VIETNAM GENERAL CONFEDERATION OF LABOR

TON DUC THANG UNIVERSITY

FACULTY OF INFORMATION TECHNOLOGY

INFORMATION TECHNOLOGY PROJECT

TEXT ERROR CORRECTION PROBLEM IN DEEP LEARNING

Supervisor**: Assoc. Phd. Le Anh Cuong**

Implementers**: Pham Huynh Tin**

Nguyen Trung Thang

Course: **26**

HO CHI MINH CITY, 2025

VIETNAM GENERAL CONFEDERATION OF LABOR

TON DUC THANG UNIVERSITY

FACULTY OF INFORMATION TECHNOLOGY

INFORMATION TECHNOLOGY PROJECT

TEXT ERROR CORRECTION PROBLEM IN DEEP LEARNING

Supervisor**: Assoc. Phd. Le Anh Cuong**

Implementers**: Pham Huynh Tin**

Nguyen Trung Thang

Course**: 26**

HO CHI MINH CITY, 2025

ACKNOWLEDGEMENT

Dear Mr. Le Anh Cuong,

We would like to extend our deepest gratitude to you for your unwavering dedication and invaluable guidance throughout our learning journey. Your profound knowledge, thoughtful instruction, and constant encouragement have not only helped us grasp complex concepts but also ignited a genuine passion for the subject.

Your patience, insight, and inspiring teaching style have opened new perspectives for us, giving us the confidence to embrace challenges and strive for excellence. Each lesson with you was not just a transfer of knowledge, but a source of motivation and personal growth. Thank you for your generous mentorship and for being such a remarkable source of inspiration. We are truly honored to have had the opportunity to learn from you.

With sincere appreciation and respect.

THE WORK IS COMPLETED
AT TON DUC THANG UNIVERSITY

I hereby declare that this is my own research project and is under the scientific guidance of Assoc. PhD. Le Anh Cuong. The research content and results in this topic are honest and have not been published in any form before. The data in the tables for analysis, comments, and evaluation were collected by the author from different sources and clearly stated in the reference section.

Project also uses a number of comments, assessments as well as data from other authors and other organizations, all with citations and source notes.

If any fraud is detected, I will take full responsibility for the content of my Project. Ton Duc Thang University is not involved in copyright violations caused by me during the implementation process (if any).

Ho Chi Minh City, March 24th, 2025

Tin

Pham Huynh Tin

Thang

Nguyen Trung Thang

ABSTRACT

In recent years, the growing demand for automated text correction systems has become increasingly essential, especially for low-resource languages like Vietnamese. This study proposes a deep learning-based approach for Vietnamese grammatical and spelling error correction using pre-trained transformer models, namely ViT5 and BARTPHO-Syllable. Due to the scarcity of publicly available parallel datasets, we design a comprehensive rule-based data augmentation pipeline to synthetically generate diverse noisy-clean sentence pairs from clean text. The proposed system fine-tunes ViT5 and BARTPHO-Syllable on this synthetic dataset in a sequence-to-sequence framework, enabling the models to learn to correct various types of real-world text errors, such as tone mark mistakes, Telex input errors, colloquial abbreviations, and structural inconsistencies. Experimental results and extended evaluations demonstrate the effectiveness of the models in improving text quality, with strong performance across standard metrics such as Word Error Rate (WER), Character Error Rate (CER), precision, recall, and F1-score. This work contributes a practical and scalable solution for Vietnamese text correction and highlights the potential of leveraging synthetic data and pretrained language models in low-resource language settings.

TABLE OF CONTENT

ABBREVIATIONS

NLP	Natural Language Processing
FP16	16-bit Floating Point Precision
WER	Word Error Rate
CER	Character Error Rate
CER	Character Error Rate
F1	F1 Score (Harmonic mean of Precision and Recall)
BERT	Bidirectional Encoder Representations from Transformers
GPT	Generative Pre-trained Transformer
GQA	Grouped Query Attention
T5	Text-to-Text Transfer Transformer
ViT5	Vietnamese T5
BART	Bidirectional and Auto-Regressive Transformers
TP	True Positives
FP	False Positives
FN	False Negatives
POS	Parts of Speech

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. OVERVIEW OF THE PROBLEM

1.1. Introduction

In this day and age, the volume of user-generated text data from social networking platforms, forums, emails, and messaging applications has expanded exponentially. However, this explosion of informal communication has also led to a significant decline in text quality. A vast majority of this data is unedited and replete with errors, including typographical mistakes, non-standard spelling, colloquial abbreviations, and grammatical inconsistencies. This "noisy" text poses significant challenges for downstream NLP tasks such as machine translation, sentiment analysis, and information retrieval, while also hindering effective communication.

Consequently, the demand for automated tools capable of checking and correcting text errors is more pressing than ever. While robust solutions for high-resource languages like English have become mature, Vietnamese-a language with unique tonal and orthographic characteristics-remains an underserved and compelling case. The development of an effective text correction system for Vietnamese not only holds practical value in improving digital communication and data quality but also presents a significant scientific challenge.

This project addresses this challenge by researching and developing a deep learning-based system specifically for Vietnamese text error correction. Our central objective is to leverage the power of Transformer architectures. Recognizing the critical bottleneck of data scarcity, our core contribution lies in the design and implementation of a comprehensive data augmentation pipeline to synthetically generate a large-scale, diverse dataset of incorrect-correct sentence pairs. This data-centric approach enables us to effectively fine-tune and evaluate pre-trained models, paving the way for a practical and scalable solution.

1.2. Key Challenges

Developing an effective text correction system for Vietnamese presents unique challenges due to the inherent diversity of this language and as well as data scarcity. The challenges can be mentioned as:

1.2.1. Linguistic Complexity

Vietnamese is a tonal language where the meaning of a word can change significantly based on its tone marks. For example, the word "ma" (ghost), "mà" (but), and "má" (mother) are distinct in meaning despite their similar spellings. Errors often arise from incorrect tones, confusion between consonants (e.g., "x" vs. "s" or "n" vs. "l"), or vowel variations (e.g., "ê" vs. "e"). These nuances require the system to not only detect spelling errors but also understand the contextual appropriateness of words.

1.2.2. The lack of Parallel Data

Unlike widely spoken languages such as English, Vietnamese lacks large-scale, publicly available datasets of incorrect-correct sentence pairs. This scarcity of labeled data makes it difficult to train supervised machine learning models effectively. Without sufficient training data, models may struggle to generalize and accurately correct diverse errors.

1.2.3. Variety of Error Types

Errors in Vietnamese text are not limited to simple misspellings. They encompass a broad range of phenomena, including:

- **Informal Slang and Abbreviations:** The use of teencode (e.g., "ko" for "không", "dc" for "được") is ubiquitous in online communication.
- **Telex Typing Errors:** Mistakes arising from common Vietnamese input methods, such as forgetting a character to complete a diacritic (e.g., chuongw trinhf instead of chương trình).

- **Grammatical and Structural Errors:** Incorrect word order, missing words, or duplicated words that violate grammatical rules.

  A robust system must be capable of identifying and correcting all these error types, which requires more than just spelling knowledge; it demands a model of language structure and usage.

1.3. Proposed Approach

To overcome the multifaceted challenges outlined above, this project proposes a data-centric, sequence-to-sequence (Seq2Seq) solution built upon the Transformer architecture. We frame the task of text error correction as a "translation" problem: translating a "noisy" (incorrect) sentence into a "clean" (correct) one. This paradigm is well-suited to handle the complex, context-dependent transformations required for Vietnamese.

The cornerstone of our methodology is a data augmentation pipeline. To combat the lack of parallel data, we have designed a rule-based system to programmatically introduce a diverse set of realistic errors into a large corpus of clean Vietnamese text. This pipeline simulates the common mistakes identified in our analysis, including tone errors, Telex typing errors, phonetic misspellings, structural inconsistencies (using Part-of-Speech tags for more realistic modifications), and the replacement of words with common abbreviations.

By fine-tuning pre-trained, Vietnamese-specific language models—namely **ViT5** and **BARTPHO-Syllable**—on this large-scale synthetic dataset, we aim to teach them the mapping from incorrect to correct text. This approach allows us to effectively circumvent the data scarcity problem while building a robust model capable of handling a wide array of real-world errors. The final phase of our project involves a rigorous evaluation and comparative analysis of these models to determine the most effective solution for the Vietnamese text correction task.

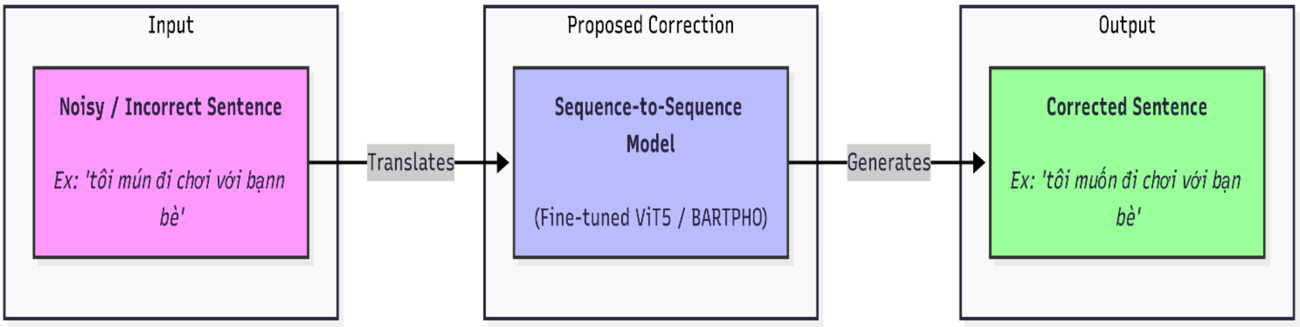

Input
Noisy / Incorrect Sentence
Ex: 'tôi mún đi chơi với bạnn bè'
Translates
Proposed Correction
Sequence-to-Sequence Model
(Fine-tuned ViT5 / BARTPHO)
Generates
Output
Corrected Sentence
Ex: 'tôi muốn đi chơi với bạn bè'

***Figure 1.1:** Overview of the proposed text correction system.*

1.4. Related Work

The problem of Text Error Correction has attracted significant attention in the Natural Language Processing community. In recent years, many studies have focused on text error correction using deep learning techniques, especially for languages such as English and Chinese. However, for Vietnamese—a language with unique phonological and grammatical characteristics—the number of studies remains limited.

One of the foundational works is **BARTpho** by Nguyen Luong Tran [1]. They introduced pre-trained sequence-to-sequence models specifically for Vietnamese, namely *BARTpho-word* and *BARTpho-syllable*. By pre-training on a large Vietnamese corpus with a denoising objective, BARTpho established a strong baseline for various text generation tasks, including error correction. Their approach demonstrated the effectiveness of adapting large language models to the phonological structure of Vietnamese, particularly at the syllable level. Our project utilizes and evaluates one of these models, BARTpho-syllable, as a key component of out comparative analysis.

Another research focuses on tailoring the Transformer architecture for this specific task. Hieu Tran proposed a **Hierarchical Transformer Encoder** [2] that operates at both character and word levels. This hierarchical structure allows the model to capture both local spelling patterns (characters) and broader contextual information (words), which is particularly useful for resolving ambiguities in Vietnamese. Similarly, Do Dinh Truong presented **VSEC** [3], another Transformer-based model that utilizes byte-pair encoding tokenization and was trained on a dataset of real-world errors, highlighting the importance of domain-specific training data.

The existing research provides a strong foundation, demonstrating the power of Transformer-based models for Vietnamese error correction. However, a significant bottleneck remains: the scarcity of large-scale, publicly available parallel corpora of incorrect-correct sentence pairs. While some studies have released datasets, they are often limited in size or scope. Our project addresses this gap directly. Instead of solely relying on existing datasets, we introduce a comprehensive and systematic data augmentation pipeline designed to synthetically generate a diverse range of realistic errors, from phonetic and Telex-typing mistakes to structural and abbreviation errors. By fine-tuning established models like ViT5 and BARTPHO on this large-scale synthetic dataset, we aim to build a more robust and generalized correction system. Our primary contribution lies not in proposing a novel model architecture, but in developing a practical and scalable data-centric methodology to overcome the data scarcity challenge, and in conducting a rigorous, comparative analysis of leading models on this new data paradigm.

CHAPTER 2. IMPLEMENTATION METHODOLOGY

This chapter details the technical approach used to build the Vietnamese text correction system. Our methodology is centered around two key components: a comprehensive data augmentation pipeline to create a synthetic training corpus, and the fine-tuning of a state-of-the-art pre-trained Transformer model for the correction task.

2.1. Data Preparation and Augmentation

A crucial prerequisite for our data-centric approach is a high-quality, diverse, and clean source corpus. Recognizing the limitations of relying on a single domain, we constructed a custom Vietnamese text corpus by aggregating data from multiple reliable sources.

The final dataset consists of over 20,000 unique and clean Vietnamese sentences. This diversity is essential for training a model that can generalize well across different contexts and text types.

To ensure a fair and unbiased evaluation, we partitioned this corpus *before* any data augmentation took place. Specifically, we randomly selected 10% of the sentences (approximately 2,000 sentences) to form our test set. This test set was kept separate and was used exclusively for the final evaluation of the trained models. The remaining 90% of the sentences constituted the training corpus, which served as the clean source for our data augmentation pipeline. This strict separation prevents any data leakage from the training process to the evaluation phase, guaranteeing the integrity of our results.

Then, we developed a rule-based data augmentation pipeline to generate a large dataset of *(incorrect, correct)* sentence pairs from a clean Vietnamese text. The core idea is to programmatically introduce errors into clean sentences, creating realistic "noisy" counterparts that mimic the mistakes commonly made by human users.

To create a realistic incorrect sentence, our pipeline does not simply apply one type of error. Instead, for each clean sentence, it determines whether or not to create a noise for the sentence or not, if any then select randomly the augmentation techniques to apply to the original sentence based on the probability of being defined from the beginning. This ensures there will be various errors created in the final data set. The specific noise creation techniques, forming the construction blocks of this pipe, are detailed below.

2.1.1. Tone error

First, to simulate Vietnamese tone error in text, reflecting common mistakes that occur during typing or due to regional pronunciation variations. We check the validity of the input sentence to make sure it is processing on a string of characters, then determine the location of all vowels (both with and without tone marks) in the words. If no vowel is found, the word will be kept the same. In case of vowel containing, we will randomly select one or two vowel positions to apply the error. For each selected positions we will remove tone, change tone or add tone. The output is a version of the original word with altered tone marks, effectively simulating common mistyping tone error, contributing to a richer and more realistic training dataset for the spell correction model.

2.1.2. Typing Telex error

To simulate common typing error that occur when using Vietnamese Telex input method. The Telex method uses Latin characters and special characters ('w', 'a', 'e', 'o', 's', 'f', 'r', 'x', 'j') following vowels or consonants to represent Vietnamese characters with tone marks or special letters (e.g., 'aa' for 'â', 's' after a vowel for the acute tone). This function introduces errors by replacing a Vietnamese character with a tone mark or a special character with its corresponding Telex sequence. We identify the positions of characters in the word that represented by a Telex sequence (e.g., 'ă', 'â', 'đ', 'ê', 'ô', 'ơ', 'ư') then randomly selects one or two positions to apply the error. For each chosen position, we look up the corresponding Telex character sequence for the

Vietnamese letter at that position based on the map that we define in the configuration.

2.1.3. Spelling error

To simulate common spelling error, particularly those related to confusion between initial consonants, final consonants, or vowels that have similar sounds or are easily mixed up. We replace parts of a word (initial consonant, final consonant) with their corresponding misspelled variants.

2.1.4. Similar meaning error

To simulate errors related to the misuse of homophones, near-homophones, or words with similar meanings that are used incorrectly in context, leading to semantic or spelling errors within a sentence. Instead of just changing characters within a word, this function operates at the word level, replacing an original word with another from a predefined list of easily confused words.

2.1.5. Structure error

To simulate the sentence structure, we perform basic operations such as deleting words, duplicating words, or swapping the positions of two adjacent words randomly, we leverage **Part-of-Speech (POS) tags** to make more informed decisions about which words to target for these operations, aiming to simulate grammatically incorrect structures that are more likely to occur in real-world Vietnamese.

In case delete word, if the token list has at least two words, we will randomly select an index prioritize prepositions, conjuntions, adverbs by using information from POS tags and deletes the word at that position.

In case duplicate word, if the token list has at least one word, we randomly select an index and inserts a copy of the word at that position immediately after the original word, prioritize duplicating certain parts of speech (e.g., adjectives, prepositions, nouns).

In case swap adjacent words, we randomly select an index (not the last word) and swaps the position of that word with the word immediately following it.

By utilizing POS tags, we move beyond purely random structural changes, generating more realistic and linguistically-motivated grammatical errors to enhance the training data.

2.1.6. Abbreviation error

To simulate the use of common abbreviations or slang in informal communication, particularly in messaging or online chats, we replace full words or phrases with their corresponding abbraviated or slang versions based on predefined mapping table.

2.2. Model Architecture

2.2.1. Model ViT5

ViT5 (Vietnamese Text-to-Text Transformer) is the Vietnamese variant of the T5 model (Text-to-Text Transfer Transformer) developed by Google. Different from the classification models, T5 and ViT5 operate according to model Text-to-Text: all NLP tasks from the machine, summarize to classifying or fixing errors which is expressed in the form of conversion of the input chain into the output chain.

ViT5 is trained by VietAI on a large-scale Vietnamese data warehouse (including Wikipedia, press, forum), helping it master the grammar, bar, vocabulary as well as the characteristics of Vietnamese phonemic and syntax.

⁕ **The reason for choosing Vit5:**

Trained entirely in Vietnamese, so it understands grammar, structure, vocabulary, and Vietnamese language characteristics (including bar marks, homonyms, ...). Seq2Seq architecture helps VIT5 very suitable for the problem of converting the wrong sentence to the correct sentence, similar to a "language translation" task.

Very easy to fine-tune, available on Hugging Face with versions such as vit5-base, vit5-large.

⁕ **Fine-Tune process:**

ViT5 is loaded from “VietAI/vit5-base”. The input data is the wrong pairs of sentences and the correct sentences born from Pipeline Data Augmentation.

Preprocessing: Each input sentence is added to "sửa lỗi:" to define the task. Both wrong sentences and correct sentences are tokenized, padding and trucation in terms of maximum length.

Set Trainer: Use Seq2SeqTrainer with the main parameters:

- Epochs: 3
- Batch size: 8
- Beam: 2

⁕ **Training process:**

The model is trained through many loops (Epochs), with batch size and super parameters such as learning rate, weight decay and warmup ratio are reasonably set.

Gradient accumulation mechanism is used to fake larger-sized processors when the GPU limits memory.

The mixed-precision mechanism (FP16) is applied, increasing and reducing memory consumption.

After each epoch or after a certain step, the model of the sentence fixes on the evaluation, and measurements such as loss, the number of correct correction errors are monitored for adjustment.

The checkpoints are periodically saved, limiting the total number of storages to save space. Having “best_model_checkpoint”, if activated, the model will choose the most appreciated results on the test set will be marked and used for Inference.

⁕ **Inference process:**

The model receiving sentences containing errors, standardizing and adding “sửa lỗi” to suggest duties, then produces bug corrections through beam search with beams digital configuration and maximum length to collect many proposals.

From this result, we choose the optimal output based on the highest probability. The result was then assessed by the main indicators: WER (Word Error Rate) and CER (Character Error Rate) measuring different characters and words; Along with Precision, Recall, F1 on the token level modifications to accurately assess the correct editing number, excess and lack of correction.

Finally, the entire evaluation results and details of each case are saved to analyze and improve the model.

To illustrate the effectiveness of ViT5, we perform a test on a series of common errors. Example: "chương trỉnhnh được páht sóng vào lúc 19h", "chúc mừng bạnn đã trúng giải nhất", "công nghề thônngg tin đáng phát chiển rất nhanh", "tôi mún đi chơi với bạn bè cuỗi tuần này”, … These examples show that the model has the ability to identify and edit the bar, telex, word, and grammar errors in Vietnamese.

⁕ **Conclusion:**

The process of using VIT5 in the problem of fixing Vietnamese errors includes specialized pre-training knowledge, defining the text-to-text task, careful training through many Epochs with continuous evaluation mechanism, and finally inference with beam search and comprehensive evaluation. This structure ensures an accurate mapping model between the sentence containing the error and the corrected sentence.

2.2.2. Model BARTPHO

BARTPHO-Syllable is a Vietnamese-adapted variant of the BART architecture, released by VinAI, that operates at the syllable level to capture nuanced tone and syllable patterns in Vietnamese text.

BART itself is a denoising sequence-to-sequence model combining a bidirectional encoder (inspired by BERT) and an autoregressive decoder (inspired by GPT).

During pre-training, BART learns to reconstruct original text from corrupted inputs, making it well suited to error correction tasks.

⁎ **The reason for choosing BARTPHO:**

It leverages a syllable-level tokenization tailored for Vietnamese, enabling precise handling of tone marks and diacritics.

Its denoising pre-training objective directly aligns with the task of correcting noisy text, as the model has been trained to recover clean text from various forms of corruption.

⁎ **Fine-Tune process:**

Initializing from the pretrained checkpoint “vinai/bartpho-syllable-base” and fine-tune on the same synthetic dataset of erroneous and correct sentence pairs used for ViT5.

Inputs are normalized and prefixed with “sửa lỗi:” to indicate the correction task. Tokenization and sequence preparation mirror the ViT5 process but employ a syllable-centric tokenizer.

Set Trainer: Use Seq2SeqTrainer with the main parameters:

- Epochs: 3
- Batch size: 8
- Beam: 2

⁎ **Training process:**

Fine-tuning runs for three epochs with a batch size of eight. During validation, we apply beam search with two beams to generate correction candidates.

Optimization employs gradient accumulation for performance under GPU memory constraints and mixed-precision training where supported.

Checkpoints are saved periodically, and the best-performing checkpoint on the validation set is selected for inference.

⁕ Inference process:

For inference, the normalized, prefixed input is tokenized and passed through the model to produce multiple candidate corrections via beam search.

The highest-scoring candidate is detokenized and post-processed. Evaluation metrics include WER and CER for token- and character-level accuracy, alongside precision, recall, and F1 computed on token-level edit actions to assess correction accuracy, overcorrections, and omissions.

All results, including summary metrics and detailed per-example logs, are stored for further analysis.

To illustrate the effectiveness of BARTPHO-Syllable, we perform a test on a series of common errors. Example: "chương trỉnhnh được páht sóng vào lúc 19h", "chúc mừng bạnn đã trúng giải nhất", "công nghề thônngg tin đáng phát chiển rất nhanh", "tôi mún đi chơi với bạn bè cuỗi tuần này", … These examples show that the model has the ability to identify and edit the bar, telex, word, and grammar errors in Vietnamese.

⁕ Conclusion:

BARTPHO-Syllable complements ViT5 by combining denoising pre-training with syllable-level tokenization. Its fine-tuned adaptation naturally suits the text-correction objective, offering an alternative yet equally effective approach to Vietnamese error correction.

CHAPTER 3. RESULT AND ANALYSIS

To evaluate the effectiveness of our trained models, we conducted a comprehensive analysis on the held-out test set. As described in Chapter 2, this test set comprises approximately 2,000 sentences sourced from our diverse, multi-domain corpus and was not seen by the models during training. The evaluation relies on a combination of standard metrics to provide a holistic view of model performance.

3.1. Evaluation Metrics

To evaluate the effectiveness of the Vietnamese spelling and grammar correction model, we employed a set of standard metrics commonly used in Natural Language Processing, particularly for error correction and machine translation tasks. These metrics can be broadly categorized into two groups: Edit Distance-based metrics and Error Analysis-based metrics.

3.1.1. Edit Distence-based Metrics

Character Error Rate (CER): The percentage of the total number of character insertions, deletions, and substitutions required to transform the predicted sentence into the reference (correct) sentence, divided by the total number of characters in the reference sentence. CER is a useful indicator of accuracy at the character level and is highly sensitive to typing or tone mark errors.

Word Error Rate (WER): The percentage of the total number of word insertions, deletions, and substitutions required to transform the predicted sequence of words into the reference sequence, divided by the total number of words in the reference sentence. WER evaluates accuracy at the word level, reflecting the model's ability to correct words and sentence structure.

3.1.2. Error Analysis-based Metrics

These metrics analyze the differences between the input sentence (erroneous), the model's predicted sentence (corrected), and the reference sentence (perfect) to determine whether the model correctly fixed original errors or introduced new ones.

- **True Positives (TP):** The number of original errors present in the input sentence that the model successfully corrected.
- **False Positives (FP):** The number of changes the model made to the input sentence that were not actually errors or introduced new errors.
- **False Negatives (FN):** The number of original errors present in the input sentence that the model failed to correct.
- **Total Errors:** The total number of original errors present in the input sentence (TP + FN).
- **Total Model Edits:** The total number of changes the model made to the input sentence (TP + FP). From TP, FP, and FN, we calculate aggregate metrics:
- **Precision:** The proportion of the model's changes that were actually correct error fixes (TP / (TP + FP)). This metric indicates the reliability of the modifications suggested by the model.
- **Recall:** The proportion of original errors in the input sentence that the model successfully detected and corrected (TP / (TP + FN)). This metric indicates the model's ability to find and fix all errors.
- **F1-score:** The harmonic mean of Precision and Recall (2 * (Precision * Recall) / (Precision + Recall)). F1-score provides a balanced view of the model's performance, especially useful when the error distribution is uneven.

3.2. Rationale for Metric Selection

The combined selection of Edit Distance-based metrics (CER, WER) and Error Analysis-based metrics (Precision, Recall, F1-score) is crucial for a Vietnamese spelling and grammar correction project because:

- **Comprehensive Evaluation:** CER and WER offer an overall view of the output sentence's accuracy compared to the reference sentence at both character and word levels, reflecting the overall quality of the correction.

- **Analysis of Correction Behavior:** TP, FP, FN, Precision, Recall, and F1-score provide a deeper analysis of how the model interacts with original errors. They help answer key questions such as: How many actual errors did the model fix? Does the model tend to "over-correct" and introduce many new errors (high FP)? Does the model miss many original errors (high FN)?
- **Suitability for Vietnamese Error Nature:** Vietnamese errors are diverse, ranging from tone mark errors, consonant/vowel confusions (affecting CER, WER) to misuse of words in context and abbreviations (primarily affecting WER and requiring error analysis for deeper understanding). The combination of metrics helps capture different facets of errors and the model's correction capabilities.
- **Balance between Error Correction and Correct Text Preservation:** Precision and Recall help balance the ability to fix errors (Recall) with the need to avoid corrupting correct text (Precision). A model with high Recall but low Precision might fix many errors but introduce many new ones; conversely, a model with high Precision but low Recall might introduce few new errors but miss many original ones. F1-score helps find this balance.

3.3. Analysis of Achieved Result

We evaluated the performance of two models: ViT5-base and BARTPHO-syllable on the same test dataset for comparison. The detailed evaluation results are summarized in **Table 3.1**, and a visual comparison is provided in **Figure 3.1**.

***Table 3.1**: Evaluation results of ViT5 vs BARTPHO-syllable.*

Metric	**ViT5-base**	**BARTPHO-syllable**
CER	0.0599	0.0793
WER	0.2057	0.3398
Precision	0.3079	0.1607
Recall	0.5838	0.4607
F1-score	0.4031	0.2383
True Positives	4004	3144
False Positives	9002	16419
False Negatives	2855	3681

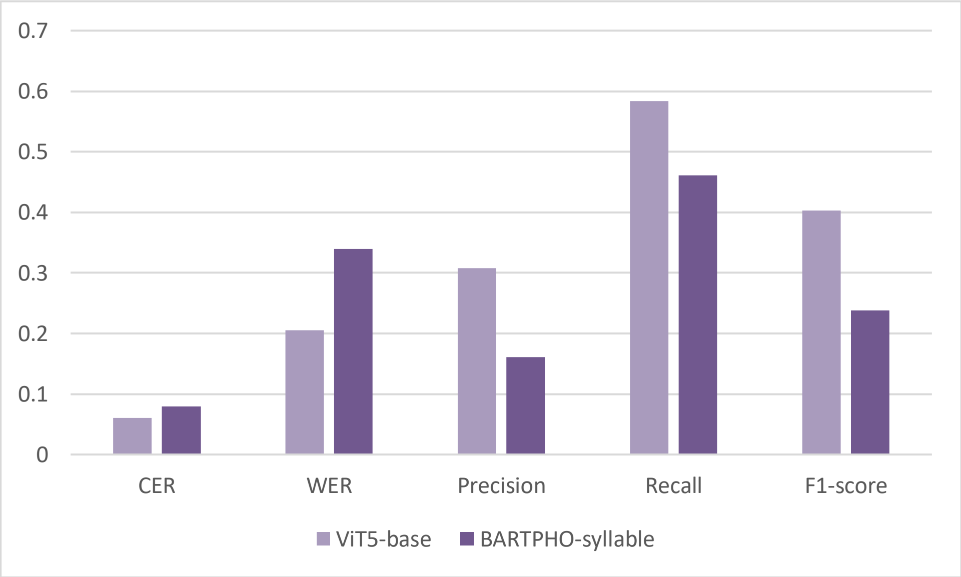


***Figure 3.1**: Overall performance comparison of ViT5-base and BARTPHO-syllable.*

Based on these results, we can make the following analysis:

CER (0.0599 vs 0.0793) and WER (0.2057 vs 0.3398): The ViT5 model demonstrates significantly superior performance compared to BART at both character and word levels. BART's WER is substantially higher, indicating that BART struggles more to produce accurate output sentences relative to the reference. This could be due to BART introducing more word additions/deletions/substitutions or incorrect sentence structures.

Error Analysis (TP, FP, FN):

- Both models faced a comparable number of original errors in the test set (around 6800 errors).
- The ViT5 model correctly fixed **4004** original errors, whereas BART only managed to fix **3144**. ViT5 has a more effective error detection and correction capability.
- The biggest difference lies in the number of **False Positives**. BART generated a staggering **16419** unnecessary changes, nearly double the **9002** of ViT5. This indicates that BART has a much more severe tendency to "over-correct" and introduce new errors.
- Regarding **False Negatives**, BART missed **3681** original errors, more than the **2855** errors missed by ViT5. BART not only makes more incorrect changes but also misses more errors.
- The **Total Model Edits** reinforce this observation: BART performed 19563 changes, while ViT5 performed only 13006. A large portion of BART's changes were incorrect.

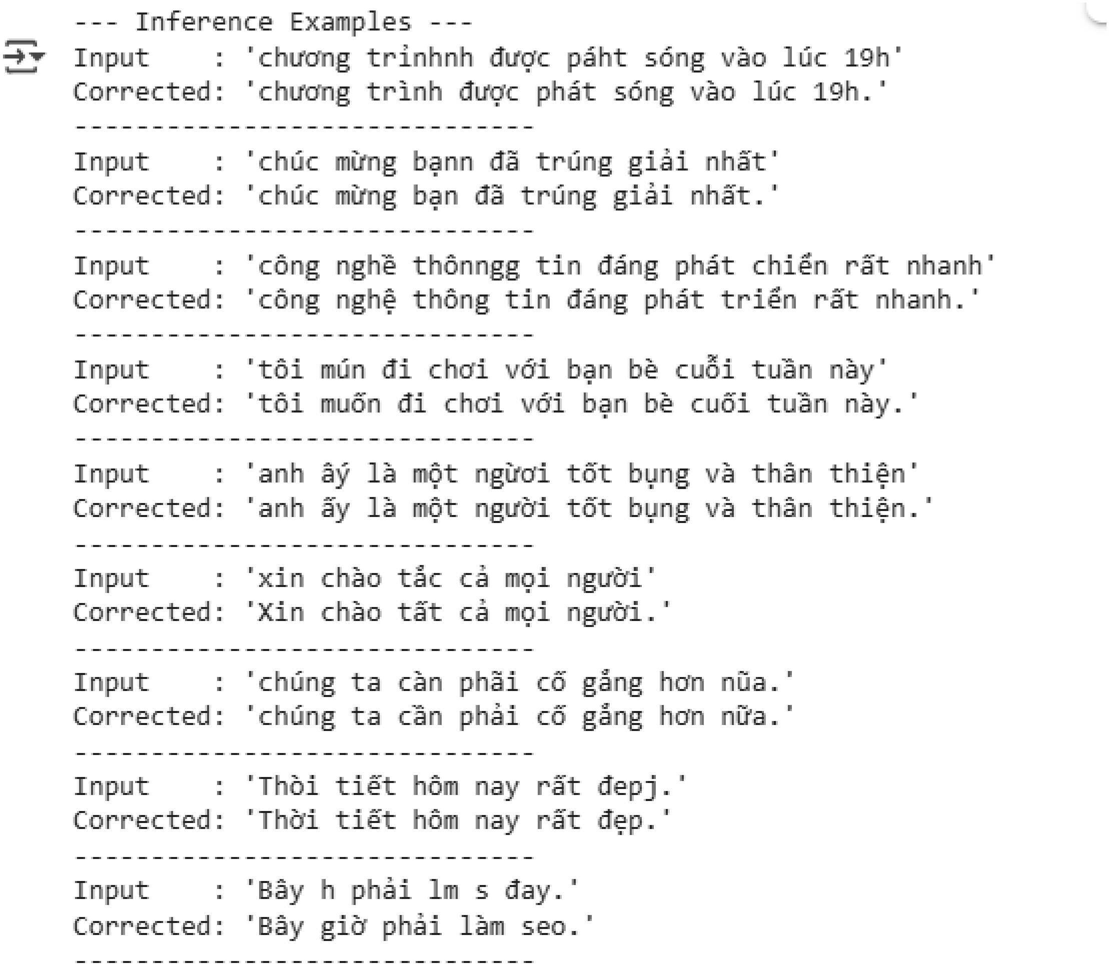
Precision (0.3079 vs 0.1607), Recall (0.5838 vs 0.4607), F1-score (0.4031 vs 0.2383):

- The ViT5 model shows a significantly better balance between Precision and Recall compared to BART.
- **Recall:** ViT5 has higher Recall (0.5838 vs 0.4607), confirming its better ability to detect and fix original errors.
- **Precision:** This is a clear weakness for both models, but BART is significantly worse. BART's Precision is only 0.1607, while ViT5 achieves 0.3079. This means that among all changes BART made, only about 16% were correct error fixes, while the remaining 84% were incorrect or unnecessary. Although ViT5's Precision is still low, it is much more reliable than BART.

- **F1-score:** ViT5's F1-score (0.4031) is considerably higher than BART's (0.2383), indicating that ViT5's overall performance is significantly superior for this task.

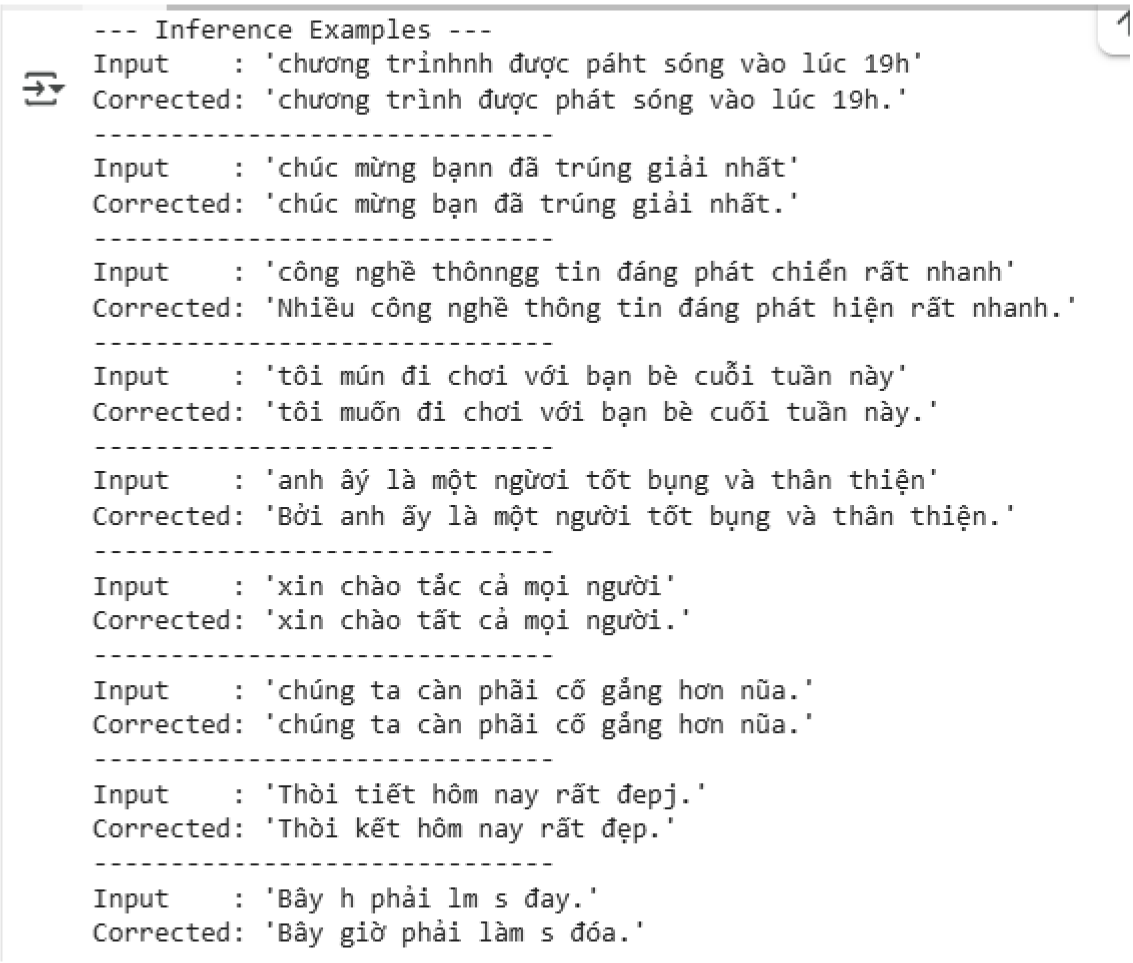
Both the ViT5 and BARTPHO-syllable models demonstrate a certain level of capability in Vietnamese spelling and grammar correction on the test dataset. However, the ViT5 model has shown clear superior performance across most metrics. ViT5 not only has a better ability to detect and correctly fix more original errors (higher Recall, TP) but, more importantly, introduces significantly fewer new errors or incorrect modifications to correct parts (lower Precision, FP, lower Total Model Edits). Consequently, ViT5's CER, WER, and F1-score are much better than BART's.

A common weakness for both models is the relatively low Precision, especially for BART, indicating that there is still a challenge in ensuring that every change the model makes is accurate and does not corrupt the original correct text. However, ViT5 has achieved a much better balance between correcting errors and preserving the correctness of the text.



```
--- Inference Examples ---
Input    : 'chương trỉnhnh được páht sóng vào lúc 19h'
Corrected: 'chương trình được phát sóng vào lúc 19h.'
------------------------------
Input    : 'chúc mừng bạnn đã trúng giải nhất'
Corrected: 'chúc mừng bạn đã trúng giải nhất.'
------------------------------
Input    : 'công nghề thônngg tin đáng phát chiển rất nhanh'
Corrected: 'công nghệ thông tin đáng phát triển rất nhanh.'
------------------------------
Input    : 'tôi mún đi chơi với bạn bè cuỗi tuần này'
Corrected: 'tôi muốn đi chơi với bạn bè cuối tuần này.'
------------------------------
Input    : 'anh âý là một ngừơi tốt bụng và thân thiện'
Corrected: 'anh ấy là một người tốt bụng và thân thiện.'
------------------------------
Input    : 'xin chào tắc cả mọi người'
Corrected: 'Xin chào tất cả mọi người.'
------------------------------
Input    : 'chúng ta càn phãi cố gẳng hơn nũa.'
Corrected: 'chúng ta cần phải cố gẳng hơn nữa.'
------------------------------
Input    : 'Thòi tiết hôm nay rất đepj.'
Corrected: 'Thời tiết hôm nay rất đẹp.'
------------------------------
Input    : 'Bây h phải lm s đay.'
Corrected: 'Bây giờ phải làm seo.'
------------------------------
```

***Figure 3.2**: Inference examples from the ViT5-base model*



```
--- Inference Examples ---
Input    : 'chương trỉnhnh được páht sóng vào lúc 19h'
Corrected: 'chương trình được phát sóng vào lúc 19h.'
------------------------------
Input    : 'chúc mừng bạnn đã trúng giải nhất'
Corrected: 'chúc mừng bạn đã trúng giải nhất.'
------------------------------
Input    : 'công nghề thônngg tin đáng phát chiển rất nhanh'
Corrected: 'Nhiều công nghề thông tin đáng phát hiện rất nhanh.'
------------------------------
Input    : 'tôi mún đi chơi với bạn bè cuỗi tuần này'
Corrected: 'tôi muốn đi chơi với bạn bè cuối tuần này.'
------------------------------
Input    : 'anh âý là một ngừơi tốt bụng và thân thiện'
Corrected: 'Bởi anh ấy là một người tốt bụng và thân thiện.'
------------------------------
Input    : 'xin chào tắc cả mọi người'
Corrected: 'xin chào tất cả mọi người.'
------------------------------
Input    : 'chúng ta càn phãi cố gắng hơn nũa.'
Corrected: 'chúng ta càn phãi cố gắng hơn nũa.'
------------------------------
Input    : 'Thòi tiết hôm nay rất đepj.'
Corrected: 'Thòi kết hôm nay rất đẹp.'
------------------------------
Input    : 'Bây h phải lm s đay.'
Corrected: 'Bây giờ phải làm s đóa.'
```

Figure 3.3*: Inference examples from the BARTPHO-syllable model.*

To provide a more granular, qualitative insight into the performance differences, we present a selection of inference outputs from both models. **Figure 3.2** showcases the typical behavior of the **ViT5** model, while **Figure 3.3** illustrates the outputs from **BARTPHO**-syllable on the same set of incorrect inputs.

Looking closely at the results, it's clear that **ViT5-base** performs better in most situations. It consistently gives more accurate and sensible corrections. However, it's not perfect and still has some weaknesses. A good example is the sentence 'Bây h phải lm s đay.'. Here, the model did a decent job fixing some of the abbreviated words, but its final answer, 'Bây giờ phải làm seo.', was still not completely correct due to a spelling mistake. This shows that ViT5 can sometimes struggle with very informal slang.

On the other hand, the **BARTPHO-syllable** model shows a more serious and frequent problem. We noticed that it often adds extra, unnecessary words to sentences. This is a big issue because it ends up “correcting” parts of a sentence that were already right, turning them into mistakes. This unreliable behavior is the main reason why BARTPHO has such a high number of ‘False Positives’ in our results, as it often does more harm than good.

CHAPTER 4. CONCLUSION AND FUTURE WORK

4.1. Conclusion

Our project researches and develops a spelling error system for Vietnamese based on Transformer architecture. By enhancing data with diverse error simulation techniques, we have built a set of data sets and appropriate evaluation.

Detailed evaluation results on test data shows that Vit5-Base model has a better performance when compared to the Bart model on most measurements, showing that VIT5 has the ability to identify and significantly repair spelling errors that exist in the text.

However, through the True Positives, False Positives we realize that both models, especially the bart tends to overdo it and create new errors, this indicates that a significant rate of changes that the model is inaccurate or unnecessary. In general, the models have shown the potential for application in fixing Vietnamese errors, but there should be improvements to better balance between the ability to fix errors and ensure the accuracy of the original text.

4.2. Future work

To improve performance and solve the current limitations, we propose some future development directions:

Improving Data Augmentation:

- Diversify errors: Research and add more complex and less common simulation techniques, based on actual analysis of user data.
- Error quality control: Tinning probability and rules in the error functions to ensure that errors are created more realistic and less excessive interference.
- Using actual data: Searching and integrating data pairs of sentences (wrong, right) from sources such as forums, social networks, or projects collecting community spelling data to supplement strengthening data.

Model and Training Refinement:

- Testing other model architecture: Explore alternative Seq2Seq model variants or architectures that might be better suited for the correction task (e.g., models with better attention mechanisms focused on error regions).
- Hyperparameter tuning (HyperParameter Tuning): Conduct systematic hyperparameter search to optimize the training process, specifically focusing on parameters affecting the "over-correction" tendency.
- Apply regularization techniques: Employ techniques like Dropout, stronger Weight Decay, or other regularization methods to reduce overfitting and improve generalization ability.
- Multi-Task Learning: Combine spelling correction training with other related tasks such as error detection or error type classification to help the model gain a deeper understanding of the nature of mistakes.

Advanced Evaluation:

- Classification of errors: Building a toolkit or process to classify the original errors and errors created by the model by type (wrong sign, wrong consonant, grammar, etc.) to have a more detailed view of the strength/weakness of the model with each specific type of error.
- Evaluation by people: Make a manual assessment of the results of the model by people to grasp the aspects of the natural, fluency and semantics that automatic measurements can be missed.

CHAPTER 5: PROJECT MANAGEMENT AND TEAMWORK

5.1. Team Collaboration and Communication

Primary Communication Channel: Zalo, Messenger

For daily communication, task coordination, and rapid problem-solving, our team using Zalo and Messenger platform. They were chosen for its ubiquity, ease of use, and real-time notification capabilities, which proved essential for maintaining a dynamic workflow. The group chat served as our virtual workspace for:

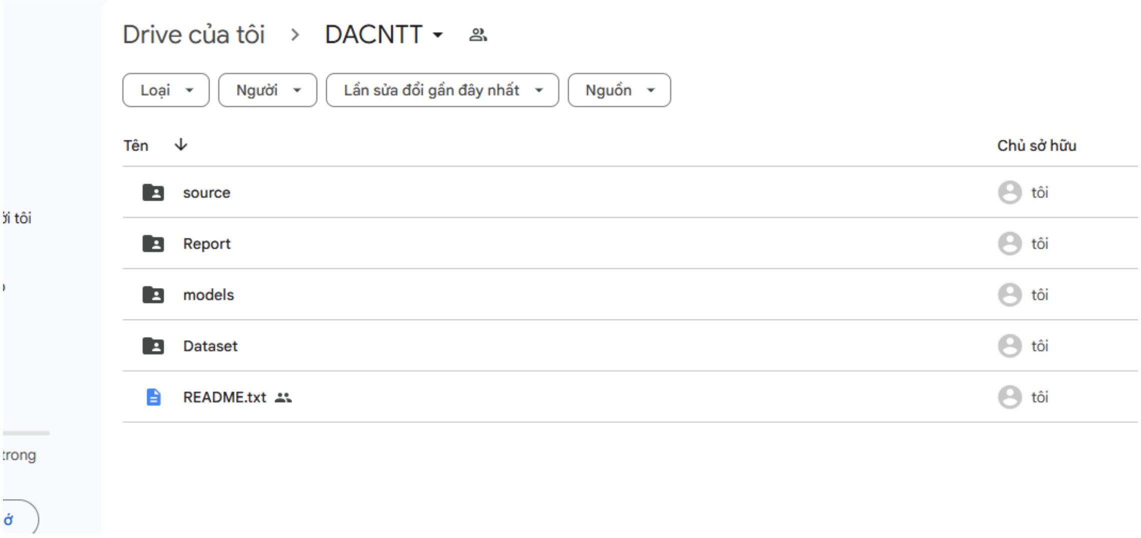
- **Daily Stand-ups:** Members briefly shared their progress, discussed any roadblocks, and planned their tasks for the day.
- **Brainstorming Sessions:** Key decisions regarding model selection, data augmentation techniques, and evaluation strategies were discussed and finalized within the group.
- **Instant Support:** Whenever a member encountered a technical issue or a conceptual question, the platform allowed for immediate assistance from the rest of the team.

5.2. Resource and Code Management

Centralized Repository: Google Drive

To manage all project-related assets, including source code, datasets, research papers, and progress reports, we established a shared Google Drive folder. This centralized repository served as our single source of truth and was structured to maintain organization and version control

By using Google Drive, we ensured that every team member had access to the latest versions of all files, which minimized conflicts and streamlined the process of integrating individual contributions into the final project.



Tên ↓	Chủ sở hữu
source	tôi
Report	tôi
models	tôi
Dataset	tôi
README.txt	tôi

Figure 5.1: *Management of Project Assets via Google Drive.*

REFERENCES

[1] N. L. T. e. a. (VinAI), “BARTpho: Pre-trained Sequence-to-Sequence Models for Vietnamese,” 20 Sep 2021. [Trực tuyến]. Available: https://arxiv.org/abs/2109.09701.

[2] Hieu Tran, Cuong V. Dinh, Long Phan, Son T. Nguyen (2021) , ““Hierarchical Transformer Encoders for Vietnamese Spelling Correction”,” 21 May 2021. [Trực tuyến]. Available: https://arxiv.org/abs/2105.13578.

[3] Do Dinh Truong, Nguyen Ha Thanh, Bui Thang Ngoc, Vo Hieu Dinh , “VSEC: Transformer-Based Model for Vietnamese Spelling Correction”,” [Trực tuyến]. Available: https://arxiv.org/abs/2111.00640.

[4] Long Phan, Hieu Tran, Hieu Nguyen, Trieu H. Trinh, 16 02 2022. [Trực tuyến]. Available: https://research.vietai.org/vit5/.

[5] “BART: Sự kết hợp giữa BERT và GPT,” [Trực tuyến]. Available: https://trituenhantao.io/kien-thuc/bart-su-ket-hop-giua-bert-va-gpt/.

[6] B. Q. Manh, “Thử áp dụng mô hình dịch máy vào bài toán tự động sửa lỗi tiếng Việt,” [Trực tuyến]. Available: https://viblo.asia/p/thu-ap-dung-mo-hinh-dich-may-vao-bai-toan-tu-dong-sua-loi-tieng-viet-maGK7vJB5j2.

[7] Rohit Raju1,2, Peeta Basa Pati*,2, SA Gandheesh2, Gayatri Sanjana Sannala2 & Suriya KS2 , “Grammatical vs Spelling error correction: An”.

[8] N. C. Thang, "“Oánh giá” model AI theo cách Mì ăn liền – Chương 2. Precision, Recall và F Score,” [Trực tuyến]. Available: https://miai.vn/2020/06/16/oanh-gia-model-ai-theo-cach-mi-an-lien-chuong-2-precision-recall-va-f-score/.

[9] Long Phan1,2, Hieu Tran1, Hieu Nguyen1,2, Trieu H. Trinh1,3, “ViT5: Pretrained Text-to-Text Transformer for Vietnamese Language,” 2022.

[10] Tamanna, “Deciphering Accuracy: Evaluation Metrics in NLP and OCR- A Comparison of Character Error Rate (CER) and Word Error Rate (WER),” 8 Feb 2024. [Trực tuyến]. Available: https://medium.com/@tam.tamanna18/deciphering-accuracy-evaluation-metrics-in-nlp-and-ocr-a-comparison-of-character-error-rate-cer-e97e809be0c8.

[11] C. Bronsdon, “Evaluating AI Models: Understanding the Character Error Rate (CER) Metric,” 26 Mar 2025. [Trực tuyến]. Available: https://galileo.ai/blog/character-error-rate-cer-metric.

[12] “Under The Sea - Vietnamese NLP Toolkit,” [Trực tuyến]. Available: https://underthesea.readthedocs.io/en/v1.1.4/readme.html.

[13] M. Neri, “Part-of-speech (POS) Tagging In NLP,” 24 Jan 2023. [Trực tuyến]. Available: https://spotintelligence.com/2023/01/24/part-of-speech-pos-tagging-in-nlp-python/.

[14] S. Mudadla, “What is Parts of Speech (POS) Tagging Natural Language Processing?In what kind of applications we can use Parts of Speech (POS) Tagging in Natural Language Processing.,” 9 Nov 2023. [Trực tuyến]. Available: https://medium.com/@sujathamudadla1213/what-is-parts-of-speech-pos-tagging-natural-language-processing-in-2b8f4b07b186.

[15] D. N. T. e. al, “Vietnamese Spelling Error Detection and Correction Using BERT and N-gram Language Model,” Jan 2022. [Trực tuyến]. Available: https://www.researchgate.net/publication/362085698_Vietnamese_Spelling_Error_Detection_and_Correction_Using_BERT_and_N-gram_Language_Model.

[16] T. H. N. e. al., “A Combination of BERT and Transformer for Vietnamese Spelling Correction,” 4 May 2024. [Trực tuyến]. Available: https://arxiv.org/abs/2405.02573.