BỘ CÔNG THƯƠNG

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP
NGÀNH CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI: NGHIÊN CỨU PHÁT TRIỂN ỨNG DỤNG HỖ TRỢ SINH CHÚ THÍCH HÌNH ẢNH DU LỊCH Ở HÀ NỘI BẰNG MÔ HÌNH BLIP

Giảng viên hướng dẫn	:	TS. Nguyễn Mạnh Cường
Sinh viên thực hiện	:	Trần Văn Tài
Mã sinh viên	:	2021604872
Lớp	:	CNTT05 – K16

Hà Nội - 2025

BỘ CÔNG THƯƠNG

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP
NGÀNH CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI: NGHIÊN CỨU PHÁT TRIỂN ỨNG DỤNG HỖ TRỢ SINH CHÚ THÍCH HÌNH ẢNH DU LỊCH Ở HÀ NỘI BẰNG MÔ HÌNH BLIP

Giảng viên hướng dẫn	:	TS. Nguyễn Mạnh Cường
Sinh viên thực hiện	:	Trần Văn Tài
Mã sinh viên	:	2021604872
Lớp	:	CNTT05 – K16

Hà Nội - 2025

MỤC LỤC

DANH MỤC NHỮNG TỪ VIẾT TẮT

CNN	Convolutional Neural Networks
BLIP	Bootstrapping Language-Image Pre-training
NLP	Natural Language Processing
RNN	Recurrent Neural Network
LSTM	Long short-term memory

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG BIỂU

LỜI CẢM ƠN

Lời đầu tiên cho phép em gửi lời cảm ơn sâu sắc tới các thầy cô trong Trường Đại Học Công Nghệ Thông Tin Và Truyền Thông – Đại Học Công Nghiệp Hà Nội, những người đã hết mình truyền đạt và chỉ dẫn cho em những kiến thức, những bài học quý báu và bổ ích. Đặc biệt em xin được bày tỏ sự tri ân và xin chân thành cảm ơn giảng viên TS. Nguyễn Mạnh Cường người trực tiếp hướng dẫn, chỉ bảo em trong suốt quá trình học tập, nghiên cứu và hoàn thành được đồ án.

Trong quá trình nghiên cứu và làm đề tài, do năng lực, kiến thức, trình độ bản thân em còn hạn hẹp nên không tránh khỏi những thiếu sót và em mong nhận được sự thông cảm và những góp ý từ quý thầy cô cũng như các bạn trong lớp.

Em xin trân trọng cảm ơn!

Sinh viên thực hiện

Trần Văn Tài

LỜI MỞ ĐẦU

Trong thời đại công nghệ số phát triển mạnh mẽ như hiện nay, công nghệ thông tin đóng vai trò vô cùng quan trọng trong mọi lĩnh vực của đời sống, từ giáo dục, y tế, đến công nghiệp và du lịch. Đặc biệt, trong lĩnh vực thị giác máy tính (Computer Vision), các mô hình học sâu (Deep Learning) ngày càng thể hiện rõ vai trò ưu việt của mình trong việc xử lý và hiểu nội dung hình ảnh. Một trong những ứng dụng nổi bật của thị giác máy tính là khả năng sinh chú thích hình ảnh (image captioning), giúp các hệ thống không chỉ nhìn thấy mà còn hiểu được nội dung trong ảnh. Điều này mở ra tiềm năng to lớn trong việc phát triển các ứng dụng thông minh, hỗ trợ người dùng tiếp cận và khai thác thông tin một cách trực quan và thuận tiện hơn.

Song song với đó, trí tuệ nhân tạo (AI) và học máy (Machine Learning) đang ngày càng phát triển mạnh mẽ, đặc biệt là với sự xuất hiện của các mô hình đa phương thức (multimodal) như BLIP (Bootstrapping Language-Image Pre-training). BLIP được đánh giá là một trong những mô hình tiên tiến trong việc kết hợp giữa xử lý ngôn ngữ tự nhiên và thị giác máy tính, cho phép sinh ra các chú thích chính xác, tự nhiên và phù hợp với ngữ cảnh của hình ảnh. Việc ứng dụng các mô hình như BLIP không chỉ là một bước tiến kỹ thuật, mà còn mang đến giá trị thực tiễn trong nhiều lĩnh vực như giáo dục, du lịch, thương mại điện tử, và truyền thông.

Xuất phát từ tiềm năng đó, đề tài **"Nghiên cứu phát triển ứng dụng hỗ trợ sinh chú thích hình ảnh du lịch ở Hà Nội bằng mô hình BLIP"** được lựa chọn và thực hiện. Đề tài tập trung vào việc nghiên cứu và ứng dụng mô hình BLIP để xây dựng một hệ thống có khả năng tự động sinh chú thích cho các hình ảnh du lịch tại Hà Nội – một trong những địa điểm du lịch nổi tiếng và giàu giá trị văn hóa lịch sử. Với ứng dụng này, người dùng có thể dễ dàng hiểu được nội dung và bối cảnh của các hình ảnh, từ đó tăng cường trải nghiệm du lịch, đồng thời có thể hỗ trợ trong các hoạt động quảng bá du lịch địa phương. Ngoài ra, đề tài cũng góp

phần vào việc tiếp cận công nghệ mới, mở rộng hiểu biết và ứng dụng các mô hình học sâu trong thực tế.

Báo cáo được chia thành **4 chương** như sau:

- **Chương I: Tổng quan về bài toán sinh chú thích hình ảnh**. Chương này giới thiệu tổng quan về lĩnh vực thị giác máy tính và xử lý ngôn ngữ tự nhiên, trình bày sự phát triển của học sâu trong bài toán sinh chú thích hình ảnh. Nội dung chương cũng làm rõ định nghĩa bài toán, mô tả chi tiết, các thách thức phải đối mặt, cũng như các ứng dụng thực tế của công nghệ này.
- **Chương II: Sinh chú thích hình ảnh bằng mô hình BLIP**. Đây là chương trọng tâm của báo cáo, trình bày sâu về các phương pháp sinh chú thích hình ảnh, đặc biệt là cách tiếp cận sử dụng mô hình Transformer. Chương này phân tích chi tiết kiến trúc của Transformer và đi sâu vào mô hình BLIP, giới thiệu cấu trúc mô hình, cách biểu diễn đặc trưng và luồng dữ liệu trong mô hình này.
- **Chương III: Thực nghiệm.** Chương này trình bày các thực nghiệm được thực hiện để đánh giá hiệu quả của mô hình. Nội dung bao gồm mô tả dữ liệu thực nghiệm, quy trình huấn luyện mô hình và các phương pháp đánh giá được sử dụng để xác định hiệu suất của mô hình.
- **Chương IV: Xây dựng ứng dụng**. Chương cuối cùng tập trung vào việc xây dựng sản phẩm demo ứng dụng mô hình BLIP trong thực tế. Chương này giới thiệu các framework được sử dụng, phân tích thiết kế hệ thống, mô tả giao diện và các chức năng chính của hệ thống.

Báo cáo trình bày tổng quan, phương pháp, thực nghiệm và ứng dụng của mô hình BLIP trong bài toán sinh chú thích hình ảnh, cho thấy tiềm năng lớn của mô hình trong việc kết nối hình ảnh và ngôn ngữ.

CHƯƠNG I: TỔNG QUAN VỀ BÀI TOÁN SINH CHÚ THÍCH HÌNH ẢNH

1.1. Tổng quan về thị giác máy tính và xử lý ngôn ngữ tự nhiên

Trong những năm gần đây, trí tuệ nhân tạo (AI) đã có những bước tiến vượt bậc, đặc biệt là trong các lĩnh vực như thị giác máy tính (Computer Vision) và xử lý ngôn ngữ tự nhiên (Natural Language Processing – NLP). Hai lĩnh vực này đóng vai trò quan trọng trong việc giúp máy tính có thể "hiểu" thế giới xung quanh và "giao tiếp" hiệu quả hơn với con người[1].

Thị giác máy tính tập trung vào việc phân tích và hiểu nội dung hình ảnh và video, như nhận diện khuôn mặt, phân loại vật thể, phát hiện hành vi,... Trong khi đó, xử lý ngôn ngữ tự nhiên giúp máy hiểu và sinh ngôn ngữ của con người dưới dạng văn bản hoặc giọng nói[2]. Sự kết hợp của hai lĩnh vực này đã mở ra một hướng đi mới cho các hệ thống thông minh – cụ thể là các hệ thống có khả năng tạo ra mô tả ngữ nghĩa từ hình ảnh đầu vào, hay còn gọi là bài toán sinh chú thích hình ảnh.

Việc cho máy tính khả năng nhìn và nói – tức là hiểu một hình ảnh và mô tả nó bằng ngôn ngữ tự nhiên – không chỉ là một thành tựu về mặt kỹ thuật, mà còn có ý nghĩa thực tiễn sâu sắc trong nhiều lĩnh vực như giáo dục, du lịch, truyền thông, hỗ trợ người khiếm thị, và các hệ thống trợ lý ảo.

1.2. Học sâu và sự phát triển của bài toán sinh chú thích hình ảnh

Sự bùng nổ của học sâu (Deep Learning) với các mô hình mạnh mẽ như mạng nơ-ron tích chập (CNN), mạng tuần tự (RNN, LSTM) và đặc biệt là các mô hình Transformer đã tạo nền tảng vững chắc cho sự phát triển của các hệ thống sinh chú thích hình ảnh hiện đại[5].

Một hệ thống sinh chú thích ảnh điển hình bao gồm hai thành phần chính:

- Bộ mã hóa hình ảnh (Image Encoder): thường là mạng CNN (ResNet, ViT,...) dùng để trích xuất đặc trưng thị giác từ hình ảnh.

- Bộ giải mã ngôn ngữ (Text Decoder): thường là LSTM hoặc Transformer, nhận đầu vào là vector đặc trưng hình ảnh và tạo ra một chuỗi từ mô tả nội dung bức ảnh.

Các mô hình hiện đại như BLIP (Bootstrapped Language Image Pretraining) đã nâng cấp khả năng của các hệ thống này thông qua việc huấn luyện đa nhiệm trên dữ liệu lớn, kết hợp giữa sinh chú thích, tìm kiếm ảnh bằng văn bản và học biểu diễn hình ảnh – văn bản một cách thống nhất.

Sự phát triển của các mô hình học sâu không chỉ giúp cải thiện độ chính xác trong việc nhận diện và mô tả ảnh, mà còn giúp mô tả trở nên tự nhiên hơn, mang ngữ nghĩa đầy đủ và phong phú hơn.

1.3 Bài toán sinh chú thích hình ảnh

1.3.1 Giới thiệu bài toán

Bài toán sinh chú thích hình ảnh (Image Captioning) là quá trình tự động tạo ra một câu mô tả nội dung của hình ảnh dưới dạng ngôn ngữ tự nhiên. Đây là một bài toán đặc biệt vì nó yêu cầu kết hợp giữa khả năng hiểu hình ảnh và khả năng sinh ngôn ngữ – một dạng bài toán đa mô thức (multimodal task).

Không giống như các bài toán phân loại ảnh thông thường chỉ yêu cầu dự đoán nhãn, sinh chú thích ảnh đòi hỏi hệ thống phải nắm bắt được nội dung chính, bối cảnh, hành động (nếu có), và tạo ra một câu mô tả tự nhiên, chính xác.

Ví dụ: Một bức ảnh chụp Văn Miếu – Quốc Tử Giám có thể được mô tả như: “Du khách tham quan Văn Miếu – Quốc Tử Giám trong một buổi chiều mùa thu tại Hà Nội.”Mô tả chi tiết bài toán

1.3.2 Mô tả chi tiết bài toán

Mô hình thực hiện nhiệm vụ theo cấu trúc đầu vào và đầu ra như sau:

- Đầu vào: Một hình ảnh kỹ thuật số, có thể là ảnh chụp địa điểm du lịch, cảnh quan, công trình kiến trúc hoặc các hoạt động tại Hà Nội.
- Đầu ra: Một hoặc nhiều câu mô tả ngắn bằng tiếng Việt, phản ánh nội dung và bối cảnh của hình ảnh.

Một số yêu cầu đối với đầu ra:

- Ngôn ngữ tự nhiên, dễ hiểu và phù hợp với ngữ cảnh.
- Không có lỗi chính tả hay cú pháp.
- Phản ánh được đúng các đối tượng và hành động trong ảnh.
- Ưu tiên các thông tin mang tính mô tả du lịch, địa danh, văn hóa.

1.3.3 Các thách thức của bài toán

Dưới đây là một số thách thức chính mà bài toán sinh chú thích hình ảnh trong lĩnh vực du lịch phải đối mặt:

- Đa dạng hình ảnh: Hình ảnh trong lĩnh vực du lịch rất đa dạng về bố cục, ánh sáng, mùa trong năm, thời tiết, số lượng người và đối tượng trong ảnh.
- Hiểu ngữ cảnh: Một số ảnh có thể khó phân biệt nếu không hiểu rõ ngữ cảnh văn hóa – lịch sử (ví dụ: Đình, Chùa, Miếu,...).
- Sinh ngôn ngữ tự nhiên: Việc tạo câu mô tả không chỉ chính xác mà còn phải mượt mà và mang tính mô tả, không khô khan hay máy móc.
- Tối ưu tốc độ và hiệu năng: Hệ thống cần xử lý ảnh nhanh, đặc biệt nếu triển khai trên nền tảng web hoặc thiết bị di động.

1.3.4 Ứng dụng thực tế

Một số ứng dụng thực tế nổi bật của công nghệ sinh chú thích hình ảnh bao gồm:

- Hệ thống hướng dẫn du lịch thông minh: Mô tả tự động các địa điểm du lịch qua ảnh giúp khách du lịch hiểu thêm về nơi họ đang đến.
- Trợ lý ảo du lịch: Có thể kết hợp với chatbot để hỗ trợ trả lời câu hỏi hoặc gợi ý lịch trình.
- Thư viện ảnh thông minh: Tự động gán nhãn và mô tả cho ảnh, hỗ trợ tìm kiếm và phân loại ảnh hiệu quả.
- Mạng xã hội và ứng dụng chia sẻ: Tạo mô tả tự động cho ảnh khi người dùng tải lên, giúp tăng tương tác và khả năng truy xuất

CHƯƠNG II: SINH CHÚ THÍCH HÌNH ẢNH BẰNG MÔ HÌNH BLIP

2.1 Tổng quan về các phương pháp sinh chú thích hình ảnh

Sinh chú thích hình ảnh (Image Captioning) là một bài toán quan trọng trong lĩnh vực kết hợp giữa thị giác máy tính (Computer Vision) và xử lý ngôn ngữ tự nhiên (Natural Language Processing). Nhiệm vụ chính của bài toán này là xây dựng một hệ thống có khả năng phân tích nội dung của một hình ảnh và tạo ra một hoặc nhiều câu mô tả bằng ngôn ngữ tự nhiên, phản ánh chính xác các đối tượng, hành động và mối quan hệ xuất hiện trong hình ảnh đó.

2.1.1 Đặc điểm của bài toán sinh chú thích hình ảnh

Tính đa mô thức

Sinh chú thích hình ảnh là một bài toán đa mô thức (multimodal) đòi hỏi hệ thống phải xử lý đồng thời hai loại dữ liệu khác nhau: dữ liệu thị giác (hình ảnh) và dữ liệu ngôn ngữ (văn bản). Sự kết hợp này tạo ra nhiều thách thức độc đáo về mặt kỹ thuật.

Yêu cầu về hiểu ngữ cảnh

Để tạo ra các chú thích có ý nghĩa, hệ thống không chỉ cần nhận diện được các đối tượng riêng lẻ trong ảnh mà còn phải hiểu được mối quan hệ giữa chúng, bối cảnh không gian, thời gian, và các hoạt động đang diễn ra. Ví dụ, thay vì chỉ nhận diện "người" và "xe đạp", hệ thống cần hiểu rằng "một người đang đạp xe trên con đường ven hồ".

Yêu cầu về ngôn ngữ tự nhiên

Chú thích được tạo ra phải đúng về mặt ngữ pháp, có cấu trúc rõ ràng, và nghe tự nhiên như được viết bởi con người. Điều này đòi hỏi hệ thống không chỉ "hiểu" nội dung hình ảnh mà còn phải "biết" cách diễn đạt thông tin đó bằng ngôn ngữ tự nhiên, phù hợp với ngữ cảnh và văn hóa.

2.1.2 Phân loại các phương pháp sinh chú thích hình ảnh

Các phương pháp sinh chú thích hình ảnh có thể được phân loại thành ba nhóm chính dựa trên cách tiếp cận và cơ chế hoạt động của chúng:

Phương pháp dựa trên mẫu (Template-based Methods)

Phương pháp này sử dụng các cấu trúc câu hoặc khuôn mẫu (template) được định nghĩa sẵn, sau đó điền các thông tin được trích xuất từ hình ảnh vào các vị trí thích hợp trong khuôn mẫu đó.

Nguyên lý hoạt động:

- Nhận diện các đối tượng, hành động và thuộc tính từ hình ảnh
- Áp dụng các luật hoặc mẫu câu có sẵn
- Điền thông tin vào các vị trí trống trong mẫu

Ví dụ khuôn mẫu:

- "Có [số lượng] [đối tượng] đang [hành động] tại [địa điểm]."
- "Một [đối tượng1] và một [đối tượng2] đang [hành động] trong [bối cảnh]."

Ưu điểm:

- Đơn giản, dễ triển khai
- Tạo ra các câu đúng ngữ pháp
- Không yêu cầu dữ liệu huấn luyện lớn

Nhược điểm

- Thiếu linh hoạt, dẫn đến các mô tả máy móc và lặp lại
- Không thể mô tả các tình huống phức tạp hoặc không thông thường
- Khó khăn trong việc tạo ra sự đa dạng về phong cách ngôn ngữ

Phương pháp truy xuất (Retrieval-based Methods)

Phương pháp này hoạt động bằng cách tìm kiếm trong một cơ sở dữ liệu các ảnh tương tự với ảnh đầu vào, sau đó sử dụng hoặc điều chỉnh các mô tả đã có của những ảnh tương tự đó.

Nguyên lý hoạt động:

- Trích xuất đặc trưng của hình ảnh đầu vào

- Tìm kiếm trong cơ sở dữ liệu các hình ảnh có đặc trưng tương tự
- Lấy hoặc tổng hợp mô tả từ các ảnh tương tự nhất

Ưu điểm

- Cho kết quả tự nhiên vì sử dụng các mô tả được viết bởi con người
- Đảm bảo tính chính xác về mặt ngữ pháp và ngữ nghĩa
- Không yêu cầu huấn luyện mô hình phức tạp

Nhược điểm

- Phụ thuộc vào kích thước và chất lượng của cơ sở dữ liệu
- Khó khăn trong việc mô tả các tình huống mới, không có trong cơ sở dữ liệu
- Có thể không nắm bắt được các chi tiết đặc thù của hình ảnh đầu vào

Phương pháp sinh (Generative Methods)

Đây là phương pháp phổ biến và tiên tiến nhất hiện nay, sử dụng các mô hình học sâu để học cách sinh ra câu mô tả từ đặc trưng của hình ảnh. Các phương pháp này có thể tạo ra các mô tả mới, không bị giới hạn bởi các mẫu có sẵn.

Nguyên lý hoạt động:

- Mã hóa hình ảnh thành vector đặc trưng sử dụng các mạng nơ-ron tích chập (CNN) hoặc Transformer
- Sử dụng các mô hình ngôn ngữ để sinh ra chuỗi từ mô tả dựa trên đặc trưng hình ảnh

Các kiến trúc tiêu biểu:

- CNN + RNN: Kết hợp CNN để trích xuất đặc trưng hình ảnh và RNN (LSTM, GRU) để sinh ra văn bản mô tả.
- Attention-based Models: Áp dụng cơ chế attention để tập trung vào các vùng khác nhau trong hình ảnh khi sinh từng từ trong mô tả.
- Transformer-based Models: Sử dụng kiến trúc Transformer cho cả việc mã hóa hình ảnh và sinh văn bản, giúp nắm bắt tốt hơn các mối quan hệ phức tạp.

- Pre-trained Vision-Language Models: Các mô hình đa phương thức được huấn luyện trước trên dữ liệu lớn như BLIP, Flamingo, CLIP, và GIT.

Ưu điểm:

- Khả năng tạo ra các mô tả sáng tạo, linh hoạt và đa dạng
- Hiệu suất cao trên các bộ dữ liệu chuẩn
- Có thể mô tả các tình huống phức tạp và chi tiết

Nhược điểm:

- Yêu cầu dữ liệu huấn luyện lớn
- Đòi hỏi tài nguyên tính toán mạnh
- Có thể sinh ra các mô tả không chính xác hoặc không liên quan

2.2 Mô hình Transformer trong sinh chú thích hình ảnh

2.2.1 Giới thiệu về Transformer

Trong những năm gần đây, lĩnh vực xử lý ngôn ngữ tự nhiên (NLP) đã chứng kiến sự phát triển mạnh mẽ của các mô hình học sâu, đặc biệt là với sự ra đời của kiến trúc Transformer. Trước khi Transformer xuất hiện, các mô hình như Recurrent Neural Network (RNN) và các biến thể của nó như Long Short-Term Memory (LSTM) đã thống trị trong việc xử lý dữ liệu tuần tự. Tuy nhiên, RNN và LSTM gặp phải những hạn chế vì bản chất tuần tự của chúng, dẫn đến việc không thể tận dụng tối đa khả năng tính toán song song của GPU, khiến cho quá trình huấn luyện trở nên chậm chạp và kém hiệu quả[7].

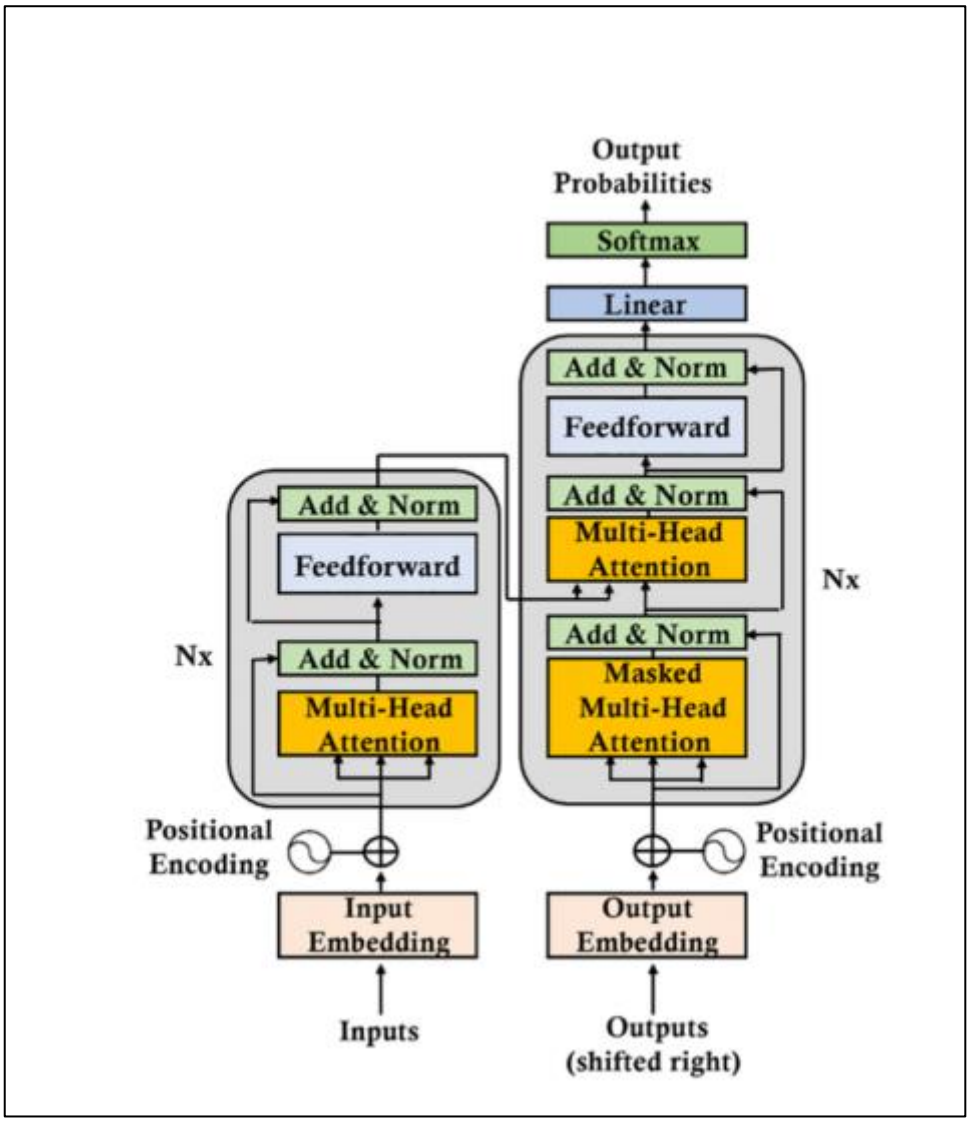
So với các phương pháp sinh chú thích hình ảnh truyền thống được trình bày trong phần 2.1, Transformer mang lại một số ưu điểm quan trọng:

- Khả năng xử lý song song: Không như RNN/LSTM phải xử lý tuần tự, Transformer cho phép tính toán song song, đẩy nhanh quá trình huấn luyện và suy luận.
- Nắm bắt tốt hơn mối quan hệ không gian: Cơ chế self-attention cho phép mô hình liên kết các vùng khác nhau trong hình ảnh một cách hiệu quả.
- Xử lý tốt phụ thuộc dài hạn: Transformer không bị giới hạn bởi vấn đề mất mát gradient như các mô hình RNN truyền thống.

- Tính mở rộng: Kiến trúc Transformer dễ dàng mở rộng quy mô và tận dụng được sức mạnh của phần cứng hiện đại (GPU, TPU).

2.2.2 Kiến trúc Transformer

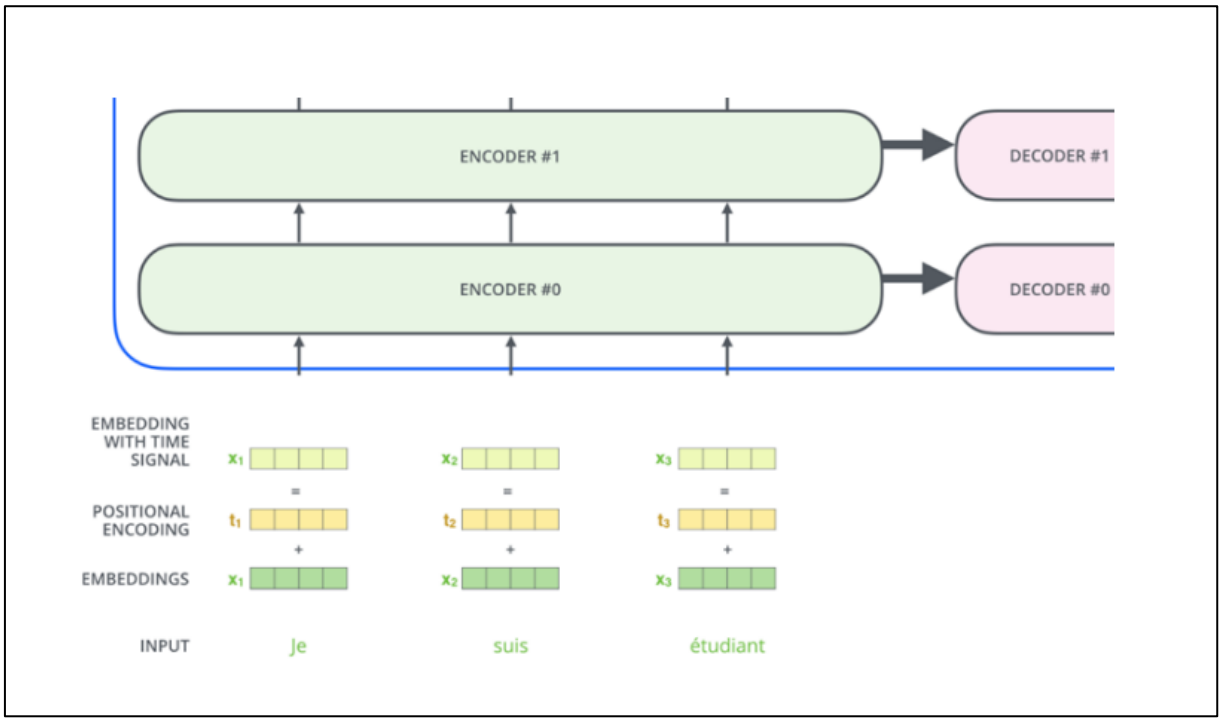
Transformer, được giới thiệu trong bài báo "Attention is All You Need" vào năm 2017 [12], hoàn toàn dựa vào cơ chế self-attention, cho phép mô hình xem xét tất cả các phần của đầu vào đồng thời mà không cần xử lý tuần tự. Điều này cải thiện khả năng học hỏi của mô hình, đồng thời tăng tốc độ huấn luyện nhờ khả năng tính toán song song. Transformer bao gồm hai phần chính: bộ mã hóa (encoder) và bộ giải mã (decoder), với nhiều lớp attention và mạng nơ-ron hồi tiếp (feed-forward networks) ở mỗi lớp.[9]



Hình 2.1: Kiến trúc Transformer[9]

Các thành phần trong kiến trúc Transformer:

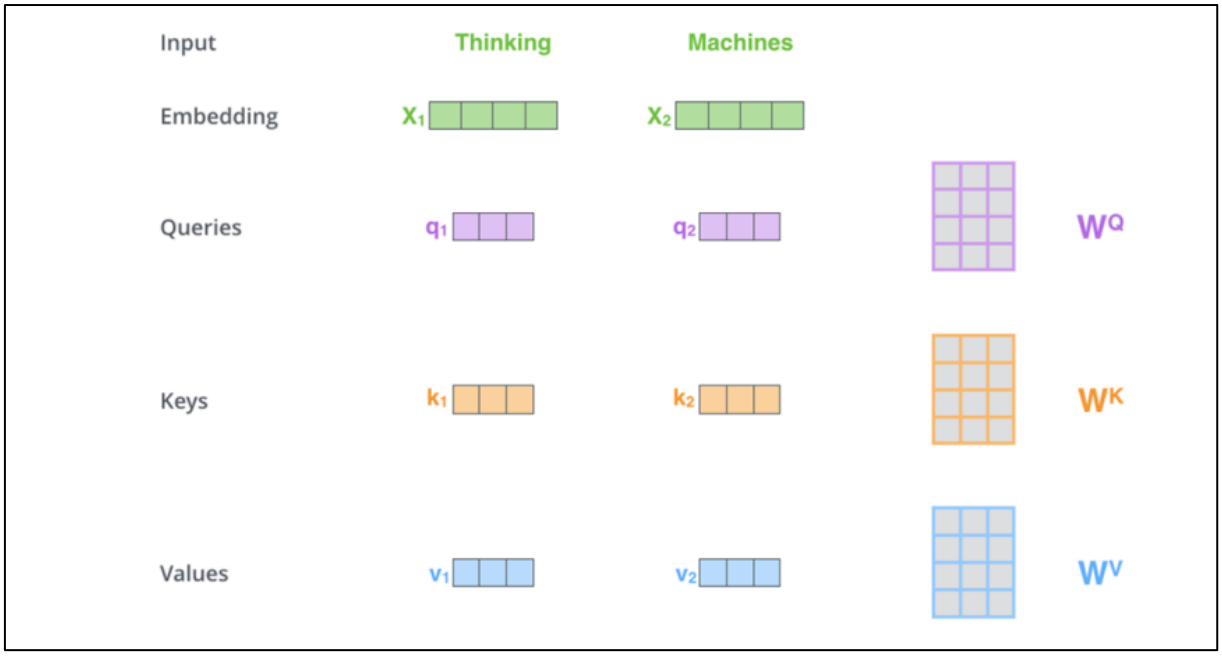
Encoder là thành phần đầu vào trong các mô hình dạng Transformer hoặc các mô hình sequence-to-sequence (chuỗi sang chuỗi), có nhiệm vụ xử lý và biến đổi dữ liệu đầu vào thành một dạng biểu diễn (representation) mà mô hình có thể hiểu và sử dụng cho các bước tiếp theo. Trong đó Positional Encoding là một kỹ thuật được dùng trong các mô hình Transformer để cung cấp thông tin về thứ tự của các phần tử trong chuỗi đầu vào. Đầu vào của mô hình transformer, tất cả các từ trong câu sẽ được đưa vào để xử lý song song, tuy nhiên phải có một cơ chế để đánh dấu vị trí của các từ đó là positional encodding. Vec-tơ chứa thông tin vị trí sẽ được thêm vào với vec-tơ từ. Việc đánh số vị trí này giúp máy có thể hiểu được rõ nghĩa của các câu hơn. Ví dụ câu "tôi là bạn của cậu" và câu "cậu là bạn của tôi" nếu không được đánh dấu thì máy sẽ hiểu là 2 câu này ngữ nghĩa giống nhau mặc dù nghĩa của chúng hoàn toàn ngược lại.



Hình 2.2: Positional Encoding[9]

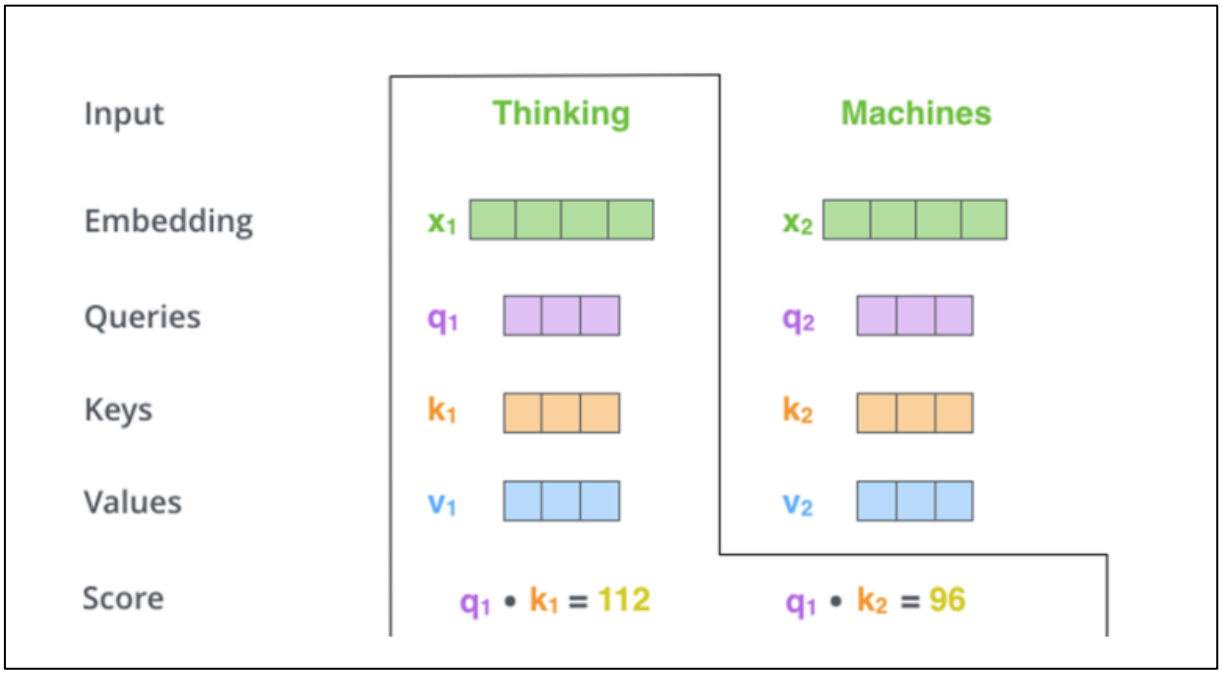
Self-Attention: là một cơ chế để xử lý và tạo ra mối quan hệ giữa các từ trong câu. Ví dụ với một câu "Bữa cơm này thực sự rất ngon vì nó toàn những món mẹ tôi làm" thì từ "nó" trong đây là biểu thị cho từ "Bữa cơm" tuy nhiên trong mô

hình RNN truyền thống thì nó được xử lý tuần tự nên các mối quan hệ xa khó biểu diễn bởi từ "nó" cách xa từ "bữa cơm". Nhưng self-attention không bị giới hạn bởi tuần tự, nó sẽ cho toàn bộ chuỗi đầu vào một lúc và xác định mối quan hệ của từng từ với các từ còn lại trong chuỗi. Xác định mối quan hệ bất kỳ từ nào với bất kỳ từ nào, bất kể khoảng cách nào. Ví dụ ở trong hình dưới là cách hoạt động self-attention:



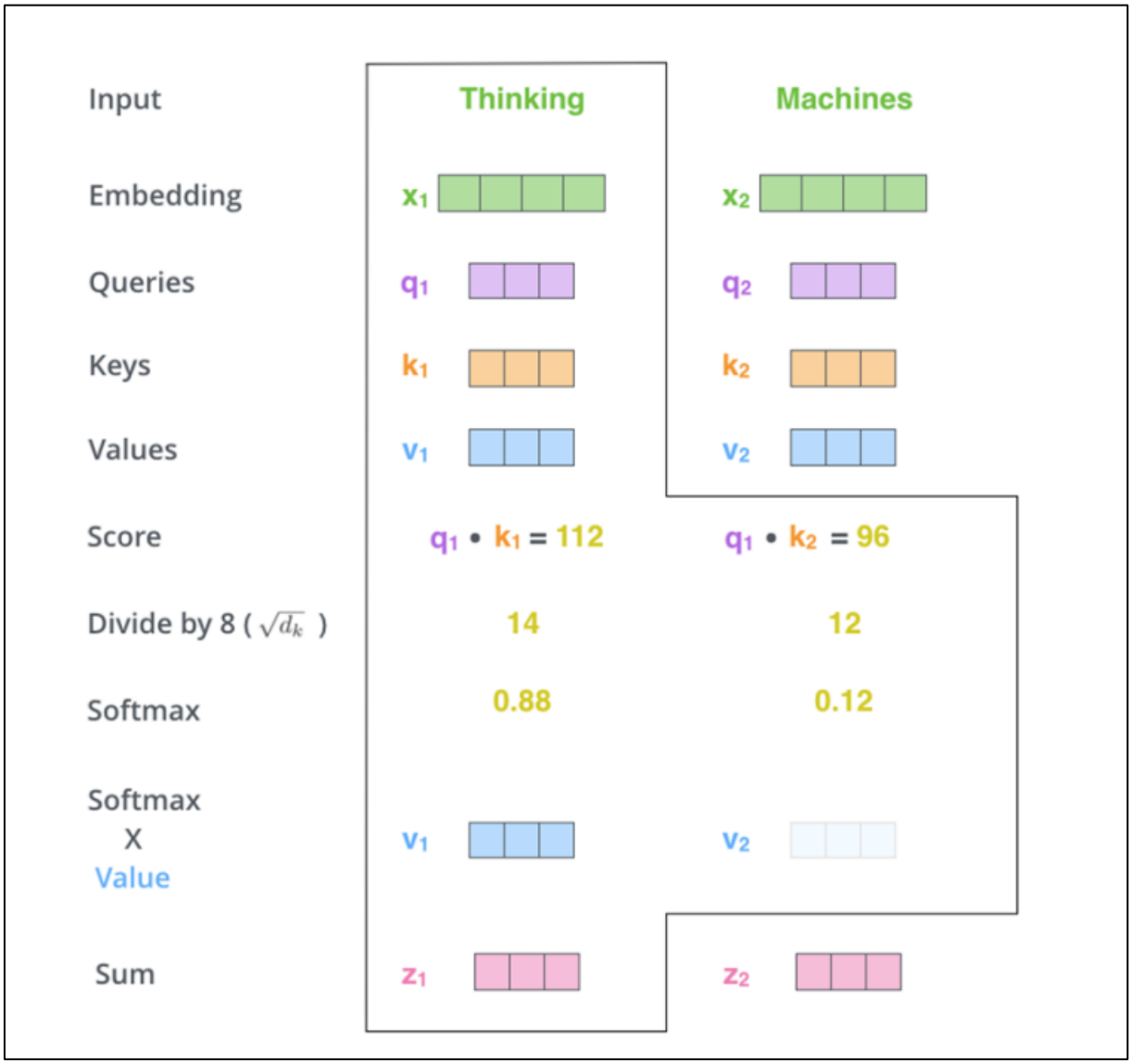
Hình 2.3: Cách hoạt động của Self-Attention[9]

Đầu tiên nó sẽ tạo các vec-tơ Query(Q), Key(K), Value(V) từ embedding đầu vào. Mỗi từ trong câu đã được ánh xạ thành vec-tơ nhúng chúng sẽ được nhân với 3 ma trận trọng số được khởi tạo ngẫu nhiên lúc đầu. Trong lúc mô hình được huấn luyện chúng sẽ được học cho phù hợp hơn. Với từng từ ví dụ như từ "Thinking" ở trên lấy Query vec-tơ của nó tính tích vô hướng với Key vec-tơ của mỗi từ khác để ra điểm cho từng từ với các từ khác ví dụ: q1.k1, q1.k2, . . .[11]



Hình 2.4: Cách hoạt động của Self-Attention[9]

Sau khi tính được điểm chia chúng cho căn bậc 2 của kích thước vec-tơ K rồi đưa áp dụng hàm softmax để đưa điểm về dạng xác suất, xác định từ có liên quan nhất. Bây giờ nhân từng vec-tơ Value với từng xác suất tương ứng từ softmax. Vec-tơ value này sẽ được cộng lại để ra vec-tơ biểu diễn mới của từ hiện tại. Vec-tơ này đã chứa đủ mức độ liên quan của từ ví dụ như từ “Thinking” với các từ còn lại trong câu. Các từ còn lại trong câu cũng thực hiện như vậy. Đây cũng chính là vec-tơ đầu ra Z của lớp seft-attention[11].



Hình 2.5: Cách hoạt động của Self-Attention[9]

Công thức của self-attention được biểu diễn như sau:

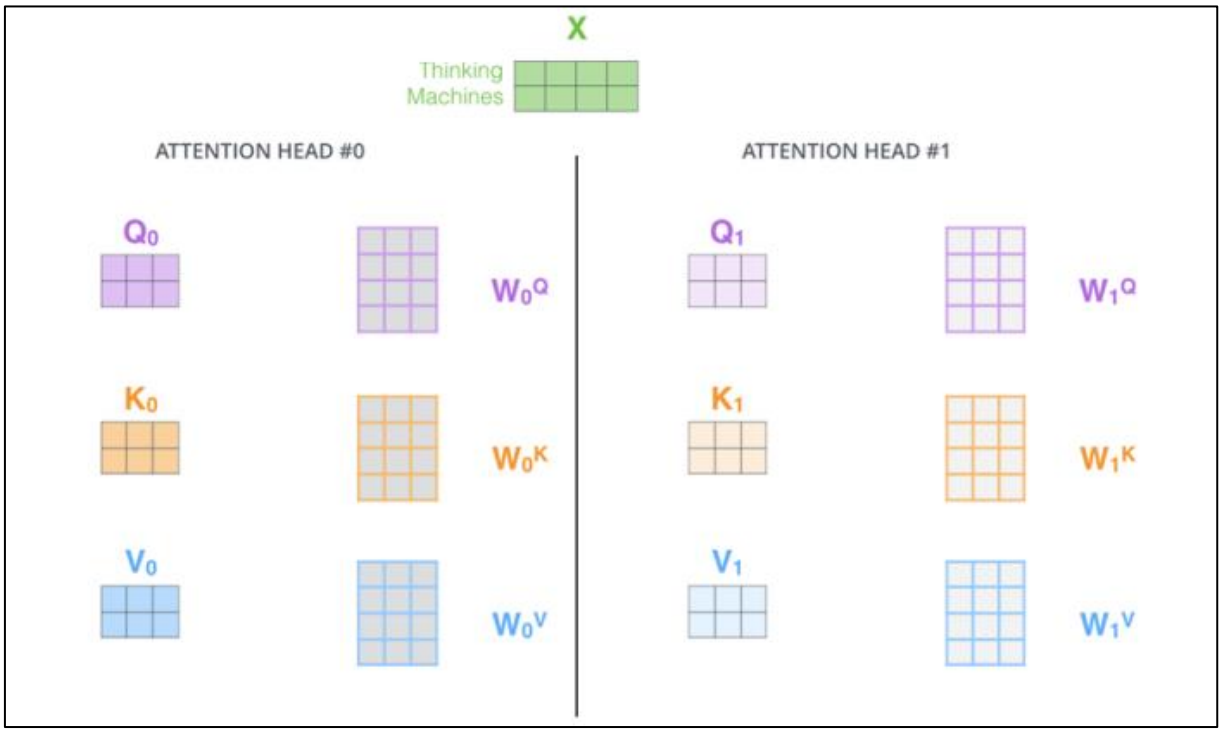
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} + M\right)V$$

Trong đó:

- Q (Query), K (Key), V (Value) là các biến đổi tuyến tính của các vector đặc trưng hình ảnh
- d_k là chiều của vector K
- $\sqrt{d_k}$ là hệ số tỷ lệ ổn định của gradient

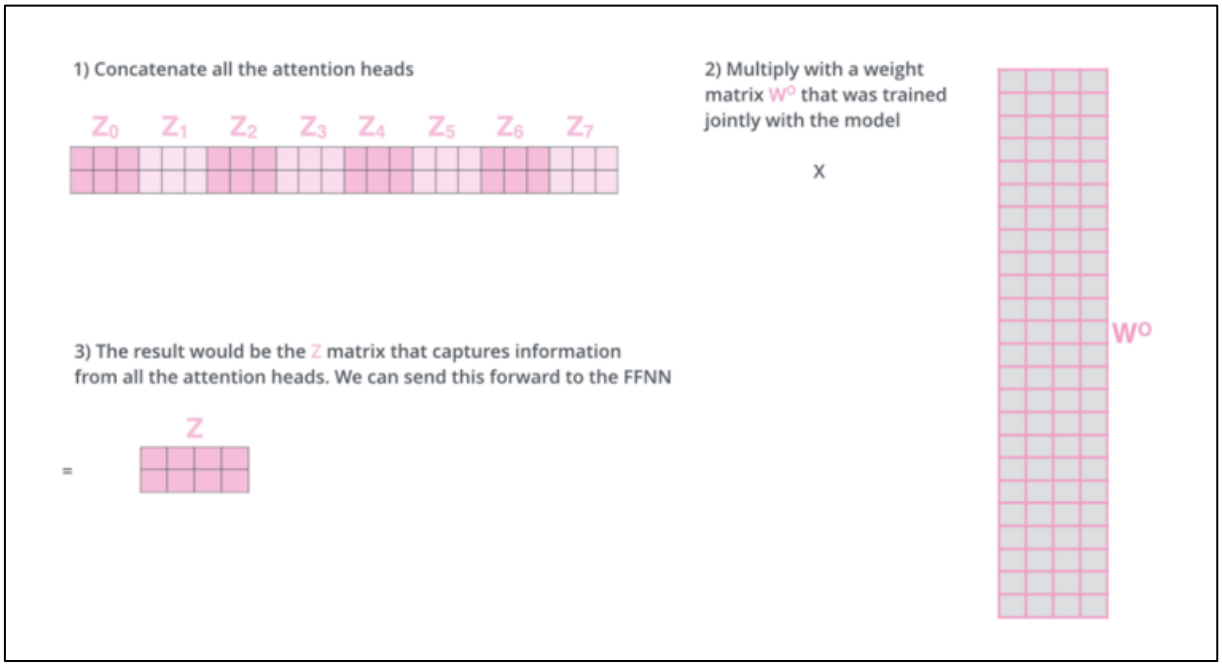
Multi-Headed Attention: là một thành phần quan trọng trong kiến trúc Transformer, giúp mô hình học được các mối quan hệ phức tạp và đa dạng giữa

các phần tử trong dữ liệu đầu vào. Như trong mô hình CNN, một ảnh đầu vào sẽ học càng được nhiều đặc trưng với bộ lọc ma trận filter. Transformer cũng vậy, multi-headed attention là cải tiến của seft-attention mở rộng khả năng tập trung vào các mối quan hệ khác nhau giữa các từ cũng như các đặc trưng khác của ảnh trong CNN vậy. Nó sẽ khởi tạo các bộ trọng số khác nhau, từ đó tính ra được các vec-tơ Z khác nhau, nối các vec-tơ Z này thành một ma trận lớn rồi nhân với ma trận W0, một ma trận trọng số được học trong quá trình training.



Hình 2.6: Các attention head khác nhau[9]

Ma trận Z cuối cùng này sẽ là vec-tơ của các từ với các quan hệ của chúng trong câu đã được thêm vào.



Hình 2.7: Đầu ra các attention head[9]

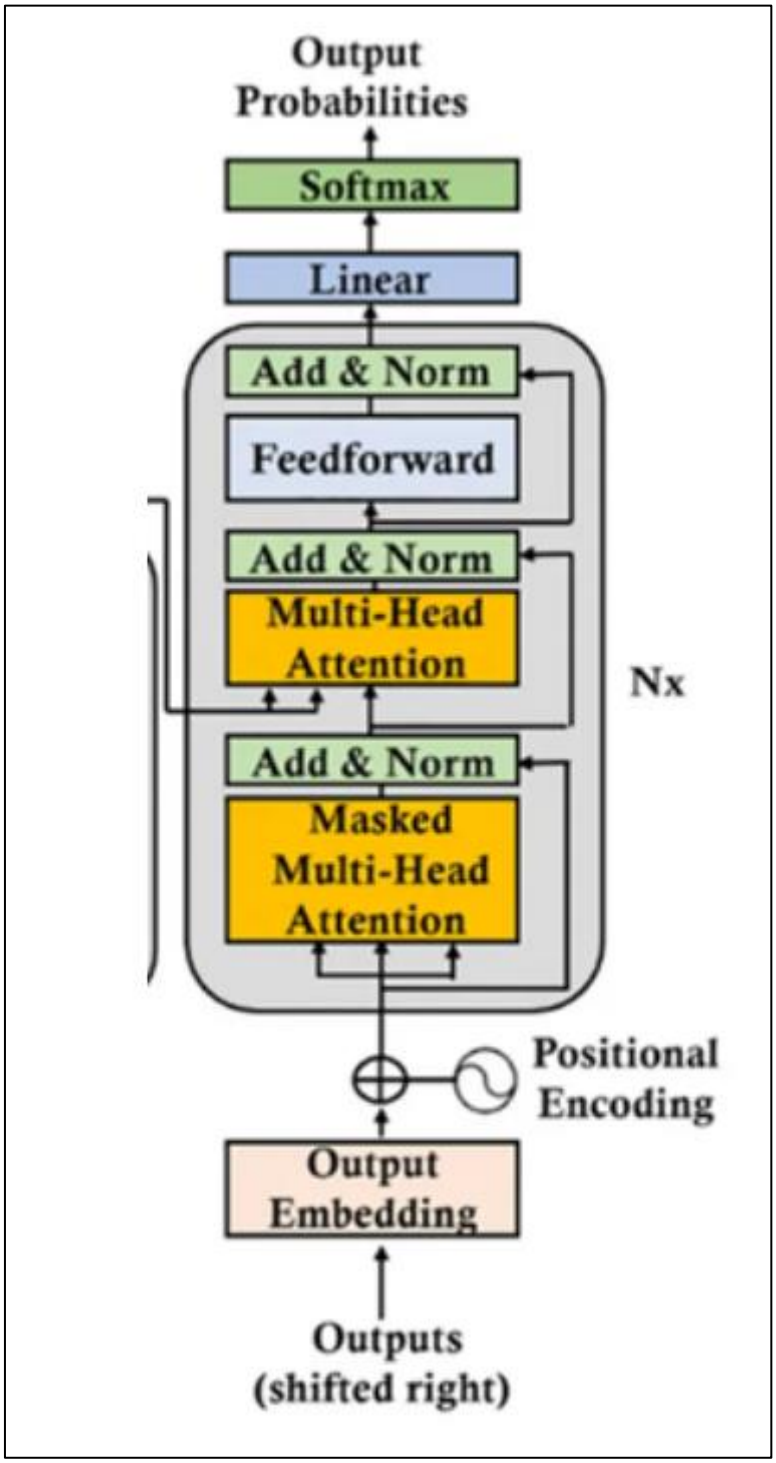
Ngoài ra, sau mỗi bước seft-attention và feed forward thì chúng có thêm một bước nữa là kết nối dữ thừa (the residual) tức là đầu vào của mỗi lớp (seft-attention, feed forward) được thêm trực tiếp vào đầu ra của mỗi lớp đó. Mục đích của việc này là để giúp bảo toàn thông tin gốc, mô hình dễ học, hội tụ nhanh hơn, ổn định quá trình huấn luyện và cải thiện khả năng học của mô hình.

Công thức:

$$\text{MultiHead (Q, K, V)} = \text{Concat (}\ head_1, \dots, head_h)W^0$$

Trong đó:

- Q, K, V: truy vấn, khóa, giá trị
- W_i^Q, W_i^K, W_i^V : trọng số học được cho từng head
- $head_i$: attention head thứ i
- W^0 : trọng số chiếu output tổng hợp

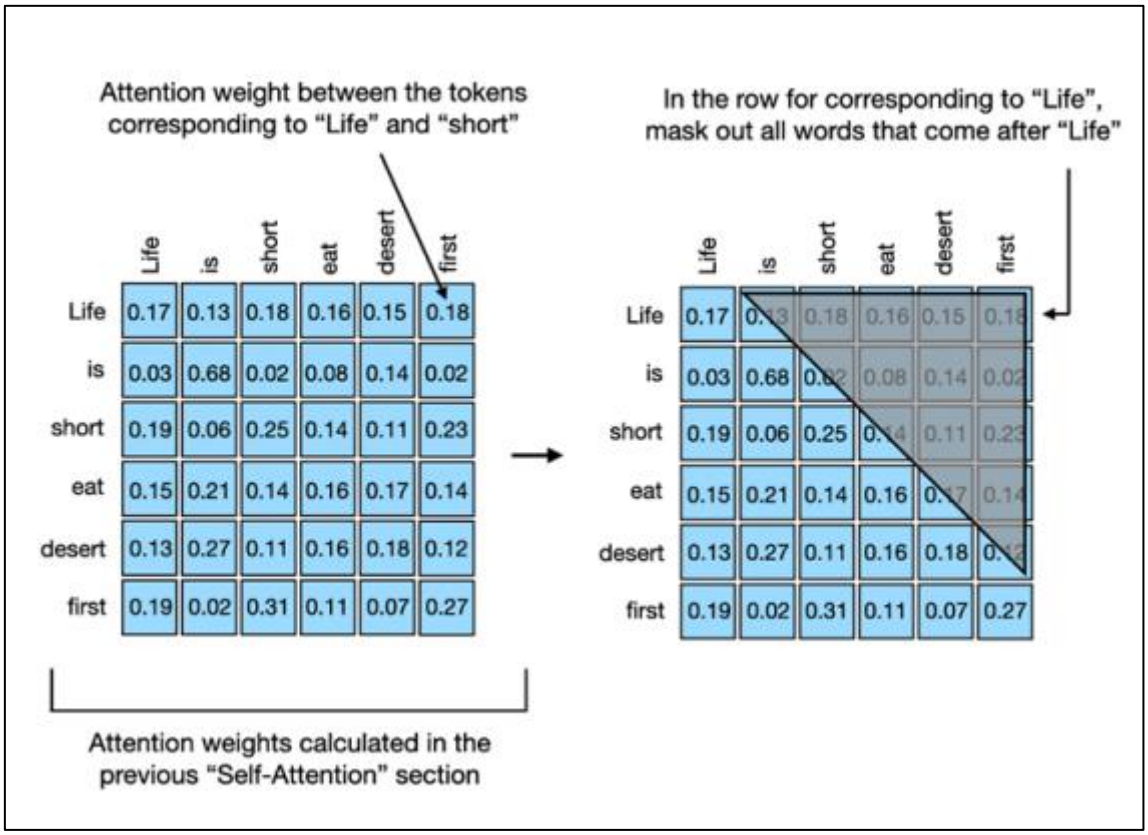


Hình 2.8: Decoder trong kiến trúc Transformer[9]

Decoder là thành phần trong các mô hình dạng Transformer hoặc sequence-to-sequence (chuỗi sang chuỗi), có nhiệm vụ tạo ra đầu ra (ví dụ như câu dịch, câu chú thích, văn bản…) dựa trên thông tin đã được mã hóa bởi Encoder. Chuỗi dữ liệu đầu ra ví dụ như câu tiếng việt câu dịch của câu tiếng anh được đưa vào

phía decoder cũng thực hiện các bước embedding và position encoding giống như phía encoder. Nhưng nó sẽ được đưa vào Masked multi-headed attention.

Masked multi-headed attention: Masked multi-headed attention khá tương đồng với multi-attention về cách tính tuy nhiên chúng khác một điểm đó là "masked". Sở dĩ chúng phải che lại phía decoder giống như trong hình để mô hình không nhìn thấy được "tương lai" mức độ liên quan của các từ. Mô hình từ đó có thể sinh từng từ dựa vào từ trước trong chuỗi mà không dựa vào các từ trong tương lai.



Hình 2.9: Masked Multi-Headed Attention trong Decoder[9]

Công thức:

$$\text{MaskedAttention } (Q, K, V) = \text{softmax } (\left(\frac{QK^T + M}{\sqrt{d_k}}\right)V$$

Trong đó:

- $Q = XW^Q$: truy vấn từ đầu vào
- $K = XW^K$: khóa từ đầu vào hiện tại
- $V = XW^V$: giá trị từ đầu vào hiện tại

- d_k : kích thước của vector truy vấn/khóa (để chuẩn hóa)
- $QK^T \epsilon R^{T \times T}$: ma trận điểm tương đồng giữa các token
- $M \epsilon R^{T \times T}$: ma trận mask tam giác dưới
- $V \epsilon R^{T \times d_v}$: vector giá trị sau khi áp dụng trọng số attention

⇨ Mục tiêu là để ẩn (mask) các thông tin trong tương lai mà mô hình không được phép nhìn thấy khi sinh từ hiện tại.

Cụ thể:

M[i, j] = 0 nếu i ≥ j (token hiện tại và các token trước)

M[i, j] = - ∞ nếu i < j (token tương lai, phải che)

Multi-headed Attention: Lớp multi-head attention trong phần decoder có vai trò quan trọng trong việc giải mã chuỗi đầu ra từ thông tin đã được mã hoá trong encoder. Đầu vào của lớp multi-head attention:

- Query (Q): Được lấy từ đầu ra của lớp trước đó trong Decode
- Key (K) và Value (V): Cả hai đều được lấy từ Encoder. Key và Value từ Encoder là các đại diện của chuỗi đầu vào, cung cấp thông tin về các từ trong đầu vào mà Decoder cần "chú ý" đến.

Cách hoạt động của lớp multi-head attention trong phần decoder tương tự như lớp multi-head attention trong phần encoder. Mục tiêu của lớp này là tìm ra sự liên hệ giữa từng từ trong câu đầu ra và các từ trong câu đầu vào. Tuy nhiên, nó không thể dự đoán từ "tương lai" mà chỉ có thể dựa vào các từ phía trước trong chuỗi đầu ra.

Công thức:

$$\text{MultiHead (Q, K, V)} = \text{Concat (} head_1, \dots, head_h)W^0$$

Trong đó:

- Q, K, V: truy vấn, khóa, giá trị
- W_i^Q, W_i^K, W_i^V : trọng số học được cho từng head
- $head_i$: attention head thứ i
- W^0 : trọng số chiếu output tổng hợp

Cross-attention là cơ chế quan trọng nhất để liên kết thông tin hình ảnh và văn bản. Trong cross-attention:

Công thức:

$$\text{CrossAttention}\ (\ Q_{dec}, K_{enc}, V_{enc}\ ) = \text{softmax}\left(\frac{Q_{dec}K_{enc}^{T}}{\sqrt{d_k}}\right)v_{enc}$$

Trong đó:

- Q_{dec} : truy vấn tại bước hiện tại trong sinh chuỗi
- K_{enc} : khóa biểu diễn đặc trưng hình ảnh
- V_{enc} : giá trị chứa thông tin nội dung hình ảnh
- d_k : dùng để chuẩn hóa khi tính attention score

Mô tả quy trình:

- Đầu vào Encoder là hình ảnh đã được chia thành patch (nếu dùng ViT) hoặc đặc trưng CNN.
- Sau Encoder, ta có ma trận biểu diễn các phần của hình ảnh:

$$F_{img} = [f_1, f_2, \dots, f_N\ ]\ \epsilon R^{N\times d}$$

- Khi Decoder đang ở bước t, token hiện tại tạo ra truy vấn q_t
- Cross-attention dùng q_t để tính attention với tất cả các patch ảnh $\{f_1, \dots ., f_N\ \}$
- Sau đó, kết hợp các đặc trưng hình ảnh (qua v_{enc}) thành biểu diễn phù hợp cho việc sinh tiếp theo

Linear Layer trong Transformer là lớp fully connected, thực hiện phép toán tuyến tính, giúp chuyển các biểu diễn ẩn (hidden representations) thành không gian phù hợp với yêu cầu của bài toán (ví dụ: dự đoán từ tiếp theo, phân loại). Softmax chuyển đầu ra của lớp Linear (logits) thành các xác suất, giúp mô hình lựa chọn từ có xác suất cao nhất (hoặc phân phối xác suất cho các lựa chọn). Cả hai lớp này phối hợp với nhau trong bước cuối của mô hình để tạo ra các dự đoán có xác suất, giúp mô hình Transformer thực hiện các nhiệm vụ như dịch ngữ nghĩa, phân loại văn bản, và nhiều ứng dụng khác.

2.3 Mô hình BLIP trong sinh chú thích hình ảnh

2.3.1 Giới thiệu mô hình BLIP

BLIP (Bootstrapping Language-Image Pre-training) là một mô hình học sâu đa phương thức tiên tiến, được phát triển bởi Junnan Li và các cộng sự tại Salesforce Research. Mô hình này được trình bày trong bài báo khoa học có tiêu đề "BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation", được công bố tại hội nghị quốc tế về học máy ICML 2022.

BLIP được thiết kế nhằm giải quyết các thách thức quan trọng trong lĩnh vực học thị giác - ngôn ngữ (vision-language learning), nơi các hệ thống trí tuệ nhân tạo cần hiểu và tạo ra ngôn ngữ dựa trên thông tin thị giác (hình ảnh). BLIP mang lại những đột phá đáng kể nhờ vào ba điểm cốt lõi:

Tận dụng hiệu quả dữ liệu web quy mô lớn nhưng nhiễu

Dữ liệu hình ảnh và chú thích trên web rất phong phú nhưng thường không đồng nhất và chứa nhiều nhiễu (ví dụ: mô tả không khớp hoàn toàn với hình ảnh, thông tin dư thừa hoặc thiếu chính xác). BLIP giải quyết vấn đề này thông qua một chiến lược tiền huấn luyện dạng bootstrapping – tức là sử dụng một mô hình tự tạo ra các cặp văn bản-hình ảnh có chất lượng cao hơn từ tập dữ liệu gốc. Cách tiếp cận này giúp BLIP tận dụng tốt hơn dữ liệu không được gán nhãn hoàn hảo, làm giảm phụ thuộc vào dữ liệu sạch và tốn kém trong thu thập.

Kiến trúc thống nhất cho nhiều nhiệm vụ

Một điểm mạnh của BLIP là khả năng kết hợp nhiều nhiệm vụ thị giác - ngôn ngữ như:

- Hiểu ngôn ngữ dựa trên hình ảnh (như phân loại câu hỏi - Visual Question Answering).
- Tạo ngôn ngữ từ hình ảnh (Image Captioning).
- Khớp hình ảnh với văn bản (Image-Text Matching).

Thay vì xây dựng mô hình riêng biệt cho từng nhiệm vụ, BLIP cung cấp một kiến trúc hợp nhất, trong đó các thành phần có thể chia sẻ tham số và học được

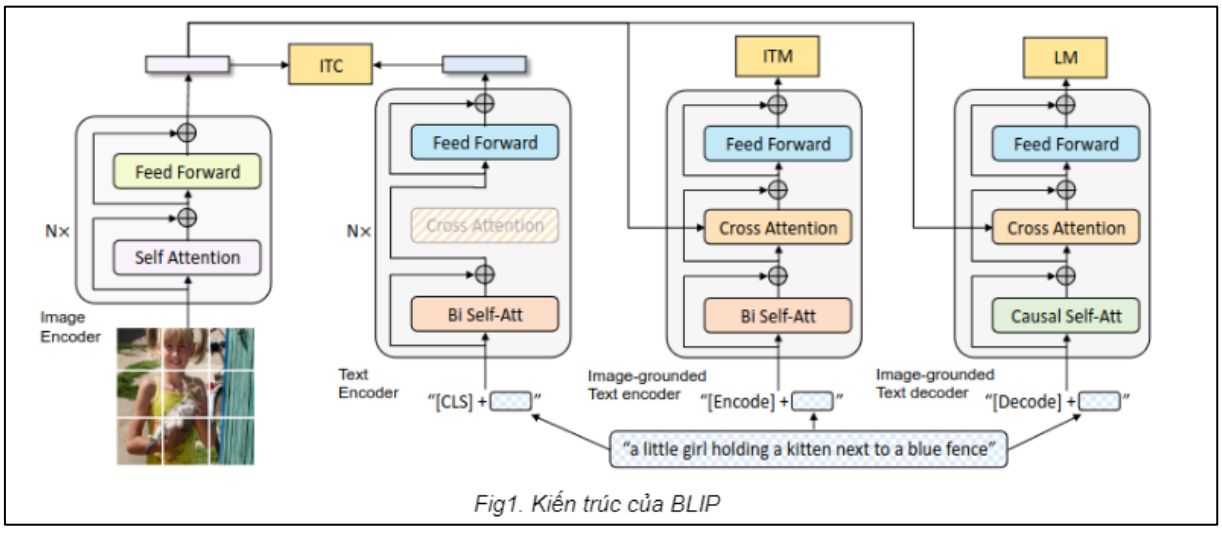
biểu diễn đa phương thức chung. Điều này giúp mô hình có hiệu suất cao hơn và dễ dàng thích nghi với các tác vụ khác nhau.

Khả năng chuyển giao giữa các nhiệm vụ

Nhờ thiết kế thống nhất và cơ chế học biểu diễn hiệu quả, BLIP cho thấy khả năng chuyển giao kiến thức tốt giữa các nhiệm vụ khác nhau. Ví dụ, khi được huấn luyện trên nhiệm vụ mô tả hình ảnh, mô hình vẫn có thể đạt hiệu suất tốt trên nhiệm vụ trả lời câu hỏi về hình ảnh hoặc ngược lại, mà không cần huấn luyện lại từ đầu.

2.3.1 Kiến trúc mô hình BLIP

Mô hình BLIP (Bootstrapping Language-Image Pre-training) được thiết kế với một kiến trúc thống nhất, cho phép thực hiện hiệu quả cả hai nhiệm vụ hiểu (understanding) và sinh (generation) trong lĩnh vực học đa phương thức thị giác-ngôn ngữ (vision-language learning). Kiến trúc của BLIP bao gồm bốn thành phần chính: Image Encoder, Text Encoder, Image-grounded Text Encoder, và Image-grounded Text Decoder, mỗi thành phần đảm nhận một vai trò riêng biệt nhưng có sự kết nối chặt chẽ nhằm khai thác tối đa thông tin từ cả hình ảnh và văn bản.



Hình 2.10: Kiến trúc mô hình BLIP[6]

Dựa trên hình ảnh mô hình gồm 4 thành phần chính:

Bộ mã hóa hình ảnh (Image Encoder)

Bộ mã hóa hình ảnh là thành phần đầu tiên trong kiến trúc BLIP, chịu trách nhiệm trích xuất đặc trưng từ ảnh đầu vào. BLIP sử dụng mô hình Vision Transformer (ViT) làm backbone cho bộ mã hóa hình ảnh nhờ khả năng học các mối quan hệ không gian toàn cục hiệu quả qua cơ chế tự chú ý (Self-Attention).

Cấu trúc tổng quát:

- Kiến trúc cốt lõi: Vision Transformer (ViT).
- Thành phần lặp: Gồm N khối Transformer, mỗi khối bao gồm:
    - Multi-head Self Attention (MSA): Học mối quan hệ toàn cục giữa các patch ảnh.
    - Feed Forward Network (MLP): Tăng khả năng biểu diễn phi tuyến.
    - Layer Normalization (LN) và Residual Connection để tăng độ ổn định trong huấn luyện.

Quy trình xử lý chi tiết:

Bước 1: Phân chia hình ảnh thành các patch

Hình ảnh đầu vào có kích thước $I \epsilon R^{H \times W \times C}$, được chia thành lưới các patch không chồng lấn, mỗi patch có kích thước $P \times P$, dẫn đến tổng cộng $N = \frac{HW}{P^2}$ patch.

$$I \rightarrow \{p_1, p_2, \dots, p_N\}, p_i \in R^{P^2.C}$$

Trong đó:

- *H, W*: chiều cao và chiều rộng của ảnh.
- *C*: số kênh màu (thường là 3).
- *P*: kích thước một patch vuông.
- p_i: patch ảnh thứ *i* sau khi làm phẳng (flatten).

Bước 2: Chiếu không gian và thêm token đặc biệt [CLS]

Mỗi patch được ánh xạ sang không gian ẩn thông qua phép chiếu tuyến tính với ma trận $E \epsilon R^{D \times (P^2.C)}$. Sau đó, chèn token đặc biệt [CLS] vào đầu chuỗi để tổng hợp thông tin toàn ảnh, đồng thời thêm embedding vị trí E_{pos}

Công thức:

$$z_0 = [x_{cls}; E.p_1; E.p_2; \dots; E.p_N] + E_{pos}$$

Trong đó:

- x_{cls}: vector khởi tạo học được cho token [CLS].
- E_{pos}: vị trí tương ứng từng patch

Bước 3: Truyền qua các khối Transformer

Với mỗi khối Transformer (lặp lại N lần), áp dụng tuần tự Self-Attention và Feed Forward với residual connections

Công thức:

$$z'_l = \text{MSA}(\text{LN}(z_{l-1})) + z_{l-1}$$
$$z_l = \text{MLP}(\text{LN}(z'_l)) + z'_l$$

Trong đó:

- z_{l-1}: đầu vào của khối thứ l.
- z'_l: đầu ra sau Self-Attention.
- LN: chuẩn hóa từng lớp.
- MSA: Multi-head Self-Attention.
- MLP: mạng nhiều lớp phi tuyến (thường gồm hai lớp fully-connected kèm ReLU hoặc GELU).

Bước 4: Xuất ra đặc trưng hình ảnh

Sau N khối Transformer, đầu ra là chuỗi đặc trưng hình ảnh:

Công thức:

$$V = z_L = \{v_{cls}, v_1, v_2, \dots, v_N\}$$

Trong đó:

- v_{cls}: đại diện cho toàn bộ hình ảnh – được sử dụng cho các nhiệm vụ phân loại như ITC và ITM.
- $\{v_1, \dots, v_N\}$: đặc trưng cục bộ từng patch, sử dụng cho các tác vụ chi tiết như sinh văn bản (captioning).

Bộ mã hóa văn bản (Text Encoder)

Bộ mã hóa văn bản trong mô hình BLIP đóng vai trò quan trọng trong việc biểu diễn ngữ nghĩa của văn bản đầu vào và thiết lập mối liên kết giữa văn bản và

hình ảnh. BLIP sử dụng kiến trúc Transformer hai chiều (Bidirectional Transformer) làm nền tảng, cho phép mô hình học ngữ cảnh đầy đủ của từng từ trong câu, đồng thời tích hợp thông tin từ hình ảnh thông qua Cross-Attention với đầu ra của Image Encoder.

Cấu trúc tổng quan:

- Embedding lớp từ: chuyển đổi mỗi từ trong câu thành vector.
- Bidirectional Self-Attention: cho phép mỗi từ tương tác với toàn bộ các từ khác trong chuỗi.
- Cross-Attention: kết nối thông tin giữa đặc trưng văn bản và đặc trưng hình ảnh.
- Feed Forward Network (FFN): tầng phi tuyến giúp tăng khả năng biểu diễn.

Chi tiết các bước xử lý:

Bước 1: Embedding văn bản

Văn bản đầu vào được biểu diễn thành chuỗi các token $w_1, w_2, \dots, w_M$, sau đó được ánh xạ sang không gian ẩn bằng embedding từ E_{word}, kết hợp với token [CLS] và embedding vị trí E_{pos}.

Công thức:

$$t_0 = [t_{cls}; E_{word}.w_1; E_{word}.w_2; \dots; E_{word}.w_M] + E_{pos}$$

Trong đó:

- t_{cls}: token đặc biệt tổng hợp thông tin toàn chuỗi.
- E_{word}: ma trận embedding từ vựng.
- E_{pos}: embedding vị trí cho mỗi token trong chuỗi.

Bước 2: Bidirectional Self-Attention

Trong mỗi khối Transformer, đầu tiên là cơ chế Self-Attention hai chiều, nơi mỗi token có thể tương tác với tất cả các token còn lại để học ngữ cảnh toàn diện

Công thức:

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

Trong đó:

- $Q = W_Q.t_{l-1}, K = W_K.t_{l-1}, V = W_V.t_{l-1}$
- W_Q, W_K, W_V: các ma trận chiếu học được.
- t_{l-1}: đầu vào từ tầng trước đó

Bước 3: Cross-Attention với đặc trưng hình ảnh

Sau bước self-attention, mô hình thực hiện Cross-Attention để đưa thông tin từ ảnh vào văn bản. Cụ thể, các đặc trưng hình ảnh V (đầu ra của Image Encoder) được sử dụng làm key và value

Công thức:

$$CrossAttention(Q_t, K_v, V_v) = softmax\left(\frac{Q_t K_v^T}{\sqrt{d_k}}\right)V_v$$

Trong đó:

- Q_t: truy vấn từ văn bản.
- K_v, V_v: đặc trưng từ ảnh.
- Kết quả: mô hình học được sự liên kết giữa nội dung ảnh và từng từ trong câu.

Bước 4: Feed Forward Network

Mỗi khối Transformer kết thúc bằng một mạng phi tuyến hai lớp (FFN) giúp tăng khả năng học các biểu diễn phức tạp

Công thức:

$$FFN(x) = GELU(xW_1 + b_1)W_2 + b_2$$

Trong đó:

- W_1, W_2: trọng số tuyến tính.
- b_1, b_2: hệ số bias.
- GELU: hàm kích hoạt Gaussian Error Linear Unit, hoạt động mượt hơn ReLU.

Bước 5: Kết hợp các tầng với Residual và Layer Normalization

Trong mỗi khối Transformer, các thành phần trên được kết hợp tuần tự với residual connection và LayerNorm để ổn định quá trình huấn luyện

Công thức:

$$t_l' = LayerNorm\ (SelfAttention(t_{l-1}) + t_{l-1})$$

$$t_l'' = LayerNorm\ (CrossAttention(t_l', V) + t_l')$$

$$t_1 = LayerNorm\ (FFN(t_l'') + t_l'')$$

Trong đó:

- t_l: đầu ra cuối cùng của khối thứ l.
- Mỗi bước đều có nhánh shortcut để giữ lại thông tin gốc

1. Bộ mã hóa văn bản dựa trên hình ảnh (Image-grounded Text Encoder)

Cấu trúc và chức năng

Bộ mã hóa văn bản dựa trên hình ảnh (Image-grounded Text Encoder) là một thành phần cốt lõi trong các mô hình đa phương thức như BLIP, đóng vai trò mã hóa văn bản dưới ngữ cảnh hình ảnh. Mục tiêu của encoder này không chỉ là hiểu nội dung văn bản, mà còn liên kết và đồng bộ hóa thông tin từ hình ảnh đi kèm để xác định xem văn bản có mô tả chính xác nội dung hình ảnh hay không. Đây là một bước quan trọng trong bài toán Image-Text Matching (ITM).

Cấu trúc:

Bộ mã hóa này có kiến trúc gần giống với Text Encoder thông thường (Transformer-based), nhưng được mở rộng để xử lý thông tin hình ảnh:

- Bi-directional Self-Attention (Bi-Self-Att):

Cho phép các token văn bản trao đổi thông tin theo cả hai chiều để nắm bắt ngữ cảnh toàn cục.

- Cross-Attention Layers:

Các lớp này giúp văn bản "chú ý" đến các đặc trưng hình ảnh (image features) đầu vào. Điều này cho phép các token văn bản được điều chỉnh dựa trên nội dung hình ảnh. Đặc trưng hình ảnh thường được trích xuất bởi Image Encoder (như ViT Vision Transformer) từ trước.

- Feed-Forward Networks:

Các lớp tuyến tính phi tuyến tính (MLPs) giúp tăng tính biểu diễn và phi tuyến tính của mạng.

Kiểm soát bởi ITM Module:

- Trong giai đoạn huấn luyện, ITM module chỉ kích hoạt cross-attention giữa đặc trưng hình ảnh và văn bản khi ảnh và văn bản khớp nhau (positive pair), để mô hình học cách tích hợp thông tin hiệu quả và tránh nhiễu từ cặp không khớp.

Đặc điểm kỹ thuật

- Nhiệm vụ chính:

Tính toán mức độ phù hợp (matching score) giữa hình ảnh và văn bản, xem chúng có mô tả cùng một nội dung hay không.

- Chia sẻ tham số:

Một số tham số, đặc biệt là ở các lớp attention hoặc embedding, có thể được chia sẻ với Text Encoder nhằm giảm số lượng tham số và tận dụng kiến thức học được từ các nhiệm vụ đơn phương (văn bản đơn lẻ).

- Đầu ra:

Vector đại diện cuối cùng (thường là token [CLS]) chứa thông tin tổng hợp từ cả văn bản và đặc trưng hình ảnh. Vector này sẽ được sử dụng để tính toán điểm phù hợp thông qua một mạng tuyến tính và hàm sigmoid.

Công thức toán học

Mục tiêu: Tính điểm phù hợp $S_{ITM}(I,T)$ giữa một hình ảnh I và một đoạn văn bản T. Công thức được định nghĩa như sau:

$$S_{ITM}(I,T) = \sigma(W.t_{cls})$$

Trong đó:

- $t_{cls} \epsilon R^d$

Đây là vector đầu ra của token [CLS] từ lớp cuối cùng của Image-grounded Text Encoder. Nó là một biểu diễn kết hợp giữa văn bản và hình ảnh. Token này được dùng làm đại diện cho toàn bộ đầu vào.

- $W \in R^{1\times d}$

Là ma trận tham số học được trong quá trình huấn luyện, đóng vai trò như một lớp tuyến tính chiếu t_{cls} thành một giá trị scalar.

- σ

Là hàm sigmoid:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

dùng để ánh xạ giá trị đầu ra vào khoảng [0, 1], thể hiện xác suất phù hợp giữa hình ảnh và văn bản.

2. Bộ giải mã văn bản dựa trên hình ảnh (Image-grounded Text Decoder)

Cấu trúc và chức năng

Bộ giải mã văn bản dựa trên hình ảnh (Image-grounded Text Decoder) là một bộ giải mã dạng Transformer một chiều, được thiết kế đặc biệt để thực hiện nhiệm vụ sinh mô tả văn bản từ hình ảnh – tức là sinh ra câu văn tự nhiên diễn tả nội dung hình ảnh đầu vào.

Chức năng chính:

- Nhận vào một đoạn văn bản một phần (ví dụ: một vài từ đầu tiên hoặc token <BOS> bắt đầu câu).
- Dưới sự điều khiển của Language Modeling (LM) module, nó thực hiện quá trình tự sinh tiếp các token còn lại từng bước một.
- Trong quá trình sinh, nó liên kết với đặc trưng hình ảnh để tạo ra văn bản sát nghĩa, chính xác và giàu ngữ nghĩa.

Cấu trúc bao gồm 3 loại lớp chính:

- Feed-Forward Networks (FFN):

Các lớp phi tuyến giúp tăng năng lực biểu diễn của mô hình, thường gồm hai lớp tuyến tính với hàm kích hoạt ReLU hoặc GELU ở giữa.

- Causal Self-Attention:

Khác với self-attention thông thường, causal attention chỉ cho phép mỗi token nhìn thấy các token trước nó (không nhìn thấy tương lai). Điều này đảm bảo tính chất tự hồi quy cần thiết cho quá trình sinh văn bản tuần tự.

- Cross-Attention:

Các lớp này cho phép mô hình "chú ý" tới các đặc trưng hình ảnh từ Image Encoder, giúp bộ giải mã sinh ra các từ có liên hệ trực tiếp đến nội dung hình ảnh.

Đặc điểm kỹ thuật

- Kiến trúc nền tảng:

Dựa trên các mô hình sinh văn bản mạnh mẽ như GPT (Generative Pre-trained Transformer) hoặc các kiến trúc Transformer Decoder khác.

- Đầu vào:

Một phần chuỗi token văn bản (ví dụ: [<BOS>, "A", "man"])

Đặc trưng hình ảnh được trích xuất từ Image Encoder

- Chiều sinh:

Tự hồi quy từ trái sang phải, tức là mỗi token được sinh dựa trên tất cả các token trước đó và thông tin hình ảnh.

- Đầu ra:

Một phân phối xác suất trên toàn bộ từ vựng tại mỗi bước, dùng để chọn ra token tiếp theo. Mô hình tiếp tục sinh cho đến khi gặp token kết thúc (<EOS>) hoặc đạt đến độ dài tối đa.

Công thức toán học

(1) Causal Self-Attention:

Causal self-attention đảm bảo rằng mỗi token chỉ được "nhìn thấy" các token phía trước nó trong chuỗi.

Công thức:

$$CausalAttention(Q, K, V) = softmax\left(\frac{QK^T + M}{\sqrt{d_k}}\right)V$$

Trong đó:

- Q, K, V: là các ma trận truy vấn (Query), khóa (Key) và giá trị (Value) được tính từ các token đầu vào
- d_k: là chiều không gian của vector khóa (key vector), dùng để chuẩn hóa.
- M: là ma trận mặt nạ (mask) định nghĩa như sau:

$$M_j = 0, nếu\ i \geq j$$

$$M_j = -\infty, nếu\ i < j$$

Điều này đảm bảo rằng token ở vị trí iii chỉ có thể "chú ý" đến các token trước hoặc tại chính nó, không truy cập tương lai.

(2) Tính xác suất sinh từ tiếp theo (Auto-regressive Text Generation):

Tại mỗi thời điểm iii, mô hình tính xác suất để sinh token wiw_iwi dựa trên các token trước đó và đặc trưng hình ảnh:

$$P(w_i|w_{<i}, I) = softmax(W_{LM}.h_i)$$

Trong đó:

- $w_{<i}$: là chuỗi token từ đầu đến i – 1
- h_i: là biểu diễn ẩn tại vị trí iii, thu được sau khi qua các lớp attention và feed-forward.
- $W_{LM} \epsilon R^{|V| \times d}$: là ma trận ánh xạ từ không gian ẩn sang không gian từ vựng |V|.
- *softmax*: biến đổi vector đầu ra thành phân phối xác suất trên toàn bộ từ vựng.

⇨ Dựa trên phân phối này, mô hình có thể lấy token có xác suất cao nhất hoặc sampling ngẫu nhiên để sinh câu tiếp theo.

Tương tác đa phương thức (Cơ chế Cross-Modal Interaction)

Tương tác đa phương thức là yếu tố cốt lõi giúp mô hình BLIP kết hợp thông tin từ hai nguồn dữ liệu khác nhau: hình ảnh và văn bản. Mục tiêu là để biểu diễn văn bản không chỉ dựa vào ngữ cảnh ngôn ngữ, mà còn được định hướng bởi đặc trưng hình ảnh tương ứng, giúp tăng cường độ chính xác và tính ngữ nghĩa của mô tả sinh ra.

Cơ chế chú ý chéo (Cross-Attention Mechanism)

Cơ chế chú ý chéo (Cross-Attention Mechanism) là cơ chế quan trọng cho phép biểu diễn từ một modality (văn bản) truy vấn và thu nhận thông tin từ biểu diễn của modality còn lại (hình ảnh). Trong BLIP, Cross-Attention được sử dụng ở nhiều vị trí:

- Trong Text Encoder: để biểu diễn văn bản có thể hấp thụ thông tin từ hình ảnh.
- Trong Image-grounded Text Encoder: để tích hợp đặc trưng hình ảnh vào trong quá trình mã hóa văn bản.
- Trong Image-grounded Text Decoder: để mô hình có thể sinh văn bản mô tả dựa trên hình ảnh đầu vào.

Công thức toán học của Cross-Attention**:**

$$CrossAttention(Q_T, K_v, V_v) = softmax\left(\frac{Q_t K_v^T}{\sqrt{d_k}}\right)V_v$$

Trong đó:

- $Q_t = W_Q.t$: ma trận truy vấn (Query), được tính từ biểu diễn văn bản t
- $K_v = W_K.V$: ma trận khóa (Key), tính từ biểu diễn hình ảnh V
- $V_v = W_V.V$: ma trận giá trị (Value), cũng từ biểu diễn hình ảnh
- d_k: kích thước chiều của vector khóa, dùng để chuẩn hóa ma trận điểm tương đồng.

Cross-Attention về bản chất cho phép mỗi token trong chuỗi văn bản "nhìn thấy" toàn bộ thông tin của ảnh để làm rõ nghĩa hoặc bổ sung bối cảnh, từ đó cải thiện chất lượng hiểu hoặc sinh văn bản.

Để tăng cường khả năng tương tác giữa hai modality (văn bản và hình ảnh), mô hình sử dụng nhiều tầng Cross-Attention liên tiếp kết hợp với các khối Self-Attention và Feed-Forward Network (FFN). Việc kết hợp này cho phép thông tin được lan truyền và hòa trộn sâu hơn giữa các tầng, thay vì chỉ tích hợp một lần.

Quy trình kết hợp trong một khối Transformer (dành cho văn bản có điều kiện hình ảnh):

- Self-Attention trong văn bản:

Token trong chuỗi văn bản tương tác với nhau để học ngữ cảnh thuần ngôn ngữ:

$$t_l' = LayerNorm(SelfAttention(t_{l-1}) + t_{l-1})$$

- Cross-Attention giữa văn bản và hình ảnh:

Văn bản tiếp nhận thông tin từ đặc trưng hình ảnh:

$$t_l'' = LayerNorm(CrossAttention(t_l', V) + t_l')$$

- Feed-Forward để học phi tuyến:

$$t_l = LayerNorm(FFN(t_l'') + t_l'')$$

Vai trò của tương tác đa phương thức trong BLIP

- Cải thiện khả năng sinh mô tả hình ảnh: thông tin từ hình ảnh được đưa vào quá trình sinh văn bản, giúp mô tả chính xác nội dung trực quan.
- Tăng cường tính liên kết giữa hai loại dữ liệu: thông qua việc kết hợp thông tin nhiều tầng, giúp mô hình học được các mối quan hệ phức tạp giữa hình ảnh và ngôn ngữ.
- Linh hoạt trong ứng dụng: như trả lời câu hỏi về hình ảnh (VQA), sinh chú thích ảnh (image captioning), truy xuất hình ảnh theo mô tả (image-text retrieval), v.v.

2.3.2 Biểu diễn đặc trưng trong BLIP

Biểu diễn đặc trưng (Feature Representations) là trung tâm của quá trình xử lý trong mô hình BLIP. Các biểu diễn này mang thông tin từ ảnh và văn bản dưới dạng vector, giúp mô hình hiểu, liên kết và sinh ngôn ngữ dựa trên nội dung hình ảnh. BLIP xây dựng và sử dụng các loại biểu diễn sau:

Biểu diễn hình ảnh

BLIP sử dụng ViT (Vision Transformer) để trích xuất đặc trưng từ ảnh đầu vào. Việc mã hóa ảnh được thực hiện thông qua các patch ảnh, mỗi patch được biểu diễn như một token trong transformer.

Global Representation (Biểu diễn toàn cục)

- Ký hiệu: v_{cls}

- Đây là biểu diễn của token đặc biệt [CLS] trong đầu ra của Vision Transformer.
- Mã hóa thông tin tổng thể của toàn bộ ảnh, đại diện cho ngữ nghĩa chung nhất.
- Được dùng cho các nhiệm vụ cần hiểu khái quát nội dung ảnh, chẳng hạn như đối sánh ảnh - văn bản (image-text retrieval).

Local Representations (Biểu diễn cục bộ)

- Ký hiệu: $\{v_1, v_2, \dots, v_N\}$
- Là tập các biểu diễn tương ứng với từng patch nhỏ trong ảnh.
- Mỗi v_i chứa thông tin về một phần cụ thể của hình ảnh, hữu ích cho các tác vụ yêu cầu chi tiết (như sinh mô tả chi tiết).

Projected Image Representation (Biểu diễn ảnh đã chiếu)

- Ký hiệu: $h_I = f_I(v_{cls})$
- Đây là biểu diễn toàn cục v_{cls} sau khi được chiếu (project) qua một mạng tuyến tính f_I để chuyển sang không gian phù hợp cho nhiệm vụ Contrastive Learning (học tương phản).
- Giúp biểu diễn ảnh có thể so sánh và tính độ tương đồng với biểu diễn văn bản.

Biểu diễn văn bản

Văn bản đầu vào được xử lý bởi Text Encoder trong mô hình BLIP (thường là BERT hoặc một biến thể Transformer khác). Mỗi từ (token) trong chuỗi được ánh xạ thành vector thông qua embedding, sau đó trải qua nhiều lớp transformer.

Global Representation (Biểu diễn toàn cục)

- Ký hiệu: t_{cls}
- Là đầu ra của token [CLS] từ Text Encoder, tương tự như v_{cls} ở phía ảnh.
- Biểu diễn ý nghĩa tổng thể của toàn bộ câu.
- Sử dụng trong các tác vụ như truy hồi (retrieval) hoặc phân loại.

Token-level Representations (Biểu diễn mức từ)

- Ký hiệu: $\{t_1, t_2, \dots, t_M\}$

- Biểu diễn chi tiết cho từng token (từ, dấu câu, v.v.)
- Cần thiết cho các nhiệm vụ như sinh mô tả (captioning), trả lời câu hỏi (VQA), vì chúng yêu cầu mô hình xử lý và hiểu cấu trúc ngôn ngữ.

Projected Text Representation (Biểu diễn văn bản đã chiếu)

- Ký hiệu: $h_T = f_T(t_{cls})$
- Tương tự ảnh, t_{cls} được đưa qua một mạng chiếu f_T để đưa về không gian tương thích với ảnh, phục vụ cho contrastive learning.

Biểu diễn kết hợp (Multimodal Representations)

Chuyển đổi hình ảnh

- Đây là biểu diễn thu được từ Image-grounded Text Encoder sau khi văn bản đã tương tác với hình ảnh qua các tầng Cross-Attention.
- Ký hiệu: vẫn dùng t_{cls}, nhưng lúc này đã mang thông tin liên kết giữa văn bản và hình ảnh.
- Được dùng trong các tác vụ yêu cầu hiểu ngữ cảnh cả hai modality, như matching hoặc VQA.

Biểu diễn cho sinh ngôn ngữ

- Là các trạng thái ẩn tại mỗi bước thời gian trong Image-grounded Text Decoder.
- Các biểu diễn này không chỉ mang thông tin ngôn ngữ (ngữ pháp, cú pháp), mà còn có thông tin từ hình ảnh đã ảnh hưởng vào từng bước sinh thông qua Cross-Attention.
- Dùng để sinh câu mô tả ảnh (image captioning), từng từ được sinh ra dựa trên biểu diễn tại thời điểm đó.

2.3.3 Luồng dữ liệu trong BLIP

BLIP (Bootstrapping Language-Image Pre-training) là một mô hình đa phương thức, xử lý và học từ cả hình ảnh và văn bản. Tùy vào nhiệm vụ huấn luyện, luồng dữ liệu sẽ thay đổi để phù hợp với mục tiêu: học tương phản (ITC),

học phân biệt (ITM), hoặc sinh ngôn ngữ (LM). Ba luồng dữ liệu chính trong BLIP bao gồm:

Luồng dữ liệu cho nhiệm vụ ITC (Image-Text Contrastive Learning)

Mục tiêu: Học ánh xạ chung giữa hình ảnh và văn bản vào cùng một không gian embedding để các cặp ảnh-văn bản đúng có khoảng cách gần nhau, còn các cặp sai thì xa nhau.

Các bước:

1. Mã hóa hình ảnh:

- Hình ảnh đầu vào được chia thành các patch và đưa vào Image Encoder (ViT).
- Kết quả là tập biểu diễn:

$$V = \{v_{cls}, v_1, \dots, v_N\}$$

Trong đó:

- v_{cls}: biểu diễn toàn cục (global) của hình ảnh (từ token [CLS]).
- $\{v_1, \dots, v_N\}$: biểu diễn cục bộ (local) cho từng patch.

2. Mã hóa văn bản:

- Văn bản được đưa qua Text Encoder (Transformer).
- Kết quả là:

$$T = \{t_{cls}, t_1, \dots, t_M\}$$

Trong đó:

- t_{cls}: biểu diễn toàn cục của văn bản (từ token [CLS]).
- $\{t_1, \dots, t_M\}$: biểu diễn cho từng token.

3. Chiếu vào không gian nhúng chung

Biểu diễn toàn cục của ảnh và văn bản được chiếu (projected) qua hai hàm tuyến tính f_I và f_T để ánh xạ về cùng không gian:

$$h_I = f_I(v_{cls}), h_T = f_T(t_{cls})$$

Trong đó:

- h_I: vector nhúng của ảnh.
- h_T: vector nhúng của văn bản.

4. Tính độ tương đồng ảnh-văn bản

Sử dụng cosine similarity để đo độ tương đồng giữa ảnh và văn bản:

$$s_{ITC}(I,T) = \frac{h_I^T h_T}{\|h_I\| \cdot \|h_T\|}$$

Trong đó:

- $h_I^T h_T$: tích vô hướng giữa hai vector nhúng.
- $\|h_I\|$: chuẩn (độ dài) của vector h_I
- $\|h_T\|$: chuẩn của vector h_T
- Giá trị $s_{ITC}(I,T) \epsilon [-1,1]$, càng gần 1 tức là cặp ảnh-văn bản càng giống nhau.

Luồng dữ liệu cho nhiệm vụ ITM (Image-Text Matching)

Mục tiêu:

Dự đoán liệu ảnh và văn bản có khớp nội dung hay không. Đây là một bài toán nhị phân (binary classification).

Các bước:

1. Mã hóa hình ảnh

Tương tự như ITC:

$$V = \{v_{cls}, v_1, \dots, v_N\}$$

2. Kết hợp đặc trưng ảnh vào văn bản

Ảnh và văn bản được đưa vào một bộ mã hóa gọi là Image-grounded Text Encoder. Tại đây:

- Văn bản được xem là chuỗi chính.
- Hình ảnh đóng vai trò tham chiếu qua Cross-Attention.
- Thông tin ảnh được tích hợp vào các biểu diễn văn bản.

3. Lấy token đầu ra [CLS]

Sau khi encode, lấy token đầu tiên của văn bản:

$$t_{cls}$$

⇨ biểu diễn toàn bộ thông tin từ ảnh và văn bản đã tích hợp.

4. Dự đoán xác suất cặp khớp

Đầu ra được đưa qua một lớp tuyến tính và hàm sigmoid:

$$s_{ITM}(I,T) = \sigma(W.t_{cls})$$

Trong đó:

- W: ma trận trọng số học được
- σ: hàm sigmoid, để chuyển đầu ra về khoảng [0, 1], đại diện cho xác suất.
- $s_{ITM}(I,T)\epsilon[0,1]$: càng gần 1 → ảnh và văn bản càng khớp.

Luồng dữ liệu cho nhiệm vụ LM (Language Modeling / Captioning)

Mục tiêu:

Sinh mô tả (caption) cho hình ảnh — đây là bài toán dịch ngược từ ảnh sang ngôn ngữ tự nhiên.

Các bước:

1. Mã hóa hình ảnh

Giống các nhiệm vụ trên:

$$V = \{v_{cls}, v_1, \dots, v_N\}$$

2. Nhập chuỗi đầu vào ban đầu

- Chuỗi token đầu tiên là:

$$w_{<i} = \{[BOS], w_1, \dots, w_{i-1}\}$$

Trong đó:

- *[BOS]:* Beginning Of Sentence — token bắt đầu.
- Các token còn lại là phần mô tả đã sinh ra.

3. Image-grounded Text Decoder

Bộ giải mã (decoder) nhận đầu vào gồm:

- Chuỗi token đã sinh.
- Đặc trưng ảnh từ image encoder.

Dự đoán token tiếp theo:

$$P(w_i|w_{<i} < I)$$

4. Sinh chuỗi hoàn chỉnh

- Mỗi token dự đoán được chèn vào đầu vào cho bước tiếp theo.
- Quá trình lặp lại đến khi gặp token kết thúc và đạt độ dài tối đa

CHƯƠNG III: THỰC NGHIỆM

3.1 Dữ liệu thực nghiệm

Quá trình thực nghiệm được em được tiến hành trên một bộ dữ liệu riêng để phục vụ cho việc sinh chú thích hình ảnh du lịch Hà Nội. Quy trình thu thập và xử lý dữ liệu được thực hiện như sau:

Hình ảnh các địa điểm du lịch nổi tiếng tại Hà Nội:

- Nguồn: Google Images, Unsplash, Wikimedia, trang web du lịch chính thống, và chụp thực tế.
- Số lượng: 416 ảnh cho huấn luyện, 20 ảnh cho kiểm thử.
- Yêu cầu:
    - Mỗi địa điểm sẽ có 4 ảnh.
    - Hình ảnh chất lượng cao.
    - Rõ địa điểm (tránh ảnh mờ, chứa quá nhiều người, không rõ chủ thể).
    - Ảnh phải có đuôi .jpg.
    - Tên ảnh phải đặt theo tiêu chuẩn, ví dụ: vuonQuocGiaBaVi1.jpg, baoTangPhuNuVietNam1.jpg.

Chú thích hình ảnh: Với mỗi hình ảnh sẽ có một đoạn mô tả bằng tiếng anh. Trong đoạn mô tả phải có tên địa điểm đó.

Dưới đây là một mẫu dữ liệu bảo gồm ảnh Văn miếu quốc tử giám và mô tả của ảnh.

Hình 3.1: Ảnh Văn Miếu Quốc Tử Giám

- Mô tả: “This is the first university in Viet Nam - Van Mieu Quoc Tu Giam”.

Tổ chức dữ liệu trong Excel

Trong Excel sẽ có 2 cột một cột image và một cột text. Với mỗi ảnh sẽ có một đoạn văn bản mô tả bằng tiếng anh khác nhau.

image	text
vanmieu1.jpg	This is the first university in Viet Nam - Van Mieu Quoc Tu Giam
vanmieu2.jpg	This is Van Mieu Quoc Tu Giam in the night
vanmieu3.jpg	A traditional temple gateway with a red-tiled roof and vibrant banners
vanmieu4.jpg	People are walking around the Van Mieu Quoc Tu Giam
congVienThienDuongBaoSon1.jpg	Entrance of Bao Son Paradise Theme Park in Vietnam, showcasing traditional architecture with a modern touch
congVienThienDuongBaoSon2.jpg	Visitors gathering at the entrance of Bao Son Paradise, a popular theme park in Vietnam blending tradition and modern entertai
congVienThienDuongBaoSon3.jpg	The vibrant entrance to Aladdin World at Bao Son Paradise, where fantasy meets adventure in a magical setting
congVienThienDuongBaoSon4.jpg	A lively scene at the entrance of Bao Son Paradise, where young visitors embark on an exciting journey of discovery and fun
hoTrucBach1.jpg	Truc Bach Lake in the sunny and cloudless sky
hoTrucBach2.jpg	A view from near the shore of Truc Bach Lake
hoTrucBach3.jpg	A morning between two trees and a garden on Truc Bach lake
hoTrucBach4.jpg	Truc Bach Lake has clear blue water in harmony with cycling activities
vuonQuocGiaBaVi1.jpg	The majestic natural scenery of Ba Vi National Park, where lush green mountains blend with the fresh
vuonQuocGiaBaVi2.jpg	In the heart of Ba Vi National Park, nature's grandeur unfolds in every direction
vuonQuocGiaBaVi3.jpg	A tranquil road winds through the autumn foliage of Ba Vi, a symphony of colors and light
vuonQuocGiaBaVi4.jpg	At Ba Vi, the air is crisp, the views are endless, and the soul finds solace in the embrace of nature

Hình 3.2: Ảnh dữ liệu Excel

3.3 Huấn luyện mô hình

Cấu Hình Huấn Luyện

Trong nghiên cứu này, em tiến hành fine-tuning mô hình BLIP pre-trained để tối ưu hóa khả năng sinh chú thích cho hình ảnh du lịch Hà Nội. Quá trình huấn luyện được thực hiện với các thông số cấu hình phù hợp nhằm đạt được hiệu quả tối ưu.

Khởi tạo mô hình và dữ liệu

Em sử dụng mô hình pre-trained "Salesforce/blip-image-captioning-base" làm nền tảng cho quá trình fine-tuning. Mô hình này đã được huấn luyện trên tập dữ liệu lớn và có khả năng nhận diện các đối tượng, cảnh vật cơ bản trong hình ảnh. Việc fine-tuning giúp mô hình có khả năng thích ứng với đặc trưng của hình ảnh du lịch Hà Nội và tạo ra các chú thích phù hợp.

```
from transformers import AutoProcessor, BlipForConditionalGeneration

processor = AutoProcessor.from_pretrained("Salesforce/blip-image-captioning-base")
model = BlipForConditionalGeneration.from_pretrained("Salesforce/blip-image-captioning-base")
```

Dữ liệu được chuẩn bị thông qua lớp ImageCaptioningDataset được thiết kế riêng, đảm bảo xử lý đúng định dạng và cung cấp các thông tin cần thiết cho quá trình huấn luyện:

```
from torch.utils.data import Dataset, DataLoader

class ImageCaptioningDataset(Dataset):
    def __init__(self, dataset, processor):
        self.dataset = dataset
        self.processor = processor

    def __len__(self):
        return len(self.dataset)

    def __getitem__(self, idx):
        item = self.dataset[idx]
        encoding = self.processor(images=item["image"], text=item["text"], padding="max_length", return_tensors="pt")
        # remove batch dimension
        encoding = {k:v.squeeze() for k,v in encoding.items()}
        return encoding
```

Quá Trình Huấn Luyện

Trong nghiên cứu này, em thực hiện quá trình fine-tuning mô hình BLIP để tạo ra hệ thống sinh chú thích hình ảnh du lịch Hà Nội. Quá trình huấn luyện được thiết kế như sau:

Môi trường huấn luyện

Quá trình huấn luyện được thực hiện trên môi trường máy tính có hỗ trợ GPU để tăng tốc độ xử lý. Hệ thống tự động phát hiện và sử dụng GPU nếu có sẵn, trong trường hợp ngược lại sẽ sử dụng CPU:

```
import torch

optimizer = torch.optim.AdamW(model.parameters(), lr=5e-5)

device = "cuda" if torch.cuda.is_available() else "cpu"
model.to(device)
```

1. Cấu hình tối ưu hóa

Em sử dụng bộ tối ưu hóa AdamW với learning rate 5e-5. AdamW là một biến thể của thuật toán Adam kết hợp với kỹ thuật weight decay, giúp cải thiện khả năng tổng quát hóa của mô hình:

```
optimizer = torch.optim.AdamW(model.parameters(), lr=5e-5)
```

2. Vòng lặp huấn luyện

Quá trình huấn luyện được thực hiện qua 10 epochs. Trong mỗi epoch, mô hình được huấn luyện trên toàn bộ tập dữ liệu theo các bước sau:

- Kích hoạt chế độ huấn luyện: Đặt mô hình ở chế độ huấn luyện để bật các cơ chế như dropout và batch normalization:

```
model.train()
```

- Xử lý từng batch dữ liệu: Dữ liệu được nạp theo từng batch thông qua DataLoader:
    - Chuyển dữ liệu lên thiết bị tính toán (CPU/GPU)

- Tách riêng các thành phần đầu vào (input_ids, pixel_values)

```
input_ids = batch.pop("input_ids").to(device)
pixel_values = batch.pop("pixel_values").to(device)
```

- Forward Pass: Đưa dữ liệu qua mô hình để tính toán đầu ra và hàm mất mát. Mô hình BLIP được cấu hình để đồng thời xử lý đầu vào là hình ảnh (pixel_values) và văn bản (input_ids), với input_ids cũng đóng vai trò là nhãn (labels) cần dự đoán:

```
outputs = model(input_ids=input_ids,
                pixel_values=pixel_values,
                labels=input_ids)

loss = outputs.loss
```

- Hiển thị thông tin huấn luyện: Giá trị loss được in ra sau mỗi batch để theo dõi tiến trình huấn luyện:

```
print("Loss:", loss.item())
```

- Backward Pass: Tính toán gradient của các tham số mô hình dựa trên hàm mất mát:

```
loss.backward()
```

- Cập nhật tham số: Sử dụng optimizer để cập nhật các tham số của mô hình dựa trên gradient đã tính:

```
optimizer.step()
```

- Xóa gradient: Đặt lại gradient về 0 sau mỗi bước cập nhật để chuẩn bị cho batch tiếp theo:

```
optimizer.zero_grad()
```

Dưới đây là toàn bộ code của phần huấn luyện mô hình

```
import torch

optimizer = torch.optim.AdamW(model.parameters(), lr=5e-5)

device = "cuda" if torch.cuda.is_available() else "cpu"
model.to(device)

model.train()

for epoch in range(10):
  print("Epoch:", epoch)
  for idx, batch in enumerate(train_dataloader):
    input_ids = batch.pop("input_ids").to(device)
    pixel_values = batch.pop("pixel_values").to(device)

    outputs = model(input_ids=input_ids,
                    pixel_values=pixel_values,
                    labels=input_ids)

    loss = outputs.loss

    print("Loss:", loss.item())

    loss.backward()

    optimizer.step()
    optimizer.zero_grad()
```

3. Đặc điểm của quá trình huấn luyện

- Hàm mất mát: Mô hình BLIP sử dụng hàm Cross-Entropy Loss để tính toán sự chênh lệch giữa chú thích được dự đoán và chú thích thực tế.
- Học đặc trưng địa phương: Quá trình fine-tuning cho phép mô hình học được các đặc điểm cụ thể của hình ảnh du lịch Hà Nội, như kiến trúc đặc trưng, cảnh quan đô thị, và các yếu tố văn hóa địa phương.
- Điều chỉnh dần dần: Với mỗi epoch, mô hình dần dần điều chỉnh các tham số để giảm thiểu hàm mất mát, từ đó cải thiện khả năng sinh chú thích chính xác và phù hợp với ngữ cảnh du lịch Hà Nội.

4. Quá trình hội tụ

```
Loss: 0.001898868940770626
Epoch: 9
Loss: 0.000941871665418148
Loss: 0.0010977271012961864
Loss: 0.0019659020472317934
Loss: 0.0021051650401204824
Loss: 0.006749162916094065
Loss: 0.0008921747794374824
Loss: 0.001193783013150096
Loss: 0.0028353016823530197
Loss: 0.005823872517794371
Loss: 0.006883316207677126
Loss: 0.007298650685697794
Loss: 0.008264189586043358
Loss: 0.0026631634682416916
Loss: 0.004641807172447443
Loss: 0.002018817700445652
Loss: 0.001099113840609789
Loss: 0.00286336918361485
Loss: 0.004025645554065704
Loss: 0.001578203635290265
Loss: 0.002450111787766218
Loss: 0.0010758211137726903
Loss: 0.0018297150963917375
Loss: 0.001233021030202508
Loss: 0.01052689366042614
Loss: 0.008869178593158722
Loss: 0.0022343394812196493
Loss: 0.0013920919736847281
Loss: 0.004113808274269104
Loss: 0.0016418473096564412
Loss: 0.0011274013668298721
```

Hình 3.3: Ảnh giá trị loss

Trong quá trình huấn luyện, em quan sát thấy giá trị loss có xu hướng giảm dần theo thời gian và ổn định khoảng 0.002 – 0.003 với epocho thứ 9. Có thể thấy:

- Mô hình đang hội tụ tốt - việc loss giảm dần theo thời gian cho thấy mô hình đang học tốt từ dữ liệu huấn luyện.
- Việc loss ổn định ở mức thấp (0.002-0.003) từ epoch thứ 9 chỉ ra rằng mô hình đã đạt đến điểm hội tụ và không còn cải thiện đáng kể.

3.4 Đánh giá mô hình

Phương pháp đánh giá

Để đánh giá hiệu quả của mô hình BLIP trong nhiệm vụ sinh mô tả hình ảnh, em sử dụng phương pháp đánh giá phổ biến là BLEU (Bilingual Evaluation Understudy). BLEU là một thuật toán đánh giá chất lượng của văn bản được dịch

bởi máy, so sánh độ tương đồng giữa văn bản đầu ra của mô hình với một hoặc nhiều văn bản tham chiếu do con người tạo ra. Bên cảnh đó chỉ số CIDEr (Consensus-based Image Description Evaluation) được sử dụng nhằm đo lường mức độ phù hợp giữa mô tả của mô hình và các mô tả tham chiếu (reference) do con người cung cấp. CIDEr là một thước đo đánh giá dựa trên sự đồng thuận, được thiết kế đặc biệt cho bài toán sinh chú thích ảnh (image captioning).

Điểm BLEU

Phương pháp BLEU đánh giá chất lượng văn bản dựa trên sự trùng khớp n-gram giữa văn bản được sinh ra và văn bản tham chiếu. Điểm BLEU được tính dựa trên công thức:

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

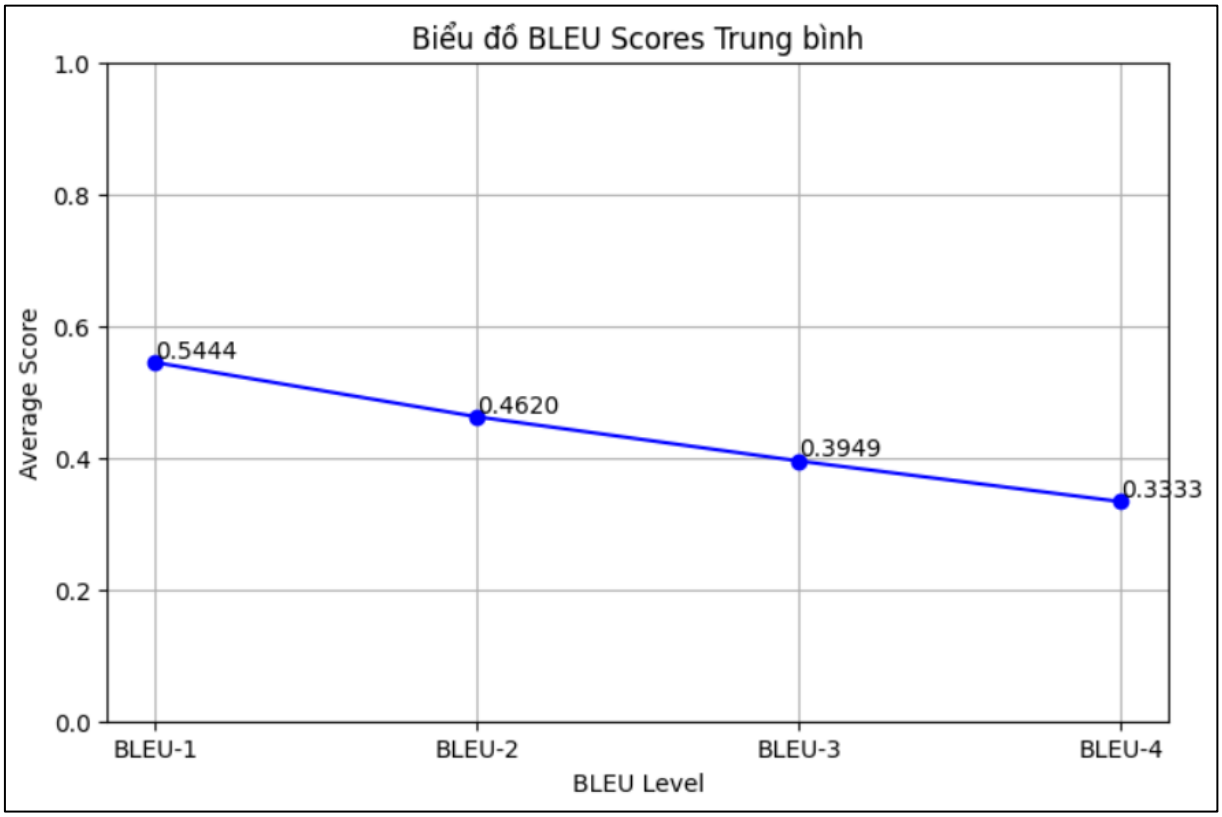
Trong đó:

- BP (Brevity Penalty): hệ số phạt độ dài, dùng để tránh việc mô hình sinh ra các câu quá ngắn chỉ nhằm đạt độ trùng khớp cao.
- w_n: trọng số của từng n-gram, thường được gán bằng nhau, tức $w_1 = w_2 = w_3 = w_4 = 0.25$
- p_n: độ chính xác (precision) của cấp độ n giữa câu sinh và câu tham chiếu.

Trong nghiên cứu này, em tính toán các điểm BLEU-1, BLEU-2, BLEU-3 và BLEU-4, tương ứng với việc đánh giá mức độ trùng khớp của unigram, bigram, trigram và 4-gram. Bên cạnh đó, em cũng tính điểm BLEU tổng hợp (cumulative BLEU), kết hợp đồng đều cả bốn cấp độ n-gram với trọng số bằng nhau.

Kết quả BLEU trung bình

Sau khi xử lý toàn bộ tập ảnh kiểm tra, em thu được các điểm BLEU trung bình như sau:



Hình 3.3: Biểu đồ BLEU Scores Trung bình

Biểu đồ trên là biểu đồ BLEU Scores Trung bình – đây là biểu đồ thể hiện điểm đánh giá BLEU trung bình cho các cấp BLEU-1 đến BLEU-4. Trục hoành (trục X) gồm 4 cấp độ BLEU-1, BLEU-2, BLEU-3, BLEU-4 mỗi cấp độ thể hiện n-gram được đánh giá trong mô hình sinh văn bản. Trục tung (trục Y) thể hiện điểm BLEU trung bình tương ứng với mỗi cấp độ, nằm trong khoảng từ 0.0 đến 1.0.

Từ kết quả có thể thấy:

- Mô hình có khả năng sinh từ vựng chính xác ở mức khá. BLEU-1 đo mức độ khớp từng từ lẻ (unigram) giữa caption sinh ra và caption tham chiếu và với BLEU-1 ≈ 0.54, mô hình có tỷ lệ khớp từ khóa ổn định (~54%) → mô hình nhận diện tốt các đối tượng, hành động chính trong ảnh.
- Mô hình có mức độ trôi chảy và ngữ cảnh ở mức trung bình khá. BLEU-4 (4-gram) = 0.3333 cho thấy mô hình không chỉ dự đoán đúng các từ, mà còn biết kết nối từ thành cụm hoặc câu hợp lý.

Điểm BLEU giảm dần khi độ dài n-gram tăng, phản ánh rằng mô hình sinh có khả năng tạo ra từ ngữ phù hợp (unigram) tốt hơn so với cấu trúc cụm từ dài hơn (trigram, 4-gram).

Điểm CIDEr

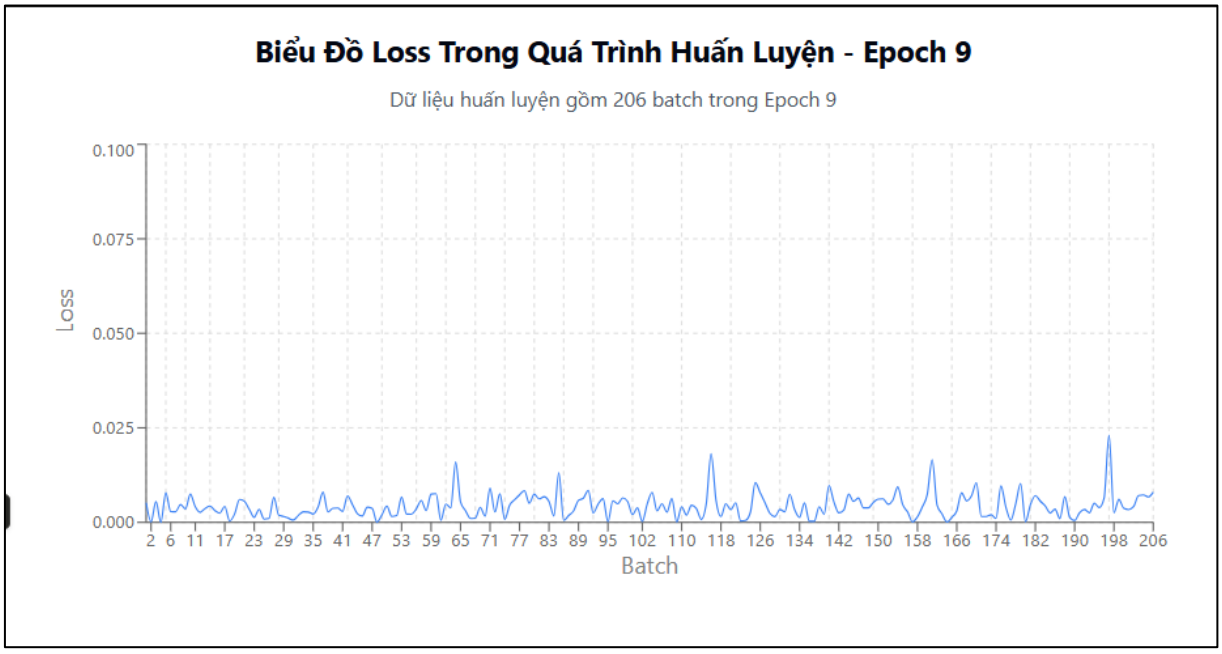
```
gts = {i: [references[i]] for i in range(len(references))}
res = {i: [hypotheses[i]] for i in range(len(hypotheses))}

# Filter out empty hypotheses if any
valid_indices = [i for i, hyp in res.items() if hyp and hyp[0].strip()]
gts_filtered = {i: gts[i] for i in valid_indices}
res_filtered = {i: res[i] for i in valid_indices}

if not res_filtered:
    print("Không có caption hợp lệ để tính CIDEr.") # Updated message
else:
    # Calculate CIDEr score
    print("\nCalculating CIDEr score...")
    cider_obj = Cider()
    cider_score, cider_scores = cider_obj.compute_score(gts_filtered, res_filtered)
    print(f"Average CIDEr score: {cider_score:.4f}")
```

```
Calculating CIDEr score...
Average CIDEr score: 1.2659
```

Hình 3.4. Điểm CIDEr

Với điểm CIDEr: 1.2659 có thể thấy mô tả của mô hình gần giống với mô tả của con người.

3.5 Kết quả đạt được



Hình 3.5: Hình ảnh biểu đồ Loss trong quá trình huấn luyện

Với hình ảnh biểu đồ Loss trên ta có những thông tin chi tiết sau:

- Trục X: là các giá trị batch từ 1 đến 206 với mỗi số thứ tự của các batch dữ liệu trong epoch 9.
- Trục Y: là các giá trị loss giá trị từ 0 đến 0.025.

Ta có thể thấy kết quả huấn luyện mô hình tại epoch thứ 9 cho thấy sự hội tụ đáng ghi nhận với giá trị loss giảm xuống mức rất thấp. Dưới đây là phân tích chi tiết về hiệu suất mô hình dựa trên dữ liệu thu được:

Đánh giá chỉ số Loss:

- Giá trị Loss trung bình: Khoảng 0.003-0.004, thể hiện mức độ sai số rất thấp trong quá trình huấn luyện.
- Độ dao động Loss: Phần lớn giá trị dao động trong khoảng 0.001-0.005, chứng tỏ sự ổn định của mô hình.
- Giá trị Loss thấp nhất: Xấp xỉ 0.0009, cho thấy khả năng dự đoán chính xác cao trong một số trường hợp.
- Một số điểm ngoại lệ: Xuất hiện một vài giá trị cao hơn (0.0254, 0.0225, 0.0182) nhưng không ảnh hưởng đến xu hướng hội tụ chung.

Đánh giá hiệu suất:

- Tính ổn định: Mô hình thể hiện sự ổn định cao với các giá trị loss nhất quán qua nhiều bước huấn luyện liên tiếp.
- Khả năng hội tụ: Mức loss thấp ở epoch 9 chứng minh mô hình đã hội tụ hiệu quả, phản ánh quá trình tối ưu hóa thành công.
- Độ chính xác: Với mức loss dưới 0.01 trong hầu hết các trường hợp, mô hình đạt độ chính xác cao trong việc dự đoán/phân loại.

CHƯƠNG IV: XÂY DỰNG ỨNG DỤNG

4.1 Giới thiệu các framework sử dụng

Backend – Flask

Trong đề tài, Flask được chọn làm framework chính để xây dựng hệ thống backend vì tính nhẹ, dễ mở rộng và khả năng tích hợp tốt với nhiều thư viện Python hiện đại như Hugging Face Transformers, MongoDB, JWT,... Flask chịu trách nhiệm:

1. Nhận ảnh từ người dùng (React gửi lên).
2. Gọi mô hình BLIP để sinh chú thích hình ảnh.
3. Lưu trữ dữ liệu ảnh và chú thích vào cơ sở dữ liệu MongoDB.
4. Cung cấp API phục vụ frontend như: sinh caption, hiển thị lịch sử, …

Dưới đây là các thư viện chính tích hợp với Flask và vai trò của chúng:

Thư viện	Vai trò
Flask (v1.1.4)	Web framework chính, xử lý route, request, response.
Flask-MongoEngine (v1.0.0)	Tương tác với MongoDB thông qua ORM-style model.
Flask-JWT-Extended (v4.3.1)	Xác thực người dùng qua JSON Web Token (JWT).
pymongo (v3.12.0)	Kết nối và truy vấn MongoDB ở mức thấp khi cần.
Werkzeug (v1.0.1)	Cung cấp các tiện ích hỗ trợ routing, bảo mật,... cho Flask.
python-dotenv (v0.19.1)	Đọc các biến môi trường từ file `.env`, tách biệt cấu hình ra khỏi mã nguồn.

flask-cors	Kích hoạt CORS để frontend (React) và backend có thể giao tiếp.
bcrypt	Mã hóa mật khẩu người dùng an toàn.
googletrans (v4.0.0-rc1)	Dịch caption ảnh sang tiếng Việt hoặc các ngôn ngữ khác.
transformers (Hugging Face)	Tải và chạy mô hình BLIP để sinh chú thích ảnh.

Quy trình hoạt động Backend:

5. Người dùng tải ảnh lên từ giao diện React.
6. Flask nhận ảnh → Tiền xử lý ảnh → Đưa vào mô hình BLIP.
7. Mô hình trả về caption → Lưu vào MongoDB.
8. Flask trả lại caption cho frontend → frontend hiển thị kết quả.

Frontend – ReactJS

ReactJS được lựa chọn để xây dựng giao diện người dùng vì khả năng tạo giao diện tương tác, tốc độ cập nhật trạng thái nhanh và cộng đồng phát triển mạnh. Frontend chịu trách nhiệm:

9. Hiển thị giao diện upload ảnh, nhận chú thích.
10. Gửi ảnh lên backend Flask để xử lý.
11. Giao tiếp với các API để nhận chú thích, dịch.
12. Hiển thị kết quả một cách trực quan, dễ sử dụng.

Các thư viện ReactJS:

Thư viện	Vai trò
React	Thư viện chính để xây dựng giao diện.
Axios	Gửi HTTP request (POST ảnh, GET caption) đến Flask API.
React Router DOM	Điều hướng giữa các trang (nếu có nhiều chức năng).

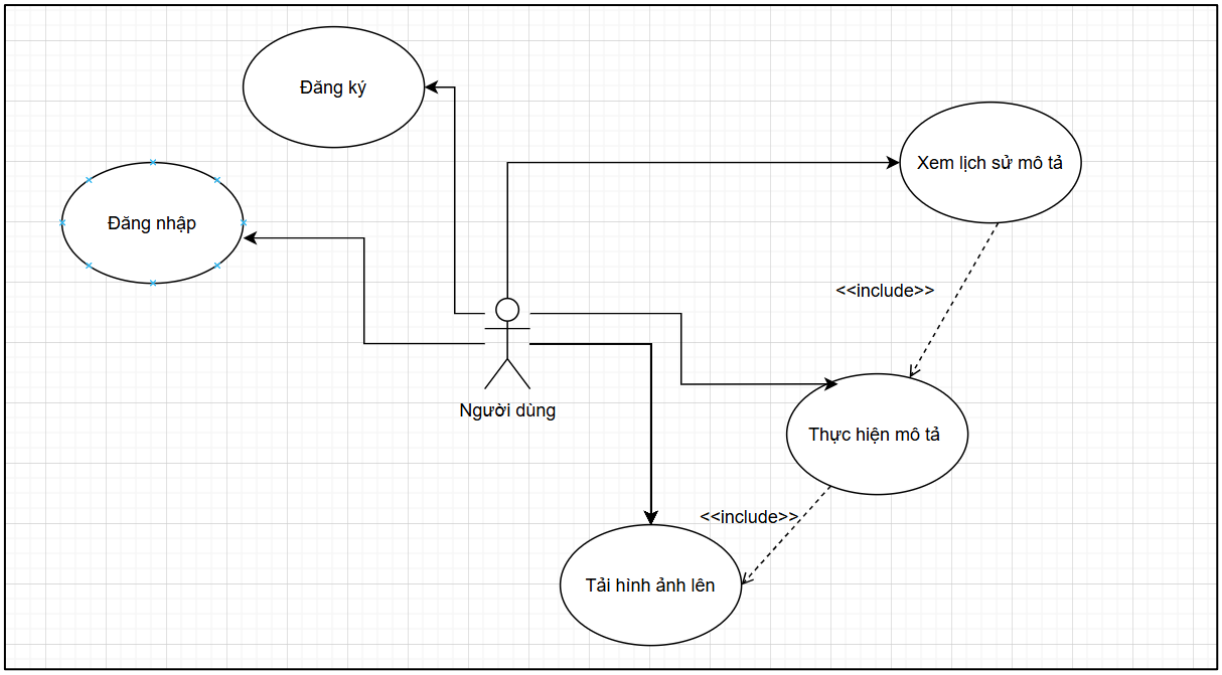
Ant Design, TailwindCSS	Hệ thống giao diện giúp xây UI nhanh chóng, đẹp mắt.

Quy trình hoạt động Frontend:

13. Người dùng chọn ảnh từ máy và nhấn nút "Tải lên".
14. Ảnh được gửi đến backend qua axios.
15. Khi nhận lại caption, frontend sẽ hiển thị chú thích.
16. Người dùng có thể chọn "Nghe caption" → Gửi request đến API → Backend trả về file âm thanh.
17. File âm thanh được phát ngay trong giao diện React.

4.2 Phân tích thiết kế hệ thống

Biểu đồ use case



Hình 4.1: Biểu đồ use case tổng quát

Mô tả chi tiết các use case

❖ **Mô tả use case đăng kí**

Bảng 4.1: Mô tả use case đăng kí

Tên use case	Đăng ký
Mục đích	Cho phép người dùng tạo tài khoản mới trong hệ thống
Tác nhân chính	Người dùng
Điều kiện tiên quyết	Người dùng chưa có tài khoản trong hệ thống
Luồng sự kiện: **1.** Người dùng truy cập vào trang đăng ký **2.** Hệ thống hiển thị form đăng kí **3.** Người dùng nhập thông tin cá nhân (tên người dùng, email, mật khẩu, xác nhận mật khẩu) **4.** Người dùng xác nhận đăng ký **5.** Hệ thống kiểm tra tính hợp lệ của thông tin **6.** Hệ thống tạo tài khoản mới và lưu thông tin **7.** Hệ thống thông báo đăng ký thành công	
Luồng thay thế: **1.** Nếu thông tin không hợp lệ hoặc đã tồn tại, hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại	

❖ **Mô tả use case đăng nhập**

Bảng 4.2: Mô tả đăng nhập

Tên use case	Đăng nhập
Mục đích	Xác thực người dùng và cấp quyền truy cập vào hệ thống
Tác nhân chính	Người dùng
Điều kiện tiên quyết	Người dùng đã có tài khoản trong hệ thống
Luồng sự kiện: **1.** Người dùng truy cập vào trang đăng nhập	

2. Hệ thống hiển thị form đăng nhập 3. Người dùng nhập tên đăng nhập và mật khẩu (email và password) 4. Người dùng xác nhận đăng nhập 5. Hệ thống xác thực thông tin đăng nhập 6. Hệ thống cấp phiên làm việc cho người dùng 7. Hệ thống chuyển người dùng đến trang chính
Luồng thay thế: 1. Nếu thông tin đăng nhập không chính xác, hệ thống hiển thị thông báo lỗi

❖ **Mô tả use case tải hình ảnh tải lên**

Bảng 4.3: Mô tả hình ảnh tải lên

Tên use case	Tải hình ảnh lên
Mục đích	Cho phép người dùng tải hình ảnh lên hệ thống
Tác nhân chính	Người dùng
Điều kiện tiên quyết	Người dùng đã có tài khoản trong hệ thống
Luồng sự kiện: 1. Người dùng chọn tùy chọn "Tải hình ảnh lên" 2. Hệ thống hiển thị giao diện tải ảnh 3. Người dùng chọn file ảnh từ thiết bị hoặc kéo thả vào khu vực quy định 4. Hệ thống tải và lưu trữ ảnh vào máy chủ 5. Hệ thống hiển thị xác nhận tải lên thành công	
Luồng thay thế: 1. Nếu quá trình tải lên thất bại, hệ thống hiển thị thông báo và cho phép thử lại	

❖ **Mô tả use case thực hiện mô tả**

Bảng 4.4: Thực hiện mô tả

Tên use case	Thực hiện mô tả
Mục đích	Cho phép người dùng tạo mô tả cho hình ảnh
Tác nhân chính	Người dùng
Điều kiện tiên quyết	Người dùng đã đăng nhập và đã có hình ảnh
Luồng sự kiện: **1.** Hệ thống hiển thị giao diện tạo mô tả với hình ảnh đã chọn **2.** Người dùng bấm thực hiện mô tả đoạn mô tả sẽ xuất hiện	
Luồng thay thế: **1.** Nếu quá trình thực hiện mô tả thất bại, hệ thống hiển thị thông báo và cho phép thử lại	

❖ **Mô tả use case xem lịch sử mô tả**

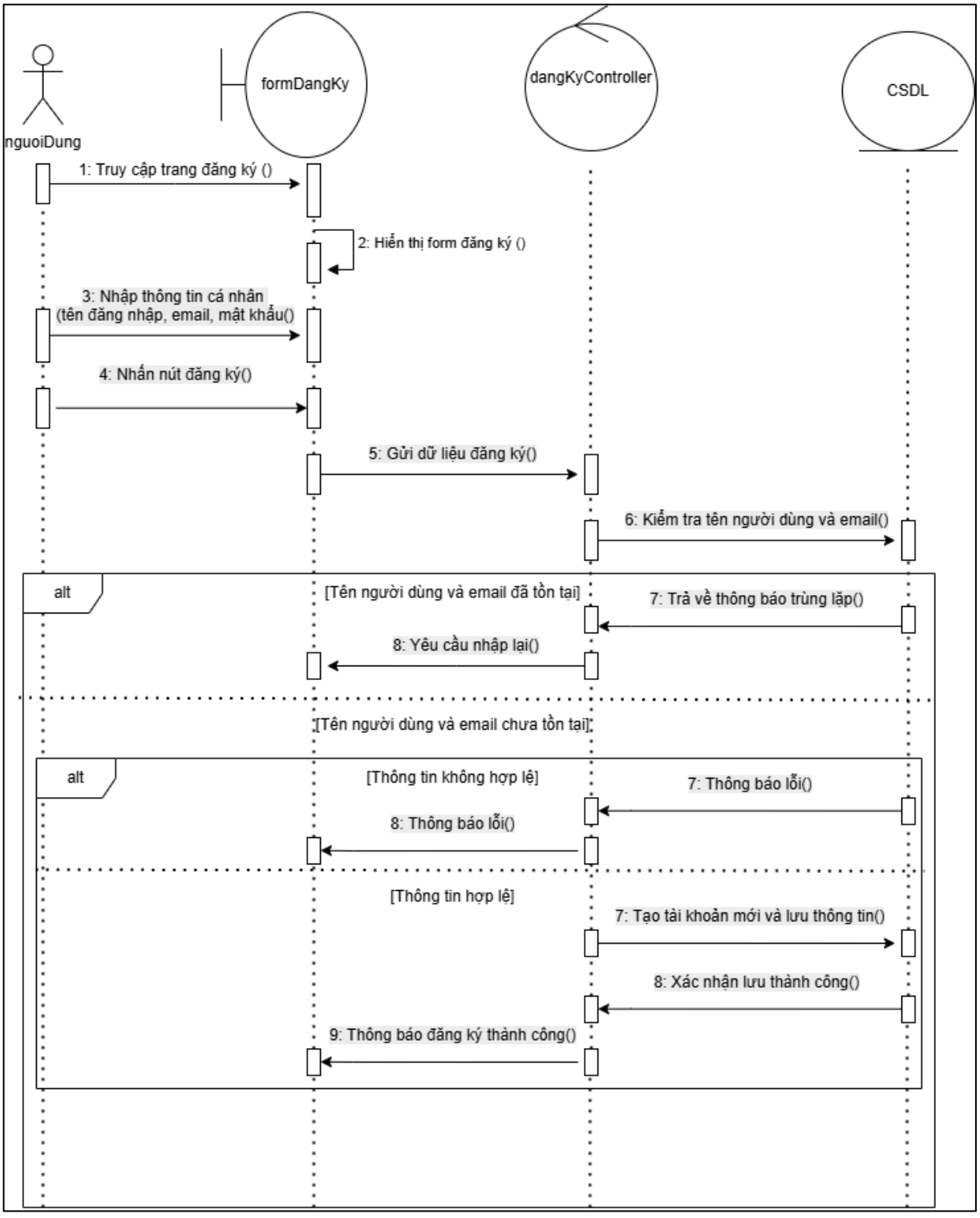
Bảng 4.5: Xem lịch sử mô tả

Tên use case	Xem lịch sử mô tả
Mục đích	Hiển thị lịch sử các mô tả đã thực hiện
Tác nhân chính	Người dùng
Điều kiện tiên quyết	Người dùng đã đăng nhập và đã có ảnh và mô tả trong hệ thống
Luồng sự kiện: **1.** Người dùng chọn mô tả ảnh **2.** Hệ thống sẽ truy xuất danh sách các ảnh và mô tả đã tạo **3.** Hệ thống sẽ hiển thị danh sách theo thứ tự thời gian	

4. Hệ thống lưu mô tả vào máy chủ
Luồng thay thế: **1.** Nếu không có mô tả nào, hệ thống thông báo và gợi ý tạo mô tả mới

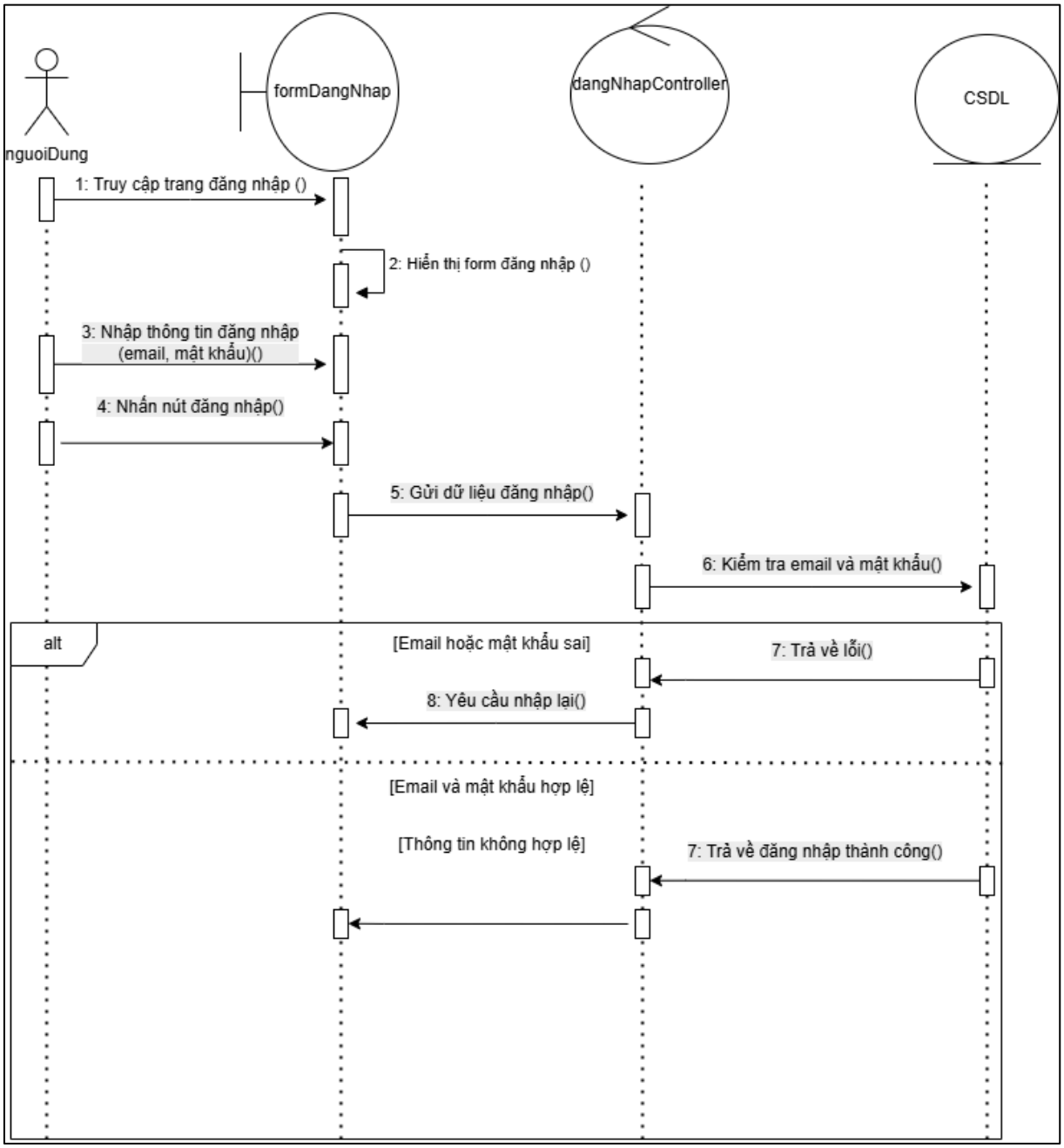
Phân tích các use case

❖ Biểu đồ trình tự use case đăng ký



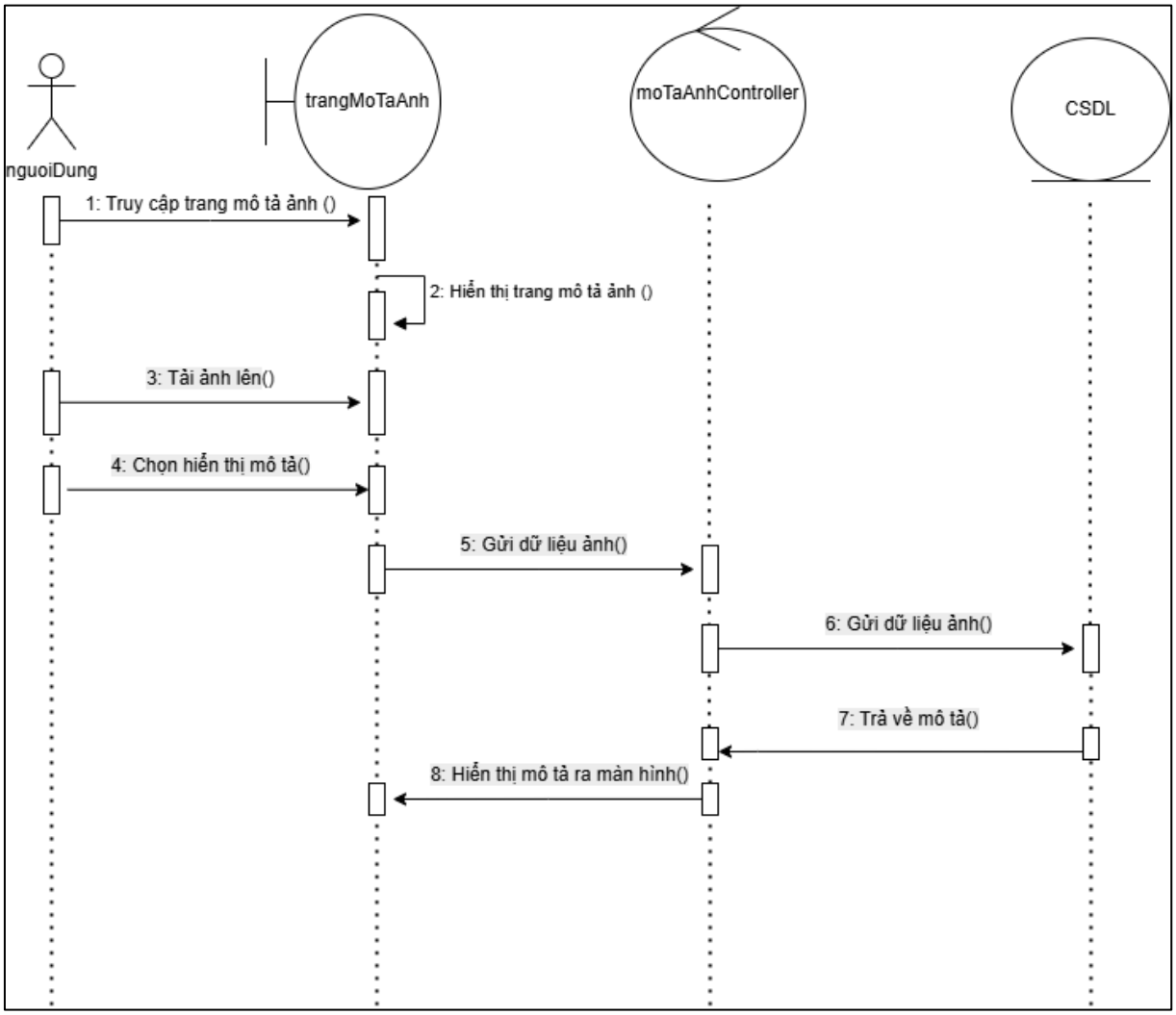
Hình 4.2: Biểu đồ trình tự use case đăng ký

❖ Biểu đồ trình tự use case đăng nhập



Hình 4.3: Biểu đồ trình tự use case đăng nhập

❖ Biểu đồ trình tự tải hình ảnh lên và thực hiện mô tả



Hình 4.4: Biểu đồ trình tự use case tải hình ảnh lên và thực hiện mô tả

Thiết kế cơ sở dữ liệu

Mô hình dữ liệu dạng tài liệu

❖ Collection: User

Bảng 4.6: Collection User

Trường	Kiểu dữ liệu
_id	ObjectId
username	String
password	String
email	String
full_name	String
is_active	Boolean

role	String
created_at	Date
last_login	Date

❖ Collection: Image

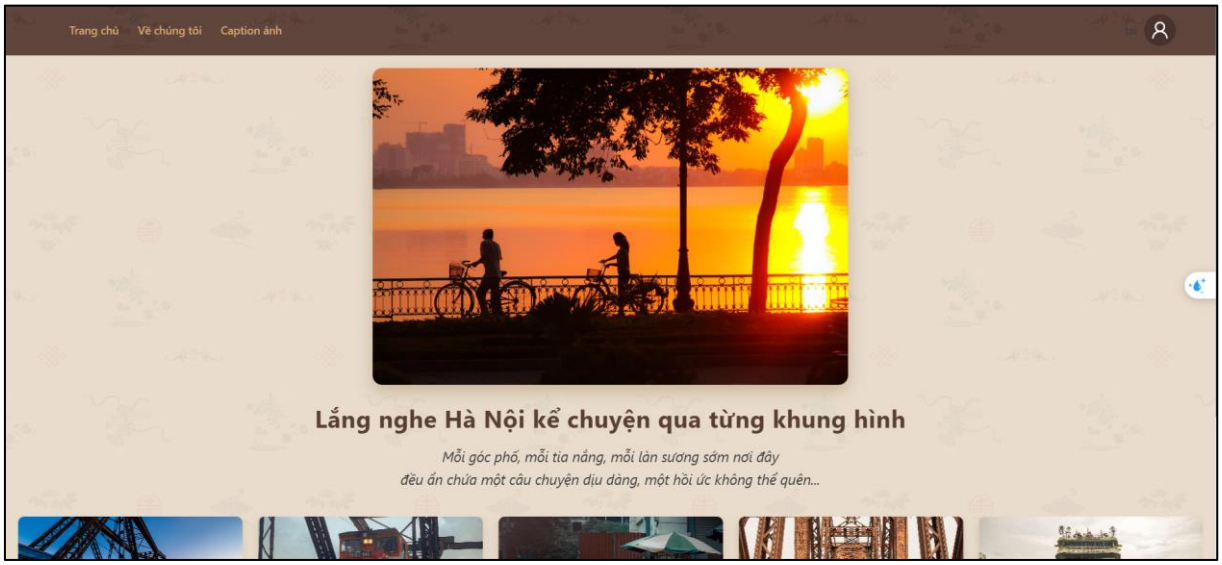
Bảng 4.7: Collection Image

Trường	**Kiểu dữ liệu**
_id	ObjectId
description	String
file_name	String
content_type	String
image_data	Binary
uploaded_by	ObjectId
created_at	Date

4.3 Giao diện hệ thống

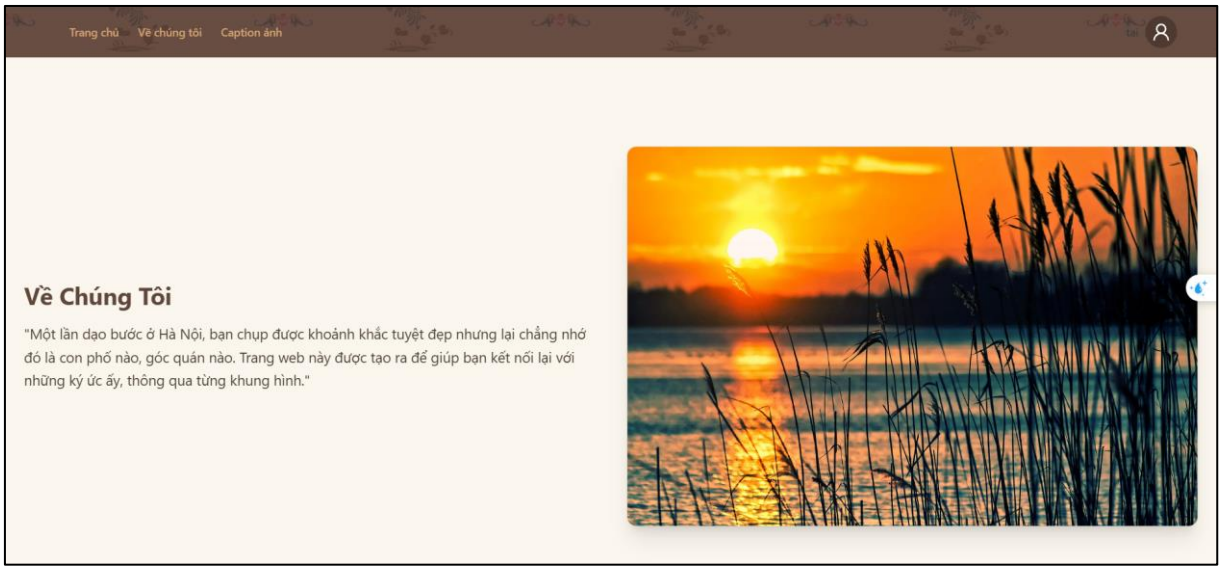
Giao diện của hệ thống được thiết kế với tông màu ấm và tự nhiên, gợi cảm giác thân thiện, gần gũi với du lịch và trải nghiệm văn hóa. Màu nâu gỗ chủ đạo kết hợp với các tông vàng đất nung và trắng ngà mang lại cảm giác cổ điển nhưng không kém phần hiện đại, tạo điểm nhấn hài hòa cho người dùng. Hiện tại hệ thống chỉ hỗ trợ ngôn ngữ Tiếng Việt với mục đích chính hướng tới người sử dụng trong nước nhưng trong tương lai hệ thống sẽ được nâng cấp để hỗ thêm một số ngôn ngữ thông dụng như tiếng Anh. Chi tiết giao diện được trình bày các hình bên dưới.

❖ Trang chủ

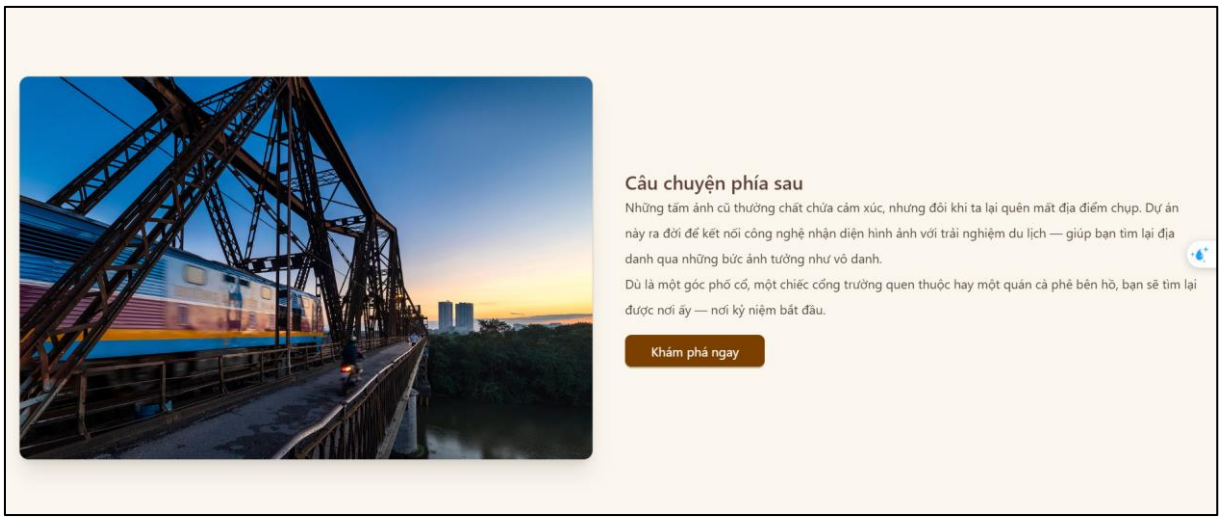


Hình 4.5: Trang chủ website

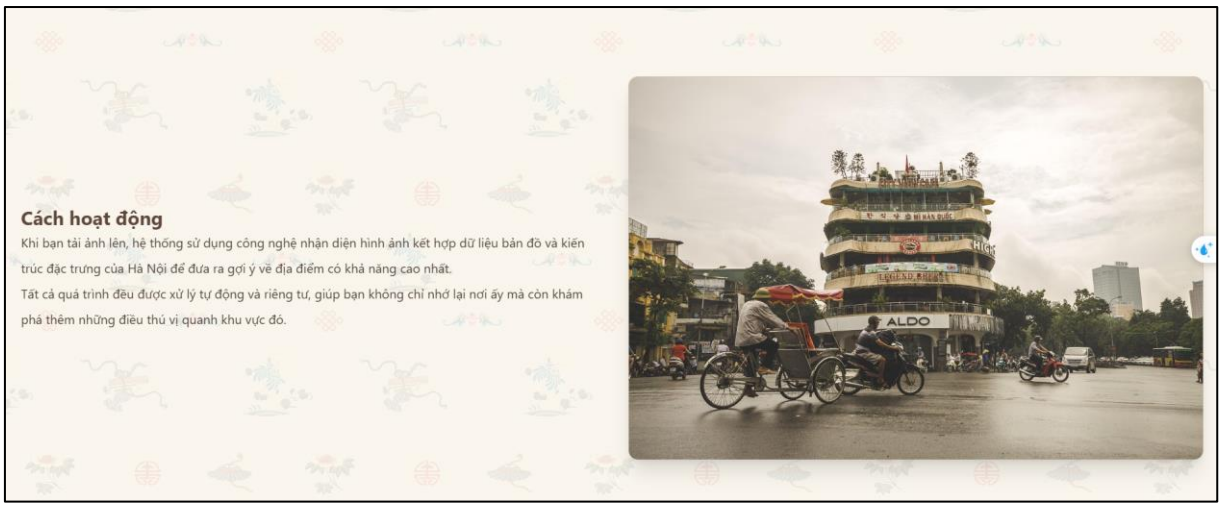
❖ Trang về chúng tôi



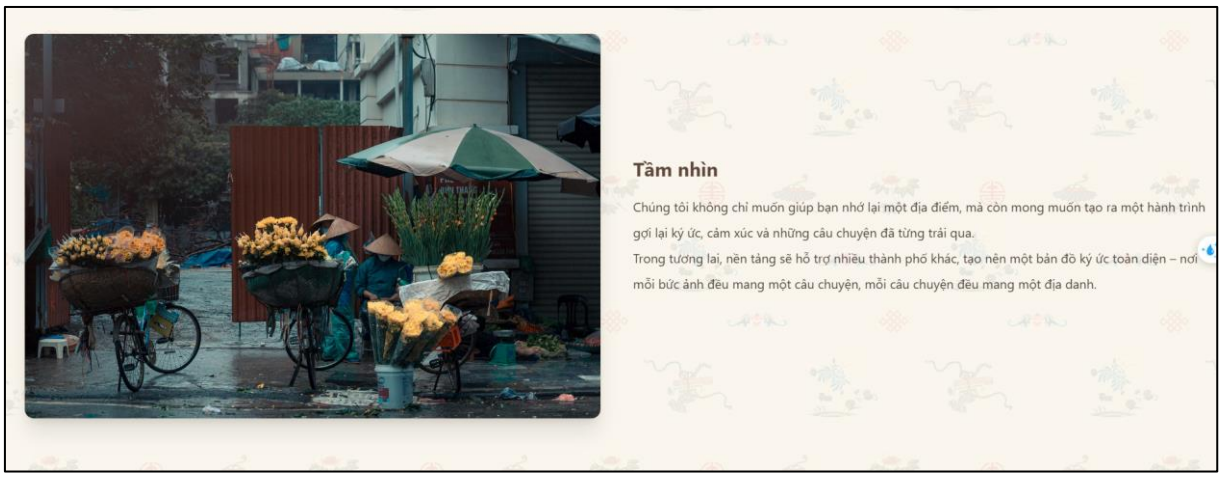
Hình 4.6: Trang về chúng tôi (1)



Hình 4.7: Trang về chúng tôi (2)



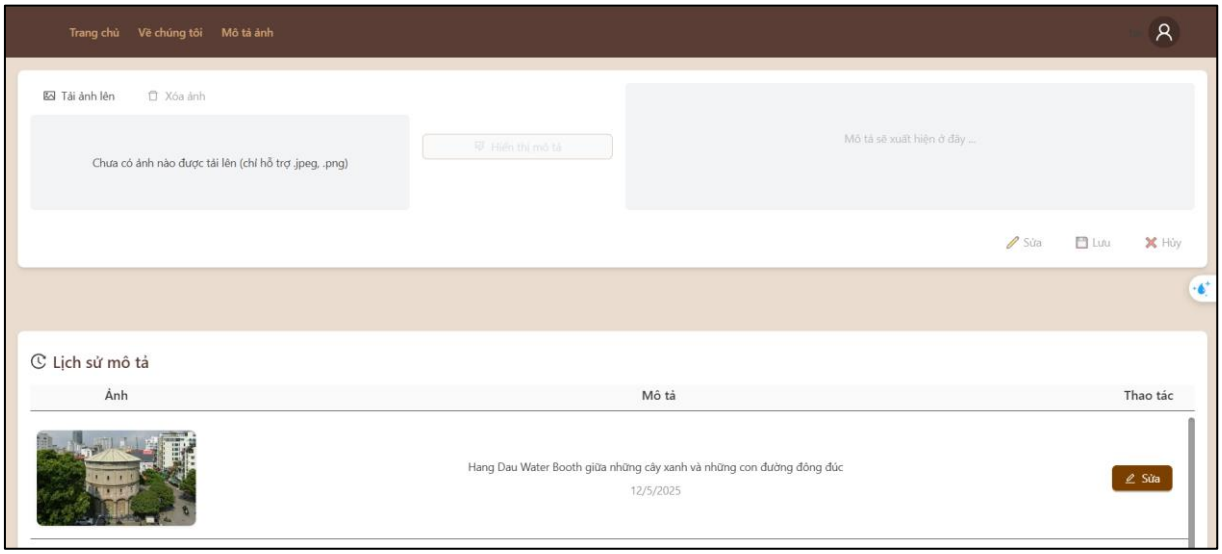
Hình 4.8: Trang về chúng tôi (3)



Hình 4.9: Trang về chúng tôi (4)

Hình 4.10: Trang về chúng tôi (5)

❖ Trang mô tả ảnh



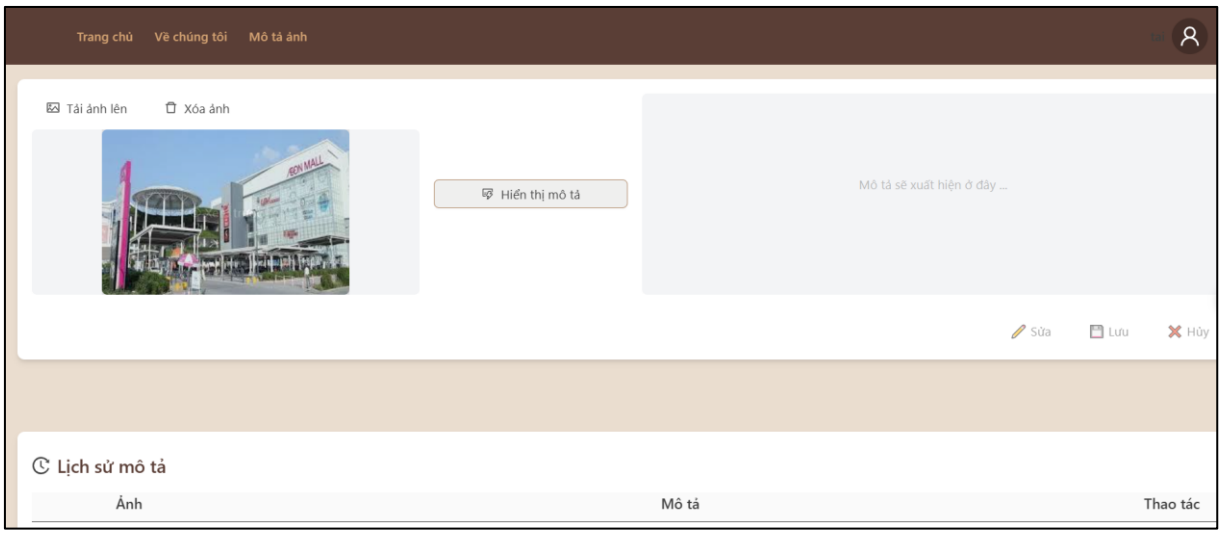
Hình 4.11: Trang mô tả ảnh

4.4 Các chức năng của hệ thống

Xử lý hình ảnh và sinh caption tự động là chức năng cốt lõi của hệ thống, giúp người dùng tải lên hình ảnh du lịch và nhận được mô tả tự động bằng ngôn ngữ tự nhiên thông qua mô hình BLIP (Bootstrapped Language-Image Pretraining).

1. **Tải ảnh lên**

❖ Người dùng chọn hoặc kéo-thả hình ảnh từ thiết bị cá nhân vào giao diện web.



Hình 4.12: Người dùng tải ảnh lên hoặc kéo thả vào

❖ Bấm hiển thị mô tả

❖ Ảnh được gửi đến backend (Flask) thông qua API RESTful.

❖ Thông tin ảnh, bao gồm người đăng, thời gian và mô tả liên quan được lưu trữ trong cơ sở dữ liệu (MongoDB).


```
/**
* Paste one or more documents here
*/
{
  "description": "vietnam national fine arts museum features
  "file_name": "e0e8c077-5d80-44f7-80d3-af16b3bf8e6e_baoTangN
  "content_type": "image/jpeg",
  "image_data": {
    "$binary": {
      "base64": "/9j/4AAQSkZJRgABAQEASABIAAD/2wCEAAUDBAQEAwUE
      "subType": "00"
    }
  },
  "uploaded_by": {
    "$oid": "68201ea3ff03a69196108202"
  },
  "created_at": {
    "$date": "2025-05-11T10:52:59.574Z"
  }
}
```


Hình 4.13: Ảnh được lưu vào MongoDB

2. Phân tích ảnh và sinh mô tả bằng BLIP

❖ Sau khi ảnh được tiếp nhận, hệ thống sử dụng mô hình BLIP để phân tích nội dung hình ảnh.

```python
@classmethod
def _load_model_if_needed(cls):
    if cls._model is None or cls._processor is None:
        if cls._is_loading:
            import time
            while cls._is_loading and (cls._model is None or cls._processor is None):
                time.sleep(0.5)
            return

        cls._is_loading = True
        try:
            if not os.path.exists(cls._model_path):
                raise FileNotFoundError(f"Không tìm thấy đường dẫn mô hình: {cls._model_path}")

            print(f"Đang tải mô hình BLIP từ {cls._model_path}...")
            cls._processor = BlipProcessor.from_pretrained(cls._model_path, use_fast=True)
            cls._model = BlipForConditionalGeneration.from_pretrained(cls._model_path)
            cls._model = cls._model.to(cls._device)
            cls._model.eval()
            print(f"Tải mô hình thành công trên thiết bị {cls._device}")
        finally:
            cls._is_loading = False
```

Mã nguồn chạy mô hình BLIP khi ảnh được tải lên

❖ Mô hình sẽ sinh một caption mô tả nội dung chính của ảnh một cách tự động, tự nhiên và ngắn gọn.

```python
def generate_caption_from_binary(cls, image_data, max_length=30, num_beams=5, speak=False):
    """
    Tạo caption cho ảnh từ dữ liệu nhị phân.
    Nếu speak=True, sẽ dịch caption sang tiếng Việt.
    """
    try:
        cls._load_model_if_needed()

        # Chuyển dữ liệu nhị phân thành đối tượng PIL Image
        image = Image.open(io.BytesIO(image_data)).convert("RGB")
        inputs = cls._processor(image, return_tensors="pt")
        for k, v in inputs.items():
            inputs[k] = v.to(cls._device)

        with torch.no_grad():
            output_ids = cls._model.generate(
                **inputs,
                max_length=max_length,
                num_beams=num_beams,
                min_length=5
            )

        caption_en = cls._processor.decode(output_ids[0], skip_special_tokens=True)
        print("📸 Caption tiếng Anh:", caption_en)
```

Mã nguồn sinh mô tả khi ảnh được tải lên

Hình 4.14: Sinh mô tả khi cho ảnh tải lên

3. Tùy chỉnh và chỉnh sửa caption

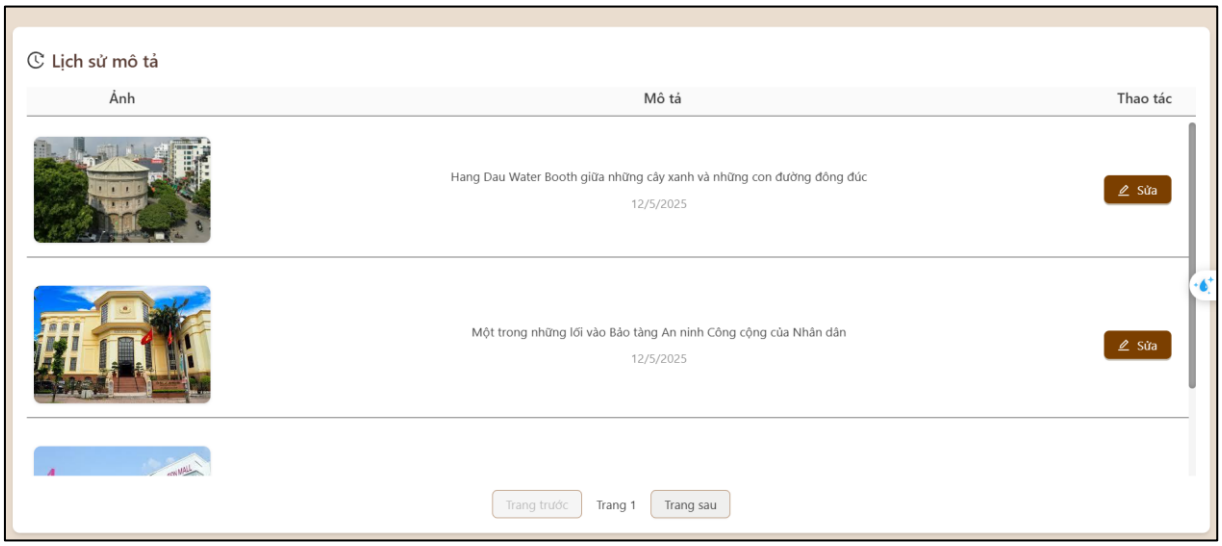
❖ Sau khi caption được tạo, người dùng có thể chỉnh sửa thủ công nếu muốn diễn đạt khác hoặc bổ sung thông tin

❖ Nhấn lưu để hệ thống lưu caption đã sửa

Hình 4.15: Chỉnh sửa mô tả khi cho ảnh tải lên

4. Lưu trữ và quản lý caption

- ❖ Mỗi ảnh sau khi được xử lý sẽ được lưu trữ cùng với caption tương ứng.
- ❖ Người dùng có thể xem lại danh sách ảnh đã tải lên và caption của từng ảnh
- ❖ Chỉnh sửa lại caption nếu cần thiết



Hình 4.16: Lịch sử mô tả của người dùng

KẾT LUẬN

Trong quá trình thực hiện luận án này, em đã tập trung nghiên cứu và triển khai một hệ thống hỗ trợ sinh chú thích hình ảnh du lịch tại Hà Nội, dựa trên mô hình BLIP (Bootstrapping Language-Image Pre-training). Đây là một hướng tiếp cận mới mẻ, kết hợp giữa khả năng hiểu ngôn ngữ tự nhiên và xử lý hình ảnh, nhằm mang lại trải nghiệm tiện ích và thông minh hơn cho du khách trong việc tìm hiểu thông tin về các địa điểm du lịch thông qua hình ảnh.Bên cạnh đó, việc sử dụng Flask và các công nghệ web hiện đại trong hệ thống giúp xây dựng giao diện người dùng thân thiện, dễ sử dụng, hỗ trợ du khách nhanh chóng nhận được thông tin mô tả chi tiết và chính xác từ hình ảnh các điểm du lịch tại Hà Nội. Tuy nhiên, một số hạn chế về thời gian xử lý khi tải lượng truy cập cao và yêu cầu kết nối internet ổn định đã được ghi nhận, làm tiền đề cho các cải tiến trong tương lai.

Em đã xây dựng thành công một hệ thống có khả năng sinh chú thích tự động cho hình ảnh du lịch, sử dụng mô hình BLIP và triển khai trên nền tảng web với Flask. Hệ thống không chỉ tạo ra các mô tả chính xác, phù hợp với ngữ cảnh văn hóa và lịch sử đặc trưng của các địa điểm tại Hà Nội, mà còn đạt được các chỉ số đánh giá cao như BLEU, METEOR và CIDEr. Điều này chứng minh mô hình hoạt động hiệu quả và có tiềm năng ứng dụng thực tiễn cao. Bên cạnh đó, việc kết hợp giao diện web thân thiện đã giúp người dùng dễ dàng tương tác và nhận được thông tin mô tả nhanh chóng từ hình ảnh.

Tuy vậy, hệ thống vẫn còn tồn tại một số hạn chế. Thời gian xử lý có thể bị kéo dài khi có nhiều người truy cập cùng lúc, và việc vận hành yêu cầu kết nối internet ổn định để đảm bảo hiệu suất. Đây là những yếu tố cần cải thiện nếu muốn mở rộng hệ thống ra phạm vi ứng dụng lớn hơn.

Trong thời gian tới, em sẽ tiếp tục phát triển hệ thống theo nhiều hướng. Trước hết là mở rộng phạm vi dữ liệu huấn luyện để bao phủ thêm nhiều địa điểm du lịch khác không chỉ ở Hà Nội mà còn trên khắp Việt Nam. Đồng thời, em sẽ tập trung tối ưu hóa mô hình BLIP nhằm rút ngắn thời gian phản hồi và cải thiện khả năng xử lý song song nhiều yêu cầu. Việc áp dụng các kỹ thuật giảm nhẹ mô

hình như Knowledge Distillation hay Quantization cũng sẽ được xem xét để giảm thiểu tài nguyên cần thiết. Cuối cùng, em mong muốn tích hợp khả năng sinh chú thích đa ngôn ngữ nhằm phục vụ cả du khách trong nước và quốc tế. Em tin rằng, với những định hướng này, hệ thống sẽ ngày càng hoàn thiện và trở thành một công cụ hữu ích trong lĩnh vực du lịch thông minh, góp phần quảng bá hình ảnh đất nước Việt Nam đến bạn bè quốc tế.

TÀI LIỆU THAM KHẢO

[1]. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., & Zhang, L. (2018). Bottom-up and top-down attention for image captioning and visual question answering. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6077-6086.

[2]. Chen, J., Guo, H., Yi, K., Li, B., & Elhoseiny, M. (2022). VisualGPT: Data-efficient adaptation of pretrained language models for image captioning. *Conference on Computer Vision and Pattern Recognition*.

[3]. Huang, L., Wang, W., Chen, J., & Wei, X. Y. (2019). Attention on attention for image captioning. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4634-4643.

[4]. Lu, J., Xiong, C., Parikh, D., & Socher, R. (2017). Knowing when to look: Adaptive attention via a visual sentinel for image captioning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 375-383.

[5]. Li, J., Li, D., Xiong, C., & Hoi, S. (2022). BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *International Conference on Machine Learning*, 12888-12900.

[6]. Dai, B., Zhu, C., & Chen, B. (2023). DialogCC: A large-scale natural dialogue corpus for BLIP research. *Conference on Empirical Methods in Natural Language Processing*.

[7]. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations*.

[8]. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 5998-6008.

[9]. Jay Alammar, "The Illustrated Transformer," Jalammar’s Blog, 2018, [Accessed: 22/05/2025], https://jalammar.github.io/illustrated-transformer/.

[11]. Đặng Xuân Thành, Trần Quốc Long. (2022). Phương pháp sinh chú thích hình ảnh sử dụng mô hình Transformer đa phương thức. *Tạp chí Khoa học và Công nghệ*, 65(7), 45-52.