with correction using a fine-tuned BARTpho model.
- We construct a hybrid training dataset integrating synthetic noise and real-world errors from administrative texts, better reflecting

[1]https://github.com/undertheseanlp/underthesea

error patterns encountered in practical deployments.

- We conduct comprehensive experiments with multiple metrics and analyses, showing improved accuracy and stability and providing explicit measurements of over-correction compared to standard seq2seq baselines and traditional dictionary-based approaches.

The rest of this paper is organized as follows: Section 2 reviews related work on Vietnamese spelling correction, including rule- and dictionary-based methods, neural correction models, and data construction for low-resource settings; Section 3 describes the proposed two-stage pipeline, covering preprocessing, entity masking, and BARTpho-based correction with post-processing control; Section 4 presents the dataset, experimental setup, evaluation metrics, and results, with additional analyses on detection conservativeness and over-correction; Section 5 discusses key findings and practical considerations for deployment. Finally, Section 6 concludes the paper and outlines future directions.

2. Related Work

2.1. Rule- and Dictionary-Based Vietnamese Spelling Correction

Rule- and dictionary-based spelling correction methods are among the earliest approaches and remain widely used in practical text processing systems. These methods typically detect errors through lexical lookup to identify out-of-vocabulary words or character sequences that violate basic morphological constraints. In the Vietnamese NLP ecosystem, toolkits such as VnCoreNLP provide core processing components (e.g., word segmentation, part-of-speech tagging, and named entity recognition), and are often employed as preprocessing or normalization layers before spelling correction [7]. The main advantages of rule-based approaches are their speed and stability, making them suitable for applications that require high reliability.

However, these methods struggle with context-dependent errors, real-word errors, and error patterns that fall outside predefined dictionaries. In administrative and legal documents, domain-specific symbols and abbreviations (e.g., "15/2023/ND-CP") are frequently mis-flagged as errors, suggesting that practical systems require explicit entity-preservation mechanisms rather than relying solely on lexical lookup.

2.2. Neural Approaches to Spelling Correction

The development of Transformer architectures has enabled a wide range of neural approaches to spelling correction. The first group employs encoder-only models such as BERT to detect anomalies or score candidate replacements based on contextual compatibility [8]. While effective at identifying contextual inconsistencies, these models do not directly generate corrected sentences.

The second group consists of encoder–decoder sequence generation models, which directly map noisy input text to its corrected form. Models such as BART, mBART, and T5 leverage denoising pre-training objectives and achieve strong performance in text editing tasks [1, 2, 9]. However, pure sequence generation models are prone to over-correction under domain shift, especially in specialized texts containing strict entities that must remain unchanged.

A third line of work focuses on edit-based or "seq2edit" approaches, which restrict the output space to a predefined set of edit operations (e.g., KEEP, DELETE, INSERT, and REPLACE). By constraining the generation process, these methods reduce over-correction and improve inference efficiency. Encode–Tag–Realize is a representative framework in this category [4].

2.3. Vietnamese-Specific Models and Challenges

Vietnamese spelling correction presents several language-specific challenges: (i) the rich diacritic system means that missing or misplaced

tone marks can substantially alter word meaning; (ii) whitespace segmentation often follows syllables rather than lexical words, which complicates tokenization and alignment between noisy and clean text; and (iii) administrative and legal documents contain many strict tokens that must be preserved verbatim, including identifiers, dates, and domain-specific abbreviations.

Among Vietnamese pre-trained language models, PhoBERT is a strong encoder-only model for contextual representation and anomaly detection [10]. For generative correction, BARTpho is a sequence-to-sequence model pre-trained on large-scale Vietnamese corpora and naturally fits the "noisy-to-clean" paradigm [3]. In practical pipelines, Underthesea is widely used as a rule-based preprocessing layer, supporting normalization and structural preservation through sentence segmentation, tokenization, and named entity recognition.

Beyond foundation models, several studies have directly addressed Vietnamese spelling correction. Notably, VSEC proposes a Transformer-based correction model and releases a real-error dataset at the syllable level [5]. Another line of work combines BERT with Transformer architectures and reports strong empirical performance on Vietnamese spelling correction [6]. Nevertheless, when transferring these models to administrative and legal domains, neural approaches still face the risk of corrupting critical entities and lack explicit mechanisms for controlling edits. Studies that directly exploit administrative text further highlight strong domain specificity and the need for tailored correction strategies [11]. These observations point to the necessity of a controlled correction pipeline that integrates error detection, contextual generation, and structural preservation.

2.4. Data Design, Noise Injection, and Training

The scarcity of annotated spelling error data is a major bottleneck for spelling correction, especially in low-resource languages. As a result, many studies rely on synthetic noise generation to simulate realistic errors, ranging from simple character-level perturbations to diverse back-translation strategies [12]. Research on robust word recognition also demonstrates the benefits of introducing recognition or normalization stages before downstream neural models [13]. For Vietnamese, effective noise modeling must capture keyboard typos, diacritic errors, and inconsistencies in Unicode normalization. Combining synthetic data with domain-specific real errors (e.g., from administrative or legal texts) has been shown to improve robustness in real-world deployment.

2.5. Related Directions: Noisy Channel and Low-Resource Languages

The Noisy Channel framework is a classical approach to spelling correction, in which performance depends on both the error model and the language model. Brill and Moore extended the error model with string-to-string edit operations, laying the foundation for many subsequent methods [14]. In low-resource settings, dictionary- and heuristic-based algorithms such as SymSpell and its variants remain popular due to their speed and efficiency [15].

In summary, rule-based approaches offer stability but lack contextual awareness, whereas neural sequence generation models provide fluent corrections but are vulnerable to over-correction under domain shift. These complementary strengths motivate our approach: a two-stage pipeline that combines conservative rule-based detection with BARTpho-based contextual correction, augmented with entity masking and post-processing to preserve critical structures in administrative and legal Vietnamese text.

3. Methodology

3.1. System Overview

Figure 1 illustrates the overall architecture of the proposed Vietnamese spelling correction system. The system is designed as a controlled two-stage pipeline comprising three main phases: (i)

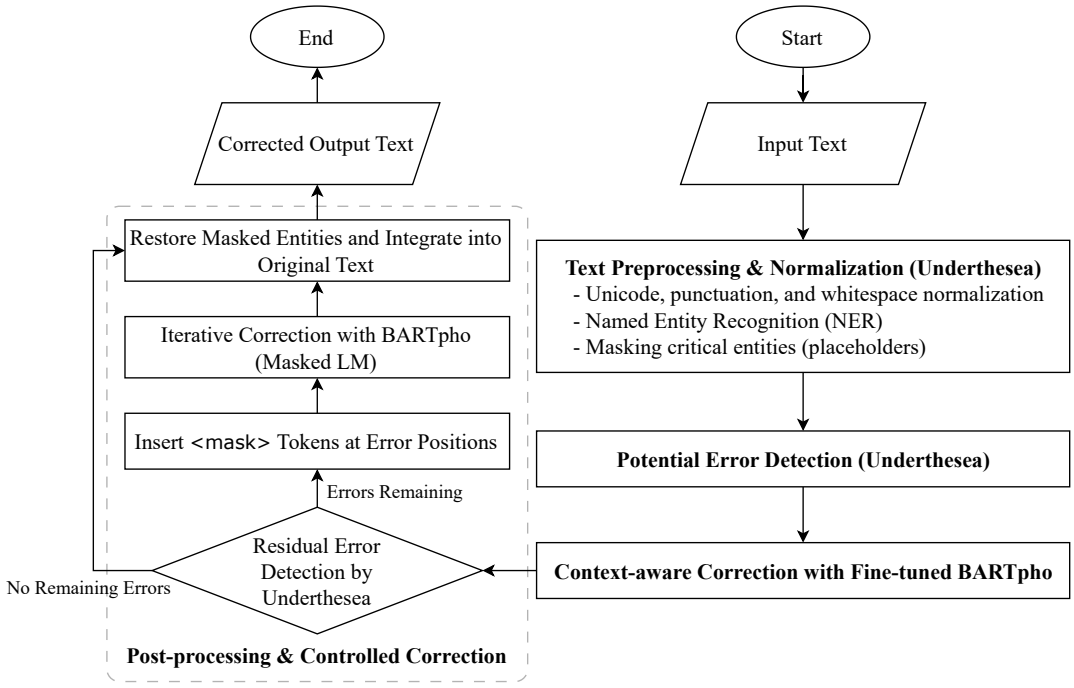


Figure 1. Overview of the proposed controlled two-stage spelling correction pipeline.

text preprocessing and conservative detection of potentially erroneous positions, (ii) context-aware correction using a fine-tuned BARTpho model, and (iii) post-processing to restore masked entities and control unnecessary edits.

This design combines the strengths of rule-based methods in preserving document structure and critical entities with the contextual understanding and fluent generation capabilities of deep neural language models. The entire pipeline follows a controlled correction paradigm, explicitly aiming to reduce over-correction, which is particularly important for administrative and legal documents where unintended edits can be costly.

3.2. Text Preprocessing and Normalization

The input text is first passed through a series of preprocessing steps designed to reduce noise, normalize textual representations, and preserve the original document structure. Specifically, we perform the following operations:

- **Unicode normalization** and removal of hidden or non-printable formatting artifacts introduced during document creation and editing.
- **Punctuation and whitespace normalization** using regular expressions to eliminate redundant characters and ensure consistent formatting.
- ***Named Entity Recognition* (NER):** we employ the NER component of Underthesea to identify entities such as personal names, locations, dates, and domain-specific identifiers.
- **Critical entity masking:** entities with rigid formats (e.g., `15/2023/ND-CP`) are replaced with placeholder tokens (e.g., `ENT_1`) to prevent the generative model from altering information that must be preserved verbatim.

After normalization and entity masking, the

text is segmented into sentences using Underthesea, yielding standardized inputs for subsequent error detection and correction stages.

3.3. Context-Aware Correction with Fine-Tuned BARTpho

We use BARTpho as the main correction model, fine-tuned on paired *noisy–clean* Vietnamese sentences. The objective of this stage is to enable the model to learn a direct mapping from sentences containing spelling errors to their contextually correct forms.

Unlike unrestricted end-to-end generation, the BARTpho model is triggered only for text segments that contain positions identified as potentially erroneous by the preceding detection stage. This design ensures that neural generation is invoked conservatively, focusing on suspicious regions rather than rewriting already-correct text unnecessarily.

3.4. Post-processing and Controlled Iterative Correction

Although the fine-tuned BARTpho model produces fluent corrections, its outputs may still contain residual errors or unnecessary edits, especially in texts with strict accuracy requirements. We therefore introduce a post-processing stage to further control and refine the final output.

First, the corrected text produced by BARTpho is re-analyzed using Underthesea to perform part-of-speech tagging and named entity recognition. Tokens labeled as `O` (non-entity), after removing punctuation, special characters, and digits, are treated as candidates for remaining spelling errors.

These candidate tokens are then validated against a normalized vocabulary derived from Underthesea. For each detected error position, we apply a *masking* operation, replacing the corresponding token with a `<mask>` symbol.

The sentence containing masked positions is fed back into the BARTpho model in inference mode, where the model predicts the most suitable token for each `<mask>` based on the full surrounding context. The highest-probability prediction is selected to produce an updated corrected sentence.

This detection–masking–correction loop is repeated iteratively until no further candidate errors are detected or a predefined stopping condition is reached. Finally, all previously masked entities are restored and reintegrated into the original text, yielding the final corrected output.

By combining rule-based linguistic analysis with contextual token prediction from a neural language model, this controlled correction strategy effectively limits over-correction while improving accuracy and consistency, making it particularly suitable for administrative and legal Vietnamese documents.

4. Experiments

4.1. Dataset

We conduct experiments on a Vietnamese spelling correction dataset containing 78,909 *noisy–clean* sentence pairs. The dataset is constructed from two complementary sources to support both general correction capability and robustness in administrative and legal text domains.

Synthetic errors: The majority of sentence pairs are generated automatically using a controlled noise injection procedure that simulates common Vietnamese spelling errors, including diacritic perturbations and keyboard typos. This synthetic component provides broad coverage of frequent error patterns, enabling models to learn generalizable correction behaviors.

Real-world errors: In addition, a small but critical subset is manually collected and annotated from real administrative documents. Although limited in size, these examples capture domain-specific characteristics such as rigid identifiers, abbreviations, and formatting constraints, which are rarely represented by purely synthetic noise.

We partition the dataset into train and test splits. To reduce bias from sentence length,

Table 1. Dataset statistics for Vietnamese spelling correction

Dataset Split	# Sentences	(%)	Avg. Length (words/sentence)	# Error Tokens	Error Rate (%)
Synthetic Errors	78,614	99.63	38.12	82,049	2.7199
Real-world Errors	295	0.37	29.88	315	3.4825
Train	70,963	89.93	38.02	74,063	2.7278
Test	7,946	10.07	38.66	8,301	2.6724

the split is stratified by length buckets, ensuring that both short and long sentences are distributed more evenly across subsets. Table 1 summarizes the overall corpus composition, average sentence length, and token-level error statistics.

4.2. Evaluation Metrics

We report a set of metrics that capture both *correction accuracy* and *surface fidelity and fluency*, as well as metrics that explicitly quantify *over-correction behavior*. Let y denote the gold (clean) sentence, $\hat{y}$ the predicted sentence, and $\text{tok}(\cdot)$ the tokenization function used for evaluation. For token-level metrics, we compare tokens positionally after applying the same tokenization to y and $\hat{y}$.

Word Accuracy (WA): Word Accuracy measures token-level correctness as the fraction of tokens that exactly match the reference:

$$\text{WA} = \frac{1}{|\text{tok}(y)|} \sum_{i=1}^{|\text{tok}(y)|} \mathbb{I}[\text{tok}(\hat{y})_i = \text{tok}(y)_i], \quad (1)$$

where $\mathbb{I}[\cdot]$ is the indicator function.

Sentence Accuracy (SA): Sentence Accuracy is a strict metric that counts a sentence as correct only if the entire prediction exactly matches the reference:

$$\text{SA} = \frac{1}{N} \sum_{n=1}^{N} \mathbb{I}[\hat{y}^{(n)} = y^{(n)}], \quad (2)$$

where N is the total number of sentences.

Levenshtein Similarity (LS): Levenshtein Similarity measures character-level similarity using normalized edit distance:

$$\text{LS} = 1 - \frac{\text{ED}(\hat{y}, y)}{\max(|\hat{y}|, |y|)}, \quad (3)$$

where $\text{ED}(\cdot,\cdot)$ denotes the Levenshtein (edit) distance and $|\cdot|$ is the number of characters.

BLEU: BLEU [16] evaluates modified n-gram precision with a brevity penalty to discourage overly short outputs:

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{K} w_n \log p_n\right), \quad (4)$$

where p_n is the modified precision of n-grams, w_n are weights (typically $w_n = \frac{1}{K}$), BP is the brevity penalty, and K denotes the maximum n-gram order:

$$\text{BP} = \begin{cases} 1, & \text{if } c > r, \\ \exp\left(1 - \frac{r}{c}\right), & \text{otherwise}, \end{cases} \quad (5)$$

with c and r denoting the total candidate and reference lengths, respectively.

ChrF: ChrF [17] computes a character n-gram F-score, which can better reflect partial matches under orthographic variations. Let P_n and R_n be character n-gram precision and recall, and define

$$P = \frac{1}{K} \sum_{n=1}^{K} P_n, \qquad R = \frac{1}{K} \sum_{n=1}^{K} R_n. \quad (6)$$

ChrF is then computed as

$$\text{ChrF} = \frac{(1+\beta^2)\, P\, R}{\beta^2 P + R}, \quad (7)$$

where β controls the relative importance of recall (we use the standard setting $\beta = 2$).

Detection-Oriented Metrics: To explicitly analyze correction behavior under a controlled pipeline, we introduce metrics computed only on tokens flagged by the detection stage. Let $\mathcal{F}$ be the set of flagged token positions, C the set of positions that are correct in the input, and $\mathcal{E}$ the set of positions that are erroneous in the input.

- **Accuracy@Flagged** measures token-level accuracy restricted to flagged positions:

$$\text{Accuracy@Flagged} = \frac{1}{|\mathcal{F}|} \sum_{i \in \mathcal{F}} \mathbb{I}[\hat{y}_i = y_i]. \tag{8}$$

- **Over-Correction** quantifies how often originally correct tokens are changed incorrectly:

$$\text{Over-Correction} = \frac{\#\{i \in \mathcal{F} \cap C : \hat{y}_i \neq y_i\}}{|\mathcal{F} \cap C|}. \tag{9}$$

- **Unchanged Error** measures the fraction of erroneous tokens left incorrect after correction:

$$\text{Unchanged Error} = \frac{\#\{i \in \mathcal{F} \cap \mathcal{E} : \hat{y}_i \neq y_i\}}{|\mathcal{F} \cap \mathcal{E}|}. \tag{10}$$

Together, these metrics provide a deployment-oriented evaluation that separates beneficial fixes from risky rewrites, which is critical for administrative and legal text processing.

4.3. Backbone Comparison and Training Setup

To evaluate the impact of the seq2seq backbone within our controlled two-stage pipeline, we experiment with three different pre-trained encoder–decoder models: BARTpho, mBART-cc25[2], and mBART-50[3].

[2]https://huggingface.co/facebook/mbart-large-cc25
[3]https://huggingface.co/facebook/mbart-large-50

In all experiments, the overall pipeline architecture remains unchanged, including Underthesea-based preprocessing, conservative detection, entity masking, and detector-guided post-processing. The only varying component is the correction backbone used in the second stage.

This controlled setup allows us to isolate the effect of backbone choice on correction accuracy, over-correction behavior, and robustness under different noise conditions.

For a fair comparison, all backbones are fine-tuned under identical optimization, scheduling, and regularization settings. The full training configuration is summarized in Table 2.

4.4. Error Detection Performance

We first evaluate the detection component to ensure that it does not "hallucinate" errors. Table 3 reports token-level Precision, Recall, and F1, along with the *Over-Flag Rate* (OFR), defined as the fraction of correct tokens incorrectly flagged as erroneous:

$$\text{OFR} = \frac{\#\{\text{correct tokens flagged}\}}{\#\{\text{correct tokens}\}}. \tag{11}$$

The results show that Underthesea achieves high precision with an extremely low OFR, making it suitable as a conservative filter prior to neural correction.

4.5. End-to-End Correction Results

Table 4 presents the main end-to-end results. All evaluated backbones substantially improve over the uncorrected input when integrated into the controlled pipeline. The two-stage design specifically targets practical deployment: by correcting only positions flagged as risky, the pipeline reduces unnecessary edits while maintaining strong overall accuracy.

4.6. Over-Correction Analysis

To explicitly analyze over-correction, we evaluate models only at *flagged positions*, i.e., tokens identified as suspicious by the detection

Table 2. Training and optimization settings for all seq2seq models

Parameter	**Value**
Optimizer	AdamW
Learning rate	3×10^{-5}
Epochs	10
Warmup ratio	6% of total steps
LR schedule	Linear warmup + cosine decay
Batch size (per device)	8
Gradient accumulation	8 steps
Effective batch size	64
Dropout	0.2
Weight decay	0.01
Label smoothing	0.1
Max sequence length	256

Table 3. Error detection performance of Underthesea

Model	**Precision**	**Recall**	**F1**	**OFR**
Underthesea	0.8173	0.4279	0.5617	0.0011

Table 4. End-to-end correction performance

Model	**WA**	**SA**	**LS**	**BLEU**	**ChrF**
Input (no correction)	0.9085	0.0240	0.9890	0.8788	0.8577
mBART-cc25	0.9255	0.6685	0.9842	0.9583	0.9126
mBART-50	0.9401	0.7077	0.9904	0.9609	0.9033
BARTpho	0.9691	0.7393	0.9903	0.9653	0.9224

stage. Table 5 reports: (i) Accuracy@Flagged; (ii) Over-Correction (the fraction of originally correct tokens changed incorrectly); and (iii) Unchanged Error (the fraction of erroneous tokens left uncorrected).

These results confirm the motivation behind a controlled pipeline: performance gains stem primarily from correcting genuine errors rather than from risky rewrites.

4.7. Performance by Error Type

We further analyze performance across major Vietnamese error categories. Table 6 reports WA for each error type.

4.8. Robustness under Different Noise Levels

Finally, we evaluate robustness under varying noise intensities. Since the generator injects a discrete number of errors per sentence, we report results for sentences containing one error and two errors. Table 7 presents WA under each setting.

4.9. Qualitative Analysis on Representative Examples

To complement the quantitative results, we analyze several representative examples to qual-

Table 5. Correction behavior on detector-flagged tokens

Model	Accuracy@Flagged	Over-Correction	Unchanged Error
mBART-cc25	0.8606	0.0120	0.1620
mBART-50	0.8728	0.0119	0.0785
BARTpho	0.8824	0.0205	0.1021

Table 6. Performance by error type

Error Type	mBART-cc25	mBART-50	BARTpho
Tone/Diacritics	0.9278	0.9409	0.9720
Spacing	0.9102	0.9113	0.9299
Missing Letter	0.9168	0.9176	0.9614
Vowel	0.9220	0.9428	0.9681
Rhyme	0.9357	0.9495	0.9730
Consonant	0.9292	0.9475	0.9676

Table 7. Robustness under different noise levels

Noise Level	mBART-cc25	mBART-50	BARTpho
1 error	0.9277	0.9406	0.9707
2 errors	0.9096	0.9245	0.9397

itatively examine how different seq2seq backbones behave within the same controlled correction pipeline when handling context-dependent Vietnamese spelling errors in administrative text.

- **Example 1:** *"...không còn tình trạng bày bán, lưu hành các xuất bản phẩm đồi trụy, mê **trín** dị đoan."*
  - **BARTpho:** trín → **tín** (Top candidates: `[tín, hoặc, sản]`).
  - **mBART-cc25:** trín → tính (Top candidates: `[và, liệt, tính]`).
  - **mBART-50:** trín → trín (Top candidates: `[chí, bá, nghệ]`).
- **Example 2:** *"...mô hình học cụ, vũ khí, trang bị; xây dựng, củng cố thao trường, bãi **tâp**;"*
  - **BARTpho:** tâp → **tập** (Top candidates: `[tập, thao, huấn]`).
  - **mBART-cc25:** tâp → tâp (Top candidates: `[trường, biển, phòng]`).
  - **mBART-50:** tâp → tập (Top candidates: `[giảng, tập, bãi]`).
- **Example 3:** *"...cơ bản đảm bảo khả năng **cập** điện cho sinh hoạt và sản xuất trên địa bàn tỉnh."*
  - **BARTpho:** cập → **cấp** (Top candidates: `[đáp, cấp, tiếp]`).
  - **mBART-cc25:** cập → cấp (Top candidates: `[phát, điện, cấp]`).
  - **mBART-50:** cập → cập (Top candidates: `[cung_cấp, cung_ứng, đủ]`).

Overall, these examples show that BARTpho produces more contextually appropriate corrections, especially for Vietnamese diacritic errors and fixed administrative expressions. In contrast, multilingual backbones display less consistent behavior, including missed corrections and occasional semantic drift.

5. Key Findings and Discussion

Our experimental results provide several practical observations for deployment-oriented Vietnamese spelling correction.

Vietnamese-specific modeling remains highly beneficial for spelling correction: All evaluated seq2seq backbones substantially improve over the uncorrected input, confirming the effectiveness of denoising-style training for Vietnamese spelling correction. Among them, BARTpho consistently achieves the strongest overall performance, suggesting that Vietnamese-specific pre-training is particularly effective for capturing fine-grained orthographic patterns such as diacritics, vowel variants, and consonant-level confusion.

Conservative detection functions as a safety-oriented filter: The detection results show that Underthesea achieves high precision with an extremely low over-flag rate, which is desirable in administrative and legal settings where false alarms may trigger harmful edits. However, its moderate recall indicates that some errors remain outside the correction path, revealing a practical trade-off between conservative triggering and error coverage.

Correction quality and correction safety do not improve in parallel: The flagged-token analysis shows that BARTpho attains the highest Accuracy@Flagged, but also exhibits a higher Over-Correction rate than the multilingual baselines. This indicates that stronger correction ability can also increase the tendency to rewrite borderline tokens. Therefore, the main advantage of the proposed framework lies not only in using a strong generator, but in constraining generation through detection, entity masking, and post-processing.

The pipeline remains strongest on local orthographic errors: Performance is highest on tone and diacritic errors, while spacing errors and multi-error sentences remain more challenging. This suggests that the current framework handles localized Vietnamese spelling distortions well, but becomes less robust when errors affect token boundaries or interact within the same sentence.

Qualitative evidence supports the quantitative trends: The representative examples show that BARTpho more consistently restores contextually appropriate Vietnamese forms, whereas multilingual backbones are less stable, sometimes leaving errors unchanged or producing contextually unsuitable substitutions. At the same time, because the current dataset is still dominated by synthetic errors, the reported results should be interpreted as strong evidence of practical potential, while broader real-world validation remains an important direction for future work.

6. Conclusion

This paper presented a deployment-oriented two-stage pipeline for Vietnamese spelling correction, designed for administrative and legal texts where strict entities and formatting must be preserved. The proposed system combines (i) Underthesea-based normalization and conservative error detection, (ii) contextual correction using a fine-tuned BARTpho seq2seq model, and (iii) post-processing with entity masking/restoration and detector-guided refinement to control unnecessary edits. Experiments on a hybrid dataset combining synthetic noise and real-world administrative errors demonstrate that fine-tuned seq2seq models achieve strong correction performance, with BARTpho yielding the best overall accuracy among the evaluated backbones. Our detection-oriented evaluation further highlights a key practical property: the Underthesea

detector maintains high precision with a very low over-flag rate, making it suitable as a safety filter in high-stakes settings, while flagged-token analysis reveals over-correction trade-offs across different generators. Qualitative examples confirm that context-aware generation is essential for resolving Vietnamese diacritic errors and fixed expressions in official documents.

Future work will focus on (i) improving detector coverage without increasing over-flagging, and (ii) introducing entity-level faithfulness metrics and human evaluation protocols to better assess semantic preservation and legal or administrative validity. We also plan to explore iterative correction strategies and richer multi-error noise modeling to enhance robustness under heavier noise conditions.

References

[1] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 7871–7880. https://doi.org/10.18653/v1/2020.acl-main.703.

[2] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, J. Mach. Learn. Res., Vol. 21, No. 1, (Jan. 2020).

[3] N. L. Tran, D. Le, D. Q. Nguyen, BARTpho: Pre-trained Sequence-to-Sequence Models for Vietnamese, in: Interspeech 2022, 2022, pp. 1751–1755. https://doi.org/10.21437/Interspeech.2022-10177.

[4] E. Malmi, S. Krause, S. Rothe, D. Mirylenka, A. Severyn, Encode, Tag, Realize: High-Precision Text Editing, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 5054–5065. https://doi.org/10.18653/v1/D19-1510.

[5] D.-T. Do, H. T. Nguyen, T. N. Bui, H. D. Vo, VSEC: Transformer-Based Model for Vietnamese Spelling Correction, in: PRICAI 2021: Trends in Artificial Intelligence: 18th Pacific Rim International Conference on Artificial Intelligence, PRICAI 2021, Hanoi, Vietnam, November 8–12, 2021, Proceedings, Part II, Springer-Verlag, Berlin, Heidelberg, 2021, p. 259–272. https://doi.org/10.1007/978-3-030-89363-7_20.

[6] T. H. Ngo, H. D. Tran, T. Huynh, K. Hoang, A Combination of BERT and Transformer for Vietnamese Spelling Correction, in: Intelligent Information and Database Systems: 14th Asian Conference, ACIIDS 2022, Ho Chi Minh City, Vietnam, November 28–30, 2022, Proceedings, Part I, Springer-Verlag, Berlin, Heidelberg, 2022, p. 545–558. https://doi.org/10.1007/978-3-031-21743-2_43.

[7] T. Vu, D. Q. Nguyen, D. Q. Nguyen, M. Dras, M. Johnson, VnCoreNLP: A Vietnamese Natural Language Processing Toolkit, in: Y. Liu, T. Paek, M. Patwardhan (Eds.), Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 56–60. https://doi.org/10.18653/v1/N18-5012.

[8] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. https://doi.org/10.18653/v1/N19-1423.

[9] Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, L. Zettlemoyer, Multilingual Denoising Pre-training for Neural Machine Translation, Transactions of the Association for Computational Linguistics, Vol. 8, 2020, pp. 726–742. https://doi.org/10.1162/tacl_a_00343.

[10] D. Q. Nguyen, A. Tuan Nguyen, PhoBERT: Pre-trained language models for Vietnamese, in: T. Cohn, Y. He, Y. Liu (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguistics, Online, 2020, pp. 1037–1042. https://doi.org/10.18653/v1/2020.findings-emnlp.92.

[11] H. T. Phung, N. V. Luong, Detecting spelling errors in Vietnamese administrative document using large language models, HO CHI MINH CITY OPEN UNIVERSITY JOURNAL OF SCIENCE - ENGINEERING AND TECHNOLOGY, Vol. 14, No. 1, 2024, pp. 31–40. https://doi.org/10.46223/HCMCOUJS.tech.en.14.1.3141.2024.

[12] Z. Xie, G. Genthial, S. Xie, A. Ng, D. Jurafsky, Noising and Denoising Natural Language: Diverse Back-

translation for Grammar Correction, in: M. Walker, H. Ji, A. Stent (Eds.), Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 619–628. https://doi.org/10.18653/v1/N18-1057.

[13] D. Pruthi, B. Dhingra, Z. C. Lipton, Combating Adversarial Misspellings with Robust Word Recognition, in: A. Korhonen, D. Traum, L. Màrquez (Eds.), Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 5582–5591. https://doi.org/10.18653/v1/P19-1561.

[14] E. Brill, R. C. Moore, An Improved Error Model for Noisy Channel Spelling Correction, in: Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Hong Kong, 2000, pp. 286–293. https://doi.org/10.3115/1075218.1075255.

[15] E. P. P. Mon, Y. K. Thu, T. T. Yu, A. Wai Oo, SymSpell4Burmese: Symmetric Delete Spelling Correction Algorithm (SymSpell) for Burmese Spelling Checking, in: 2021 16th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP), 2021, pp. 1–6. https://doi.org/10.1109/iSAI-NLP54397.2021.9678171.

[16] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a Method for Automatic Evaluation of Machine Translation, in: P. Isabelle, E. Charniak, D. Lin (Eds.), Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 2002, pp. 311–318. https://doi.org/10.3115/1073083.1073135.

[17] M. Popović, chrF: character n-gram F-score for automatic MT evaluation, in: O. Bojar, R. Chatterjee, C. Federmann, B. Haddow, C. Hokamp, M. Huck, V. Logacheva, P. Pecina (Eds.), Proceedings of the Tenth Workshop on Statistical Machine Translation, Association for Computational Linguistics, Lisbon, Portugal, 2015, pp. 392–395. https://doi.org/10.18653/v1/W15-3049.