



Đại học Quốc gia Hà Nội
Trường Đại học Công nghệ

XỬ LÝ NGÔN NGỮ TỰ NHIÊN BÁO CÁO BÀI TẬP LỚN

NHÓM 2

DỊCH MÁY NMT CHO CÁC CẶP NGÔN NGỮ
VIỆT-KHMER, VIỆT-LÀO, VIỆT-TRUNG, VIỆT-ANH

Nhóm sinh viên :

23020401 Vũ Đức Minh

23020437 Tạ Nguyên Thành

23020376 Nguyễn Đức Huy

Giảng viên :

TS. Trần Hồng Việt

Tháng 12, 2025

Contents

1	GIỚI THIỆU & ĐẶT VẤN ĐỀ	4
1.1	Bối cảnh dịch máy hiện nay	4
1.2	Ý nghĩa của bài toán dịch Việt - Lào, Việt - Khmer, Việt - Trung, Việt - Anh	4
1.3	Thách thức và khó khăn đặc thù	5
1.4	Mục tiêu và đóng góp của bài tập lớn	5
2	CƠ SỞ LÝ THUYẾT	6
2.1	Tổng quan về Dịch máy	6
2.1.1	Dịch máy thống kê (Statistical Machine Translation - SMT)	6
2.1.2	Dịch máy Neural (Neural Machine Translation - NMT)	6
2.2	Kiến trúc Encoder-Decoder và Attention	6
2.2.1	Encoder-Decoder với RNN	6
2.2.2	Cơ chế Attention (Chú ý)	7
2.3	Kiến trúc Transformer	7
2.3.1	Scaled Dot-Product Attention	7
2.3.2	Multi-Head Attention	7
2.3.3	Mạng Feed-Forward (FFN)	7
2.3.4	Mã hóa vị trí (Positional Encoding)	8
2.3.5	Chuẩn hóa lớp (Layer Normalization) và Kết nối dư (Residual Connections)	8
2.3.6	Chiến lược giải mã (Decoding Strategy): Beam Search	8
2.4	Các chỉ số đánh giá	8
2.4.1	BLEU (Bilingual Evaluation Understudy)	8
2.4.2	sacreBLEU	9
2.4.3	chrF (Character F-score)	9
3	CÁC CÁCH TIẾP CẬN LIÊN QUAN	10
3.1	Các mô hình NMT phổ biến	10
3.1.1	RNN-based NMT	10
3.1.2	Transformer và Pre-trained Models	10
3.2	Chiến lược cho ngôn ngữ ít tài nguyên (Low-resource NMT)	10
3.2.1	Multilingual NMT	10
3.2.2	Back-translation (Dịch ngược)	10
3.3	Nghiên cứu hiện có về Việt - Lào/Khmer	11
3.3.1	Dịch máy tiếng Việt	11
3.3.2	Dịch máy Lào và Khmer	11
3.3.3	Cuộc thi VLSP 2023	11

3.4	Phân tích ưu nhược điểm	11
4	PHÁT BIỂU BÀI TOÁN & DỮ LIỆU	12
4.1	Phát biểu bài toán	12
4.2	Đặc điểm ngôn ngữ học và Thách thức xử lý	12
4.2.1	Tiếng Việt: Ngôn ngữ đơn lập và đa thanh điệu	12
4.2.2	Tiếng Lào và Tiếng Khmer: Hệ chữ viết Scriptio Continua	12
4.2.3	Kỹ thuật Back-translation trong bối cảnh Low-resource	13
4.3	Mô tả dữ liệu	13
4.3.1	Nguồn dữ liệu	13
4.3.2	Phương pháp xử lý dữ liệu thô trực tiếp	13
4.3.3	Thống kê dữ liệu	14
4.4	Tiền xử lý dữ liệu (Preprocessing)	14
5	PHƯƠNG PHÁP & MÔ HÌNH ĐỀ XUẤT	15
5.1	Kiến trúc mô hình dựa trên Transformer	15
5.1.1	Mô hình mT5-base (580 Million Parameters)	15
5.1.2	Mô hình M2M-100 (418 Million Parameters)	15
5.2	Quy trình huấn luyện đề xuất (4-Phase Pipeline)	16
5.2.1	Phase 1: Full Fine-tuning	16
5.2.2	Phase 2: Encoder Freezing	16
5.2.3	Phase 3: Partial Encoder Fine-tuning	16
5.2.4	Phase 4: Low Learning Rate Full Fine-tuning	17
5.2.5	Tổng quan về quá trình huấn luyện qua nhiều phase	17
5.2.6	Thiết lập tham chiếu trên cặp ngôn ngữ high-resource (Phụ lục)	17
6	CÀI ĐẶT HỆ THỐNG & THỰC NGHIỆM	19
6.1	Cấu hình cho hệ thống	19
6.2	General Training Setup	19
6.2.1	Phase 1 Training Configuration	19
6.2.2	Phase 2 Training Configuration	20
6.2.3	Phase 3 Training Configuration	20
6.2.4	Phase 4 Training Configuration	20
6.2.5	Phụ lục: Cấu hình fine-tune cho cặp ngôn ngữ high-resource	21
6.3	Tổng kết về việc lựa chọn siêu tham số và cấu hình trong từng phase	21
7	ĐÁNH GIÁ & PHÂN TÍCH KẾT QUẢ	22
7.1	Kết quả thực nghiệm trên các mô hình	22
7.2	Phân tích hiệu suất mô hình	23
7.2.1	Đặc điểm của mT5-base	23
7.2.2	Hiệu quả của M2M-100	23
7.3	Phân tích định tính (Qualitative Analysis)	23
7.4	Phân tích lỗi và Thách thức	24
7.5	Đánh giá tổng quát	24
8	KẾT LUẬN & HƯỚNG PHÁT TRIỂN	25
8.1	Tổng kết	25

8.2	Hạn chế	25
8.3	Hướng phát triển	25

Chapter 1

GIỚI THIỆU & ĐẶT VẤN ĐỀ

1.1 Bối cảnh dịch máy hiện nay

Trong kỷ nguyên số hóa và toàn cầu hóa, nhu cầu giao tiếp và trao đổi thông tin giữa các quốc gia ngày càng trở nên cấp thiết. Dịch máy (Machine Translation - MT), đặc biệt là Dịch máy Neural (Neural Machine Translation - NMT), đã đạt được những bước tiến vượt bậc nhờ sự phát triển của Deep Learning và phần cứng máy tính (GPU). Các mô hình như Transformer [29], BERT [6], mBART [13] đã giải quyết tốt các bài toán dịch giữa các ngôn ngữ phổ biến (Anh, Pháp, Đức, Trung).

Tuy nhiên, đối với các cặp ngôn ngữ ít tài nguyên (low-resource languages) như Tiếng Việt - Tiếng Lào hay Tiếng Việt - Tiếng Khmer, bài toán dịch máy vẫn còn nhiều thách thức lớn do sự khan hiếm về dữ liệu song ngữ chất lượng cao và sự phức tạp về mặt ngôn ngữ học.

1.2 Ý nghĩa của bài toán dịch Việt - Lào, Việt - Khmer, Việt - Trung, Việt - Anh

Việt Nam, Lào và Campuchia là ba nước láng giềng có mối quan hệ hợp tác kinh tế, văn hóa và chính trị sâu rộng từ lâu trong lịch sử. Việc xây dựng một hệ thống dịch máy hiệu quả giữa Tiếng Việt và hai ngôn ngữ này có ý nghĩa thực tiễn to lớn đối với cả ba nước trong bối cảnh hiện nay:

- Hỗ trợ giao tiếp, du lịch và thương mại biên giới.
- Hỗ trợ việc tra cứu tài liệu, văn bản hành chính và giáo dục.
- Bảo tồn và phát triển tài nguyên ngôn ngữ số cho các ngôn ngữ Đông Nam Á.

Tiếng Trung và tiếng Anh là hai ngôn ngữ được sử dụng rộng rãi trên thế giới. Sự phát triển của các mô hình dịch máy dựa trên học sâu đã mang lại chất lượng dịch ngày càng cao. Tuy nhiên, hiệu năng của các mô hình này có thể khác nhau tùy thuộc vào kiến trúc và tập dữ liệu. Vì vậy, việc so sánh chất lượng dịch giữa các mô hình trên một tập dữ liệu cụ thể là cần thiết để đánh giá khách quan hiệu quả của từng mô hình.

1.3 Thách thức và khó khăn đặc thù

Dịch máy cho cặp Việt - Lào/Khmer gặp phải những khó khăn đặc thù:

1. **Thiếu dữ liệu song ngữ (Parallel Corpus):** Không giống như tiếng Anh hay tiếng Trung, lượng dữ liệu cặp câu (sentence pairs) cho Việt-Lào/Khmer rất hạn chế và thường bị nhiễu.
2. **Khác biệt về hệ chữ viết:** Tiếng Việt sử dụng chữ Latin, trong khi tiếng Lào và Khmer sử dụng hệ chữ tượng thanh gốc Brahmic (Abugida). Điều này gây khó khăn cho việc học các biểu diễn từ chung.
3. **Vấn đề Tokenization:** Tiếng Lào và Khmer thường viết liền không có dấu cách giữa các từ, việc tách từ (word segmentation) đòi hỏi các công cụ chuyên biệt và những thư viện để tokenize đòi hỏi nhiều thời gian, gây ảnh hưởng lớn đến chất lượng dịch.

Đối với cặp ngôn ngữ Việt-Trung/Anh, phần này được nhóm bỏ sung nhằm phục vụ việc so sánh chất lượng dịch giữa hai mô hình pretrained đã được sử dụng trước đó cho các cặp ngôn ngữ Việt-Lào và Việt-Khmer [26]. Do đó, nội dung trình bày trong phần này không đi sâu vào phân tích chi tiết, mà đóng vai trò như một phần bổ trợ nhằm cung cấp thêm thông tin tham khảo.

1.4 Mục tiêu và đóng góp của bài tập lớn

Bài tập lớn này của nhóm tập trung chủ yếu vào bài toán dịch máy thần kinh (Neural Machine Translation – NMT) cho các cặp ngôn ngữ ít tài nguyên (low-resource), cụ thể là Việt-Lào và Việt-Khmer. Bên cạnh đó, nhóm thực hiện thêm một phần thực nghiệm bổ trợ trên các cặp ngôn ngữ giàu tài nguyên (high-resource) nhằm cung cấp góc nhìn so sánh giữa các mô hình pretrained. Các mục tiêu và đóng góp chính của dự án bao gồm:

- Nghiên cứu cơ sở lý thuyết của các phương pháp NMT hiện đại, với trọng tâm là các thách thức và đặc điểm của bài toán dịch máy cho ngôn ngữ ít tài nguyên.
- Xây dựng tập dữ liệu song ngữ Việt-Lào và Việt-Khmer bằng cách mở rộng dữ liệu song ngữ ban đầu với dữ liệu đơn ngữ thông qua phương pháp *back-translation*, nhằm cải thiện chất lượng và quy mô dữ liệu huấn luyện.
- Thiết kế và triển khai pipeline fine-tuning cho các mô hình pretrained, phù hợp với bối cảnh low-resource, và áp dụng pipeline này cho hai mô hình M2M-100 418M và mT5-base 580M.
- Thực hiện fine-tuning hai mô hình M2M-100 418M và mT5-base trên các tập dữ liệu song ngữ Việt-Trung và Việt-Anh có quy mô nhỏ, với mục đích so sánh chất lượng dịch giữa các mô hình trong bối cảnh ngôn ngữ giàu tài nguyên. Phần thực nghiệm này mang tính bổ trợ và không sử dụng pipeline được thiết kế cho bài toán low-resource.
- Đánh giá chất lượng dịch của các mô hình bằng các chỉ số định lượng (sacreBLEU) kết hợp với phân tích định tính, từ đó rút ra nhận xét về hiệu quả của từng mô hình trong các bối cảnh khác nhau.

Chapter 2

CƠ SỞ LÝ THUYẾT

2.1 Tổng quan về Dịch máy

2.1.1 Dịch máy thống kê (Statistical Machine Translation - SMT)

Trước kỷ nguyên Deep Learning, SMT là phương pháp chủ đạo [12, 3]. SMT dựa trên định lý Bayes:

$$\hat{y} = \arg \max_y P(y|x) = \arg \max_y P(x|y)P(y) \quad (2.1)$$

Trong đó $P(x|y)$ là mô hình dịch (translation model) và $P(y)$ là mô hình ngôn ngữ (language model). Tuy nhiên, SMT phụ thuộc nhiều vào việc trích xuất đặc trưng thủ công và khó nắm bắt ngữ cảnh dài.

2.1.2 Dịch máy Neural (Neural Machine Translation - NMT)

NMT mô hình hóa trực tiếp xác suất có điều kiện $P(y|x)$ thông qua một mạng nơ-ron duy nhất [23, 2]:

$$P(y|x) = \prod_{t=1}^T P(y_t|y_{<t}, x; \theta) \quad (2.2)$$

Trong đó θ là tham số của mạng nơ-ron, x là câu nguồn và y là câu đích.

2.2 Kiến trúc Encoder-Decoder và Attention

2.2.1 Encoder-Decoder với RNN

Mô hình Seq2Seq cơ bản sử dụng hai mạng RNN (hoặc LSTM/GRU). Encoder chuyển câu đầu vào $x = (x_1, \dots, x_n)$ thành vector ngữ cảnh c . Decoder sinh ra từ dự đoán y_t dựa trên trạng thái ẩn h_t và c .

Công thức cập nhật trạng thái của RNN:

$$h_t = f(h_{t-1}, x_t) \quad (2.3)$$

2.2.2 Cơ chế Attention (Chú ý)

Attention giúp giải quyết vấn đề "nút thắt cổ chai" (bottleneck) của vector ngữ cảnh cố định [2, 15]. Tại mỗi bước thời gian i của decoder, mô hình tính toán trọng số chú ý α_{ij} tới các trạng thái ẩn của encoder:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \quad (2.4)$$

Trong đó $e_{ij} = \text{score}(s_{i-1}, h_j)$ là hàm tính độ tương đồng.

2.3 Kiến trúc Transformer

Transformer [29] là kiến trúc hiện đại nhất hiện nay, loại bỏ hoàn toàn hồi quy (recurrence) và chỉ sử dụng cơ chế Self-Attention.

2.3.1 Scaled Dot-Product Attention

Đây là thành phần cốt lõi của Transformer:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \quad (2.5)$$

Trong đó Q (Query), K (Key), V (Value) là các ma trận biểu diễn từ.

2.3.2 Multi-Head Attention

Cho phép mô hình tập trung vào các vị trí khác nhau trong không gian biểu diễn:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \quad (2.6)$$

với $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$.

2.3.3 Mạng Feed-Forward (FFN)

Mỗi lớp trong Encoder và Decoder chứa một mạng Feed-Forward kết nối hoàn toàn, được áp dụng riêng biệt và đồng nhất cho từng vị trí. Nó bao gồm hai phép biến đổi tuyến tính với một hàm kích hoạt ReLU ở giữa:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (2.7)$$

Mạng này giúp mô hình học được các đặc trưng phi tuyến tính phức tạp từ các biểu diễn thu được sau cơ chế Attention.

2.3.4 Mã hóa vị trí (Positional Encoding)

Vì kiến trúc Transformer không chứa các vòng lặp (recurrence) hay phép nhân chập (convolution), nó không có thông tin về thứ tự tương đối của các từ trong chuỗi. Để khắc phục điều này, chúng tôi thêm vào các "positional encodings" ở dưới cùng của các ngăn Encoder và Decoder. Chúng tôi sử dụng các hàm sine và cosine với các tần số khác nhau:

$$PE_{(pos, 2i)} = \sin(pos/10000^{2i/d_{model}}) \quad (2.8)$$

$$PE_{(pos, 2i+1)} = \cos(pos/10000^{2i/d_{model}}) \quad (2.9)$$

Trong đó pos là vị trí và i là chiều. Việc sử dụng các hàm lượng giác cho phép mô hình dễ dàng học cách tập trung vào các vị trí tương đối, vì với bất kỳ khoảng cách cố định k nào, PE_{pos+k} có thể được biểu diễn dưới dạng hàm tuyến tính của PE_{pos} .

2.3.5 Chuẩn hóa lớp (Layer Normalization) và Kết nối dư (Residual Connections)

Để ổn định quá trình huấn luyện và cho phép mô hình sâu hơn, mỗi thành phần (Self-Attention, FFN) đều được bao quanh bởi một kết nối dư, sau đó là bước chuẩn hóa lớp:

$$\text{Output} = \text{LayerNorm}(x + \text{Sublayer}(x)) \quad (2.10)$$

Cơ chế này giúp giảm thiểu hiện tượng triệt tiêu gradient (vanishing gradient) trong các mô hình lớn như mT5 hay M2M-100.

2.3.6 Chiến lược giải mã (Decoding Strategy): Beam Search

Trong giai đoạn suy luận (Inference), thay vì chọn từ có xác suất cao nhất tại mỗi bước (Greedy Search), chúng tôi sử dụng Beam Search. Thuật toán này giữ lại k giả thuyết (beams) có xác suất tích lũy cao nhất tại mỗi bước thời gian:

$$\beta = \arg \max_{y_1, \dots, y_t} \sum_{i=1}^t \log P(y_i | y_{<i}, x) \quad (2.11)$$

Điều này giúp tránh việc rơi vào cực trị cục bộ và tạo ra các câu dịch có tính trôi chảy cao hơn.

2.4 Các chỉ số đánh giá

2.4.1 BLEU (Bilingual Evaluation Understudy)

BLEU [17] đo lường độ trùng khớp n-gram giữa câu máy dịch và câu tham chiếu:

$$\text{BLEU} = BP \cdot \exp \left(\sum_{n=1}^N w_n \log p_n \right) \quad (2.12)$$

Trong đó BP là Brevity Penalty (phạt câu ngắn), p_n là độ chính xác n-gram.

2.4.2 sacreBLEU

sacreBLEU [19] là phiên bản chuẩn hoá của BLEU, nhằm đảm bảo tính tái lập (reproducibility) khi so sánh các hệ thống dịch máy. Chỉ số này sử dụng cùng công thức với BLEU truyền thống, nhưng cố định các thành phần như cách tokenize, xử lý chữ hoa/thường và tham chiếu.

$$\text{sacreBLEU} = BP \cdot \exp \left(\sum_{n=1}^N w_n \log p_n \right) \quad (2.13)$$

Trong đó BP là Brevity Penalty, p_n là độ chính xác n-gram, và w_n là trọng số tương ứng. Việc chuẩn hoá quy trình tính toán giúp sacreBLEU trở thành thước đo được khuyến nghị trong các nghiên cứu dịch máy hiện đại.

2.4.3 chrF (Character F-score)

chrF [18] là chỉ số đánh giá dựa trên mức ký tự (character-level), đo lường độ tương đồng giữa câu dịch và câu tham chiếu thông qua F-score của n-gram ký tự. Chỉ số này đặc biệt hiệu quả đối với các ngôn ngữ có hình thái phong phú hoặc không phân tách từ rõ ràng.

$$\text{chrF}_\beta = (1 + \beta^2) \cdot \frac{\text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}} \quad (2.14)$$

Trong đó Precision và Recall được tính trên các n-gram ký tự, và β điều chỉnh tầm quan trọng tương đối giữa Recall và Precision (thường $\beta = 2$).

Chapter 3

CÁC CÁCH TIẾP CẬN LIÊN QUAN

3.1 Các mô hình NMT phổ biến

3.1.1 RNN-based NMT

Các nghiên cứu ban đầu [23, 5] sử dụng LSTM đa lớp. Mặc dù cải thiện so với SMT, mô hình này gặp khó khăn trong việc huấn luyện song song và ghi nhớ các câu quá dài.

3.1.2 Transformer và Pre-trained Models

Sự ra đời của Transformer đã thay đổi hoàn toàn bài toán dịch máy. Các mô hình Pre-trained lớn như mBART [13], T5 [20], và M2M-100 [7] đã chứng minh hiệu quả vượt trội nhờ cơ chế Transfer Learning [32]: huấn luyện trước trên dữ liệu đơn ngữ khổng lồ, sau đó fine-tune trên dữ liệu song ngữ ít ỏi.

3.2 Chiến lược cho ngôn ngữ ít tài nguyên (Low-resource NMT)

3.2.1 Multilingual NMT

Thay vì huấn luyện mô hình riêng lẻ (One-to-One), cách tiếp cận đa ngôn ngữ (Many-to-Many) [10, 1] cho phép chia sẻ tham số giữa các cặp ngôn ngữ giàu tài nguyên và ít tài nguyên. Ví dụ: huấn luyện chung Việt-Anh và Việt-Lào để mô hình học được cấu trúc tiếng Việt tốt hơn.

3.2.2 Back-translation (Dịch ngược)

Sennrich et al. [21] đề xuất sử dụng dữ liệu đơn ngữ (monolingual data) của ngôn ngữ đích, dùng một mô hình dịch ngược để tạo ra dữ liệu song ngữ giả (synthetic data), từ đó làm giàu tập huấn luyện. Kỹ thuật này đã được áp dụng thành công trong các cuộc thi VLSP [26].

3.3 Nghiên cứu hiện có về Việt - Lào/Khmer

3.3.1 Dịch máy tiếng Việt

Nhiều nghiên cứu trong nước [16, 30] đã áp dụng Transformer cho tiếng Việt, tập trung vào kỹ thuật tách từ (Word Segmentation) sử dụng công cụ như VnCoreNLP hoặc PyVi.

3.3.2 Dịch máy Lào và Khmer

Đối với tiếng Lào và Khmer, các nghiên cứu còn hạn chế. Một số cách tiếp cận sử dụng BPE [22] để xử lý vấn đề không có dấu cách trong tiếng Lào/Khmer, cho kết quả khả quan nhưng bị giới hạn bởi dữ liệu. Các thử nghiệm gần đây trong VLSP 2023 [28] cho thấy các mô hình pretrained như mBART và M2M-100 đạt hiệu quả tốt hơn so với các mô hình vanilla Transformer.

3.3.3 Cuộc thi VLSP 2023

VLSP 2023 (Vietnamese Language and Speech Processing) [28] tổ chức shared task về dịch máy cho cặp ngôn ngữ Việt–Lào và Lào–Việt nhằm thúc đẩy nghiên cứu NMT cho ngôn ngữ ít tài nguyên. Ban tổ chức cung cấp dữ liệu song ngữ (100,000 cặp câu) và đơn ngữ, cho phép áp dụng các kỹ thuật như multilingual NMT và back-translation; hệ thống được đánh giá bằng cả chỉ số tự động (BLEU, SacreBLEU) và đánh giá con người [26]. Tương tự, VLSP 2022 [27] đã tổ chức shared task cho cặp Việt–Trung với 300,000 cặp câu huấn luyện.

3.4 Phân tích ưu nhược điểm

Phương pháp	Ưu điểm	Nhược điểm
SMT	Dễ hiểu, cần ít tài nguyên tính toán	Chất lượng dịch thấp, khó xử lý ngữ pháp phức tạp
RNN-NMT	Xử lý chuỗi tốt hơn SMT	Huấn luyện chậm, khó song song hóa
Transformer	Huấn luyện nhanh, chất lượng cao	Cần lượng dữ liệu lớn, dễ bị Overfitting nếu dữ liệu ít
Fine-tuning mBART	Tận dụng tri thức ngôn ngữ học sẵn có	Cần tài nguyên phần cứng lớn (GPU VRAM cao)

Table 3.1 – So sánh các cách tiếp cận dịch máy

Chapter 4

PHÁT BIỂU BÀI TOÁN & DỮ LIỆU

4.1 Phát biểu bài toán

Bài toán đặt ra là xây dựng một hàm ánh xạ $f : \mathcal{X} \rightarrow \mathcal{Y}$ sao cho với một câu đầu vào bằng tiếng Việt $x \in \mathcal{X}$, mô hình sinh ra câu $y \in \mathcal{Y}$ (tiếng Lào hoặc Khmer) có ý nghĩa tương đương. Mục tiêu tối ưu hóa hàm mất mát Cross-Entropy:

$$\mathcal{L}(\theta) = - \sum_{i=1}^N \log P(y^{(i)} | x^{(i)}; \theta) \quad (4.1)$$

4.2 Đặc điểm ngôn ngữ học và Thách thức xử lý

4.2.1 Tiếng Việt: Ngôn ngữ đơn lập và đa thanh điệu

Tiếng Việt là ngôn ngữ đơn lập, ranh giới từ không phải lúc nào cũng trùng với khoảng trắng. Ví dụ, "học sinh" là một từ đơn vị nhưng có khoảng trắng giữa hai âm tiết. Việc sử dụng thư viện PyVi là cần thiết để thực hiện tách từ (Word Segmentation), giúp mô hình hiểu được các thực thể ngữ nghĩa trọn vẹn thay vì các âm tiết rời rạc.

4.2.2 Tiếng Lào và Tiếng Khmer: Hệ chữ viết Scriptio Continua

Cả tiếng Lào và tiếng Khmer đều thuộc hệ chữ Abugida, điểm khó khăn nhất trong xử lý ngôn ngữ tự nhiên (NLP) cho hai ngôn ngữ này là chúng không sử dụng dấu cách để phân tách từ (*scriptio continua*).

- **Tiếng Lào:** Khoảng trắng chỉ được dùng để phân tách các mệnh đề hoặc câu. Do đó, việc sử dụng LaoNLP giúp xác định ranh giới từ dựa trên các quy tắc âm tiết học.
- **Tiếng Khmer:** Có hệ thống phụ âm phức tạp với các ký tự chân (subscript). Thư viện `khmer-nltk` và các bộ tokenizer tự động được sử dụng để xử lý việc tách từ dựa trên từ điển và học máy, nhằm cung cấp đầu vào chuẩn xác nhất cho mô hình NMT.

4.2.3 Kỹ thuật Back-translation trong bối cảnh Low-resource

Vì dữ liệu song ngữ Việt-Lào/Khmer cực kỳ khan hiếm, Chúng tôi sử dụng Back-translation [21] như một phương pháp tăng cường dữ liệu (Data Augmentation). Chúng tôi sử dụng google translation để dịch ngược dữ liệu đơn ngữ từ tập **Leipzig** và **MOT**. Quá trình này giúp mô hình mục tiêu học được cách cấu trúc câu đích dựa trên phân phối dữ liệu thực tế (real target data), ngay cả khi phần nguồn là dữ liệu tổng hợp (synthetic source).

4.3 Mô tả dữ liệu

4.3.1 Nguồn dữ liệu

Đối với cặp Việt - Lào/Lkhmer, dữ liệu được thu thập từ các nguồn dưới đây, sau đó sử dụng phương pháp Back-translation đối với tập đích đã được đề cập ở trên để tăng tổng số lượng dữ liệu:

- **Wortschatz Leipzig / Leipzig Corpora Collection** [8]: Tập dữ liệu tin tức, báo chí bằng tiếng Việt vào năm 2022 bao gồm khoảng 700,000 dòng, mỗi dòng là một câu hoàn chỉnh. Tập dữ liệu này được nhóm chúng tôi Back-translate sang tiếng Lào và tiếng Khmer để phục vụ cho phần dịch máy NMT từ Lào → Việt và Khmer → Việt.
- **Multilingual Open Text (MOT)**: Tập dữ liệu chứa cả bài báo, âm thanh, hình ảnh và video. Dữ liệu cho ngôn ngữ Lào và Khmer là dữ liệu bài báo, được trích xuất ra khoảng 700,000 dòng, mỗi dòng là một câu hoàn chỉnh. Tập dữ liệu này được nhóm chúng tôi Back-translate ngược lại sang tiếng Việt để phục vụ cho phần dịch máy NMT từ Việt → Lào và Việt → Khmer.

Đối với cặp Việt-Trung/Anh, các tập dữ liệu được sử dụng đều là dữ liệu song ngữ chất lượng cao đã được tiền xử lý sẵn:

- **IWSLT'15 English–Vietnamese** [4]: Tập dữ liệu song ngữ Anh–Việt được xây dựng trong khuôn khổ hội nghị IWSLT 2015, chủ yếu gồm các câu hội thoại và bài nói (TED Talks).
- **VLSP 2022 Chinese–Vietnamese** [27, 26]: Tập dữ liệu song ngữ Trung–Việt do ban tổ chức VLSP 2022 cung cấp, bao gồm 300,000 cặp câu thuộc nhiều lĩnh vực khác nhau như tin tức và văn bản tổng quát.

4.3.2 Phương pháp xử lý dữ liệu thô trực tiếp

Với Việt - Lào/Khmer, để xử lý dữ liệu được lấy từ các nguồn trên, với mỗi tập dữ liệu, nhóm chúng tôi đã xử lý theo một cách riêng:

- **Wortschatz Leipzig / Leipzig Corpora Collection**: Dữ liệu được lưu dưới dạng file txt, được xử lý qua để tạo ra file chứa 700,000 dòng. Sau đó nó được đem qua dịch sang lần lượt tiếng Lào và tiếng Khmer theo batch, mỗi batch chứa 5000 câu, tổng cộng 140 batch. Dữ liệu sau khi được dịch ra thì được tổng hợp lại và được shuffle để đảm bảo tính ngẫu nhiên.
- **Multilingual Open Text (MOT)**: Dữ liệu là một Github Repository chứa file .tar, giải nén file để ra được folder chứa data. Trong folder này là các file JSON

chứa link dẫn tới bài báo, thu thập và xử lý để tạo ra file txt chứa 700,000 dòng. Tương tự như trên, lượng dữ liệu tiếng Lào và Khmer này được Back-translate về tiếng Việt và tổng hợp lại, sau đó shuffle để đảm bảo tính ngẫu nhiên.

Đối với cặp Việt–Trung/Anh, các tập dữ liệu được sử dụng đều là dữ liệu song ngữ chất lượng cao đã được tiền xử lý sẵn, do đó nhóm chúng tôi không áp dụng các bước xử lý và mở rộng dữ liệu phức tạp như đối với các cặp ngôn ngữ ít tài nguyên nêu trên.

4.3.3 Thống kê dữ liệu

Tập dữ liệu	Cặp Việt - Lào	Cặp Việt - Khmer	Cặp Việt - Trung	Cặp Việt - Anh
Train	694,000 cặp câu	694,000 cặp câu	300,000 cặp câu	130,000 cặp câu
Validation	5,000 cặp câu	5,000 cặp câu	1,000 cặp câu	1,200 cặp câu
Test	1,000 cặp câu	1,000 cặp câu	1,000 cặp câu	1,000 cặp câu

Table 4.1 – Thống kê số lượng câu trong tập dữ liệu (Số liệu giả định)

4.4 Tiền xử lý dữ liệu (Preprocessing)

Đây là bước quan trọng nhất quyết định chất lượng mô hình:

1. **Đồng nhất dữ liệu (Uniformity):** Đọc dữ liệu với encoding="utf-8", giữ nguyên dấu cho tiếng Việt và giữ nguyên ký tự Unicode cho tiếng Lào.
2. **Tokenization:**
 - **Tiếng Việt:** Sử dụng thư viện PyVi để tách từ ghép.
 - **Tiếng Lào:** Sử dụng thư viện LaoNLP để tách ra từ
 - **Tiếng Khmer:** Sử dụng thư viện `khmer-nltk` và thư viện `khopilot/km-tokenizer-khmer` trực tiếp trên văn bản thô để học cách tách từ tự động.
 - **Tiếng Trung, tiếng Anh:** Sử dụng tokenize có sẵn từ mô hình.
3. **Chuẩn hóa Dataset:** Đưa dữ liệu vào cấu trúc dữ liệu Dataset của HuggingFace, gồm 2 cột chính là source và target.
4. **Tokenize bằng model:** Tokenization thêm một bước bằng tokenizer của chính model để chuẩn hóa độ dài batch, tránh tình trạng OOM (Out Of Memory).
5. **Gán nhãn chuẩn:** Gán nhãn -100 cho padding tokens ở trong labels.
6. **Soft Filtering:** Giữ lại câu ngắn, nhưng những câu quá dài sẽ bị truncate.

Chapter 5

PHƯƠNG PHÁP & MÔ HÌNH ĐỀ XUẤT

5.1 Kiến trúc mô hình dựa trên Transformer

Trong nghiên cứu này, chúng tôi đề xuất thực nghiệm và so sánh hai kiến trúc Pre-trained lớn dựa trên Transformer, được thiết kế chuyên biệt cho các bài toán đa ngôn ngữ.

5.1.1 Mô hình mT5-base (580 Million Parameters)

mT5 (Multilingual T5) [31] là một biến thể đa ngôn ngữ của mô hình T5 [20]. Mô hình này đã được áp dụng thành công trong các cuộc thi VLSP cho bài toán dịch máy Việt–Lào [26]. Điểm khác biệt cốt lõi là mT5 được huấn luyện trên bộ dữ liệu mC4 bao gồm hơn 101 ngôn ngữ khác nhau.

- **Cơ chế Encoder-Decoder:** Sử dụng cấu trúc Transformer chuẩn nhưng thay đổi cách đánh số vị trí (Relative Positional Bias).
- **Text-to-Text Framework:** Mọi tác vụ NLP đều được chuyển đổi về dạng chuỗi sang chuỗi (sequence-to-sequence).
- **Dung lượng:** Với 580M tham số, mT5-base có khả năng ghi nhớ ngữ nghĩa phức tạp của các ngôn ngữ ít tài nguyên như tiếng Lào và Khmer.

5.1.2 Mô hình M2M-100 (418 Million Parameters)

M2M-100 [7] là mô hình dịch máy đa ngôn ngữ đầu tiên không phụ thuộc vào tiếng Anh làm ngôn ngữ trung gian (pivot). Trong các thử nghiệm tại VLSP 2023 [26], M2M-100 đã chứng tỏ hiệu quả vượt trội cho cặp ngôn ngữ Việt–Lào, đạt điểm BLEU cao hơn đáng kể so với các mô hình vanilla Transformer.

- **Language-Specific Tokens:** Sử dụng các token đặc biệt như <2vi>, <2lo> để chỉ định ngôn ngữ nguồn và đích trực tiếp vào quá trình mã hóa.
- **Cấu trúc:** Mặc dù có số lượng tham số ít hơn mT5-base (418M so với 580M), M2M-100 được tối ưu hóa đặc biệt cho tác vụ dịch máy (Translation-centric), giúp nó hoạt động hiệu quả ngay cả khi tài nguyên dữ liệu hạn chế.

5.2 Quy trình huấn luyện đề xuất (4-Phase Pipeline)

Nhóm nghiên cứu thiết lập một quy trình 4 giai đoạn chặt chẽ để đảm bảo tính nhất quán giữa các mô hình, lấy cảm hứng từ các phương pháp thành công trong VLSP 2022-2023 [26], nhưng được cải tiến với chiến lược fine-tuning nhiều pha:

5.2.1 Phase 1: Full Fine-tuning

Trong pha đầu tiên, toàn bộ các tham số của mô hình (encoder và decoder) đều được mở (unfrozen) và cho phép cập nhật. Huấn luyện được khởi tạo từ bộ trọng số pretrained $\theta^{(0)}$ của mô hình **mT5-base** hoặc **M2M-418M**.

Mục tiêu chính của pha này là cho phép mô hình nhanh chóng thích nghi với phân phối dữ liệu mới của cặp ngôn ngữ mục tiêu, đặc biệt trong bối cảnh ngôn ngữ tài nguyên thấp, nơi sự khác biệt về cú pháp, trật tự từ và hình thái học có thể chưa được mô hình hóa đầy đủ trong giai đoạn pretraining đa ngôn ngữ.

Quá trình cập nhật tham số được mô tả như sau:

$$\theta^{(1)} = \theta^{(0)} - \eta_1 \nabla_{\theta} \mathcal{L}(\theta), \quad (5.1)$$

trong đó η_1 là learning rate tương đối lớn, nhằm thúc đẩy khả năng học nhanh trong giai đoạn đầu. Tuy nhiên, việc fine-tune toàn bộ mô hình với learning rate cao cũng tiềm ẩn nguy cơ catastrophic forgetting, đặc biệt đối với các biểu diễn ngôn ngữ tổng quát.

5.2.2 Phase 2: Encoder Freezing

Sau khi mô hình đã đạt được sự thích nghi ban đầu, pha thứ hai tiến hành đóng băng toàn bộ **encoder**, trong khi **decoder** vẫn được cập nhật.

Do đó, gradient chỉ được lan truyền qua các tham số decoder:

$$\nabla_{\theta_{\text{enc}}} \mathcal{L}(\theta) = 0, \quad \nabla_{\theta_{\text{dec}}} \mathcal{L}(\theta) \neq 0. \quad (5.2)$$

Pha này đóng vai trò như một bước ổn định hóa biểu diễn nguồn (source-side representation stabilization). Việc cố định encoder giúp bảo toàn các biểu diễn ngôn ngữ đã được điều chỉnh ở pha 1, trong khi decoder tiếp tục học cách ánh xạ các biểu diễn đó sang ngôn ngữ đích một cách chính xác hơn. Điều này đặc biệt hữu ích trong bối cảnh dữ liệu huấn luyện hạn chế và không đồng đều.

5.2.3 Phase 3: Partial Encoder Fine-tuning

Trong pha thứ ba, chúng tôi áp dụng chiến lược fine-tuning có chọn lọc bằng cách chỉ mở lại một phần encoder, cụ thể là nửa dưới của encoder (layers 0–5), trong khi nửa trên (layers 6–11) vẫn được giữ cố định. Cụ thể, ta định nghĩa các tập tham số như sau:

θ_f : tập tham số encoder bị freeze (layers 6–11),

θ_u : tập tham số encoder được unfreeze (layers 0–5) cùng toàn bộ decoder.

Khi đó, hàm mất mát chỉ được tối ưu đối với tập tham số θ_u :

$$\theta_u^{(3)} = \theta_u^{(2)} - \eta_3 \nabla_{\theta_u} \mathcal{L}(\theta). \quad (5.3)$$

Chiến lược này dựa trên các giả định sau:

- Các tầng encoder thấp chủ yếu học các đặc trưng ngôn ngữ cơ bản (token-level, cú pháp, hình thái).
- Các tầng encoder cao mang tính trừu tượng và đa ngôn ngữ cao hơn.

Do đó, việc chỉ fine-tune các tầng thấp giúp mô hình điều chỉnh tốt hơn các đặc trưng ngôn ngữ đặc thù của cặp ngôn ngữ mục tiêu, đồng thời giữ ổn định các biểu diễn trừu tượng đã được học trong quá trình pretraining quy mô lớn.

5.2.4 Phase 4: Low Learning Rate Full Fine-tuning

Trong pha cuối cùng, toàn bộ các tầng của mô hình được mở lại hoàn toàn và tiếp tục fine-tune với learning rate nhỏ hơn đáng kể so với pha 1:

$$\eta_4 \ll \eta_1. \quad (5.4)$$

Quá trình cập nhật tham số được mô tả như sau:

$$\theta^{(4)} = \theta^{(3)} - \eta_4 \nabla_{\theta} \mathcal{L}(\theta). \quad (5.5)$$

Mục tiêu của pha này là tinh chỉnh toàn bộ mô hình một cách nhẹ nhàng, giúp đạt được sự hội tụ ổn định và cải thiện chất lượng dịch cuối cùng. Learning rate thấp giúp hạn chế các cập nhật đột ngột, tránh phá vỡ những biểu diễn đã được ổn định và tinh lọc qua các pha trước đó.

5.2.5 Tổng quan về quá trình huấn luyện qua nhiều phase

Chiến lược huấn luyện nhiều pha được thiết kế nhằm cân bằng giữa khả năng thích nghi và tính ổn định của mô hình trong bối cảnh dịch máy với ngôn ngữ tài nguyên thấp. Pha 1 fine-tune toàn bộ mô hình để thích nghi nhanh với phân phối dữ liệu mới. Pha 2 đóng băng encoder nhằm ổn định các biểu diễn nguồn, trong khi pha 3 chỉ mở lại một phần encoder để tinh chỉnh các đặc trưng mang tính nhiệm vụ. Cuối cùng, pha 4 mở lại toàn bộ mô hình với learning rate thấp để tái đồng bộ encoder-decoder và đảm bảo sự hội tụ ổn định.

5.2.6 Thiết lập tham chiếu trên cặp ngôn ngữ high-resource (Phụ lục)

Bên cạnh pipeline huấn luyện nhiều pha được đề xuất cho các cặp ngôn ngữ tài nguyên thấp (Việt–Lào/Khmer), nhóm nghiên cứu xây dựng thêm một thí nghiệm phụ trên một cặp ngôn ngữ tài nguyên cao nhằm cung cấp bối cảnh tham chiếu cho việc đánh giá chất lượng dịch máy. Thí nghiệm này được đưa vào dưới dạng phụ lục phương pháp, với mục đích mô tả cấu hình fine-tuning và thiết lập đánh giá, không nhằm thực hiện so sánh giữa các kiến trúc mô hình khác nhau.

Trong thí nghiệm phụ này, nhóm nghiên cứu chỉ sử dụng mô hình pretrained M2M-100 (418M), vốn được thiết kế chuyên biệt cho bài toán dịch đa ngôn ngữ. Mô hình được fine-tune trên một tập dữ liệu song ngữ quy mô nhỏ được trích xuất từ một cặp ngôn ngữ high-resource, nhằm quan sát hành vi và mức chất lượng dịch đạt được trong điều kiện dữ liệu tương tự với các thí nghiệm low-resource.

Khác với bài toán low-resource, pipeline huấn luyện nhiều pha không được áp dụng trong thí nghiệm phụ này. Thay vào đó, mô hình **M2M-100** được fine-tune theo một cấu hình đơn giản với toàn bộ encoder và decoder được mở, learning rate cố định và số epoch giới hạn. Cách thiết lập này giúp tách biệt ảnh hưởng của chiến lược huấn luyện khỏi đặc tính của dữ liệu, đồng thời cung cấp một điểm tham chiếu (reference point) cho các kết quả thực nghiệm được trình bày ở phần sau.

Chapter 6

CÀI ĐẶT HỆ THỐNG & THỰC NGHIỆM

6.1 Cấu hình cho hệ thống

Toàn bộ các thí nghiệm được thực hiện trên hệ thống sử dụng GPU để tăng tốc quá trình huấn luyện. Môi trường phần cứng và phần mềm được cấu hình như sau:

- **Phần cứng:** GPU NVIDIA H200 SXM5 với bộ nhớ 141GB đủ lớn để huấn luyện các mô hình M2M-100 với 418M tham số và mô hình MT5-base với 580M tham số.
- **Hệ điều hành:** Linux trên Cloud.
- **Ngôn ngữ lập trình:** Python.
- **Thư viện chính:** PyTorch và HuggingFace Transformers.

Các mô hình pretrained `mT5-base` và `M2M-100` được tải trực tiếp từ HuggingFace Hub và được fine-tune cho tác vụ dịch máy song ngữ.

6.2 General Training Setup

Đối với tất cả các pha huấn luyện, chúng tôi sử dụng cùng một hàm mất mát là negative log-likelihood trên chuỗi đầu ra. Quá trình huấn luyện được thực hiện theo cơ chế teacher forcing, trong đó token mục tiêu tại thời điểm $t - 1$ được sử dụng làm đầu vào để dự đoán token tại thời điểm t .

Optimizer được sử dụng là Adam [11] hoặc AdamW [14], vốn phổ biến trong huấn luyện các mô hình Transformer. Learning rate được điều chỉnh tùy theo từng pha nhằm kiểm soát mức độ cập nhật tham số.

Để đảm bảo tính nhất quán của pipeline, trọng số mô hình sau mỗi phase được lưu lại và sử dụng làm điểm khởi tạo cho phase huấn luyện tiếp theo.

Ngoài ra toàn bộ các phase còn được áp dụng early stopping để tránh mô hình overfitting.

6.2.1 Phase 1 Training Configuration

Trong pha đầu tiên, mô hình được fine-tune toàn bộ tham số (full fine-tuning) bắt đầu từ trọng số pretrained ban đầu. Pha này đóng vai trò giúp mô hình nhanh chóng thích

ngiht với cặp ngôn ngữ mục tiêu.

Learning rate trong pha 1 được lựa chọn tương đối lớn so với các pha sau nhằm thúc đẩy khả năng học nhanh. Số epoch của pha này được lựa chọn sao cho mô hình đạt được sự cải thiện rõ rệt về hàm mất mát trên tập huấn luyện.

6.2.2 Phase 2 Training Configuration

Trong pha thứ hai, toàn bộ các tham số của encoder được đóng băng (freeze), trong khi các tham số của decoder vẫn được phép cập nhật. Mục tiêu của pha này là ổn định các biểu diễn nguồn đã được điều chỉnh ở pha trước, đồng thời cho phép decoder tiếp tục thích nghi với nhiệm vụ sinh ngôn ngữ đích.

Mặc dù encoder không được cập nhật, pha này vẫn được huấn luyện trong một số epoch cố định nhằm duy trì tính liên tục của quá trình tối ưu và hạn chế sự dao động đột ngột khi chuyển sang các chiến lược fine-tuning tiếp theo.

Cụ thể, pha 2 được huấn luyện trong **3 epoch** với learning rate được thiết lập ở mức vừa phải:

$$\eta_2 = 5 \times 10^{-5}. \quad (6.1)$$

Việc sử dụng learning rate này giúp decoder điều chỉnh một cách ổn định, đồng thời đảm bảo quá trình chuyển tiếp sang các pha fine-tuning có chọn lọc và fine-tuning toàn bộ sau đó diễn ra một cách mượt mà.

6.2.3 Phase 3 Training Configuration

Pha thứ ba áp dụng chiến lược fine-tuning có chọn lọc, trong đó chỉ một nửa số tầng của mô hình được phép cập nhật. Các tầng thấp được giữ cố định nhằm bảo toàn các biểu diễn ngôn ngữ tổng quát, trong khi các tầng cao được fine-tune để thích nghi tốt hơn với tác vụ dịch máy.

Pha này được huấn luyện trong **3 epoch**, với learning rate giống pha 2:

$$\eta_3 = 5 \times 10^{-5}. \quad (6.2)$$

Số epoch giới hạn giúp tránh hiện tượng overfitting trong khi vẫn cho phép mô hình điều chỉnh các biểu diễn mang tính nhiệm vụ ở các tầng cao.

6.2.4 Phase 4 Training Configuration

Trong pha cuối cùng, toàn bộ các tham số của mô hình được unfreeze và fine-tune trở lại, tương tự pha 1. Tuy nhiên, learning rate được giảm đáng kể nhằm đảm bảo quá trình hội tụ ổn định.

Cụ thể, pha 4 được huấn luyện trong **4 epoch** với learning rate:

$$\eta_4 = 1 \times 10^{-5}. \quad (6.3)$$

Learning rate thấp cho phép tinh chỉnh toàn bộ mô hình một cách nhẹ nhàng, giúp cải thiện chất lượng dịch cuối cùng mà không làm suy giảm các biểu diễn đã được học và ổn định trong các pha trước.

6.2.5 Phụ lục: Cấu hình fine-tune cho cặp ngôn ngữ high-resource

Đối với thí nghiệm phụ trên cặp ngôn ngữ tài nguyên cao, nhóm nghiên cứu sử dụng một cấu hình fine-tuning đơn giản cho mô hình pretrained M2M-100 (418M). Mục tiêu của cấu hình này là cung cấp một thiết lập tham chiếu cho bài toán dịch máy trên ngôn ngữ high-resource, đồng thời tránh sự can thiệp của các chiến lược huấn luyện nhiều pha được thiết kế riêng cho bối cảnh low-resource.

Cụ thể, toàn bộ các tham số của mô hình (encoder và decoder) đều được mở và cập nhật trong suốt quá trình huấn luyện. Không áp dụng các chiến lược đóng băng từng phần hay fine-tuning có chọn lọc như trong pipeline 4 pha.

Quá trình fine-tuning được thực hiện với learning rate cố định:

$$\eta = 5 \times 10^{-5}, \quad (6.4)$$

và được huấn luyện trong **10 epoch**. Các thành phần còn lại của thiết lập huấn luyện, bao gồm hàm mất mát negative log-likelihood, cơ chế teacher forcing, optimizer (Adam/AdamW) và early stopping, được giữ nguyên và nhất quán với cấu hình tổng quát đã mô tả ở phần *General Training Setup*.

Cấu hình fine-tuning này đóng vai trò như một thiết lập tham chiếu (reference setup) cho bài toán dịch máy trên cặp ngôn ngữ high-resource, và được sử dụng để cung cấp bối cảnh so sánh khi phân tích các kết quả thực nghiệm ở phần sau.

6.3 Tổng kết về việc lựa chọn siêu tham số và cấu hình trong từng phase

Phase	Trainable Layers	Epochs	Learning Rate
Phase 1	All layers	–	η_1
Phase 2	None (Frozen)	3	5×10^{-5}
Phase 3	Half of layers	3	5×10^{-5}
Phase 4	All layers	4	1×10^{-5}

Table 6.1 – Training configuration cho từng phase

Setting	Trainable Layers	Epochs	Learning Rate
High-resource fine-tuning	All layers	10	5×10^{-5}

Table 6.2 – Cấu hình fine-tuning cho thí nghiệm phụ trên cặp ngôn ngữ high-resource

Chapter 7

ĐÁNH GIÁ & PHÂN TÍCH KẾT QUẢ

7.1 Kết quả thực nghiệm trên các mô hình

Dựa trên quá trình huấn luyện 4 giai đoạn, chúng tôi thu được kết quả so sánh giữa mT5 và M2M-100 cho cặp ngôn ngữ Việt - Lào và Việt - Khmer.

Direction	mT5-base	M2M-100
vi \rightarrow lo	23.31	28.72
lo \rightarrow vi	42.89	52.01
vi \rightarrow khm	29.57	23.82
khm \rightarrow vi	42.45	52.34

Table 7.1 – So sánh BLEU giữa mT5-base và M2M-100 theo từng hướng dịch

Nhằm cung cấp bối cảnh tham chiếu cho các kết quả trên cặp ngôn ngữ tài nguyên thấp, nhóm nghiên cứu bổ sung một thí nghiệm phụ trên các cặp ngôn ngữ tài nguyên cao sử dụng mô hình M2M-100. Thí nghiệm này áp dụng cấu hình fine-tuning đơn giản như đã mô tả ở phần phụ lục phương pháp. Kết quả BLEU trên các cặp Việt–Trung và Việt–Anh được trình bày trong Bảng 7.2.

Direction	BLEU (M2M-100)
vi \rightarrow zh	36.97
zh \rightarrow vi	36.84
vi \rightarrow en	29.65
en \rightarrow vi	29.57

Table 7.2 – Kết quả BLEU của mô hình M2M-100 trên các cặp ngôn ngữ high-resource

7.2 Phân tích hiệu suất mô hình

7.2.1 Đặc điểm của mT5-base

Mô hình mT5-base, với kiến trúc text-to-text tổng quát và dữ liệu tiền huấn luyện đa dạng, cho thấy khả năng sinh câu dịch có độ trôi chảy tốt, đặc biệt trong các cấu trúc câu dài hoặc phức hợp. Trong một số hướng dịch, chẳng hạn Việt sang Khmer, mT5-base đạt BLEU cao hơn, cho thấy lợi thế nhất định trong việc mô hình hóa ngữ cảnh nguồn và cấu trúc cú pháp khi ngôn ngữ đích có khoảng cách ngôn ngữ lớn với tiếng Việt.

Tuy nhiên, do không được thiết kế chuyên biệt cho tác vụ dịch máy, hiệu năng của mT5-base vẫn phụ thuộc đáng kể vào hướng dịch và phân bố dữ liệu huấn luyện.

7.2.2 Hiệu quả của M2M-100

Mô hình M2M-100 thể hiện hiệu năng ổn định và vượt trội hơn trong phần lớn các hướng dịch, đặc biệt là các hướng dịch từ Lào và Khmer sang tiếng Việt, nơi mô hình đạt điểm BLEU cao hơn rõ rệt. Điều này phù hợp với mục tiêu thiết kế ban đầu của M2M-100, vốn được tối ưu trực tiếp cho bài toán dịch máy đa ngôn ngữ với nhiều cặp ngôn ngữ khác nhau.

Bên cạnh điểm BLEU cao, M2M-100 cũng cho thấy khả năng xử lý tốt các thực thể tên riêng và các thuật ngữ chuyên biệt, góp phần nâng cao tính nhất quán và độ chính xác của bản dịch trong bối cảnh ngôn ngữ tài nguyên thấp.

7.3 Phân tích định tính (Qualitative Analysis)

Dưới đây là một số mẫu kết quả từ Phase 4 (Random Samples), ví dụ với Vi \rightarrow Lo:

- **Input (VI):** “Chúng tôi đang học tại trường đại học.”
- **Reference (LO):** “ພວກເຮົາກຳລັງຮຽນຢູ່ມະຫາວິທະຍາໄລ.”
- **mT5 Prediction:** “ພວກ ຂ້າພະເຈົ້າ ກຳລັງ ຮຽນ ຢູ່ ມະຫາວິທະຍາໄລ.” (Bảo toàn nội dung nhưng sai đại từ số nhiều.)
- **M2M Prediction:** “ພວກເຮົາ ກຳລັງ ຮຽນ ຢູ່ ມະຫາວິທະຍາໄລ.” (Dịch đúng và sát câu tham chiếu.)

Với Vi \rightarrow Khm:

- **Input (VI):** “Chúng tôi đang học tại trường đại học.”
- **Reference (KHM):** “ពួកយើងកំពុងសិក្សានៅសាកលវិទ្យាល័យ.”
- **mT5 Prediction:** “យើង រៀន នៅ មហាវិទ្យាល័យ.” (Đúng ý chính nhưng mất thì tiếp diễn.)
- **M2M Prediction:** “យើង កំពុង សិក្សា នៅ សាកលវិទ្យាល័យ.” (Bảo toàn đầy đủ ngữ nghĩa.)

7.4 Phân tích lỗi và Thách thức

1. **Lỗi Tokenization:** Tiếng Lào không có dấu cách khiến việc xác định biên giới từ (word boundary) đôi khi bị sai, dẫn đến việc mô hình sinh ra các chuỗi ký tự vô nghĩa (hallucination). Vấn đề này cũng được ghi nhận trong báo cáo VLSP 2023 [26].
2. **Xử lý tên riêng và số liệu:** Các nghiên cứu từ VLSP 2022 [26] cho thấy việc áp dụng post-processing cho các giá trị số và ngày tháng có thể cải thiện đáng kể chất lượng dịch, mặc dù không trực tiếp tăng điểm BLEU.

7.5 Đánh giá tổng quát

Phương pháp tiếp cận dựa trên các mô hình pretrained kết hợp với chiến lược fine-tuning nhiều giai đoạn cho thấy tính hiệu quả rõ rệt trong bối cảnh dịch máy cho các cặp ngôn ngữ tài nguyên thấp. Quy trình huấn luyện 4 pha được thiết kế nhằm cân bằng giữa khả năng thích nghi nhanh với dữ liệu mục tiêu và tính ổn định của các biểu diễn đã được học trong giai đoạn tiền huấn luyện, từ đó cải thiện chất lượng dịch trong điều kiện dữ liệu hạn chế.

Các kết quả thực nghiệm cho thấy hiệu năng của mô hình không chỉ phụ thuộc vào quy mô tham số, mà còn chịu ảnh hưởng đáng kể từ kiến trúc và mục tiêu tiền huấn luyện ban đầu. Trong khi các mô hình tổng quát như **mT5-base** thể hiện lợi thế nhất định ở một số hướng dịch cụ thể, các mô hình được tối ưu chuyên biệt cho dịch máy đa ngôn ngữ như **M2M-100** cho thấy hiệu năng ổn định và vượt trội hơn trong phần lớn các thiết lập low-resource.

Bên cạnh đó, các thí nghiệm phụ trên cặp ngôn ngữ tài nguyên cao được sử dụng như một bối cảnh tham chiếu, giúp làm rõ sự khác biệt về mức chất lượng dịch đạt được giữa các kịch bản dữ liệu khác nhau. Tuy nhiên, các kết quả này không nhằm mục đích so sánh trực tiếp với các cặp ngôn ngữ tài nguyên thấp, mà chỉ đóng vai trò hỗ trợ cho việc diễn giải và đánh giá phương pháp đề xuất.

Chapter 8

KẾT LUẬN & HƯỚNG PHÁT TRIỂN

8.1 Tổng kết

Trong bài tập lớn này, nhóm nghiên cứu đã đạt được các mục tiêu chính sau:

1. Khảo sát các phương pháp dịch máy neural hiện đại và phân tích đặc thù của các cặp ngôn ngữ Việt–Lào và Việt–Khmer trong bối cảnh tài nguyên thấp.
2. Thu thập, xây dựng và tiền xử lý các bộ dữ liệu song ngữ từ các nguồn mở phục vụ huấn luyện mô hình.
3. Đề xuất và triển khai quy trình fine-tuning nhiều pha dựa trên các mô hình pre-trained, áp dụng cho hai cặp ngôn ngữ nghiên cứu.
4. Thực nghiệm và đánh giá cho thấy pipeline huấn luyện đề xuất giúp cải thiện chất lượng dịch, đồng thời làm rõ sự khác biệt về hiệu năng giữa các mô hình pretrained trong điều kiện dữ liệu hạn chế.

8.2 Hạn chế

- Quy mô dữ liệu huấn luyện còn hạn chế so với các nghiên cứu quốc tế, mặc dù đã áp dụng back-translation để mở rộng dữ liệu tương tự như các đội hàng đầu trong VLSP [26].
- Chưa áp dụng được các kỹ thuật nâng cao như Data Augmentation hay gói gọn việc huấn luyện nhiều chiều dịch trong một model do hạn chế về thời gian và tài nguyên.
- Mặc dù điểm BLEU đạt mức tương đối cao, cần lưu ý rằng các đánh giá tự động có thể không phản ánh đầy đủ chất lượng dịch. Kinh nghiệm từ VLSP [26] cho thấy đánh giá bởi con người thường cho kết quả khác biệt đáng kể so với các chỉ số tự động.

8.3 Hướng phát triển

Để cải thiện hệ thống trong tương lai, nhóm đề xuất:

- **Mở rộng dữ liệu:** Thu thập thêm dữ liệu đơn ngữ và sử dụng các mô hình dịch lớn để Back-translation cho việc tạo dữ liệu giả thay thế cho google translation.
- **Sử dụng Pre-trained Models:** Fine-tune các mô hình lớn như mBART-50 [24] hoặc NLLB-200 [25] (No Language Left Behind) của Meta, vốn hỗ trợ rất tốt tiếng Lào và Khmer.
- **Tối ưu hóa mô hình:** Sử dụng Knowledge Distillation [9] để tạo ra mô hình nhỏ gọn hơn (Lightweight NMT) có thể chạy trên thiết bị di động.

Table 8.1 – Bảng phân công công việc

Nội dung công việc	Vũ Đức Minh (23020401)	Tạ Nguyên Thành (23020437)	Nguyễn Đức Huy (23020376)
<i>Giai đoạn 1: Chuẩn bị</i>			
Xác định bài toán, mục tiêu	X	X	X
Tìm kiếm và tiền xử lí dataset	X	X	
Xây dựng pipeline mô hình	X	X	X
<i>Giai đoạn 2: Huấn luyện & Đánh giá</i>			
Việt - Lào	X	X	
Việt - Khmer	X	X	
Việt - Trung			X
Việt - Anh			X
<i>Giai đoạn 3: Hoàn thiện</i>			
Xây dựng hệ thống demo		X	
Viết báo cáo	X	X	
Thiết kế slide thuyết trình			X

Bibliography

- [1] Roei Aharoni, Melvin Johnson, and Orhan Firat. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3874–3884, 2019.
- [2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR)*, 2015.
- [3] Peter F Brown, Stephen A Della Pietra, Vincent J Della Pietra, and Robert L Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311, 1993.
- [4] Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. The iwslt 2015 evaluation campaign. In *Proceedings of the 12th International Workshop on Spoken Language Translation (IWSLT)*, pages 2–14, 2015.
- [5] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014.
- [6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, 2019.
- [7] Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48, 2021.
- [8] Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. Building large monolingual dictionaries at the leipzig corpora collection: From 100 to 200 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC’12)*, pages 759–765, 2012.
- [9] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
- [10] Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. Google’s multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351, 2017.

- [11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015.
- [12] Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54, 2003.
- [13] Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation. In *Transactions of the Association for Computational Linguistics*, volume 8, pages 726–742, 2020.
- [14] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.
- [15] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, 2015.
- [16] Dat Quoc Nguyen and Anh Tuan Nguyen. Vncorenlp: A vietnamese natural language processing toolkit. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 56–60, 2018.
- [17] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.
- [18] Maja Popović. chrF: character n-gram f-score for automatic mt evaluation. In *Proceedings of the tenth workshop on statistical machine translation*, pages 392–395, 2015.
- [19] Matt Post. A call for clarity in reporting bleu scores. *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, 2018.
- [20] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.
- [21] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, 2016.
- [22] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, 2016.
- [23] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.
- [24] Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. Multilingual translation with extensible multilingual pretraining and finetuning. In *arXiv preprint arXiv:2008.00401*, 2020.

- [25] NLLB Team, Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, et al. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*, 2022.
- [26] Hong Viet Tran, Minh Quy Nguyen, and Van Vinh Nguyen. ViBidirectionMT-Eval: Machine translation for vietnamese-chinese and vietnamese-lao language pair. *Journal of Computer Science and Cybernetics*, 41(3):285–304, 2025.
- [27] Hong Viet Tran and Van Vinh Nguyen. VLSP 2022 shared task: Machine translation. In *Proceedings of the 9th International Workshop on Vietnamese Language and Speech Processing (VLSP)*, 2022. Vietnamese-Chinese Machine Translation Shared Task.
- [28] Hong Viet Tran and Van Vinh Nguyen. VLSP 2023 shared task: Machine translation. In *Proceedings of the 10th International Workshop on Vietnamese Language and Speech Processing (VLSP)*, 2023. Vietnamese-Lao Machine Translation Shared Task.
- [29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.
- [30] Xuan-Son Vu, Thanh Vu, Mai-Vu Tran, and Phuong Le-Hong. Vlsr shared task: Machine translation. In *Proceedings of the 5th Workshop on Vietnamese Language and Speech Processing (VLSP)*, 2018.
- [31] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, 2021.
- [32] Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, 2016.