Báo cáo xây dựng mô hình Transformer và áp dụng cho bài toán VLSP

Trương Văn Hải
23020058@vnu.edu.vn

Trần Xuân Phong
23021657@vnu.edu.vn

Nguyễn Anh Tú
23020147@vnu.edu.vn

Tóm tắt nội dung

Bài báo cáo này trình bày việc ứng dụng mô hình Transformer trong xử lý ngôn ngữ tự nhiên, để giải quyết bài toán dịch máy nói chung và trong khuôn khổ VLSP (Vietnamese Language and Speech Processing) nói riêng. Phương pháp tiếp cận bao gồm việc xây dựng kiến trúc Transformer hoàn chỉnh từ đầu, cài đặt các thành phần cơ bản như Encoder, Decoder, cơ chế Multi-Head Attention (Sau này đổi thành Grouped Query Attention), Positional Encoding để bảo toàn thông tin vị trí, và các kỹ thuật chuẩn hóa như Layer (RMS) Normalization để ổn định quá trình huấn luyện. Sau quá trình huấn luyện kéo dài 10 epoch, mô hình đã đạt được các kết quả khả quan. Đánh giá trên tập IWSLT'15 English-Vietnamese cho thấy mô hình đạt được BLEU Score là 0.258, minh chứng cho khả năng dịch thuật của hệ thống. Về việc áp dụng cho bài toán VLSP, nhóm đã huấn luyện mô hình từ đầu và đạt chỉ số BLEU là 0.48, TER: 39.57, MEOTEOR: 0.71.

1 Giới thiệu

Dịch máy (Machine Translation - MT) đóng vai trò ngày càng quan trọng trong thế giới kết nối toàn cầu ngày nay, phá vỡ rào cản ngôn ngữ và tạo điều kiện thuận lợi cho giao tiếp, thương mại và trao đổi kiến thức xuyên biên giới. Từ việc dịch các trang web và tài liệu cho đến hỗ trợ giao tiếp tức thời, các hệ thống dịch máy hiện đại đã trở thành một công cụ không thể thiếu trong nhiều lĩnh vực.

Trong bối cảnh xử lý ngôn ngữ tự nhiên tiếng Việt, bài toán Dịch Máy Ngôn ngữ và Xử lý Tiếng nói Việt (Vietnamese Language and Speech Processing - VLSP) từ tiếng Việt sang tiếng Anh mang đến những thách thức đặc thù. Tiếng Việt là một ngôn ngữ đơn lập, giàu ngữ cảnh, với cấu trúc ngữ pháp và từ vựng khác biệt đáng kể so với tiếng Anh. Điều này đòi hỏi các mô hình dịch máy phải có khả năng hiểu sâu sắc ngữ nghĩa, xử lý các hiện tượng đa nghĩa, và tạo ra các bản dịch tự nhiên, chính xác trong một ngôn ngữ đích có cấu trúc phức tạp hơn.

Động lực chính đằng sau việc xây dựng và triển khai mô hình Transformer từ đầu trong báo cáo này xuất phát từ mong muốn hiểu sâu sắc cơ chế hoạt động bên trong của kiến trúc này, cũng như khả năng tùy chỉnh và tối ưu hóa nó cho các đặc điểm riêng của cặp ngôn ngữ Việt-Anh. Mặc dù có nhiều thư viện và framework hiện có cung cấp các triển khai Transformer sẵn sàng sử dụng, việc xây dựng mô hình từ những khối cơ bản giúp củng cố kiến thức nền tảng, cho phép kiểm soát hoàn toàn các thành phần, từ cơ chế attention, các lớp feed-forward, cho đến quy trình huấn luyện và tối ưu hóa. Điều này không chỉ phục vụ mục đích giáo dục mà còn mở ra cơ hội thử nghiệm các biến thể hoặc cải tiến cụ thể phù hợp với dữ liệu và yêu cầu của VLSP.

2 Phân tích và tiền xử lý tập dữ liệu cho mô hình nền

2.1 Về bộ dữ liệu IWSLT 2015

Chúng tôi sử dụng bộ dữ liệu dịch máy IWSLT'15 làm tập dữ liệu nền cho mô hình. Bộ dữ liệu này bao gồm các câu được tách từ các bài nói hoặc đoạn hội thoại, do đó một số câu mang tính phụ thuộc ngữ cảnh hoặc tham chiếu đến câu đứng trước. Ví dụ, cặp câu "và may mắn là anh bệnh nhân không chết." – "and fortunately he didn't die." chỉ thực sự đầy đủ ý nghĩa khi được đặt trong ngữ cảnh rộng hơn, điều này có thể ảnh hưởng đến độ chính xác của quá trình đánh giá.

2.2 Tiền xử lý dữ liệu

Dữ liệu thô ban đầu bao gồm khoảng 133,317 cặp câu. Để đảm bảo chất lượng dữ liệu đầu vào cho mô hình, chúng tôi thực hiện chuỗi các bước làm sạch sau:

- **Giải mã HTML:** Dữ liệu gốc chứa các ký tự mã hóa HTML (ví dụ: ' thay cho dấu nháy đơn). Chúng tôi sử dụng thư viện `html`

để chuyển đổi chúng về dạng ký tự văn bản chuẩn, giúp giữ nguyên ngữ nghĩa của câu.

- **Chuẩn hóa văn bản:** Toàn bộ văn bản tiếng Anh và tiếng Việt được chuyển về chữ thường và loại bỏ khoảng trắng thừa ở hai đầu.
- **Lọc bỏ dữ liệu nhiễu:** Chúng tôi phát hiện và loại bỏ các mẫu dữ liệu bị lỗi, cụ thể là các dòng mà câu tiếng Việt hoặc tiếng Anh bị rỗng (ví dụ tại chỉ số 121148 và 121635).
- **Xử lý trùng lặp:** Sau khi chuẩn hóa, chúng tôi phát hiện nhiều cặp câu bị trùng lặp hoàn toàn trong tập huấn luyện. Điển hình là cặp câu "cảm ơn ." - "thank you ." xuất hiện lặp lại 165 lần. Việc loại bỏ các bản ghi trùng lặp giúp giảm thiểu hiện tượng mô hình quá khớp (overfitting) vào các mẫu phổ biến. Sau bước này, kích thước tập huấn luyện còn lại là 132,146 cặp câu.

2.3 Phân tích khám phá dữ liệu mức từ

Trước khi áp dụng các kỹ thuật tách từ con, chúng tôi tiến hành phân tích thống kê trên các đơn vị từ vựng cơ bản (word) để hiểu rõ phân bố dữ liệu.

2.3.1 Thống kê từ vựng và OOV

Dựa trên bộ từ điển cung cấp sẵn kèm theo dữ liệu:

- Kích thước từ vựng tiếng Anh: 15,722 từ.
- Kích thước từ vựng tiếng Việt: 6,469 từ.

Chúng tôi tính toán tỷ lệ từ không nằm trong từ điển (OOV) trên tập huấn luyện. Kết quả cho thấy tỷ lệ OOV trung bình trên toàn bộ tập dữ liệu tiếng Việt rất thấp, chỉ khoảng 0.79%, còn với tiếng Anh là 2.06%. Điều này cho thấy bộ từ vựng đi kèm bao phủ tốt không gian dữ liệu huấn luyện.

2.3.2 Phân bố độ dài câu

Độ dài câu là yếu tố quan trọng ảnh hưởng đến việc lựa chọn tham số $\mathrm{max\_sequence\_length}$ cho mô hình. Thống kê trên tập tiếng Việt cho thấy:

- Độ dài trung bình: khoảng 25 từ/câu.
- Độ lệch chuẩn: 18.77, cho thấy sự biến động lớn về độ dài giữa các câu.
- Phân vị 50%: 20 từ.
- Giá trị cực đại: Có những câu rất dài lên tới 850 từ.

Biểu đồ phân bố độ dài cho thấy dữ liệu lệch phải, tập trung chủ yếu ở các câu ngắn dưới 40 từ. Tham khảo Hình 1. Còn đối với tập tiếng Anh là:

- Độ dài trung bình: khoảng 20 từ/câu.
- Độ lệch chuẩn: 15.01
- Phân vị 50%: 17 từ.
- Giá trị cực đại: Câu dài nhất của tiếng Anh là 628 từ, thấp hơn câu dài nhất tiếng Việt khá nhiều, điều này cũng cho thấy sự khác biệt đáng kể trong cấu trúc cú pháp và xu hướng sử dụng mệnh đề giữa hai ngôn ngữ.

Cũng như tiếng Việt thì biểu đồ histogram tần suất độ dài câu tiếng Anh (Tham khảo Hình 2) cũng lệch phải, cho thấy sự tương đương của hai ngôn ngữ, điều này sẽ được làm rõ hơn trong phần 2.3.4.

2.3.3 Từ phổ biến

Phân tích tần suất xuất hiện của các từ cho thấy các từ xuất hiện nhiều nhất chủ yếu là các từ chức năng hoặc các cụm từ giao tiếp xã giao thường gặp hay là những dấu câu, ví dụ như: ".", "và", "tôi", "là", "một", "có",... như minh họa trong Hình 3. Tiếng Anh thì là: ".", "the", "and", "to", "of", "a", "that",... như minh họa trong Hình 4

2.3.4 Tỉ lệ độ dài câu giữa hai ngôn ngữ

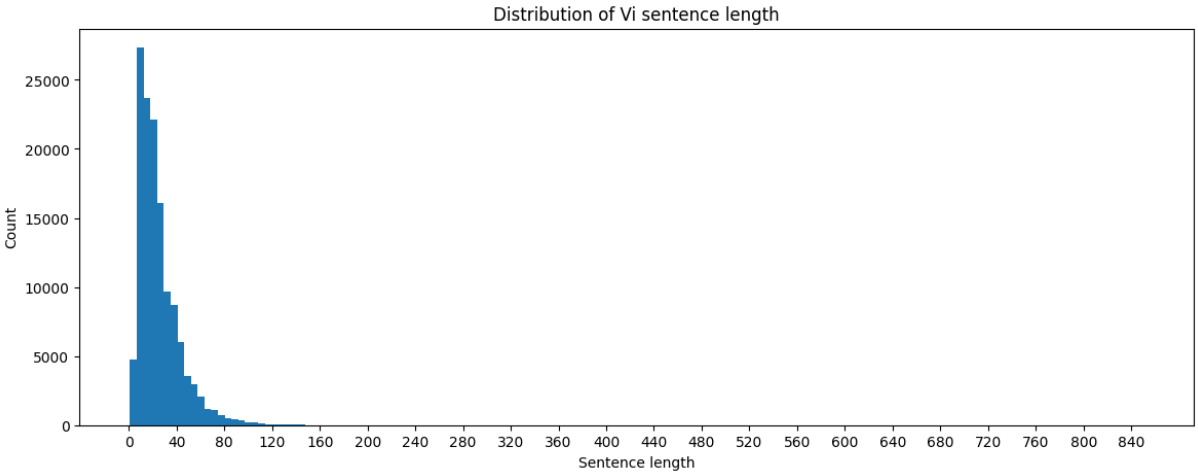
Để hiểu rõ mối tương quan về độ dài giữa hai ngôn ngữ, chúng tôi đã tiến hành phân tích tỉ lệ độ dài câu. Tỉ lệ này được tính toán cho từng cặp câu trong tập huấn luyện theo công thức:

$$R = \frac{L_{vi}}{L_{en}}$$

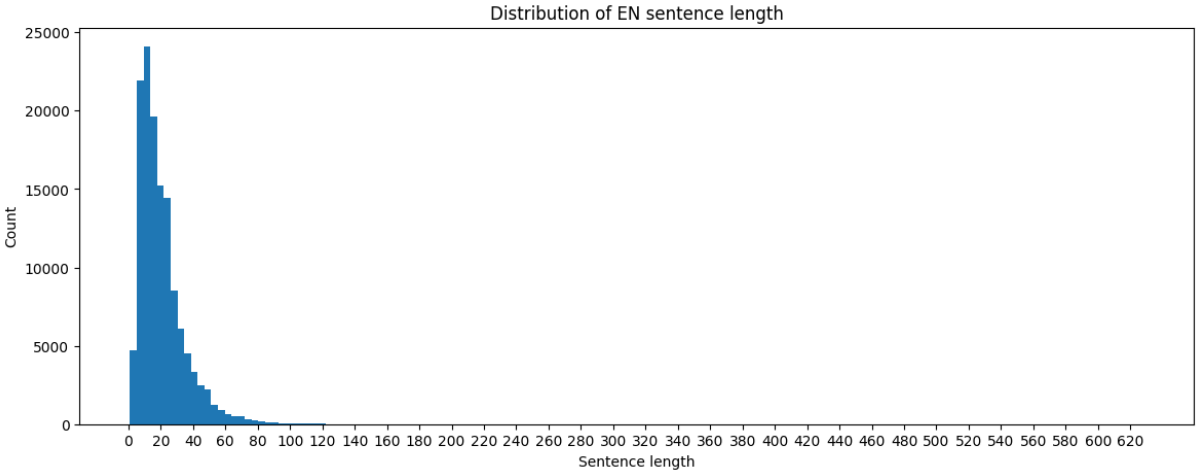
Trong đó L_{vi} và L_{en} lần lượt là số lượng từ trong câu tiếng Việt và câu tiếng Anh tương ứng.

Thống kê mô tả: Kết quả phân tích trên 132,146 cặp câu cho thấy các chỉ số thống kê như sau:

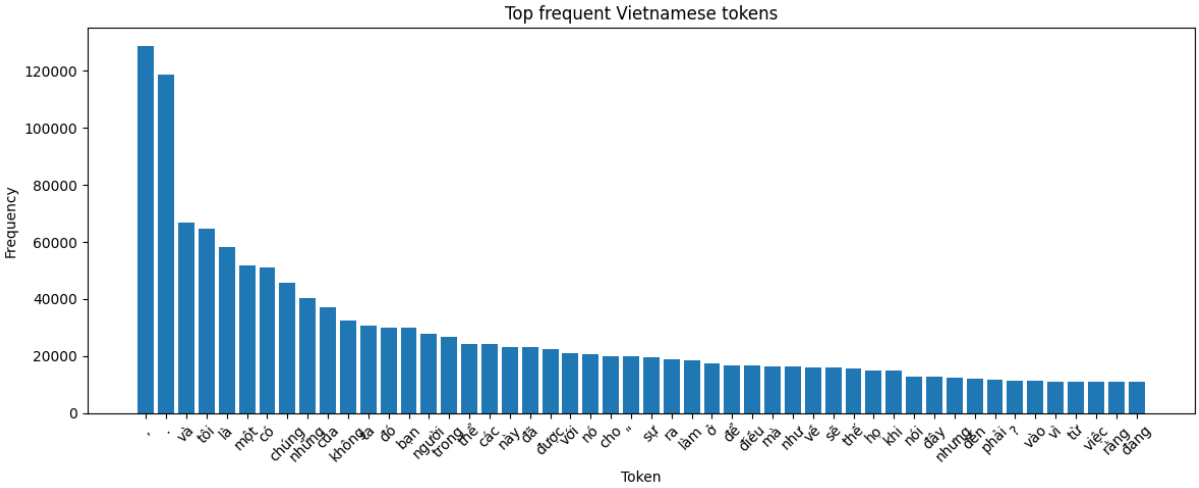
- **Tỉ lệ trung bình:** 1.24. Điều này chỉ ra rằng, trung bình một câu tiếng Việt sẽ dài hơn khoảng 24% so với câu tiếng Anh tương ứng về số lượng từ.
- **Độ lệch chuẩn:** 0.26, cho thấy sự biến động của tỉ lệ này quanh giá trị trung bình là không quá lớn.
- **Trung vị:** 1.22.



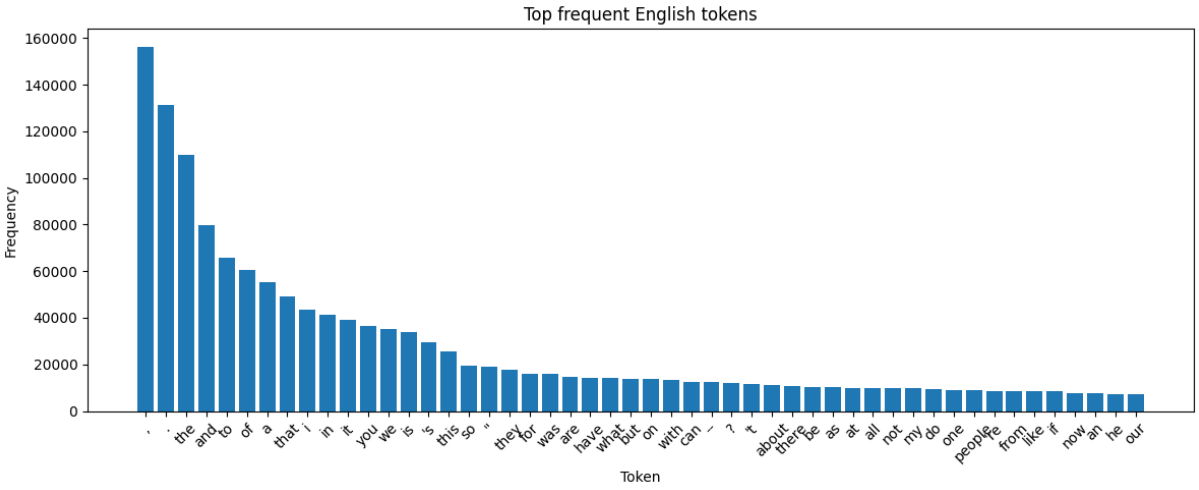
Hình 1: Phân phối của độ dài câu tiếng Việt.



Hình 2: Phân phối của độ dài câu tiếng Anh.



Hình 3: Top 50 từ tiếng Việt có tần suất xuất hiện nhiều nhất.



Hình 4: Top 50 từ tiếng Anh có tần suất xuất hiện nhiều nhất.

- **Phân vị 50%:** 50% số lượng cặp câu có tỉ lệ độ dài nằm trong khoảng từ 1.07 đến 1.38.

- **Giá trị cực đại:** 6.0. Tồn tại một số trường hợp ngoại lai khi câu tiếng Việt dài gấp 6 lần câu tiếng Anh, có thể do lỗi dữ liệu hoặc cách dịch giải thích dài dòng.

Nhận xét và Ý nghĩa: Biểu đồ phân phối (tham khảo Hình 5) của tỉ lệ độ dài câu cho thấy dạng phân phối chuẩn, tập trung chủ yếu quanh giá trị 1.2 - 1.3. Việc câu tiếng Việt có xu hướng dài hơn tiếng Anh là đặc điểm quan trọng cần lưu ý khi xây dựng mô hình dịch máy.

Kết luận: Bước phân tích mức từ cho thấy dữ liệu sạch, có độ bao phủ từ vựng tốt, nhưng tồn tại sự chênh lệch lớn về độ dài câu, đòi hỏi chiến lược cắt tỉa hoặc padding hợp lý trong giai đoạn sau.

3 Phân tích và tiền xử lý tập dữ liệu VLSP (Medical Domain)

Khác với bài toán chính sử dụng bộ dữ liệu IWSLT vốn mang đặc trưng văn phong giao tiếp và ngôn ngữ phổ thông, bài toán phụ tập trung vào việc xử lý dữ liệu thuộc chuyên ngành y tế trong khuôn khổ VLSP 2025 Shared Task. Đặc thù của miền dữ liệu này thể hiện ở hệ thống thuật ngữ chuyên môn dày đặc, cấu trúc câu phức tạp và mức độ biến thiên từ vựng cao. Do đó, chúng tôi tiến hành phân tích khám phá dữ liệu một cách chi tiết nhằm làm rõ các đặc trưng ngôn ngữ quan trọng, từ đó làm cơ sở cho việc lựa chọn chiến lược tiền xử lý và huấn luyện mô hình phù hợp.

3.1 Tổng quan bộ dữ liệu

Bộ dữ liệu được cung cấp bao gồm 358,796 cặp câu, được chia thành ba tập: Train, Validation và Test. Cụ thể:

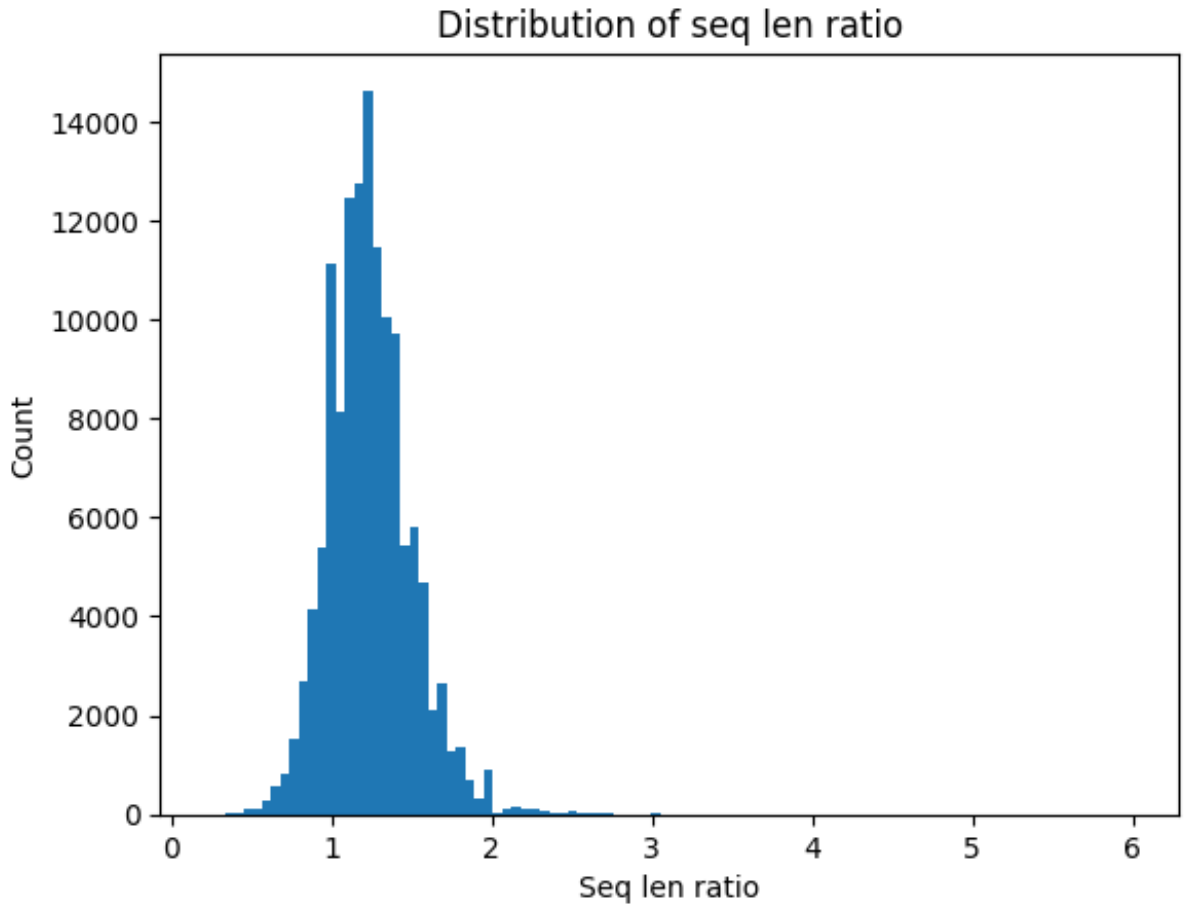
- **Tập Train:** 340,897 cặp câu (chiếm ∼95%).
- **Tập Validation:** 8,939 cặp câu (∼2.5%).
- **Tập Test:** 8,960 cặp câu (∼2.5%).

Quy mô dữ liệu này lớn hơn đáng kể so với IWSLT, qua đó đảm bảo số lượng mẫu đủ lớn để mô hình có thể học được các cấu trúc ngữ pháp phức tạp và các mẫu ngôn ngữ đặc thù của miền y tế. Tuy nhiên, kích thước dữ liệu lớn cũng đồng thời đặt ra những thách thức không nhỏ về tài nguyên tính toán, đặc biệt là yêu cầu bộ nhớ và thời gian huấn luyện khi triển khai các mô hình học sâu có số lượng tham số lớn.

3.2 Phân tích đặc trưng phân bố độ dài

Độ dài câu là một yếu tố then chốt trong việc xác định tham số $\mathrm{max\_length}$, bởi nó ảnh hưởng trực tiếp đến khả năng bao phủ ngữ cảnh của mô hình cũng như chi phí tính toán trong quá trình huấn luyện. Việc lựa chọn giá trị $\mathrm{max\_length}$ không phù hợp có thể dẫn đến hiện tượng cắt bỏ thông tin quan trọng hoặc gia tăng đáng kể lượng padding không cần thiết. Dựa trên biểu đồ phân bố độ dài câu (Hình 6) kết hợp với bảng thống kê chi tiết, chúng tôi tiến hành phân tích định lượng để đưa ra quyết định cân bằng giữa hiệu quả mô hình và tài nguyên tính toán.

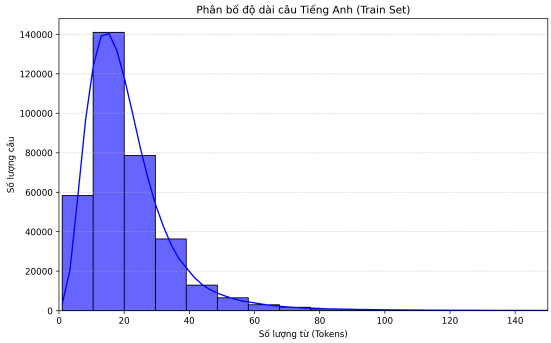
- **Sự chênh lệch độ dài:** Độ dài trung bình của câu tiếng Việt (30.20) cao gấp 1.4 lần so với



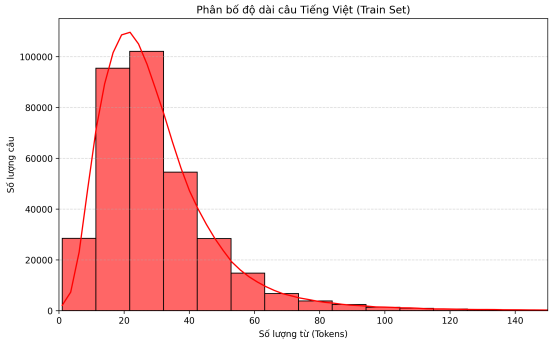
Hình 5: Phân phối tỉ lệ độ dài câu tiếng Việt và tiếng Anh.

Bảng 1: Thống kê độ dài câu (token) trên tập Train

Chỉ số	Tiếng Anh	Tiếng Việt
Trung bình	21.30	30.20
Độ lệch chuẩn	14.34	19.75
Trung vị	18	26
Phân vị 95%	46	64
Độ dài lớn nhất	477	519



(a) Phân bố độ dài câu Tiếng Anh

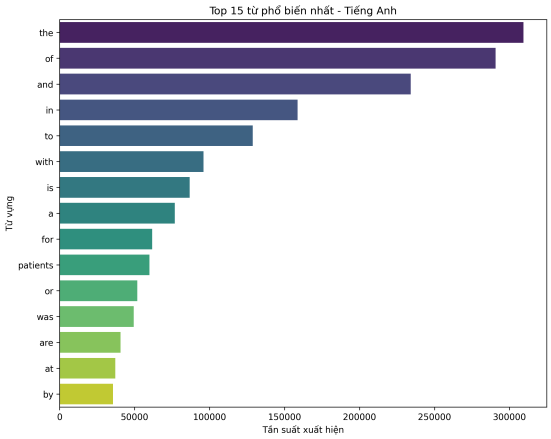


(b) Phân bố độ dài câu Tiếng Việt

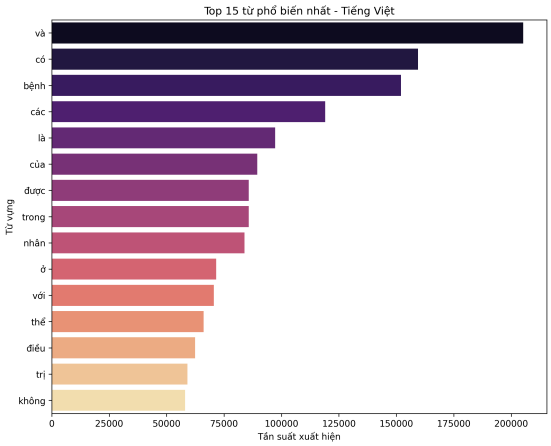
Hình 6: Biểu đồ phân bố độ dài câu trên tập Train. Dữ liệu tập trung ở khoảng ngắn nhưng tồn tại đuôi dài (long-tail).

tiếng Anh (21.30). Điều này phản ánh đặc thù của tiếng Việt khi một từ tiếng Anh (ví dụ: "patients") thường được dịch thành từ ghép ("bệnh nhân").

- **Phân bố lệch phải:** Biểu đồ cho thấy đa số các câu tập trung ở độ dài ngắn (10-30 từ).
- **Quyết định cắt tỉa:** Mặc dù độ dài lớn nhất lên tới 519 từ, nhưng 95% dữ liệu nằm dưới ngưỡng 64 từ. Để tối ưu hóa bộ nhớ GPU và tránh padding lãng phí, chúng tôi thiết lập $\max\_len = 128$. Ngưỡng này là "ngưỡng an toàn", bao phủ trọn vẹn hơn 99% tập dữ liệu.



(a) Top 15 từ tiếng Anh



(b) Top 15 từ tiếng Việt

Hình 7: Các từ xuất hiện nhiều nhất khẳng định đặc thù dữ liệu Y tế.

3.3 Phân tích từ vựng và Xác định miền dữ liệu

Để chứng minh tính chất "Medical Domain" của dữ liệu, chúng tôi phân tích tần suất xuất hiện của các từ vựng (Hình 7).

- **Tiếng Anh:** Trong Top 15 từ phổ biến nhất, bên cạnh các stopwords như "the", "of", "and", từ **"patients"** xuất hiện ở vị trí thứ 10. Đây là dấu hiệu rõ ràng của văn bản y khoa lâm sàng.
- **Tiếng Việt:** Danh sách từ phổ biến chứa nhiều từ tố tạo nên thuật ngữ y tế như: **"bệnh"**, **"nhân"** (trong bệnh nhân), **"điều"**, **"trị"** (trong điều trị), **"thuốc"**.

3.4 Chiến lược Tiền xử lý

Để chuẩn bị dữ liệu đầu vào cho mô hình dịch máy, chúng tôi triển khai quy trình tiền xử lý văn bản dựa trên đặc thù ngữ pháp của từng ngôn ngữ, thay

vì sử dụng các kỹ thuật tách từ con như BPE. Quy trình cụ thể được thực hiện như sau:

3.4.1 Chuẩn hóa và Tách từ (Tokenization)

Quy trình tách từ được thiết kế riêng biệt cho hai ngôn ngữ nhằm đảm bảo tính chính xác về mặt ngữ nghĩa:

- **Tiếng Anh:** Sử dụng thư viện NLTK (*nltk.word_tokenize*) để tách từ và dấu câu chuẩn. Toàn bộ văn bản được chuyển về dạng chữ thường để giảm chiều dữ liệu.
- **Tiếng Việt:** Sử dụng thư viện PyVi (*ViTokenizer*) để xử lý đặc trưng từ ghép của tiếng Việt. Thay vì tách rời từng âm tiết (ví dụ: "bệnh", "nhân"), PyVi giúp gom nhóm các từ ghép lại với nhau bằng dấu gạch dưới (ví dụ: "bệnh_nhân", "điều_trị"), giúp mô hình học được ngữ nghĩa tốt hơn trong ngữ cảnh y tế.

3.4.2 Xây dựng bộ từ điển

Thay vì sử dụng từ điển chung , chúng tôi xây dựng hai bộ từ điển riêng biệt cho tiếng Anh và tiếng Việt dựa trên tập dữ liệu huấn luyện. Chiến lược xử lý từ vựng bao gồm:

1. **Lọc bỏ từ hiếm:** Để giảm nhiễu và hạn chế kích thước từ điển bùng nổ, chúng tôi chỉ giữ lại các từ xuất hiện tối thiểu 2 lần ($\mathrm{min\_freq}=2$) trong tập huấn luyện.
2. **Xử lý từ chưa biết:** Tất cả các từ có tần suất xuất hiện thấp (< 2) hoặc các từ mới xuất hiện trong tập Validation/Test sẽ được quy về token đặc biệt $<\mathrm{unk}>$ (Unknown).
3. **Token đặc biệt:** Bộ từ điển được bổ sung 4 token kỹ thuật bắt buộc cho quá trình huấn luyện Sequence-to-Sequence:
   - $<\mathrm{unk}>$ (Index 0): Đại diện cho từ không có trong từ điển.
   - $<\mathrm{pad}>$ (Index 1): Dùng để đệm (padding) cho các câu có độ dài không đồng nhất.
   - $<\mathrm{s}>$ (Index 2): Đánh dấu bắt đầu câu (Start of Sentence).
   - $</\mathrm{s}>$ (Index 3): Đánh dấu kết thúc câu (End of Sentence).

3.4.3 Kết quả tiền xử lý

Sau quá trình làm sạch và chuẩn hóa, chúng tôi thu được bộ dữ liệu đã được tiền xử lý và các tệp cấu hình từ điển tương ứng. Kết quả cụ thể như sau:

a. Bộ từ điển: Hai bộ từ điển được xây dựng từ tập huấn luyện sau khi loại bỏ các từ xuất hiện ít hơn 2 lần . Các tệp JSON ánh xạ từ vựng sang chỉ số (token-to-id) bao gồm:

- **Tiếng Anh (**$\mathrm{en\_token2id.json}$**):** Kích thước từ điển đạt **69,025** từ.
- **Tiếng Việt (**$\mathrm{vi\_token2id.json}$**):** Kích thước từ điển đạt **45,864** từ.
- **Token đặc biệt:** Cả hai từ điển đều tích hợp sẵn 4 token điều khiển với định danh cố định: $<\mathrm{unk}>$:0, $<\mathrm{pad}>$:1, $<\mathrm{s}>$:2, $</\mathrm{s}>$:3.

b. Dữ liệu văn bản đã tách từ: Dữ liệu huấn luyện, kiểm thử và thẩm định được lưu trữ dưới dạng tệp CSV ($\mathrm{train.csv}$, $\mathrm{val.csv}$, $\mathrm{test.csv}$). Dữ liệu trong các tệp này đã hoàn tất bước tách từ:

- **Định dạng:** Văn bản được chuẩn hóa về chữ thường.
- **Xử lý từ ghép:** Đối với tiếng Việt, các từ ghép đã được gom nhóm tự động bằng ký tự gạch dưới (underscore) nhờ thư viện PyVi.
- **Minh họa thực tế:**
  - *Gốc:* "nghiên cứu đặc điểm lâm sàng"
  - *Sau xử lý:* "$\mathrm{nghiên\_cứu\ đặc\_điểm\ lâm\_sàng}$"

Việc lưu trữ dưới dạng token text giúp dễ dàng kiểm tra trực quan trước khi đưa vào mô hình. Quá trình ánh xạ từ văn bản sang chuỗi số nguyên sẽ được thực hiện dynamic trong quá trình tải dữ liệu dựa trên hai tệp từ điển JSON đã tạo.

4 Kiến trúc mô hình

Kiến trúc Transformer là mô hình đột phá trong xử lý ngôn ngữ tự nhiên, trở thành nền tảng cho BERT, GPT và nhiều mô hình hiện đại khác. Khác biệt chính so với RNNs/LSTMs là Transformer loại bỏ hoàn toàn cơ chế tuần tự và tích chập, thay vào đó dựa vào self-attention để nắm bắt các mối quan hệ toàn cục trong chuỗi.

4.1 Encoder-Decoder Architecture

4.1.1 Encoder

Encoder chuyển đổi chuỗi đầu vào thành biểu diễn ngữ cảnh liên tục thông qua stack N lớp giống hệt nhau. Mỗi lớp bao gồm:

1. **Multi-Head Self-Attention**: Tính toán mối quan hệ giữa tất cả các token trong chuỗi, cho phép mỗi token "chú ý" đến toàn bộ ngữ cảnh
2. **Position-wise FFN**: Áp dụng phép biến đổi phi tuyến độc lập cho từng vị trí

Mỗi sublayer được bao bọc bởi residual connection và layer normalization. Đầu ra cuối cùng là tập hợp các vector biểu diễn ngữ cảnh phong phú.

4.1.2 Decoder

Decoder sinh chuỗi đầu ra từ biểu diễn của Encoder thông qua stack N lớp, mỗi lớp gồm:

1. **Masked Self-Attention**: Ngăn các vị trí tương lai ảnh hưởng đến dự đoán hiện tại, đảm bảo tính tự hồi quy
2. **Cross-Attention**: Query từ Decoder, Key/Value từ Encoder – cho phép tập trung vào thông tin liên quan từ đầu vào
3. **Position-wise FFN**: Tương tự Encoder

Đầu ra được chiếu qua lớp linear và softmax để tạo phân phối xác suất cho token tiếp theo.

4.2 Self-Attention Mechanism

4.2.1 Scaled Dot-Product Attention

Attention tính toán biểu diễn có trọng số bằng cách tổng hợp thông tin từ toàn bộ chuỗi:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

Quy trình: (i) tính similarity scores qua dot product, (ii) scaling với $\sqrt{d_k}$ để ổn định gradient, (iii) softmax tạo attention weights, (iv) weighted sum với Values.

4.2.2 Multi-Head Attention

Chia Q, K, V thành h heads nhỏ hơn, mỗi head học các mối quan hệ khác nhau trong các không gian biểu diễn riêng. Kết quả được concat và chiếu qua linear layer.

Lợi ích: Mô hình học được đa dạng patterns attention đồng thời, tăng khả năng biểu diễn.

4.2.3 Grouped Query Attention (GQA) – Cải tiến

Vấn đề. Trong cơ chế Multi-Head Attention (MHA) truyền thống, mỗi query head đi kèm một cặp key/value (KV) head tương ứng. Thiết kế này làm gia tăng đáng kể chi phí bộ nhớ và băng thông truy cập, đặc biệt trong giai đoạn inference của các mô hình lớn và trong thiết lập (autoregressive generation.

Giải pháp. Grouped-Query Attention (GQA) đề xuất nhóm nhiều query heads để chia sẻ một tập nhỏ các KV heads, qua đó cân bằng giữa hiệu quả tính toán và chất lượng mô hình:

- **MHA**: n_{heads} query heads, n_{heads} KV heads.
- **MQA**: n_{heads} query heads, 1 KV head.
- **GQA**: n_{heads} query heads, n_{kv} KV heads, với $1 < n_{\text{kv}} < n_{\text{heads}}$.

Ưu điểm.

1. **Memory efficiency**: Giảm đáng kể số tham số và bộ nhớ dành cho KV (lên tới $\sim$75% so với MHA).
2. **Inference speed**: Giảm áp lực băng thông bộ nhớ, mang lại cải thiện rõ rệt về độ trễ trong suy luận, đặc biệt đối với sinh chuỗi dài.
3. **Quality–efficiency trade-off**: Đạt chất lượng gần tương đương MHA trong khi nhanh hơn đáng kể, đồng thời vượt trội hơn MQA về hiệu năng mô hình.

Cài đặt. Trong thiết lập của chúng tôi, 8 query heads chia sẻ 2 KV heads, tương ứng với tỷ lệ nhóm 4:1, nhằm đạt được sự cân bằng thực nghiệm tối ưu giữa tốc độ suy luận và chất lượng đầu ra.

4.3 Positional Encoding

Do attention xử lý song song, cần tiêm thông tin vị trí vào embeddings:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d}}\right) \quad (2)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d}}\right) \quad (3)$$

Lợi ích: (i) ngoại suy cho chuỗi dài hơn, (ii) mã hóa vị trí tương đối, (iii) không tăng tham số, (iv) deterministic.

4.4 Feed-Forward Networks

FFN áp dụng biến đổi phi tuyến độc lập cho từng vị trí:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (4)$$

với $d_{ff} = 4 \times d_{\text{model}}$ trong Transformer gốc.

4.4.1 SwiGLU Activation – Cải tiến

Động lực: ReLU/GELU activation đơn giản nhưng chưa tối ưu cho LLMs.

Giải pháp: SwiGLU sử dụng gating mechanism:

$$\text{SwiGLU}(x) = (xW_1) \otimes \text{SiLU}(xW_2) \quad (5)$$

với $\text{SiLU}(x) = x \cdot \sigma(x) = \frac{x}{1+e^{-x}}$

Cơ chế:

- Nhánh 1 (xW_1): content stream
- Nhánh 2 ($\text{SiLU}(xW_2)$): gate stream kiểm soát information flow
- Element-wise product kết hợp hai nhánh

Ưu điểm: Empirically tốt hơn ReLU/GELU trong LLaMA, PaLM. Trade-off: tăng 33% tham số FFN nhưng cải thiện quality đáng kể.

Triển khai này sử dụng $d_{ff} = \frac{8d}{3}$ để cân bằng giữa capacity và efficiency.

4.5 Normalization và Residual Connections

4.5.1 Residual Connections

Skip connection cho phép gradient flow trực tiếp:

$$\text{Output} = \text{Input} + \text{Sublayer}(\text{Input}) \quad (6)$$

Lợi ích: Giải quyết vanishing gradients, sublayer chỉ học residual, bảo toàn thông tin.

4.5.2 RMSNorm – Cải tiến

Động lực: LayerNorm chuẩn hóa cả mean và variance, tốn $\sim 4d$ operations:

$$\text{LayerNorm}(x) = \gamma \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \quad (7)$$

Giải pháp: RMSNorm loại bỏ mean centering, chỉ chuẩn hóa magnitude:

$$\text{RMSNorm}(x) = \gamma \frac{x}{\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2 + \epsilon}} \quad (8)$$

Ưu điểm:

1. **Efficiency**: Chỉ $3d$ ops/token, giảm 25% tính toán
2. **Quality**: Hiệu suất $\geq$ LayerNorm (LLaMA, T5)
3. **Lý thuyết**: Residual connections cung cấp implicit centering; scale quan trọng hơn shift cho stability

4.5.3 Pre-Layer Normalization – Cải tiến

Post-LN (Transformer gốc):

$$x_{l+1} = \text{Norm}(x_l + \text{Sublayer}(x_l)) \quad (9)$$

Vấn đề: Gradient flow không ổn định, cần warm-up dài.

Pre-LN (Triển khai này):

$$x_{l+1} = x_l + \text{Sublayer}(\text{Norm}(x_l)) \quad (10)$$

Ưu điểm:

- Gradient truyền trực tiếp qua residual path
- Không cần learning rate warm-up
- Ổn định hơn với mạng sâu

5 Các Cải tiến So với Mô hình Gốc

Chúng tôi tiến hành đánh giá từng cải tiến kiến trúc một cách độc lập nhằm phân tích ảnh hưởng của chúng tới chất lượng dịch và hiệu quả huấn luyện. Mô hình baseline sử dụng kiến trúc Transformer gốc với Post-Layer Normalization, LayerNorm, Multi-Head Attention (MHA) và ReLU trong Feed-Forward Network. Các biến thể còn lại lần lượt tích hợp từng cải tiến đã được mô tả trong phần trước .

5.1 Pre-Layer Normalization

Thay thế Post-LN bằng Pre-LN giúp cải thiện đáng kể độ ổn định trong quá trình huấn luyện. Như thể hiện trong Bảng 2, Pre-LN mang lại mức tăng lớn về BLEU trong decoding greedy (từ 0.1529 lên 0.2200), trong khi gần như không ảnh hưởng tới thời gian huấn luyện mỗi epoch. Kết quả này cho thấy Pre-LN đặc biệt hiệu quả trong việc cải thiện khả năng tối ưu hóa mô hình.

5.2 RMSNorm

Việc thay thế LayerNorm bằng RMSNorm giúp giảm chi phí tính toán mà vẫn duy trì chất lượng mô hình. RMSNorm đạt thời gian huấn luyện nhanh nhất trong tất cả các cấu hình (4m10s/epoch), đồng thời cải thiện nhẹ BLEU so với Pre-LN. Điều này xác nhận RMSNorm là một lựa chọn hiệu quả về mặt tính toán cho các mô hình Transformer lớn.

5.3 Grouped Query Attention (GQA)

Chúng tôi áp dụng Grouped Query Attention với cấu hình 8 query heads và 2 key/value heads (tỷ lệ 4:1). GQA giúp giảm đáng kể chi phí bộ nhớ

liên quan đến KV cache trong quá trình suy luận. Như kết quả thực nghiệm cho thấy, GQA duy trì chất lượng gần tương đương MHA (BLEU beam giảm không đáng kể từ 0.2509 xuống 0.2498), trong khi rút ngắn thời gian huấn luyện xuống còn 4m15s/epoch. Điều này cho thấy GQA đạt được trade-off hợp lý giữa hiệu suất và độ chính xác.

5.4 SwiGLU Activation

Thay thế ReLU bằng SwiGLU trong Feed-Forward Network mang lại cải thiện nhất quán về chất lượng dịch. SwiGLU đạt BLEU cao hơn so với các cấu hình trước đó (0.2531 với beam search), với chi phí tính toán tăng không đáng kể. Kết quả này phù hợp với các quan sát trước đó trên các mô hình ngôn ngữ lớn như LLaMA và PaLM.

5.5 Label Smoothing

Cuối cùng, Label Smoothing được áp dụng như một kỹ thuật regularization trong huấn luyện. Đây là cải tiến mang lại hiệu quả cao nhất về chất lượng, đạt BLEU 0.2580 với beam search. Mặc dù không cải thiện tốc độ huấn luyện, Label Smoothing giúp mô hình tổng quát hóa tốt hơn và giảm hiện tượng over-confidence trong phân phối xác suất đầu ra.

5.6 Tổng hợp kết quả

Bảng 2 tổng hợp kết quả của tất cả các cải tiến, hình 8 và hình 9 so sánh quá trình huấn luyện của từng cải tiến. Nhìn chung, các cải tiến kiến trúc (Pre-LN, RMSNorm, GQA, SwiGLU) chủ yếu cải thiện tính ổn định và hiệu quả tính toán, trong khi Label Smoothing đóng vai trò then chốt trong việc nâng cao chất lượng đầu ra. Sự kết hợp các kỹ thuật này tạo nên một kiến trúc Transformer tối ưu hơn so với mô hình gốc cả về hiệu suất lẫn độ chính xác.

Bảng 2: So sánh hiệu suất các phiên bản mô hình

Phiên bản	**BLEU (Greedy)**	**BLEU (Beam)**	**Thời gian/ Epoch**
Baseline	0.1529	0.2506	4m40s
Pre-LN	0.2200	0.2507	4m30s
RMSNorm	0.2217	0.2509	4m10s
GQA	0.2188	0.2498	4m15s
SwiGLU	0.2252	0.2531	4m20s
Label Smooth	0.2311	0.2580	4m30s

Bảng 3: So sánh các cải tiến so với Transformer gốc

Thành phần	**Transformer gốc**	**Mô hình đề xuất**
Attention	MHA (8 heads)	GQA (8 Q / 2 KV)
Normalization	LayerNorm	RMSNorm
Vị trí chuẩn hóa	Post-LN	Pre-LN
FFN activation	ReLU	SwiGLU
FFN dimension	$4d$	$\frac{8d}{3}$
Optimizer	Adam	AdamW (8-bit)
Learning rate	Noam scheduler	OneCycleLR
Decoding	Greedy	Beam Search
Evaluation	BLEU	BLEU + LLM-Judge

6 So sánh với Baseline

6.1 Phân tích theo thành phần

Encoder Layer (1 layer = 951K tham số):

- RMSNorm $\times 2$: 1.0K
- GQA attention: 163.8K
- SwiGLU FFN: 786.4K

Decoder Layer (1 layer = 1.11M tham số):

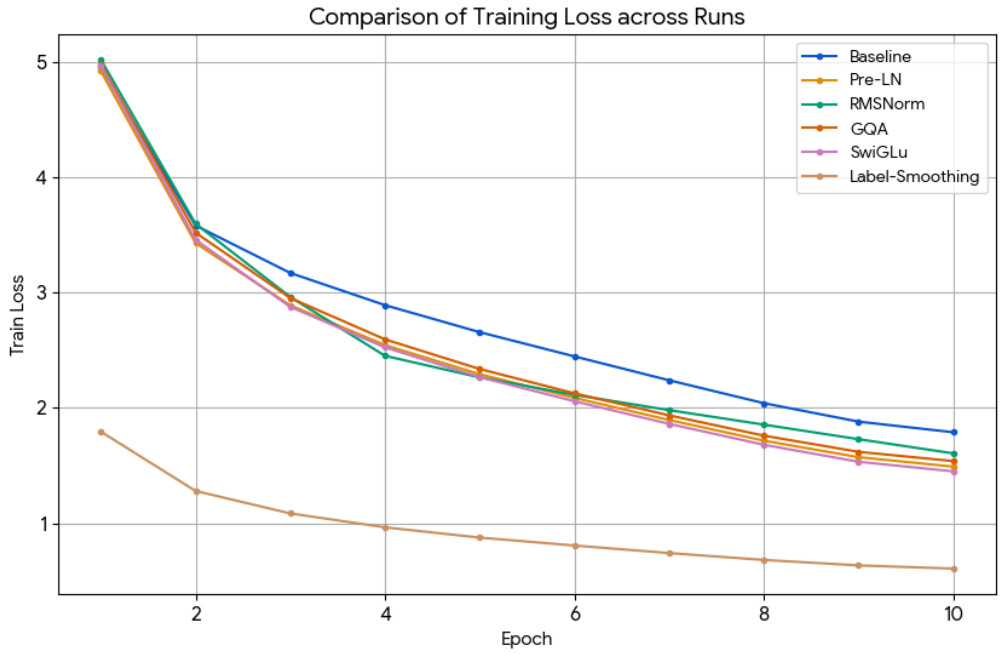
- RMSNorm $\times 3$: 1.5K
- Self-attention GQA: 163.8K
- Cross-attention GQA: 163.8K
- SwiGLU FFN: 786.4K

Bảng 4: Phân bố tham số theo thành phần

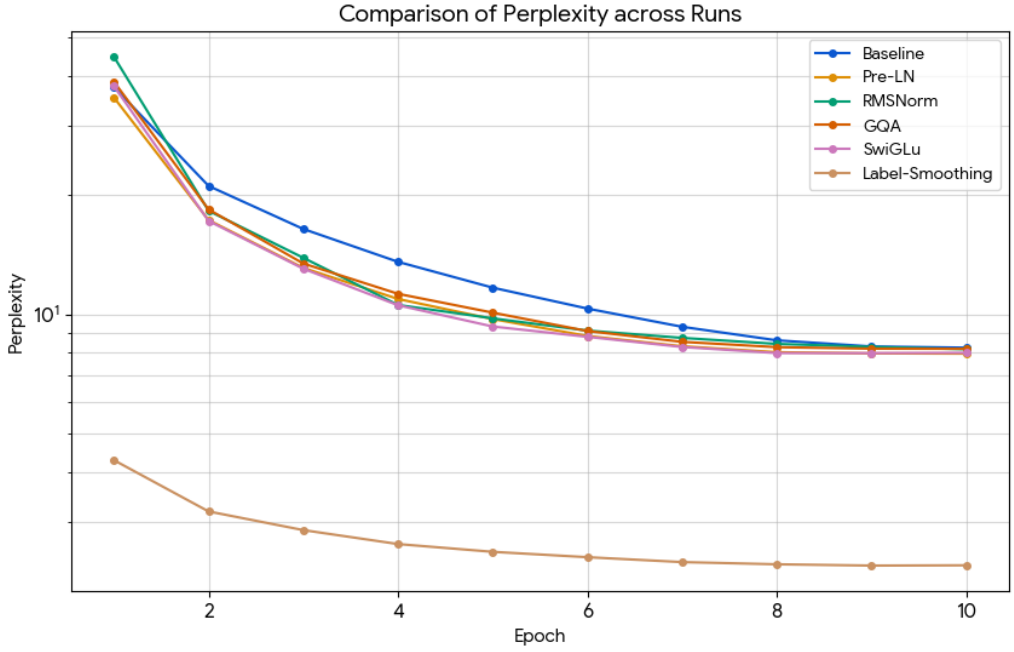
Thành phần	**Tham số**
Source embeddings	1.66M
Target embeddings	4.02M
Encoder (3 layers)	2.85M
Decoder (3 layers)	3.35M
Generator	4.04M
Tổng	**15.93M**

Bảng 5: So sánh cấu hình với Transformer gốc

Cấu hình	**Gốc**	**Ours**
d_{model}	512	256
Layers	6	3
d_{ff}	2048	1024
Attention	MHA	GQA
Tổng tham số	**$\sim$65M**	**$\sim$16M**



Hình 8: Train loss của từng quá trình cải tiến



Hình 9: Perplexity của từng quá trình cải tiến

7 Kết quả và Phân tích

7.1 Thiết lập bài toán

Chúng tôi đánh giá mô hình trên tập IWSLT'15 English–Vietnamese, so sánh hai chiến lược giải mã phổ biến: Greedy Search và Beam Search. Phân tích được thực hiện ở cả hai góc độ định lượng (BLEU) và định tính (phân tích mẫu đầu ra).

7.2 Phân tích định lượng

Với Beam Search, mô hình đạt BLEU 0.2580, cải thiện đáng kể so với Greedy Search (0.2311). Mức cải thiện này phản ánh khả năng của Beam Search trong việc mở rộng không gian tìm kiếm, từ đó tăng mức độ trùng khớp n-gram với bản dịch tham chiếu. Tuy nhiên, giá trị BLEU vẫn dưới 0.3, cho thấy chất lượng dịch còn khoảng cách lớn so với bản dịch do con người thực hiện, đặc biệt ở các câu có cấu trúc ngữ nghĩa và suy luận phức tạp.

7.3 Phân tích định tính

7.3.1 Khôi phục ngữ nghĩa ở câu đơn giản

Ở các câu ngắn và cấu trúc đơn giản, mô hình tạo ra các bản dịch tương đối chính xác. Ví dụ, câu "tôi bước vào xe , trong lòng cảm thấy rất rất choáng ngợp" được dịch thành "i walked into my car , and i felt very overwhelmed", bảo toàn đầy đủ nội dung và sắc thái cảm xúc so với bản tham chiếu. Điều này cho thấy Encoder đã học được các ánh xạ ngữ nghĩa cơ bản tương đối tốt.

7.3.2 Hạn chế về lựa chọn từ vựng

Một số lỗi dịch cho thấy hạn chế trong vốn từ và khả năng phân biệt nghĩa. Trong ví dụ "họ bán các chứng chỉ bảo mật", mô hình sinh ra "security evidence" thay vì "certificates". Đây là lỗi thay thế từ có embedding gần nhau nhưng khác ngữ nghĩa, phản ánh hạn chế của biểu diễn từ vựng và thiếu thông tin ngữ cảnh đủ mạnh để phân biệt các khái niệm chuyên biệt.

7.3.3 Thất bại trong suy luận số và logic

Các câu chứa cấu trúc so sánh và dữ liệu số phức tạp là điểm yếu rõ rệt của hệ thống. Với câu nói về án tử hình, mô hình sinh ra chuỗi trôi chảy về hình thức nhưng sai lệch nghiêm trọng về mặt logic và số lượng, ví dụ "22 times as black or six times as the black". Điều này cho thấy Decoder thiếu khả năng duy trì tính nhất quán toàn cục khi phải kết hợp nhiều mệnh đề và con số phụ thuộc lẫn nhau.

7.3.4 Ảnh hưởng của chiến lược giải mã

Greedy Search thường bị mắc kẹt trong các chuỗi con kém tối ưu hoặc lặp token, trong khi Beam Search tạo ra văn bản mạch lạc hơn nhưng có xu hướng "tự tin sai". Beam Search ưu tiên các chuỗi có xác suất cao cục bộ, dẫn đến việc khuếch đại các lỗi hệ thống của mô hình, đặc biệt trong các câu giàu thông tin. Hiện tượng này minh họa rõ ràng cho vấn đề *hallucination* trong các mô hình sinh chuỗi.

7.3.5 Tác động của nhiễu và từ hiếm

Sự xuất hiện của token $<\mathrm{unk}>$ trong kết quả Beam Search cho thấy vấn đề không nằm ở thuật toán giải mã mà ở Encoder và tập từ vựng. Các khái niệm hiếm hoặc ngoài miền huấn luyện không được mã hóa hiệu quả, dẫn đến mất mát thông tin không thể khắc phục ở giai đoạn suy luận.

7.4 Đánh giá tổng hợp

Tổng thể, Beam Search cải thiện rõ rệt độ trôi chảy và tính tự nhiên của bản dịch so với Greedy Search, nhưng đồng thời bộc lộ hạn chế về tính trung thực thông tin, đặc biệt đối với các câu yêu cầu suy luận logic và xử lý số lượng. Mô hình có xu hướng tạo ra các bản dịch trông hợp lý về hình thức nhưng chứa thông tin sai lệch, cho thấy nhu cầu cấp thiết về các cơ chế kiểm soát nhất quán và biểu diễn ngữ nghĩa mạnh hơn trong các nghiên cứu tiếp theo.

8 Áp dụng mô hình đã xây cho bài toán VLSP

Sau khi hoàn thiện và kiểm chứng kiến trúc Transformer trên bài toán cơ bản, chúng tôi tiến hành áp dụng mô hình này vào bài toán thách thức hơn: **Dịch máy lĩnh vực Y tế (VLSP 2025 Shared Task)**. Đây là bài toán có tính chuyên biệt cao với nhiều thuật ngữ phức tạp. Dưới đây là chi tiết quá trình tinh chỉnh siêu tham số và chiến lược huấn luyện.

8.1 Chiến lược Tinh chỉnh Siêu tham số

Quá trình chuyển đổi từ bài toán cơ bản sang bài toán chuyên biệt (Medical Domain) đòi hỏi nhiều nỗ lực thực nghiệm. Ngoài việc thay đổi cấu trúc dữ liệu, chúng tôi đã thực hiện một loạt các thí nghiệm so sánh để tìm ra điểm cân bằng giữa hiệu suất và tài nguyên. Chiến lược cuối cùng được đúc kết là: **"Mở rộng Bề rộng - Tối ưu Độ sâu - Ổn định Hội tụ"**. Dưới đây là chi tiết các thực nghiệm đã thực hiện:

- **Thực nghiệm về Độ sâu mô hình ($N = 3$ vs. $N = 5$):**
  - *Giả thuyết:* Với độ phức tạp ngữ nghĩa cao của văn bản y tế, một mô hình sâu hơn ($N = 5$ layers) sẽ có khả năng trích xuất đặc trưng tốt hơn.
  - *Thực tế:* Kết quả cho thấy mô hình $N = 5$ có hiệu suất **kém hơn** mô hình $N = 3$. Loss trên tập Validation của mô hình $N = 5$ bắt đầu tăng ngược lại sau epoch thứ 4.
  - *Phân tích:* Với lượng dữ liệu hạn chế , mô hình sâu hơn chịu tác động nặng nề của hiện tượng **Overfitting**. Ngoài ra, vấn đề vanishing gradient khiến mô hình khó hội tụ nhanh trong vòng 10 epochs. Chúng tôi quyết định giữ $N = 3$ như một biện pháp Regularization về mặt kiến trúc.
- **Tinh chỉnh tỷ lệ Dropout (0.1 vs. 0.3 vs. 0.15):**
  - *Thí nghiệm:* Để chống lại Overfitting khi dùng mô hình trên tập dữ liệu nhỏ, chúng tôi đã thử tăng Dropout lên 0.3 (so với mặc định 0.1).
  - *Kết quả:* Với Dropout 0.3, mô hình hội tụ quá chậm và rơi vào trạng thái **Underfitting** (Loss giảm rất chậm). Ngược lại, Dropout 0.1 vẫn để lộ dấu hiệu học vẹt nhẹ.
  - *Quyết định:* Chúng tôi thiết lập tỷ lệ Dropout ở mức trung gian 0.15 cho cả FeedForward và Attention. Đây là con số tối ưu giúp mô hình vừa khái quát hóa tốt, vừa không đánh mất quá nhiều thông tin trong quá trình lan truyền xuôi.
- **Kiểm soát phân phối xác suất với Label Smoothing trong không gian từ vựng lớn:**
  - *Thách thức:* Việc mở rộng từ điển lên hàng chục nghìn token (69k/45k) dẫn đến không gian dự đoán rất loãng (sparse). Mô hình có xu hướng bị "Overconfident" vào các từ thông dụng và bỏ qua các thuật ngữ y tế hiếm gặp.
  - *Chiến lược:* Chúng tôi **quyết định duy trì** kỹ thuật **Label Smoothing** ($\epsilon = 0.1$) đã sử dụng ở bài toán cơ sở, nhưng vai trò của nó ở đây trở nên thiết yếu hơn. Kỹ thuật này giúp phân phối lại xác suất, ngăn mô hình gán 100% tin tưởng vào một từ duy nhất, từ đó giúp mô hình học được các sắc thái ngữ nghĩa tốt hơn và cải thiện khả năng tổng quát hóa trên các mẫu câu y tế chưa từng gặp (Unseen medical contexts).
- **Tối ưu phần cứng với AdamW8bit:** Để hiện thực hóa chiến lược "Từ điển lớn" trên GPU T4 giới hạn bộ nhớ, chúng tôi bắt buộc thay thế Adam chuẩn bằng $\mathrm{bitsandbytes.optim.AdamW8bit}$. Kỹ thuật này giảm 75% bộ nhớ optimizer state mà không làm suy giảm độ chính xác hội tụ, cho phép duy trì $\mathrm{Batch\ Size} = 16$ (thay vì phải giảm xuống 8 nếu dùng Adam thường, gây nhiễu gradient).

8.2 Chiến lược Huấn luyện: Phân tích và Lựa chọn

Trong quá trình chuyển giao mô hình từ bài toán dịch phổ quát sang miền dữ liệu y tế chuyên biệt (VLSP), nhóm nghiên cứu đã đối mặt với câu hỏi chiến lược: Liệu nên áp dụng phương pháp *Fine-tuning* hay *Train from Scratch*? Sau khi cân nhắc kỹ lưỡng các yếu tố kỹ thuật và đặc thù dữ liệu, chúng tôi quyết định lựa chọn hướng tiếp cận **Huấn luyện từ đầu**. Quyết định này được củng cố bởi các phân tích chi tiết sau:

1. Sự bất tương thích nghiêm trọng về không gian biểu diễn: Đây là trở ngại kỹ thuật lớn nhất khi chuyển đổi giữa hai bài toán.

- Ở bài toán chính, mô hình được huấn luyện trên bộ từ điển nhỏ gọn (khoảng 10.000 token cho tiếng Anh và 5.000 cho tiếng Việt) phù hợp với văn phong giao tiếp chung.
- Ngược lại, dữ liệu y tế VLSP yêu cầu một không gian từ vựng lớn hơn gấp nhiều lần để bao phủ các thuật ngữ chuyên ngành (khoảng 69.025 token nguồn và 45.864 token đích).
- *Hệ quả:* Lớp Embedding và lớp Generator (Output Projection) chiếm tỷ trọng lớn trong tổng số tham số của mô hình Transformer ($V \times d_{model}$). Việc thay đổi kích thước từ điển buộc chúng tôi phải khởi tạo lại ngẫu nhiên toàn bộ các ma trận trọng số này. Nếu thực hiện Fine-tune, phần lớn mô hình thực chất vẫn phải học lại từ con số 0, làm mất đi ý nghĩa của việc tận dụng trọng số tiền huấn luyện.

2. Khác biệt về phân phối dữ liệu và Nguy cơ "Negative Transfer":

- Dữ liệu huấn luyện ban đầu mang đậm tính chất văn nói, câu ngắn và cấu trúc ngữ pháp linh hoạt. Trong khi đó, văn bản y tế VLSP đòi hỏi sự chính xác tuyệt đối về thuật ngữ, cấu trúc câu phức hợp (câu thụ động, mệnh đề quan hệ lồng nhau) và văn phong khoa học.

- Việc áp dụng trọng số đã học từ văn phong đời thường có thể gây ra hiện tượng *"Negative Transfer"* , nơi mô hình bị nhiễu bởi các thói quen dịch thuật cũ, dẫn đến việc sinh ra các bản dịch nghe có vẻ tự nhiên nhưng sai lệch về sắc thái chuyên môn. Huấn luyện từ đầu giúp cô lập mô hình khỏi các bias không mong muốn này.

3. Tối ưu hóa quỹ đạo hội tụ trên dữ liệu chuyên biệt:

- Với tập dữ liệu VLSP được chuẩn hóa tốt và tập trung cao độ vào một domain, việc huấn luyện từ đầu cho phép các cơ chế Attention tự do thiết lập các mối quan hệ ngữ nghĩa đặc thù của y tế ngay từ những bước đầu tiên.

- Thay vì mất thời gian và tài nguyên tính toán để quên các tri thức cũ không phù hợp, mô hình tập trung toàn bộ năng lực để tối ưu hóa hàm mất mát trên không gian dữ liệu mới, giúp quá trình hội tụ diễn ra sạch và hiệu quả hơn.

Kết quả thực nghiệm: Chiến lược này đã chứng minh hiệu quả rõ rệt. Kết hợp với kỹ thuật **Label Smoothing** ($\epsilon = 0.1$) để kiểm soát sự tự tin thái quá của mô hình trên không gian từ vựng lớn, quá trình huấn luyện đã hội tụ ổn định. Các chỉ số đánh giá chuyên sâu (như BLEU, METEOR, TER và Gemini Score) đạt mức khả quan ở phần kết quả chính là minh chứng rõ ràng nhất, khẳng định rằng mô hình đã hội tụ tốt và nắm bắt được văn phong y tế đặc thù mà không cần phụ thuộc vào tri thức pre-trained .

9 Kết quả Bài toán VLSP

Mô hình Transformer (với tổng số tham số khoảng 47.4 triệu) được huấn luyện trong 10 epochs và đánh giá trên tập kiểm thử của VLSP. Quá trình sinh văn bản sử dụng kỹ thuật **Beam Search** với kích thước $k = 5$.

9.1 Độ đo đánh giá

Để đánh giá toàn diện chất lượng dịch, chúng tôi kết hợp các độ đo truyền thống (BLEU, TER, METEOR) và đánh giá dựa trên mô hình ngôn ngữ lớn (LLM-Judge Evaluation). Các chỉ số được hiện thực hóa thông qua các thư viện mã nguồn mở chuẩn với cấu hình cụ thể như sau:

1. BLEU Score: Được tính toán sử dụng thư viện $\mathrm{torchmetrics}$. Trước khi tính điểm, văn bản đầu ra được sinh ra bằng thuật toán giải mã **Beam Search** với kích thước $k = 5$ (beam width) để tối ưu hóa xác suất chuỗi từ. BLEU đo lường độ trùng khớp n-gram (từ 1 đến 4) giữa bản dịch máy và bản tham chiếu, phản ánh độ chính xác về mặt từ vựng.

2. TER: Sử dụng thư viện $\mathrm{sacrebleu}$. TER đo lường tỷ lệ các thao tác chỉnh sửa cần thiết (Thêm, Xóa, Thay thế, và Dịch chuyển - Shift) để biến câu dự đoán thành câu tham chiếu.

$$TER = \frac{\#edits}{L_{ref}} \times 100 \tag{11}$$

Chỉ số này phản ánh nỗ lực hậu chỉnh sửa của con người; giá trị càng thấp càng tốt.

3. METEOR: Sử dụng thư viện NLTK. Khác với BLEU, METEOR tính điểm dựa trên sự hài hòa giữa độ chính xác (Precision) và độ phủ (Recall), có xét đến từ đồng nghĩa (synonyms) và biến thể từ (stemming). Điểm số được tính dựa trên F-mean có trọng số nghiêng về Recall:

$$F_{mean} = \frac{10PR}{R + 9P} \tag{12}$$

4. Gemini Evaluation: Chúng tôi sử dụng mô hình $\mathrm{gemini\text{-}2.5\text{-}flash\text{-}lite}$ thông qua Google GenAI SDK để đánh giá chất lượng ngữ nghĩa. Hệ thống được thiết lập với vai trò "chuyên gia đánh giá dịch thuật", chấm điểm trên thang **1 - 10** dựa trên prompt cụ thể:

- **Input:** Câu nguồn, Câu máy dịch, Câu tham chiếu.

- **Tiêu chí:** Độ chính xác ý nghĩa, ngữ pháp và độ tự nhiên.

- **Thang điểm:** Từ 1 (Vô nghĩa/Ảo giác) đến 10 (Hoàn hảo/Như người bản xứ).

Điểm số cuối cùng là trung bình cộng của các đánh giá trên toàn bộ tập kiểm thử.

Bảng 6: Tổng hợp kết quả đánh giá mô hình

Phương pháp đánh giá	Thang đo	Kết quả
BLEU Score (Beam Search k=5)	0 – 1	**0.4852**
METEOR Score (NLTK)	0 – 1	**0.7108**
TER Score (SacreBLEU)	0 – 100	**39.57**
Gemini Evaluation	1 – 10	**8.10**

9.2 Đánh giá định lượng

Kết quả tổng hợp các chỉ số trên tập Test VLSP được trình bày tại Bảng 6.

Nhận xét chi tiết:

- **BLEU và METEOR:** BLEU đạt 0.4852 cho thấy độ khớp n-gram cao. Đặc biệt, chỉ số METEOR đạt **0.7108** khẳng định mô hình không chỉ học vẹt các từ vựng mà còn nắm bắt tốt các biến thể từ và từ đồng nghĩa, do cơ chế so khớp linh hoạt của METEOR (như đã trình bày ở trên).

- **TER Score (39.57):** Với việc sử dụng sacrebleu để tính toán cả các thao tác dịch chuyển vị trí từ, kết quả 39.57 ngụ ý rằng người dịch cần thực hiện các thao tác chỉnh sửa trên khoảng 40% nội dung văn bản để đạt bản dịch hoàn thiện. Đây là mức hiệu suất khả quan đối với dữ liệu y tế chuyên ngành.

- **Gemini Score (8.10/10):** Điểm đánh giá ngữ nghĩa từ LLM tương đồng với kết quả cao của METEOR, xác nhận tính trôi chảy và dễ hiểu của bản dịch.

9.3 Đánh giá định tính và Phân tích lỗi

Dựa trên các mẫu dịch thử nghiệm, chúng tôi phân tích các ưu điểm và hạn chế (lỗi sai) cụ thể của mô hình:

1. Khả năng nắm bắt cấu trúc câu: Mô hình dịch khá mượt mà các cấu trúc câu y tế phức tạp.

- *Input:* "methodology : a cross sectional study was used among..."

- *Model:* "đối tượng và phương pháp nghiên cứu : nghiên cứu mô tả cắt ngang được thực hiện trên..."

- *Nhận xét:* Mô hình không chỉ dịch đúng "cross sectional study" là "nghiên cứu mô tả cắt ngang" mà còn tự động bổ sung cụm từ chuyên ngành "đối tượng và phương pháp" rất tự nhiên.

2. Lỗi dịch sai Thực thể Tên riêng: Đây là điểm yếu lớn nhất của mô hình hiện tại. Mô hình có xu hướng dịch sát nghĩa các tên riêng địa danh thay vì giữ nguyên hoặc phiên âm.

- *Input:* "...in **Phone Hong** and..." (Tên địa danh)

- *Model:* "...tại **điện thoại hồng**..."

- *Phân tích:* Mô hình tách từ "Phone" (điện thoại) và "Hong" (hồng) và dịch nghĩa đen, làm sai lệch hoàn toàn ý nghĩa địa danh.

3. Hiện tượng Ảo giác: Mô hình đôi khi sinh ra các từ vựng không có trong câu gốc hoặc sai lệch hoàn toàn ngữ cảnh.

- *Input:* "...influencing factors in vientiane, lao"

- *Model:* "...các yếu tố ảnh hưởng tại **cơ sở giết mổ** , lào"

- *Phân tích:* Cụm từ "vientiane" (thủ đô Viêng Chăn) bị dịch sai nghiêm trọng thành "cơ sở giết mổ". Lỗi này có thể do nhiễu trong dữ liệu huấn luyện hoặc mô hình bị nhầm lẫn giữa các token hiếm gặp.

Kết luận: Mô hình đạt hiệu suất tốt về mặt chỉ số tổng quát và độ trôi chảy, phù hợp để hỗ trợ dịch thuật sơ bộ. Tuy nhiên, để ứng dụng thực tế trong y tế, cần cải thiện khâu xử lý tên riêng (Named Entity Recognition) và kiểm soát lỗi ảo giác bằng cách mở rộng dữ liệu hoặc áp dụng các kỹ thuật ràng buộc khi sinh từ.

10 Thảo luận và Hướng nghiên cứu tương lai

Mặc dù các cải tiến kiến trúc đã mang lại hiệu quả rõ rệt về chất lượng và hiệu suất, một số hướng mở quan trọng vẫn chưa được khai thác, đặc biệt là tối ưu inference và khả năng mở rộng theo độ dài chuỗi.

10.1 KV Cache Implementation

Trong thiết lập hiện tại, mô hình chưa sử dụng *Key–Value cache*, khiến Key và Value phải được tính lại cho toàn bộ chuỗi tại mỗi bước sinh token, dẫn đến chi phí suy luận $O(T^2)$. Việc tích hợp KV cache cho phép tái sử dụng các cặp (K, V) đã tính toán, giảm độ phức tạp xuống $O(T)$ và cải thiện đáng kể độ trễ khi sinh chuỗi dài. Trong tương lai, chúng tôi hướng tới kết hợp KV cache với Grouped Query Attention nhằm giảm footprint bộ nhớ và hạn chế bottleneck về memory bandwidth trong suy luận.

10.2 Rotary Position Embedding (RoPE)

Mô hình hiện tại sử dụng sinusoidal positional encoding, vốn còn hạn chế trong việc mã hóa quan hệ vị trí tương đối và ngoại suy sang chuỗi dài. Rotary Position Embedding (RoPE) là một hướng thay thế tiềm năng, tích hợp trực tiếp thông tin vị trí vào attention và đã cho thấy khả năng ổn định hơn trong các thiết lập long-context.

11 Bài học kinh nghiệm (Lessons Learned)

Qua quá trình thiết kế, triển khai và đánh giá mô hình, chúng tôi rút ra một số bài học quan trọng như sau.

Thứ nhất, *kiến trúc mô hình đóng vai trò then chốt*. Các cải tiến kiến trúc hiện đại như Grouped Query Attention, RMSNorm và SwiGLU mang lại lợi ích rõ rệt cả về độ ổn định lẫn chất lượng đầu ra. Đặc biệt, việc sử dụng Pre-Layer Normalization giúp quá trình huấn luyện ổn định hơn đáng kể so với thiết kế Transformer gốc. Kết quả này cho thấy việc bám sát kiến trúc ban đầu mà không cập nhật các cải tiến mới có thể dẫn đến hiệu suất kém tối ưu.

Thứ hai, *các kỹ thuật huấn luyện có ảnh hưởng lớn không kém kiến trúc*. Label smoothing cho thấy hiệu quả nhất quán trong việc cải thiện khả năng tổng quát hóa của mô hình. Bên cạnh đó, lựa chọn optimizer và chiến lược điều chỉnh learning rate có tác động mạnh tới tốc độ hội tụ và độ ổn định trong huấn luyện, nhấn mạnh tầm quan trọng của việc tối ưu hóa pipeline huấn luyện một cách toàn diện.

Thứ ba, *đánh giá mô hình cần được tiếp cận đa chiều*. Việc chỉ dựa vào một chỉ số duy nhất như BLEU là chưa đủ để phản ánh đầy đủ chất lượng dịch. Phương pháp đánh giá dựa trên LLM-as-Judge cung cấp góc nhìn bổ sung về mức độ trôi chảy và phù hợp ngữ nghĩa, trong khi việc kiểm tra thủ công các mẫu đầu ra giúp phát hiện những lỗi hệ thống mà các chỉ số tự động thường bỏ sót.

Cuối cùng, *bộ nhớ thường là nút thắt chính khi mở rộng mô hình*. Việc sử dụng optimizer 8-bit cho phép tăng kích thước batch và cải thiện hiệu quả huấn luyện trong điều kiện tài nguyên hạn chế. Đồng thời, các cơ chế attention hiệu quả về bộ nhớ, chẳng hạn như GQA, trở nên ngày càng quan trọng khi mô hình và độ dài chuỗi tiếp tục tăng. Do đó, việc profiling và theo dõi mức sử dụng bộ nhớ ngay từ sớm là cần thiết.

hyperref

Acknowledgments

Source code có sẵn trên GitHub repository