Mô hình dịch máy Anh–Việt dựa trên Transformer Code from Scratch và ứng dụng cho bài toán VLSP Medical MT

Lê Minh Đức
MSV: 23020047

Lê Văn Khoa
MSV: 23020092

***Tóm tắt nội dung*—Báo cáo trình bày quá trình xây dựng một hệ thống dịch máy Anh–Việt gồm hai phần: (i) xây dựng mô hình Transformer Seq2Seq theo hướng *Code from Scratch* (tự xây dựng Multi-Head Attention, Encoder/Decoder, Positional Encoding, Label Smoothing, Noam Scheduler, Beam Search, ...) và huấn luyện trên tập IWSLT2015 En–Vi; (ii) áp dụng mô hình Transformer cho bài toán dịch máy lĩnh vực y tế của VLSP Medical MT bằng cách fine-tune mô hình MarianMT `Helsinki-NLP/opus-mt-en-vi` trên dữ liệu được cung cấp. Kết quả thực nghiệm cho thấy mô hình Transformer Code from Scratch đạt BLEU = 13.180246 khi dịch 500 câu đầu của tập test IWSLT, trong khi MarianMT sau khi fine-tune đạt BLEU = 47.4958 (đánh giá bằng Trainer) và BLEU = 47.1984 (tính BLEU thủ công trên 3000 câu test VLSP).**

***Index Terms*—Machine Translation, Transformer, Seq2Seq, Code from Scratch, MarianMT, VLSP, BLEU**

I. Giới thiệu

Dịch máy (Machine Translation – MT) là một bài toán cốt lõi trong Xử lý ngôn ngữ tự nhiên. Theo yêu cầu bài tập lớn môn NLP 2025, nhóm thực hiện hai nhiệm vụ: **(1)** xây dựng mô hình dịch máy Seq2Seq dựa trên Transformer theo hướng *Code from Scratch*; **(2)** áp dụng mô hình/kinh nghiệm cho bài toán phụ VLSP Shared Task về dịch máy miền y tế.

Transformer [1] là kiến trúc tiêu chuẩn trong MT nhờ khả năng mô hình hoá phụ thuộc dài, huấn luyện song song, và dễ mở rộng. Tuy nhiên, việc dùng trực tiếp lớp "Transformer" có sẵn thường che khuất các chi tiết quan trọng (mask, attention, scheduler, decoding), làm giảm khả năng hiểu sâu. Vì vậy, phần I của báo cáo tập trung xây dựng đầy đủ pipeline và mô hình theo hướng Code from Scratch. Phần II đặt trong bối cảnh dữ liệu chuyên ngành (y tế), nhiều thuật ngữ và ràng buộc dữ liệu, nên fine-tune một mô hình đã tiền huấn luyện (MarianMT) thường cho hiệu quả vượt trội.

II. Tổng quan phương pháp

A. Bài toán

Cho câu nguồn tiếng Anh $x = (x_1, \ldots, x_S)$, mô hình sinh câu đích tiếng Việt $y = (y_1, \ldots, y_T)$ bằng cách cực đại hoá xác suất có điều kiện:

$$p(y|x) = \prod_{t=1}^{T} p(y_t \mid y_{<t}, x). \tag{1}$$

B. Hai yêu cầu chính

(i) Transformer Code from Scratch trên IWSLT2015 En–Vi: tự xây dựng các thành phần lõi (Attention, Encoder/Decoder, PE, masking, loss, scheduler, decoding) và huấn luyện từ dữ liệu phổ quát.

(ii) Fine-tune MarianMT cho VLSP Medical MT: dùng mô hình MarianMT đã tiền huấn luyện trên corpora lớn, tinh chỉnh trên dữ liệu y tế VLSP để thích nghi theo miền.

III. Bài toán chính: Transformer Code from Scratch cho IWSLT2015 En–Vi

A. Dữ liệu và tiền xử lý

Nguồn dữ liệu. Sử dụng IWSLT2015 English–Vietnamese qua HuggingFace (`thainq107/iwslt2015-en-vi`): Train 133,317; Validation 1,268; Test 1,268 cặp câu.

Tokenizer BPE. Huấn luyện BPE bằng `tokenizers`, huấn luyện riêng cho Anh/Việt với `vocab_size=20000`, tách từ bằng `Whitespace()`, token đặc biệt `[PAD],[UNK],[BOS],[EOS]`.

Chuẩn hoá độ dài. Mỗi câu thêm `[BOS]` và `[EOS]`, cắt tối đa `max_src_len = max_tgt_len = 80` (kể cả BOS/EOS), padding theo batch. Với phía đích: `tgt_input` là chuỗi dịch phải, `tgt_output` là nhãn (bỏ BOS).

Bucket batch sampler. Sắp xếp theo độ dài (xấp xỉ số token), gom batch size 64 và trộn thứ tự batch mỗi epoch để giảm padding.

B. Masking

Source mask: kích thước $[B, 1, 1, S]$, che vị trí PAD trong encoder self-attention.
Target mask: kết hợp padding mask $[B, 1, 1, T]$ và causal mask $[1, 1, T, T]$ (tam giác dưới) để decoder không truy cập token tương lai.

C. Kiến trúc Transformer Code from Scratch

Multi-Head Attention. Cấu hình $d_{\text{model}} = 512$, $h = 8$ nên $d_k = 64$. Scaled dot-product attention:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V. \tag{2}$$

Mask áp dụng bằng `masked_fill` với $-\infty$, có dropout 0.1.

Positional Encoding (sinusoidal).

$$\begin{aligned} \mathrm{PE}_{(pos,2i)} &= \sin\left(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}\right), \\ \mathrm{PE}_{(pos,2i+1)} &= \cos\left(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}\right). \end{aligned} \tag{3}$$

Encoder/Decoder layer. EncoderLayer: Self-Attn → Add&Norm → FFN $(512 \to 2048 \to 512)$ → Add&Norm. DecoderLayer: Masked Self-Attn → Add&Norm → Cross-Attn → Add&Norm → FFN → Add&Norm.

Cấu hình tổng thể. `num_encoder_layers=4`, `num_decoder_layers=4`, $d_{\text{model}} = 512$, $h = 8$, dropout 0.1.

D. Loss và tối ưu

Label smoothing. Hệ số $\epsilon = 0.1$; bỏ qua `[PAD]` trong phân phối mục tiêu.

Noam scheduler. Adam với `betas=(0.9,0.98)`, `eps=1e-9`, warmup 4000:

$$lr = \text{factor} \cdot d_{\text{model}}^{-0.5} \cdot \min(step^{-0.5},\ step \cdot warmup^{-1.5}). \tag{4}$$

Gradient clipping `max_grad_norm=1.0`.

E. Thiết lập huấn luyện và đánh giá

Huấn luyện 10 epoch, batch size 64 trên GPU (Colab). Sau mỗi epoch tính loss/perplexity trên validation và lưu checkpoint tốt nhất theo `valid_loss`.

Diễn biến loss/ppl. Bảng I là log epoch 1, 5, 10 trong notebook.

Bảng I
LOSS VÀ PERPLEXITY TRÊN IWSLT2015 (TRÍCH LOG NOTEBOOK)

Epoch	Train Loss	Train PPL	Valid Loss	Valid PPL
1	5.7880	326.35	4.8352	125.86
5	3.9272	50.77	4.0108	55.19
10	**3.5231**	**33.89**	**3.8607**	**47.50**

BLEU trên test IWSLT. Dùng Beam Search (beam size 5, max length 80) để dịch **500 câu đầu** của tập test IWSLT, tính BLEU bằng `sacrebleu` [4]. Kết quả:

$$\text{BLEU} = 13.180246. \tag{5}$$

IV. BÀI TOÁN PHỤ: FINE-TUNE MARIANMT CHO VLSP MEDICAL MT

A. Dữ liệu VLSP và tiền xử lý

Dữ liệu do ban tổ chức cung cấp gồm `train.en.txt`, `train.vi.txt`, `public_test.en.txt`, `public_test.vi.txt`. Đọc theo dòng-đối-dòng, bỏ dòng rỗng và đảm bảo alignment :

- Train: 500,000 cặp câu.
- Test: 3,000 cặp câu.

Tập train chia 95/5: 475,000 (train) và 25,000 (validation).

Token hoá bằng `AutoTokenizer` của `Helsinki-NLP/opus-mt-en-vi`, `max_length=128`, `truncation=True`. PAD trong nhãn thay bằng `-100` để loss bỏ qua.

B. Thiết lập fine-tune

Mô hình: `AutoModelForSeq2SeqLM.from_pretrained("Hels NLP/opus-mt-en-vi")` [2], [3]. Thiết lập huấn luyện :

- Batch size: 8; Epoch: 2.
- Learning rate: $5 \cdot 10^{-5}$; Weight decay: 0.01.
- Generate: beam size 4; `max_length=128`.
- Chọn best model theo BLEU validation (`load_best_model_at_end=True`).

C. Kết quả trên test VLSP

- `trainer.evaluate(test)`: BLEU = 47.4958, loss khoảng 1.16.
- Tự dịch 3000 câu test và tính BLEU thủ công: BLEU = 47.1984.

V. SO SÁNH, PHÂN TÍCH LỖI VÀ THẢO LUẬN

A. So sánh hai hướng tiếp cận

Bảng II tóm tắt kết quả chính.

Bảng II
SO SÁNH KẾT QUẢ

Mô hình	Tập test	Pretrain	BLEU
Transformer Code from Scratch	IWSLT (500 câu đầu)	Không	13.180246
MarianMT fine-tune	VLSP (3000 câu)	Có	47.1984–47.4958

Sự chênh lệch BLEU phản ánh sức mạnh của transfer learning: MarianMT đã được tiền huấn luyện trên corpora lớn, nên khi fine-tune theo miền y tế với vài epoch, mô hình nhanh chóng thích nghi với thuật ngữ và văn phong chuyên ngành.

B. Phân tích lỗi điển hình (trích từ quan sát trong notebook)

Một số lỗi đáng chú ý:

- Cụm "card's holders" đôi khi bị dịch thành "người quản lý thẻ bảo hiểm" thay vì "người có thẻ bảo hiểm" (hiểu sai quan hệ sở hữu).
- Tên riêng như "Phone Hong" có thể bị dịch thành "Điện thoại Hồng" do mô hình cố dịch nghĩa từng từ thay vì giữ nguyên.

Nhìn chung, MarianMT fine-tune dịch tốt cấu trúc báo cáo y khoa và nhiều thuật ngữ chuyên ngành, nhưng vẫn có hạn chế ở tên riêng/địa danh và một số câu rất dài.

VI. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Báo cáo đã mô tả: **(1)** mô hình Transformer Seq2Seq theo hướng *Code from Scratch* cho IWSLT2015, đạt BLEU = 13.180246 (500 câu test đầu); **(2)** fine-tune MarianMT cho VLSP Medical MT, đạt BLEU = 47.4958 (Trainer) và 47.1984 (tính trực tiếp) trên 3000 câu test.

Hướng phát triển:

- Với Code from Scratch: tăng số lớp encoder/decoder (ví dụ 6–6), tăng epoch, thử vocab/độ dài câu, và regularization.

- Với VLSP: thử domain adaptation (tiếp tục pretrain theo miền y tế), tinh chỉnh hyperparameter, và hậu xử lý cho tên riêng/đơn vị/viết tắt y khoa.

Lời cảm ơn

Nhóm xin cảm ơn giảng viên và các bạn đã góp ý trong quá trình thực hiện đồ án.

Tài liệu

[1] A. Vaswani *et al.*, "Attention is All You Need," *NeurIPS*, 2017.
[2] M. Junczys-Dowmunt *et al.*, "Marian: Fast Neural Machine Translation in C++," *ACL*, 2018.
[3] J. Tiedemann and S. Thottingal, "OPUS-MT – Building open translation services for the world," *EAMT*, 2020.
[4] M. Post, "A Call for Clarity in Reporting BLEU Scores," *WMT*, 2018.