ĐẠI HỌC QUỐC GIA TPHCM

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

Báo cáo đồ án cuối kỳ

Project: Vietnamese Spelling Correction

Môn học: Nhập môn xử lý ngôn ngữ tự nhiên

Sinh viên thực hiện:

Nguyễn Lê Quang (23127109)

Đỗ Ngọc Minh Tuấn (23127137)

Lê Quốc Thiện (23127481)

Phạm Quang Thịnh (23127485)

Giảng viên hướng dẫn:

PGS.TS. Đinh Điền

TS. Nguyễn Hồng Bửu Long

TS. Lương An Vinh

Ngày 6 tháng 1 năm 2026

Mục lục

Lời cảm ơn

Lời đầu tiên, nhóm chúng em xin gửi lời cảm ơn chân thành đến Ban Giám hiệu và quý Thầy Cô trường Đại học Khoa học Tự nhiên, ĐHQG-HCM đã tạo điều kiện môi trường học tập và nghiên cứu tốt nhất cho chúng em.

Đặc biệt, chúng em xin bày tỏ lòng biết ơn sâu sắc đến quý Thầy phụ trách bộ môn **Nhập môn Xử lý ngôn ngữ tự nhiên (Lớp 23CNTThuc2)**:

- **Thầy Đinh Điền**
- **Thầy Nguyễn Hồng Bửu Long**
- **Thầy Lương An Vinh**

Cảm ơn quý Thầy đã tận tình giảng dạy, truyền đạt những kiến thức nền tảng quý báu cũng như cập nhật những xu hướng công nghệ mới nhất trong lĩnh vực NLP. Những bài giảng tâm huyết và sự hướng dẫn của quý Thầy chính là cơ sở quan trọng giúp chúng em định hình phương pháp và hoàn thành đồ án này.

Chúng em cũng xin gửi lời cảm ơn đến chị **Hương Lâm** (trợ giảng) đã nhiệt tình hỗ trợ, giải đáp thắc mắc và đưa ra những định hướng quý báu trong suốt quá trình thực hiện đồ án. Sự giúp đỡ của chị đã giúp nhóm vượt qua nhiều khó khăn kỹ thuật quan trọng.

Bên cạnh đó, nhóm xin gửi lời tri ân đến đội ngũ nghiên cứu **VinAI Research** và tác giả **bmd1905** vì đã công khai các mô hình và bộ dữ liệu mã nguồn mở giá trị, tạo nền tảng vững chắc cho cộng đồng nghiên cứu NLP tiếng Việt.

Mặc dù đã nỗ lực hết mình, nhưng do giới hạn về thời gian và kinh nghiệm, đồ án khó tránh khỏi thiếu sót. Chúng em rất mong nhận được những ý kiến đóng góp từ quý Thầy để hoàn thiện hơn trong tương lai.

Chúng em xin chân thành cảm ơn!

Tóm tắt nội dung

Báo cáo này trình bày phương pháp tiếp cận bài toán sửa lỗi chính tả tiếng Việt dựa trên kiến trúc Sequence-to-Sequence của mô hình Bartpho. Nghiên cứu tập trung vào việc tận dụng hạ tầng tính toán hiệu năng cao (NVIDIA A100) để huấn luyện mô hình trên bộ dữ liệu cặp câu (Input-Output) trích xuất từ văn bản sách. Mục tiêu là xây dựng một hệ thống có khả năng phục hồi văn bản nhiễu về định dạng chuẩn với độ chính xác cao.

1 Giới thiệu

1.1 Tính cấp thiết của đề tài

Trong kỷ nguyên số hóa, dữ liệu văn bản tiếng Việt bùng nổ trên các nền tảng mạng xã hội và thương mại điện tử đi kèm với tình trạng nhiễu loạn thông tin, đặc biệt là lỗi chính tả và lỗi gõ máy (typo). Việc chuẩn hóa văn bản là bước tiền xử lý bắt buộc để đảm bảo độ chính xác cho các bài toán xử lý ngôn ngữ tự nhiên (NLP) phía sau như Dịch máy hay Chatbot.

Trước đây, các phương pháp truyền thống như N-gram hay các mạng nơ-ron hồi quy (RNN, LSTM) thường gặp hạn chế về khả năng ghi nhớ ngữ cảnh dài và tốc độ huấn luyện chậm. Tuy nhiên, sự ra đời của kiến trúc **Transformer** và các mô hình ngôn ngữ tiền huấn luyện (Pre-trained Language Models) đã tạo ra bước đột phá lớn trong việc mô hình hóa ngôn ngữ tự nhiên, cho phép nắm bắt các phụ thuộc xa và ngữ nghĩa phức tạp của văn bản. Ví dụ tiêu biểu nhất là các mô hình nền tảng như BERT, GPT, BART hay BARTpho.

Đặc biệt, **BARTpho** - một mô hình Sequence-to-Sequence dựa trên kiến trúc BART được VinAI huấn luyện riêng cho tiếng Việt - đã đạt kết quả vượt trội (State-of-the-art) trên nhiều tác vụ. Việc tận dụng tri thức ngôn ngữ khổng lồ mà BARTpho đã học được để giải quyết bài toán sửa lỗi chính tả (thông qua kỹ thuật Fine-tuning) là hướng đi hiện đại, giúp tiết kiệm tài nguyên tính toán và đạt độ chính xác cao hơn so với việc huấn luyện mô hình từ đầu.

1.2 Tuyên bố bài toán

Bài toán được định nghĩa là một tác vụ "dịch" từ văn bản lỗi sang văn bản chuẩn (Text Correction as Translation).

- **Đầu vào:** Chuỗi văn bản chứa lỗi chính tả X (ví dụ: "tri tue nhan tao").

- **Đầu ra:** Chuỗi văn bản chính xác Y (ví dụ: "trí tuệ nhân tạo").

Mô hình sẽ học xác suất có điều kiện $P(Y|X)$ để sinh ra câu đúng dựa trên ngữ cảnh toàn cục của câu sai.

1.3 Mục tiêu nghiên cứu

Đề tài tập trung vào việc ứng dụng mô hình ngôn ngữ lớn cho tiếng Việt với các mục tiêu cụ thể:

1. **Nghiên cứu lý thuyết:** Tìm hiểu kiến trúc Transformer, cơ chế Denoising Autoencoder của BART và thư viện Hugging Face.
2. **Chuẩn bị dữ liệu:** Xử lý bộ dữ liệu huấn luyện song ngữ (câu lỗi - câu đúng) quy mô 60.000 cặp câu.
3. **Thực nghiệm mô hình:** Thực hiện tinh chỉnh (Fine-tuning) mô hình `vinai/bartpho-syllable` để thích nghi với tác vụ sửa lỗi chính tả.
4. **Đánh giá:** Đo lường hiệu suất mô hình thông qua các chỉ số định lượng: BLEU score, Character Error Rate (CER), Word Error Rate (WER) và ROUGE score.

1.4 Đối tượng và Phạm vi nghiên cứu

- **Đối tượng nghiên cứu:** Các loại lỗi chính tả tiếng Việt (không dấu, sai dấu, sai phụ âm) và mô hình BARTpho.
- **Phạm vi dữ liệu:** Bộ dữ liệu *Vietnamese Correction 60k* bao gồm các câu văn thuộc nhiều lĩnh vực khác nhau.
- **Phạm vi kỹ thuật:** Sử dụng kỹ thuật Transfer Learning trên nền tảng PyTorch và thư viện Transformers.

1.5 Đóng góp của đề tài

Nghiên cứu này đóng góp vào lĩnh vực xử lý ngôn ngữ tiếng Việt trên các khía cạnh chính sau:

- **Về mặt phương pháp:** Chứng minh tính hiệu quả vượt trội của việc áp dụng mô hình ngôn ngữ tiền huấn luyện (Pre-trained Model) **BARTpho** kết hợp với kỹ thuật tinh chỉnh (Fine-tuning) cho bài toán sửa lỗi chính tả, thay thế các phương pháp N-gram hay LSTM truyền thống.
- **Về mặt thực tiễn:** Cung cấp một mô hình hoàn chỉnh (fine-tuned checkpoint) và mã nguồn mở, có khả năng tích hợp thực tế vào các hệ thống gõ tiếng Việt, Chatbot hoặc công cụ chuẩn hóa văn bản (Text Normalization).
- **Về mặt thực nghiệm:** Thực hiện đánh giá toàn diện mô hình dựa trên bộ chỉ số đa chiều (BLEU, ROUGE, CER, WER). Kết quả thực nghiệm cho thấy mô hình đề xuất đạt độ chính xác cao trong việc khôi phục văn bản, giảm thiểu đáng kể tỷ lệ lỗi ký tự và lỗi từ so với mô hình gốc (Baseline). Từ đó làm cơ sở cho các nghiên cứu cải thiện sau này.

1.6 Cấu trúc báo cáo

Báo cáo được trình bày một cách hệ thống qua 7 phần chính, với nội dung cụ thể của từng chương như sau:

- **Phần 1: Giới thiệu:** Trình bày tổng quan về bối cảnh đề tài, tính cấp thiết của bài toán sửa lỗi chính tả tiếng Việt, đồng thời xác định tuyên bố bài toán, mục tiêu, đối tượng, phạm vi và những đóng góp chính của nghiên cứu.
- **Phần 2: Yêu cầu về Dữ liệu:** Phân tích các nguồn dữ liệu được sử dụng, mô tả quy mô, cách phân chia tập dữ liệu (Train/Val/Test) và cấu trúc định dạng của các cặp câu song ngữ.
- **Phần 3: Phương pháp luận:** Trình bày cách mô hình hóa bài toán dưới dạng Sequence-to-Sequence. Chương này đi sâu vào kiến trúc BARTpho, phân tích các thách thức đặc thù của tiếng Việt, chiến lược tinh chỉnh (Fine-tuning) và hàm tối ưu hóa được áp dụng.
- **Phần 4: Thực nghiệm:** Mô tả chi tiết môi trường thực nghiệm, cấu hình phần cứng/phần mềm, các tham số huấn luyện (hyperparameters), chiến lược huấn luyện cụ thể và định nghĩa các độ đo (metrics) dùng để đánh giá mô hình.

- **Phần 5: Kết quả thực nghiệm:** Trình bày bảng giá trị Loss trong quá trình huấn luyện, phân tích chi tiết các chỉ số định lượng trên tập kiểm thử và thực hiện đánh giá định tính dựa trên các mẫu dữ liệu thực tế.
- **Phần 6: Thảo luận:** Đi sâu phân tích các vấn đề chuyên môn phát hiện được trong quá trình nghiên cứu, bao gồm: vai trò của dấu cách, hiện tượng ảo giác ở mô hình gốc, khả năng phục hồi lỗi của mô hình sau tinh chỉnh và hiện tượng "quá tự tin" (over-confidence).
- **Phần 7: Kết luận:** Đối chiếu kết quả đạt được với mục tiêu ban đầu, tổng kết những thành tựu, nhìn nhận thẳng thắn các hạn chế còn tồn tại và đề xuất hướng phát triển trong tương lai.
- **Phụ lục (Source Code & Resources):** Cung cấp các đường dẫn đến mã nguồn dự án, hướng dẫn cài đặt chi tiết, liên kết tải xuống bộ dữ liệu và trọng số mô hình (Model Weights) đã được huấn luyện.

2 Yêu cầu về Dữ liệu

2.1 Nguồn dữ liệu

Để đảm bảo cả độ chính xác theo miền cụ thể (domain-specific) và khả năng hiểu ngôn ngữ tổng quát (generalization), nhóm đã xây dựng bộ dữ liệu gồm 60,000 mẫu bằng cách kết hợp hai nguồn:

Nguồn Internal (∼20,000 mẫu): Dữ liệu được thu thập bởi nhóm trong đồ án giữa kỳ, trích xuất từ bộ sách *"Lịch sử Việt Nam"*, Tập 9 (Từ năm 1930 đến năm 1945), do Tạ Thị Thúy chủ biên, xuất bản bởi Viện Sử học thuộc Viện Hàn lâm Khoa học Xã hội Việt Nam (NXB Khoa học Xã hội). Nguồn này tập trung vào các sự kiện lịch sử Việt Nam với các lỗi OCR từ quá trình scan sách.

Nguồn External (∼40,000 mẫu): Dữ liệu được lấy từ dataset `bmd1905/error-correction-vi` trên HuggingFace, bao gồm tin tức, bài báo và các chủ đề xã hội. Mục đích của nguồn này là cải thiện khả năng tổng quát hóa của mô hình, giúp mô hình hoạt động tốt trên nhiều loại văn bản khác nhau, không chỉ giới hạn trong miền lịch sử.

2.2 Quy mô và Phân chia dữ liệu

Tổng bộ dữ liệu gồm 60,000 cặp câu, được xáo trộn (shuffle) và phân chia thành ba tập con. Tập Test được giữ riêng hoàn toàn để đánh giá khách quan hiệu suất cuối cùng của mô hình.

Tập dữ liệu	**Số lượng mẫu**	**Tỷ lệ**
Train	50,000	83.33%
Validation	5,000	8.33%
Test	5,000	8.33%
Tổng cộng	**60,000**	**100%**

Bảng 1: Bảng phân chia dữ liệu

2.3 Cấu trúc dữ liệu

Bộ dữ liệu được tổ chức theo dạng cặp song song (parallel pairs), trong đó mỗi mẫu bao gồm hai thành phần:

Input (x): Câu văn có lỗi, bao gồm lỗi OCR, lỗi gõ phím (typos), thiếu dấu thanh, và lỗi ngữ pháp.

Target (y): Câu văn chuẩn tương ứng, đã được chuẩn hóa hoàn toàn về chính tả và ngữ nghĩa, đóng vai trò là nhãn để mô hình học cách ánh xạ từ câu lỗi sang câu đúng.

Ví dụ minh họa:

- **Input:** *"toiii la sinhviến đai học khoa hoc tunhien"*
- **Output:** *"tôi là sinh viên đại học khoa học tự nhiên. "*

3 Phương pháp luận

3.1 Mô hình hóa bài toán

Trong nghiên cứu này, chúng em tiếp cận bài toán sửa lỗi chính tả tiếng Việt (Vietnamese Spelling Correction) dưới góc độ của một bài toán **Dịch máy (Machine Translation)**, cụ thể là mô hình chuyển đổi chuỗi sang chuỗi (Sequence-to-Sequence). Đây là phương pháp coi văn bản chứa lỗi là "input" và văn bản chính xác là "target".

Giả sử chúng ta có một tập dữ liệu $\mathcal{D} = \{(X^{(i)}, Y^{(i)})\}_{i=1}^{N}$, trong đó mỗi cặp dữ liệu bao gồm:

- **Chuỗi đầu vào (Input sequence):** $X = (x_1, x_2, ..., x_T)$ đại diện cho câu văn bản chứa lỗi (lỗi chính tả, lỗi dấu, lỗi OCR...).
- **Chuỗi đầu ra (Target sequence):** $Y = (y_1, y_2, ..., y_{T'})$ đại diện cho câu văn bản chính xác tương ứng.

Lưu ý rằng độ dài của chuỗi đầu vào T và chuỗi đầu ra T' có thể khác nhau (ví dụ trong trường hợp lỗi dính từ hoặc tách từ sai).

Mục tiêu của mô hình là học một phân phối xác suất có điều kiện $P(Y|X;\theta)$ để tìm ra chuỗi Y có khả năng xuất hiện cao nhất với đầu vào X cho trước. Về mặt toán học, quá trình suy diễn (inference) nhằm tìm ra chuỗi kết quả $\hat{Y}$ thoả mãn:

$$\hat{Y} = \underset{Y}{\operatorname{argmax}}\, P(Y|X;\theta) \tag{1}$$

Trong đó θ là tập hợp các tham số của mô hình được tối ưu hóa trong quá trình huấn luyện.

Theo nguyên lý chuỗi (Chain rule) trong xác suất, xác suất có điều kiện của toàn bộ chuỗi đầu ra $P(Y|X)$ được phân rã thành tích xác suất của từng token tại mỗi bước thời gian t, phụ thuộc vào ngữ cảnh đầu vào X và các token đã được sinh ra trước đó $y_{<t}$:

$$P(Y|X;\theta) = \prod_{t=1}^{T'} P(y_t|y_{<t}, X;\theta) \tag{2}$$

Trong đó $y_{<t} = (y_1, ..., y_{t-1})$ biểu thị lịch sử các từ đã được dự đoán. Mô hình sẽ thực hiện sinh văn bản theo cơ chế tự hồi quy (Auto-regressive), nghĩa là dự đoán tại bước hiện tại sẽ trở thành đầu vào cho bước tiếp theo.

3.2 Kiến trúc mô hình BARTpho

Để giải quyết bài toán Sequence-to-Sequence đã định nghĩa ở trên, chúng em lựa chọn sử dụng **BARTpho**, mô hình ngôn ngữ tiền huấn luyện (Pre-trained Language Model) đầu tiên được thiết kế chuyên biệt cho tiếng Việt dựa trên kiến trúc Sequence-to-Sequence.

3.2.1 Tổng quan kiến trúc

BARTpho kế thừa kiến trúc của mô hình **BART** (Bidirectional and Auto-Regressive Transformers) do Lewis et al. (2020) đề xuất. Đây là một kiến trúc lai ghép, kết hợp những ưu điểm mạnh nhất của hai dòng mô hình phổ biến là BERT và GPT. Cụ thể, BARTpho hoạt động như một bộ **Denoising Autoencoder** (Tự mã hóa khử nhiễu) tiêu chuẩn với hai thành phần chính:

- **Bộ mã hóa hai chiều (Bidirectional Encoder):** Tương tự như BERT, thành phần này có khả năng đọc và mã hóa chuỗi đầu vào X (câu chứa lỗi) theo cả hai chiều (từ trái sang phải và từ phải sang trái). Điều này cho phép mô hình nắm bắt được ngữ cảnh toàn cục (global context) của câu, giúp phát hiện các lỗi sai dựa trên các từ xung quanh nó.

  Công thức tổng quát của Encoder:

$$H_{enc} = \text{Encoder}(x_1, x_2, ..., x_T) \tag{3}$$

  Trong đó H_{enc} là các biểu diễn ngữ nghĩa (hidden states) của chuỗi đầu vào.

- **Bộ giải mã tự hồi quy (Auto-regressive Decoder):** Tương tự như GPT, thành phần này chịu trách nhiệm sinh ra chuỗi văn bản đích Y (câu đúng). Tại mỗi bước thời gian, Decoder sử dụng cơ chế *Cross-Attention* để tham chiếu đến toàn bộ thông tin ngữ nghĩa từ Encoder (H_{enc}), đồng thời sử dụng cơ chế *Masked Self-Attention* để chỉ nhìn thấy các từ đã sinh ra trước đó.

  Công thức tổng quát của Decoder:

$$h_t = \text{Decoder}(y_{<t}, H_{enc}) \tag{4}$$

Tại sao kiến trúc này phù hợp cho sửa lỗi chính tả? BARTpho được huấn luyện tiền kỳ (pre-training) bằng cách học cách khôi phục lại văn bản gốc từ một văn bản đã bị làm nhiễu (thêm nhiễu, xóa từ, đảo từ...). Cơ chế này tương đồng hoàn toàn với bản chất của bài toán sửa lỗi chính tả: Đầu vào là một văn bản bị "nhiễu" (lỗi chính tả/OCR) và mục tiêu là khôi phục lại văn bản "sạch". Do đó, BARTpho có khả năng chuyển giao tri thức (transfer learning) rất tốt cho tác vụ này.

3.2.2 Những khó khăn về mặt tiếng Việt

Tiếng Việt là ngôn ngữ đơn âm tiết (monosyllabic), trong đó các từ có thể được cấu tạo từ một hoặc nhiều âm tiết riêng biệt được ngăn cách bởi khoảng trắng. BARTpho cung cấp hai phiên bản tiền huấn luyện: cấp độ từ (Word-level) và cấp độ âm tiết (Syllable-level). Trong nghiên cứu này, chúng em quyết định sử dụng phiên bản `vinai/bartpho-syllable` dựa trên các phân tích sau:

- **Hạn chế của mô hình cấp độ từ (Word-level) đối với dữ liệu nhiễu:**
  Các mô hình Word-level yêu cầu văn bản đầu vào phải được đi qua một bộ tách từ (Word Segmenter, ví dụ: VnCoreNLP) trước khi đưa vào mô hình. Tuy nhiên, các bộ tách từ này thường dựa trên giả định rằng văn bản đầu vào là chính xác về mặt chính tả và ngữ pháp. Khi văn bản chứa lỗi (ví dụ: lỗi dính từ *"họcsinh"*, lỗi telex *"truong hoc"*), bộ tách từ sẽ hoạt động sai lệch, dẫn đến việc sinh ra các token không xác định (`<unk>`) hoặc tách sai ngữ nghĩa. Điều này làm mất mát thông tin ngay từ bước đầu vào.

- **Ưu điểm của mô hình cấp độ âm tiết (Syllable-level):**
  Phiên bản Syllable-level xử lý văn bản dựa trên các âm tiết được phân tách bằng khoảng trắng, kết hợp với kỹ thuật mã hóa đoạn con (Sub-word tokenization) sử dụng SentencePiece (Byte-Pair Encoding).

  - **Tính bền vững (Robustness):** Mô hình không phụ thuộc vào công cụ tách từ bên ngoài. Nó có thể xử lý trực tiếp các chuỗi ký tự bị lỗi, dính từ hoặc thiếu dấu mà không gặp lỗi tách từ.
  - **Từ vựng mở:** Với cơ chế BPE, ngay cả khi gặp một âm tiết viết sai chưa từng xuất hiện trong từ điển (ví dụ: *"nghiêp"* thay vì *"nghiệp"*), mô hình vẫn có thể phân rã nó thành các đơn vị nhỏ hơn để xử lý thay vì gán nhãn `<unk>`, giúp bảo toàn thông tin tối đa cho quá trình sửa lỗi của Decoder.

Do đó, kiến trúc Syllable-level là lựa chọn tối ưu cho bài toán sửa lỗi chính tả tiếng Việt, nơi mà đầu vào mang tính chất nhiễu cao.

3.3 Chiến lược Tinh chỉnh

Trong nghiên cứu này, chúng em áp dụng chiến lược **Tinh chỉnh có giám sát toàn vẹn (Full Supervised Fine-tuning)**. Thay vì đóng băng (freeze) các lớp dưới và chỉ huấn luyện lớp đầu ra,

nhóm cập nhật trọng số của toàn bộ mô hình BARTpho (bao gồm cả Encoder và Decoder) để mô hình thích nghi tốt nhất với phân phối của dữ liệu lỗi chính tả tiếng Việt.

Quy trình tinh chỉnh được thực hiện theo cơ chế **Học End-to-End (End-to-End Learning)** như sau:

- **Tiền xử lý đầu vào:**

  Mỗi cặp câu (X, Y) từ tập huấn luyện được đưa qua bộ Tokenizer của BARTpho. Bộ tokenizer này tự động thêm các token đặc biệt để đánh dấu điểm bắt đầu và kết thúc của mỗi chuỗi, đảm bảo tính nhất quán cho mô hình Seq2Seq (ví dụ: `<s>` và `</s>`). Để đảm bảo tính toán song song theo batch trên GPU, các câu được đệm (padding) hoặc cắt ngắn (truncation) về cùng một độ dài cố định (Max_Length = 256).

- **Cơ chế ánh xạ linh hoạt:**

  Một ưu điểm vượt trội của chiến lược Seq2Seq so với các phương pháp gán nhãn chuỗi (Sequence Labeling) truyền thống là khả năng xử lý **độ dài biến thiên**.

  - Trong bài toán sửa lỗi chính tả, độ dài của câu đầu ra Y thường xuyên khác với câu đầu vào X (ví dụ: lỗi dính từ *"họcsinh"* [1 token] $\rightarrow$ *"học sinh"* [2 token], hoặc lỗi thừa ký tự).
  - Chiến lược tinh chỉnh BARTpho cho phép mô hình tự do sinh ra số lượng token cần thiết để tái tạo câu đúng mà không bị ràng buộc bởi độ dài của câu sai.

- **Cập nhật tham số:**

  Mô hình nhận đầu vào là các token ID của câu sai và học cách dự đoán token ID của câu đúng. Trọng số θ của mô hình được tối ưu hóa dựa trên sự sai biệt giữa phân phối từ vựng dự đoán và từ vựng thực tế (Ground Truth).

3.4 Hàm mất mát và Tối ưu hóa

Quá trình huấn luyện mô hình được thực hiện dựa trên ba thành phần cốt lõi sau:

- **Hàm mất mát (Loss Function):**

  Bản chất của việc sinh văn bản là bài toán phân loại từ: Tại mỗi vị trí, mô hình phải chọn ra 1 từ đúng nhất trong bộ từ điển. Chúng em sử dụng hàm **Cross-Entropy Loss**. Hàm này

đo lường độ sai lệch giữa phân phối xác suất do mô hình dự đoán và từ đúng thực tế (Ground Truth). Giá trị Loss càng nhỏ, mô hình dự đoán càng chính xác.

$$\mathcal{L} = -\log P(\text{từ đúng} \mid \text{ngữ cảnh}) \tag{5}$$

- **Cơ chế Teacher Forcing:**
  Để tăng tốc độ hội tụ, chúng em áp dụng kỹ thuật **Teacher Forcing**. Cơ chế này được tích hợp sẵn trong kiến trúc Transformer Decoder thông qua việc dịch chuyển chuỗi đích (Target shifting) và Masked Self-Attention. Trong thư viện Hugging Face, việc này được xử lý tự động khi tính toán Loss Thay vì để mô hình tự dùng kết quả dự đoán (có thể sai) của bước trước làm đầu vào cho bước sau, chúng em "mớm" cho mô hình token đúng từ dữ liệu huấn luyện. Điều này giúp mô hình không bị trượt dài theo các lỗi sai dây chuyền trong quá trình học.

- **Thuật toán tối ưu (Optimizer):**
  Nhóm sử dụng **AdamW**, thuật toán tối ưu hóa tiêu chuẩn cho các mô hình Transformer hiện nay. AdamW giúp điều chỉnh tốc độ học linh hoạt cho từng tham số và hạn chế hiện tượng quá khớp (overfitting) hiệu quả hơn so với Adam truyền thống.

4 Thực nghiệm

4.1 Môi trường thực nghiệm

Phần cứng: Thực nghiệm được thực hiện trên GPU NVIDIA A100 (40GB VRAM) thông qua nền tảng Vast.ai.

Việc lựa chọn GPU A100 mang lại nhiều lợi ích quan trọng cho quá trình huấn luyện:

- **VRAM 40GB:** Cho phép sử dụng batch size lớn hơn, giúp mô hình học được các pattern tổng quát hơn và ổn định hơn trong quá trình tối ưu.

- **Tensor Cores thế hệ 3:** Hỗ trợ tính toán BF16/FP16, tăng tốc độ huấn luyện lên đến 2-3 lần so với FP32 mà không làm giảm đáng kể độ chính xác.

- **Băng thông bộ nhớ cao (1. 6 TB/s):** Giảm thời gian truyền dữ liệu giữa GPU và VRAM, đặc biệt hiệu quả với các mô hình Transformer có nhiều tham số như BARTpho.

Phần mềm: Môi trường thực nghiệm được xây dựng trên nền tảng Docker Container, sử dụng **Python 3.10** làm ngôn ngữ chính. Các thư viện cốt lõi bao gồm: **PyTorch 2.1.2** (hỗ trợ CUDA 12.1), **Transformers 4.35.0** (cho mô hình BARTpho), **Datasets 2.14.0** và **Accelerate 0.24.0**. Việc đánh giá mô hình sử dụng các gói thư viện chuẩn: `sacrebleu`, `jiwer`, và `rouge_score`.

4.2 Cấu hình mô hình

Mô hình sử dụng là BARTpho-syllable (`vinai/bartpho-syllable`) với kiến trúc BART Encoder-Decoder, kích thước khoảng 1.58 GB, được load thông qua class `AutoModelForSeq2SeqLM` của thư viện Transformers.

4.3 Tham số huấn luyện

Các tham số huấn luyện được trình bày trong bảng sau:

Tham số	**Giá trị**
Learning rate	3e-5
Epochs	5
Train batch size	32 (per device)
Eval batch size	64 (per device)
Gradient accumulation steps	2
Effective batch size	$32 \times 2 = 64$
Optimizer	AdamW
Weight decay	0.01
Max generation length	256
Mixed Precision	BF16 (bfloat16)

Bảng 2: Các tham số huấn luyện chính

Giải thích các tham số quan trọng:

- **Learning rate (3e-5):** Giá trị này được lựa chọn cho việc fine-tune các mô hình pre-trained lớn. Learning rate quá cao có thể phá vỡ các trọng số đã học từ quá trình pre-training, quá thấp sẽ khiến mô hình hội tụ chậm hoặc bị kẹt tại local minima.
- **Gradient accumulation steps (2):** Kỹ thuật này cho phép mô phỏng batch size lớn hơn mà không cần thêm VRAM. Với batch size 32 per device và tích lũy qua 2 steps, effective batch size đạt 64. Batch size lớn giúp gradient ổn định hơn, giảm variance trong quá trình cập nhật trọng số.

- **Mixed Precision BF16:** Sử dụng BF16 (Brain Floating Point) thay vì FP16 trên kiến trúc Ampere của A100 giúp tăng tốc độ huấn luyện và giảm 50% bộ nhớ sử dụng. BF16 là đặc quyền của kiến trúc Ampere (A100) trở lên, giúp khắc phục nhược điểm mất mát độ chính xác (precision loss) của FP16 truyền thống khi huấn luyện các mô hình lớn như Transformer.
- **Weight decay (0.01):** Áp dụng regularization L2 để tránh overfitting, đặc biệt quan trọng khi fine-tune mô hình lớn trên dataset có kích thước vừa phải. Giá trị 0.01 là mức chuẩn được khuyến nghị cho các mô hình Transformer.

4.4 Chiến lược huấn luyện

Chiến lược huấn luyện bao gồm: đánh giá và lưu checkpoint sau mỗi epoch, chỉ giữ lại model tốt nhất dựa trên eval_loss (save_total_limit = 1), load model tốt nhất khi kết thúc huấn luyện (load_best_model_at_end = True), và sử dụng predict_with_generate để tính toán metrics trong quá trình evaluation. Việc chọn eval_loss làm tiêu chí giúp tránh overfitting trên các metrics cụ thể và đảm bảo mô hình tổng quát hóa tốt.

4.5 Các Metrics đánh giá

Để đánh giá hiệu quả của mô hình sửa lỗi chính tả, chúng tôi sử dụng bốn metrics phổ biến trong các bài toán sinh văn bản và sửa lỗi. Các metrics được tóm tắt trong Bảng 3.

Metric	**Thư viện**
BLEU (↑)	sacrebleu
CER (↓)	evaluate/cer
WER (↓)	evaluate/wer
ROUGE-1 F1 (↑)	rouge_score

Bảng 3: Các metrics đánh giá mô hình

Ghi chú: Ký hiệu (↑) nghĩa là giá trị càng cao càng tốt (BLEU, ROUGE-1 F1), ký hiệu (↓) nghĩa là giá trị càng thấp càng tốt (CER, WER).

Giải thích các metrics:

- **BLEU (Bilingual Evaluation Understudy):** Đo lường độ trùng khớp n-gram giữa văn bản dự đoán và văn bản tham chiếu. BLEU càng cao (thang điểm 0-100) cho thấy kết quả

sửa lỗi càng gần với ground-truth. Đây là metric tiêu chuẩn trong các bài toán dịch máy và sinh văn bản.

- **CER (Character Error Rate):** Tỷ lệ lỗi ở mức ký tự, được tính dựa trên số phép chèn, xóa và thay thế cần thiết để biến văn bản dự đoán thành văn bản tham chiếu. CER càng thấp càng tốt (0 là hoàn hảo). Metric này đặc biệt phù hợp với bài toán sửa lỗi chính tả vì đánh giá chi tiết đến từng ký tự.
- **WER (Word Error Rate):** Tương tự CER nhưng đo lường ở mức từ. WER càng thấp càng tốt. Metric này cho thấy khả năng sửa lỗi ở mức từ vựng của mô hình.
- **ROUGE-1 F1:** Đo lường độ trùng khớp unigram giữa văn bản dự đoán và tham chiếu, kết hợp cả Precision và Recall thông qua F1-score. ROUGE-1 F1 càng cao càng tốt (thang điểm 0-1). Metric này đánh giá khả năng bảo toàn nội dung của văn bản sau khi sửa lỗi.

Quy trình chuẩn hóa trước đánh giá:

Một thách thức lớn khi đánh giá các mô hình sinh văn bản tiếng Việt (đặc biệt là BARTpho) là sự không nhất quán về định dạng (ví dụ: dấu gạch dưới nối từ, khoảng trắng thừa). Để đảm bảo tính công bằng và chính xác, trước khi tính toán bất kỳ metric nào, chúng em áp dụng quy trình **Chuẩn hóa hậu kỳ (Post-processing Normalization)** cho cả văn bản dự đoán và văn bản gốc:

- Loại bỏ toàn bộ ký tự gạch dưới (_) do tokenizer sinh ra.
- Xóa bỏ các khoảng trắng thừa (multiple spaces) và khoảng trắng ở đầu/cuối câu.

Việc này giúp các chỉ số CER/BLEU phản ánh đúng năng lực sửa lỗi ngữ nghĩa của mô hình.

5 Kết quả thực nghiệm

5.1 Kết quả huấn luyện

Bảng 4: Quá trình huấn luyện và các chỉ số đánh giá trên tập Validation (5.000 mẫu)

Training Loss	Epoch	Step	Val Loss	Bleu	Cer	Wer	F1
0,3907	1,0	782	0,0338	83,8387	0,0343	0,0892	0,9626
0,0355	2,0	1564	0,0240	87,6565	0,0244	0,0669	0,9763
0,0278	3,0	2346	0,0225	88,6311	0,0216	0,0605	0,9804
0,0210	4,0	3128	0,0220	88,8898	0,0200	0,0580	0,9821
0,0187	**5,0**	**3910**	**0,0213**	**89,3574**	**0,0196**	**0,0563**	**0,9826**

Nhận xét về quá trình huấn luyện

Dựa trên kết quả từ Bảng 4, ta có thể đưa ra các nhận định tổng quát sau:

- **Tốc độ hội tụ và độ ổn định:** Mô hình Fine-tune cho thấy khả năng hội tụ rất nhanh chóng. Giá trị *Training Loss* giảm mạnh từ chu kỳ đầu tiên (0,3907) xuống còn 0,0187 tại chu kỳ thứ 5. *Training Loss* và *Validation Loss* cùng nhau giảm dần chứng minh mô hình học được các quy luật ngôn ngữ rất hiệu quả mà không gặp hiện tượng quá khớp (overfitting).
- **Hiệu năng vượt trội:** Các chỉ số đo lường ghi nhận sự tiến bộ đồng nhất qua từng chu kỳ. Điểm *Bleu* đạt mức xấp xỉ 89,36% và chỉ số *F1-score* tiệm cận mức tuyệt đối (0,9826). Đặc biệt, tỉ lệ lỗi ký tự (CER) được kiểm soát dưới mức 2%, khẳng định độ chính xác cao trong việc khôi phục dấu thanh và sửa lỗi chính tả trong tập validation.
- **Khả năng tổng quát hóa:** Việc các chỉ số duy trì đà cải thiện ổn định qua 3.910 bước huấn luyện cho thấy mô hình đã xây dựng được một "bản đồ ngữ nghĩa" vững chắc, đủ khả năng xử lý các biến thể văn bản lỗi phức tạp trên tập dữ liệu thử nghiệm.

5.2 Phân tích chỉ số trên toàn tập dữ liệu test

Kết quả đánh giá định lượng trên toàn tập thực nghiệm cho thấy sự chênh lệch đáng kể về hiệu suất giữa mô hình Baseline và mô hình Fine-tune. Các chỉ số đo lường chính bao gồm BLEU (độ tương đồng văn bản), CER (tỷ lệ lỗi ký tự) và WER (tỷ lệ lỗi từ) được tổng hợp trong Bảng 5.

Bảng 5: So sánh chỉ số trên toàn tập test giữa mô hình Baseline và Fine-tune

Mô hình	**BLEU (↑)**	**CER (↓)**	**WER (↓)**	**F1-score (↑)**
Baseline	32.97	0.5859	0.8174	0.7421
Fine-tune	**89.05**	**0.0201**	**0.0564**	**0.9825**

Mô hình Fine-tune cho thấy sự tiến bộ vượt bậc khi chỉ số BLEU tăng gấp gần 3 lần, đồng thời tỷ lệ lỗi ký tự (CER) giảm xuống mức tối thiểu (xấp xỉ 2%). Điều này chứng tỏ việc tinh chỉnh trên dữ liệu đặc thù đã giúp mô hình học được quy luật dấu thanh và cấu trúc từ vựng tiếng Việt một cách hệ thống.

5.3 Đánh giá dựa trên mẫu thực tế

Để làm rõ sự khác biệt trong khả năng xử lý, Bảng 6 trình bày 15 mẫu đầu tiên với định dạng so sánh trực tiếp giữa kết quả đích (Reference) và dự đoán của hai mô hình.

Bảng 6: So sánh kết quả dự đoán trên 15 mẫu đầu tiên

STT	**Nội dung (Ref: Đích | Base: Baseline | Fine: Fine-tune)**
1	**Ref:** Về chính trị, Mặt trận Nhân dân trên thực tế phải chấp nhận sự **Base:** vè chính tri, Măt tràn Nhàn dan trén thc té phài cháp nhàn su, **Fine:** Về chính trị, Mặt trận Nhân dân trên thực tế phải chấp nhận sự
2	**Ref:** Và những phiên thảo luận của hội nghị đang đi gần đến những thỏa thuận về tài chính cho những kế hoạch đề ra. **Base:** Và những phiên thao luậnG của hội nghđAng đi gần đến những thuận về tài chính cho những kế hoạch đề **Fine:** Và những phiên thảo luận của hội nghị đang đi gần đến những thuận về tài chính cho những kế hoạch đề ra.
3	**Ref:** hội đồng và thành viên hội đồng là các giáo viên người Pháp hay **Base:** hōi dòng và thành vien hòi dòng là các giáo vièn nguòi Pháp hay **Fine:** hội đồng và thành viên hội đồng là các giáo viên người Pháp hay
6	**Ref:** Trước đó, cầu thủ này còn lập kỷ lục cầu thủ già nhất giành danh hiệu vô địch...

STT	Nội dung (Ref: Đích | Base: Baseline | Fine: Fine-tune)
	Base: Trước đó cLầu thủ này cònlập kỷ lục... (xem clip). (xem clip). (xem clip). (xem clip). **Fine:** Trước đó cầu thủ này còn lập kỷ lục cầu thủ già nhất giành danh hiệu vô địch...
7	**Ref:** Chi hội có những gương mặt mới như Ngô Thanh Vân, Minh Đạt... **Base:** ...Chi bảo Quyền Quyền lợi quyền lợi quyền lợi quyền lợi quyền lợi quyền lợi... **Fine:** Chi hội có những gương mặt mới như Ngô Thanh Vân, Minh Đạt...
9	**Ref:** Tần và Lê Văn Lương lãnh đạo. Công nhân đình việc, tổ chức mít **Base:** Tàn và Le Văn Luong lãnh dao. Còng nhan dinh viec, tò chúrc mít **Fine: Trần** và Lê Văn Lương lãnh đạo. Công nhân đình việc, tổ chức mít
10	**Ref:** Đến nhà ai chơi, thấy tượng nào đẹp tôi liền mượn về đúc thử. **Base:** Đvn nhà ai chơi thấy tượng nào đẹp tôi lền mượnvề đúc thử **Fine: Đưa** nhà ai chơi thấy tượng nào đẹp tôi liền mượn về đúc thử.
12	**Ref:** ...Em muốn hỏi khoa Mỹ Thuật Công Nghiệp của trường Kiến Trúc năm vừa rồi lấy bao nhiêu điểm? **Base:** ...Em muốn hỏi Khoa Mỹ Thuật... (Lặp lại nội dung 4 lần liên tiếp) **Fine:** ...Em muốn hỏi Khoa Mỹ Thuật Công Nghiệp của trường Kiến Trúc năm vừa rồi lấy bao nhiêu điểm?
14	**Ref:** Chiều nay, chúng tôi chủ động tấn công ngay từ đầu. **Base:** Chiều nagy cjúng tôi chủ độBg tấn công ngaRy lừ đầu Chiều nagy cjúng tôi... **Fine:** Chiều nay, chúng tôi chủ động tấn công ngay từ đầu.
15	**Ref:** Dương cho lập Cục Khẩn hoang và triển khai một chương trình **Base:** Duong cho làp Cuc Khán hoang và trièn khai mot chuong trinh **Fine:** Dương cho lập Cục **Khắp** hoang và triển khai một chương trình

Đánh giá tổng quát

Dựa trên các mẫu thực tế, mô hình Baseline bộc lộ điểm yếu trầm trọng khi thường xuyên lặp lại các cụm từ vô nghĩa (Hallucination) và không thể khôi phục dấu thanh chính xác. Ngược lại, mô hình Fine-tune đã khắc phục hoàn toàn lỗi lặp từ, sửa được các lỗi dính chữ và lỗi gõ Telex (Sample 14). Tuy nhiên, Fine-tune vẫn mắc một số lỗi về ngữ nghĩa khi thay thế các từ ít phổ biến bằng từ

thông dụng hơn (Sample 9, 10, 15).

6 Thảo luận

Sự khác biệt về hiệu suất giữa hai mô hình phản ánh những thách thức đặc thù trong bài toán xử lý tiếng Việt:

6.1 Dấu cách

Phần này tập trung thảo luận 3 kịch bản sai lệch về dấu cách: (1) Thiếu dấu cách giữa các âm tiết, (2) Sự xuất hiện của từ mượn trong chuỗi ký tự dính liền, và (3) Phân mảnh âm tiết.

Vấn đề 1: Thiếu dấu cách (Missing Whitespace)

Kịch bản này đánh giá khả năng khôi phục ranh giới từ dựa trên tần suất xuất hiện và cấu trúc ngữ pháp.

ID	Câu gốc (Input)	Kết quả (Inference)	Đặc điểm
S1.1	Các **sinhviên** đang tích cực thảo luận.	Các **sinh viên** đang tích cực thảo luận.	Từ ghép cố định
S1.2	Trẻ em **ratthich** ăn **bánhkẹo**.	Trẻ em **rất thích** ăn **bánh kẹo**.	Sửa lỗi kép (Dấu + Cách)
S1.3	Các **họctrò** rất **chăm chỉhọc**.	Các **học trò** rất **chăm chỉ học**.	Tổ hợp tự do

Bảng 7: Thực nghiệm thiếu dấu cách giữa các âm tiết.

Phân tích từng mẫu:

- **S1.1:** "sinhviên" là một thực thể có độ kết hợp (*Semantic Cohesion*) cực cao. Do tần suất xuất hiện đồng thời của hai âm tiết này trong corpus rất lớn, mô hình khôi phục dấu cách dễ dàng nhờ xác suất chuỗi vượt trội.
- **S1.2:** Mẫu này cho thấy khả năng *Joint Correction*. Mô hình không chỉ tách "ratthich" thành hai từ mà còn đồng thời khôi phục dấu thanh ("rat" $\rightarrow$ "rất"). Điều này chứng tỏ mô hình xử lý lỗi ở mức độ biểu diễn âm tiết toàn diện.
- **S1.3:** "chăm chỉ học" là một cấu trúc cú pháp mở [Tính từ + Động từ]. Việc mô hình tách đúng "chỉ học" cho thấy nó không chỉ dựa vào bộ từ điển từ ghép mà còn hiểu được ranh giới ngữ pháp giữa bổ ngữ và động từ chính.

Vấn đề 2: Từ mượn và Chuỗi dính liền phức tạp

Kịch bản này kiểm tra khả năng nhận diện thực thể ngoại lai (*Foreign Entity Recognition*) và sự thích nghi ngôn ngữ.

ID	Câu gốc (Input)	Kết quả (Inference)	Hiện tượng
S2.1	Tôi đang bị **deadlinedí** rất nhiều.	Tôi đang bị **deadline dí** rất nhiều.	Tách biên đa ngôn ngữ
S2.2	**toithichngheradio** ở Paris...	**Tôi thích nghe radio** ở Paris...	Phân đoạn chuỗi dài
S2.3	Người **tậpgym** không thích **tập-cardio**.	Những người **tập gym** không thích **tập cardio**.	Thuật ngữ chuyên biệt
S2.4	Mạng xã hội hiện nay rất **toxic**.	Mạng xã hội hiện nay rất **độc**.	Chuyển dịch ngữ nghĩa

Bảng 8: Thực nghiệm từ mượn và chuỗi dính liền.

Phân tích từng mẫu:

- **S2.1 & S2.3:** Mô hình xác định chính xác "deadline", "gym", "cardio" là các đơn vị từ vựng độc lập dù chúng dính liền với các động từ tiếng Việt ("dí", "tập"). Điều này chứng tỏ bộ Tokenizer đã học được các đặc điểm phân phối ký tự của tiếng Anh trong ngữ cảnh Việt hóa.
- **S2.2:** Với chuỗi "toithichngheradio", mô hình thực hiện phân đoạn đa mức (multi-level segmentation). Nó khôi phục thành công cả dấu cách, viết hoa đầu câu và dấu chấm lửng, cho thấy khả năng tái cấu trúc văn bản ở mức độ câu (Sentence-level).
- **S2.4:** Đây là mẫu thú vị nhất. Thay vì sửa thành "tô xích" hay giữ nguyên "toxic", mô hình chọn từ "độc" (nghĩa tương đương). Đây là hành vi *Paraphrasing* (diễn đạt lại), cho thấy mô hình ưu tiên tính dễ hiểu của văn bản tiếng Việt hơn là tính trung thực tuyệt đối với từ mượn.

Vấn đề 3: Phân mảnh âm tiết (Syllable Fragmentation)

Kịch bản thử nghiệm giới hạn của Tokenizer khi các ký tự bị chia tách bất thường.

ID	Câu gốc (Input)	Kết quả (Inference)
S3.1	tôi không **t h i c h** học, dù tôi là một học **s i n h ch ăm** chỉ	Tôi không **tự ý bỏ** học, dù tôi là một **học sinh năng nổ, chăm** chỉ.
S3.2	tôi **r ấ t** thích **họ c** vì nó giúp tôi giảm stress	Tôi **rất** thích **họ**, có lẽ vì nó giúp tôi giảm stress.

Bảng 9: Thực nghiệm phân mảnh âm tiết.

Phân tích từng mẫu:

- **S3.1:** "t h i c h" bị biến đổi hoàn toàn thành cụm "tự ý bỏ". Việc dấu cách thừa xuất hiện giữa các chữ cái đã làm Tokenizer sụp đổ, mô hình không thể ghép nối lại thành từ "thích" mà tự ý sinh văn bản (Hallucination) để cố gắng tạo ra một câu có vẻ trôi chảy.
- **S3.2:** "họ c" bị mô hình sửa thành "họ". Dấu cách thừa đã làm thay đổi hoàn toàn ngữ nghĩa và từ loại (từ động từ "học" sang đại từ "họ"). Mô hình ưu tiên chọn một từ có nghĩa gần nhất với chuỗi ký tự nhận diện được ("họ") và loại bỏ ký tự lẻ ("c").
- **Hệ quả:** Kết quả tại bảng này cho thấy tính *Faithfulness* (Trung thực với văn bản gốc) bị giảm mạnh khi gặp lỗi phân mảnh. Mô hình ưu tiên tính *Fluency* (Trôi chảy) dẫn đến việc thay đổi ý định ban đầu của người dùng.

Kết luận chung

Mô hình thể hiện sự vượt trội trong việc giải quyết các lỗi **thiếu dấu cách** nhờ vào khả năng học sâu về cấu trúc ngữ pháp và từ vựng đa ngôn ngữ. Tuy nhiên, rủi ro về **ảo giác ngữ nghĩa** rất cao khi dữ liệu đầu vào bị **phân mảnh âm tiết**. Điều này đặt ra yêu cầu cần bổ sung các kỹ thuật *Data Augmentation* tập trung vào việc ghép nối ký tự rời rạc trong tương lai.

6.2 Về hiện tượng ảo giác và lặp từ ở Baseline:

Mô hình Baseline khi chưa được tinh chỉnh thường rơi vào trạng thái mất kiểm soát khi đối mặt với các chuỗi ký tự lỗi hoặc câu dài. Việc lặp lại các cụm từ như "Quyền lợi" (Sample 7) hoặc "xem clip" (Sample 6) cho thấy mô hình bị mắc kẹt trong không gian xác suất của các cụm từ phổ biến mà không dựa trên ngữ cảnh thực tế của câu đầu vào.

6.3 Về khả năng phục hồi và sửa lỗi của Fine-tune:

Quá trình Fine-tune đã cung cấp cho mô hình một "bản đồ ngôn ngữ" chuẩn xác hơn. Việc sửa thành công "cjúng tôi" thành "chúng tôi" (sample 14) chứng minh mô hình đã học được quy tắc gõ Telex và sự liên kết ngữ nghĩa giữa các từ trong câu. Đây là bước tiến quan trọng giúp văn bản đầu ra đạt độ tự nhiên cao.

6.4 Về vấn đề "Quá tự tin" (Over-confidence) của mô hình học sâu:

Một điểm đáng lưu ý là lỗi thay đổi ý nghĩa từ vựng của Fine-tune. Việc đổi họ "Tần" thành "Trần"(sample 9) hay "Đến" thành "Đưa"(sample 10) cho thấy mô hình có xu hướng ưu tiên các từ có tần suất xuất hiện cao trong tập huấn luyện (Predictive Bias). Khi gặp một từ không dấu có nhiều cách khôi phục, mô hình đôi khi chọn phương án phổ biến nhất thay vì phương án đúng nhất theo ngữ cảnh lịch sử hoặc tên riêng.

7 Kết luận

Nghiên cứu này đã thực hiện việc xây dựng và đánh giá hệ thống khôi phục dấu và sửa lỗi chính tả tiếng Việt dựa trên phương pháp tinh chỉnh mô hình ngôn ngữ lớn. Dựa trên các kết quả thực nghiệm, có thể đưa ra các kết luận tổng kết sau:

7.1 Về mục tiêu và phương pháp đề ra

Mục tiêu cốt lõi của bài toán là chuyển đổi các văn bản tiếng Việt bị lỗi (mất dấu, gõ sai Telex, dính chữ) về trạng thái văn bản chuẩn ngữ pháp và ngữ nghĩa. Phương pháp *Fine-tuning* được lựa chọn đã chứng minh được tính đúng đắn và hiệu quả vượt trội so với mô hình *Baseline*. Việc tinh chỉnh mô hình trên tập dữ liệu đặc thù đã giúp hệ thống không chỉ học được quy luật dấu thanh mà còn hiểu được cấu trúc từ vựng tiếng Việt trong nhiều ngữ cảnh khác nhau.

7.2 Về kết quả nhận lại

Kết quả thu được cho thấy sự tiến bộ vượt bậc về mặt định lượng lẫn định tính:

- **Hiệu năng định lượng:** Mô hình đạt được các chỉ số ấn tượng với điểm F1 xấp xỉ 0,9826 và điểm Bleu tiệm cận mức 89,36%. Tỉ lệ lỗi ký tự (CER) giảm xuống dưới mức 2%, cho thấy khả năng phục hồi văn bản gần như hoàn hảo về mặt hình thức.
- **Khả năng xử lý thực tế:** Mô hình đã khắc phục triệt để hiện tượng ảo giác rác (hallucination) và lỗi lặp từ vô hạn thường thấy ở mô hình Baseline, đồng thời xử lý tốt các biến thể lỗi gõ Telex phức tạp.

7.3 Hạn chế và Những vấn đề tồn tại

Dù đạt được các chỉ số kỹ thuật cao, nghiên cứu cũng chỉ ra một thách thức quan trọng về mặt ngữ nghĩa: **Sự tự tin sai lệch (Over-confidence)**.

- Hệ thống có xu hướng ưu tiên các từ ngữ có xác suất phổ biến cao trong tập huấn luyện thay vì bảo tồn các thực thể hiếm gặp nhưng chính xác trong văn bản gốc.
- **Hiện tượng "Sửa lỗi quá mức"**: Do mô hình được tiền huấn luyện trên lượng lớn văn bản chuẩn mực (báo chí, wiki), nó có xu hướng "chuẩn hóa" cả những từ ngữ mang sắc thái địa phương, văn nói hoặc cấu trúc câu đặc biệt thành dạng phổ biến hơn. Điều này đôi khi làm giảm tính bảo toàn nguyên tác (*Preservation of Original Intent*) của văn bản đầu vào.

7.4 Hướng phát triển tương lai

Tóm lại, phương pháp Fine-tuning là một giải pháp mạnh mẽ và khả thi cho bài toán sửa lỗi chính tả tiếng Việt. Tuy nhiên, để ứng dụng vào các lĩnh vực đòi hỏi độ chính xác tuyệt đối như hành chính hay lịch sử, hệ thống cần được tích hợp thêm các cơ chế hậu kiểm thực thể (NER) và tra cứu từ điển cứng để kiềm chế sự tự tin dự đoán của mô hình học sâu. Hướng nghiên cứu tiếp theo sẽ tập trung vào việc thiết kế các hàm mất mát (loss function) có khả năng phạt nặng hành vi thay đổi tên riêng và thực thể của mô hình.

8 Phụ lục (Source Code & Resources)

Để đảm bảo tính minh bạch và khả năng tái lập kết quả của nghiên cứu, nhóm thực hiện công khai toàn bộ mã nguồn, dữ liệu và mô hình đã huấn luyện trên các nền tảng lưu trữ mã nguồn mở.

- **Mã nguồn dự án (GitHub Repository):**
  Nơi lưu trữ toàn bộ mã nguồn huấn luyện (Notebooks), mã nguồn ứng dụng Web Demo (Gradio App), file yêu cầu môi trường và hướng dẫn sử dụng chi tiết.
  - **Đường dẫn:** https://github.com/pqthinh232/HCMUS-Vietnamese-correction-project
- **Mô hình đã huấn luyện (Hugging Face Model Hub):**
  Kho lưu trữ chứa trọng số mô hình (Model Weights - `.safetensors`), file cấu hình (Config)

và bộ tách từ (Tokenizer) của mô hình tốt nhất (Best Checkpoint tại Epoch 5). Có thể sử dụng trực tiếp qua thư viện `transformers`.

  - **Đường dẫn:** https://huggingface.co/pqthinh232/HCMUS-vietnamese-correction-project
  - **Model ID:** `pqthinh232/HCMUS-vietnamese-correction-project`

- **Bộ dữ liệu (Hugging Face Dataset):**
  Bộ dữ liệu tổng hợp (60k mẫu) đã qua các bước tiền xử lý, làm sạch và chia tập dữ liệu sẵn (Train/Validation/Test).

  - **Đường dẫn:** https://huggingface.co/datasets/pqthinh232/vietnamese-correction-60k

Tài liệu

[1] Nguyen Luong Tran, Duong Minh Le, and Dat Quoc Nguyen. "BARTpho: Pre-trained Sequence-to-Sequence Models for Vietnamese." *Proceedings of the 21st Annual Conference of the International Speech Communication Association (Interspeech)*, 2020. (Mã nguồn: https://github.com/VinAIResearch/BARTpho)

[2] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension." *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

[3] Bmd1905. "Vietnamese Error Correction Dataset (error-correction-vi)." Hugging Face Datasets Repository. https://huggingface.co/datasets/bmd1905/error-correction-vi.

[4] Thomas Wolf et al. "Transformers: State-of-the-Art Natural Language Processing." *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, 2020.

[5] Hugging Face Inc. "Hugging Face Documentation: Transformers, Datasets, and Evaluate Libraries." https://huggingface.co/docs.