Đại học quốc gia TP.HCM
Trường đại học công nghệ thông tin

KHOA KHOA HỌC MÁY TÍNH

CS406.P11 - XỬ LÝ ẢNH VÀ ỨNG DỤNG

BÁO CÁO ĐỒ ÁN CUỐI KÌ

IMAGE CAPTIONING IN VIETNAMESE CONTEXT

GV hướng dẫn: TS. Đỗ Văn Tiến
TS. Mai Tiến Dũng

Nhóm thực hiện
21522792 - Phạm Quốc Việt
21520553 - Trần Hoài An
21522443 - Lường Đại Phát

1 Tóm tắt kết quả đạt được

Trong bài toán Image Captioning này, nhóm chúng tôi đã thực hiện thí nghiệm và đánh giá hiệu suất của 5 mô hình bao gồm CNN+LSTM, CNN+LSTM (có attention), ClipCap, BLIP và BLIP-2. Dataset được sử dụng là bộ dataset ViIC, bao gồm các hình ảnh hoạt động thể thao liên quan đến bóng kèm theo câu mô tả tiếng Việt tương ứng. Kết quả thực nghiệm được so sánh dựa trên các chỉ số BLEU-1, BLEU-2, BLEU-3, BLEU-4 và ROUGE-L.

	Type	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**	**ROUGE-L**
CNN + LSTM	DenseNet201	67.21	52.49	42.76	35.39	47.21
	VGG16	61.67	45.3	35.53	28.17	45.13
	EfficientNetV2	56.84	40.06	31.68	24.68	41.60
CNN + LSTM (attention)	DenseNet201	65.98	52.58	44.92	39.04	48.00
	VGG16	56.84	44.93	38.22	32.69	41.60
	EfficientNetV2	56.84	44.93	38.22	32.69	41.60
ClipCap	ViT-g + GPT2	43.42	40.25	37.24	35.63	42.12
BLIP	ViT-b + BERT-b	63.50	52.06	43.50	36.06	49.01
BLIP-2	ViT-g + OPT 2.7B + PEFT (int8)	53.78	43.20	35.90	29.47	48.06

Bảng 1: Kết quả thực nghiệm trên tập dataset ViIC (learning rate default)

Trong các mô hình thực nghiệm, CNN+LSTM sử dụng DenseNet201 và cơ chế attention cho kết quả tốt nhất với BLEU-4 (39.04) và ROUGE-L (48.0), trong khi BLIP đạt ROUGE-L cao nhất toàn bảng (49.01). Nhìn chung, DenseNet201 (attention) là mô hình phù hợp nhất cho Image Captioning khi cân bằng cả chỉ số BLEU và ROUGE.

Link github: https://github.com/HatakaCder/Image-Captioning-in-Vietnamese

Mục lục

2 Giới thiệu

Trong thời đại công nghệ hiện nay, trí tuệ nhân tạo (AI) và học sâu (Deep Learning) đã trở thành những lĩnh vực tiên phong, mang lại nhiều ứng dụng thực tiễn trong đời sống. Nổi bật trong số đó là sự kết hợp giữa xử lý hình ảnh (Computer Vision) và xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP) đã mở ra những hướng nghiên cứu và ứng dụng mới mẻ, tiêu biểu là lĩnh vực Image Captioning (Tạo chú thích ảnh). Image Captioning là bài toán kết hợp giữa thị giác máy tính và xử lý ngôn ngữ tự nhiên, với mục tiêu tạo ra các câu văn bản mô tả nội dung trực quan của hình ảnh.

Qua một cuộc khảo sát [1], hiện nay việc sinh văn bản dựa trên hình ảnh hoặc video thu hút được sự quan tâm lớn và đạt được những thành tựu đáng kể trong nhiều lĩnh vực khác nhau như hỗ trợ người khiếm thị, hệ thống tìm kiếm hình ảnh thông minh, tự động hóa chú thích hình ảnh trong quản lý dữ liệu và thậm chí hỗ trợ giáo dục. Với xu hướng phát triển mạnh mẽ của AI, đề tài này hứa hẹn mang lại giá trị thực tiễn cao và tiếp tục là hướng nghiên cứu quan trọng trong tương lai.

Mặc dù các phương pháp tiên tiến hiện nay đã đạt được những kết quả ấn tượng, tuy nhiên hầu hết chúng chủ yếu tập trung vào ngôn ngữ Tiếng Anh. Điều này đặt ra một vấn đề trong việc nghiên cứu và phát triển các mô hình này trên các ngôn ngữ khác, đặc biệt là Tiếng Việt - một ngôn ngữ có nhiều đặc điểm ngữ pháp và ngữ nghĩa riêng biệt.

Trong nghiên cứu này, chúng tôi tập trung vào việc khai thác và mở rộng các phương pháp tạo chú thích ảnh trên bộ dữ liệu tiếng Việt, cụ thể là UIT-ViIC, một bộ dữ liệu chuyên biệt được xây dựng để hỗ trợ cho bài toán này. Kèm theo đó, chúng tôi sử dụng phương pháp truyền thống hiện nay được sử dụng phổ biến là CNN+LSTM, không có lớp chú ý và có lớp chú ý, để đánh giá được độ hiệu quả giữa các mô hình đối với dữ liệu Tiếng Việt.

Nghiên cứu này không chỉ giúp nhận diện những thách thức khi xử lý ngôn ngữ Tiếng Việt, mà còn tìm ra được phương pháp tối ưu trong việc tạo chủ thích hình ảnh trên ngôn ngữ Tiếng Việt.Từ đó áp dụng vào thực tiễn, cải thiện quy trình tạo chú thích ảnh tiếng Việt, thay thế việc tạo chú thích thủ công bằng các phương pháp tự động hiệu quả.

3 Công trình liên quan

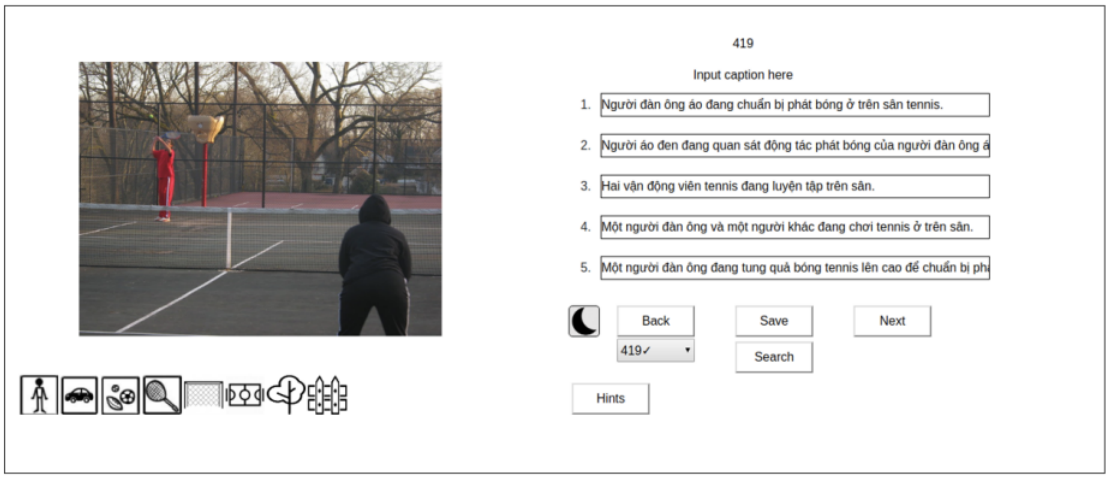
3.1 UIT-ViIC: Vietnamese Image Captioning Dataset

UIT-ViIC là một bộ dữ liệu được thiết kế cho bài toán Image Captioning với các câu mô tả bằng Tiếng Việt. Bộ dữ liệu này được xây dựng bởi nhóm nghiên cứu NLP từ Trường Đại học Công Nghệ Thông tin, Đại học Quốc gia Thành phố Hồ Chí Minh (University of Information Technology - VNUHCM). Đây là bộ dữ liệu đầu tiên đi đầu trong công cuộc xây dựng dữ liệu Tiếng Việt cho việc tạo chú thích ảnh [2].

UIT-ViIC bao gồm 3,850 hình ảnh liên quan đến thể thao chơi bằng bóng từ bộ dữ liệu MS-COCO 2017, cùng với 19,250 chú thích Tiếng Việt, mỗi hình ảnh chứa 5 câu mô tả.

Nhóm tác giả nhận thấy rằng việc sử dụng công cụ Google Dịch cho quá trình tạo bộ dữ liệu Tiếng Việt từ MS-COCO 2017 không được hiệu quả vì câu mô tả sau khi dịch từ Tiếng Anh sang Tiếng Việt không được tự nhiện và không bám sát được ngữ nghĩa từ câu Tiếng Anh gốc. Thế nên các câu mô tả được dịch thủ công từ câu mô tả MS-COCO 2017 Tiếng Anh sang Tiếng Việt và tuân thủ các luật dịch trong bài báo gốc của nhóm tác giả [2], được thực hiện bởi những người đã được huấn luyện về kiến thúc thể thao và những từ vựng chuyên môn.

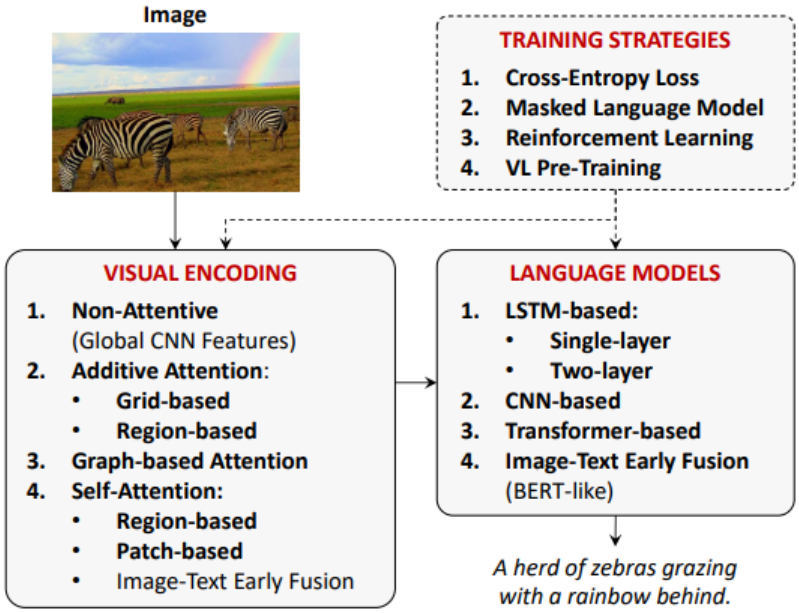
Vì tài nguyên hạn hẹp trong quá trình huấn luyện và xử lý bài toán, nên chúng tôi sử dụng bộ dữ liệu này để tập trung huấn luyện và đánh giá trên một mảng nhất định. Từ đó chúng tôi lựa chọn phương pháp hiệu quả nhất để phát triển thêm nhiều mảng khác nhau.



Hình 1: Một mẫu dữ liệu từ UIT-ViIC trích từ bài báo gốc

3.2 Các phương pháp tiếp cận Image Captioning

Qua quá trình khảo sát [1, 2], các hướng tiếp cận Image Captioning bao gồm 2 thành phần chính: Visual Encoding và Language Model.



Hình 2: Tổng quan các phương pháp tiếp cận Image Captioning

Khởi đầu cho bài toán Image Captioning, với sự xuất hiện của mô hình mạng học sâu CNN điển hình như AlexNet, ResNet và mô hình mạng học sâu sinh văn bản RNN, LSTM, phương pháp đầu tiên được công bố ở năm 2014 "Show and Tell"[4] với kiến trúc CNN+LSTM và tới thời điểm hiện tại vẫn là phương pháp được sử dụng phổ biến nhất vì triển khai đơn giản và đòi hỏi tài nguyên thấp hơn rất nhiều so với các mô hình VLP (Vision Language Pre-training).

Để cải tiến cho phương pháp ban đầu, một lớp chú ý (Attention Layer) được áp dụng giữa CNN và LSTM để giúp mô hình chú tâm sinh văn bản trên một vùng cụ thể trong ảnh. Khởi đầu với Additive Attention (Bahdanau Attention), cơ chế Attention này tính toán điểm attention giữa hidden state từ Decoder và các feature vectors của các vùng ảnh từ Encoder bằng cách sử dụng một mạng học sâu nhỏ (thường là một perceptron đa lớp - MLP), nhưng các điểm số attention này được tính theo từng ảnh riêng biệt, không xét đến mối quan hệ giữa các vùng ảnh với nhau.

Từ khi xuất hiện Transformer và các mô hình Transformer-based, Image-Text Early Fusion, Self-Attention được ra đời. Điểm attention được tính giữa chính các feature vectors của các vùng ảnh. Attention này giải quyết được vấn đề của Additive Attention, cho phép mô hình nắm bắt mối quan hệ giữa tất cả các vùng ảnh với nhau, không phụ thuộc vào vị trí, nhưng độ phức tạp cao hơn Additive Attention.

Một phiên bản Attention được xây dựng trên Self-Attention không lâu sau đó là Graph-based Attention (GAT), cơ chế là biểu diễn hình ảnh dưới dạng đồ thị, trong đó các nút (nodes) đại diện cho các vùng ảnh và các cạnh (edges) đại diện cho mối quan hệ giữa chúng. Sử dụng các mạng học sau đồ thị (Graph Neural Networks - GNNs) để lan truyền thông tin giữa các nút, từ đó tính toán điểm số attention. Theo nhóm tác giả GAT, lớp này cải thiện được độ chính xác và tận dụng thông tin về cấu trúc và mối quan hệ giữa các vùng ảnh tốt hơn Self-Attention nhưng đồng thời mô hình này tối tài nguyên nhiều nếu đồ thị lớn và cần phải định nghĩa rõ các cạnh và trọng số.

Với mô hình CNN+LSTM, chỉ cần sử dụng Cross-Entropy Loss trong quá trình huấn luyện Decoder. Với Transformer-based hoặc các mô hình Vision Language Pre-training, tùy thuộc vào cấu trúc của mô hình mà có độ đo nhất định (Cross-Entropy Loss, Masked Language Model, Reinforcement Learning, VL Pre-training).

4 Định nghĩa bài toán

4.1 Mô tả bài toán

- **Input:** Tập dataset bao gồm n ảnh số, mỗi ảnh có k câu mô tả tương ứng ($k \geq 1$). Một ảnh đầu vào I là một ma trận 3 chiều $I \in \mathbb{R}^{H \times W \times C}$. Trong đó:
  - H: Chiều cao của ảnh ($H \in \mathbb{N}^*$)
  - W: Chiều rộng của ảnh ($W \in \mathbb{N}^*$)
  - C: Số kênh màu của ảnh ($C = 3$)
- **Output:** Chuỗi văn bản $S = w_1, w_2, w_3, ..., w_T$, trong đó:
  - T: Độ dài của chuỗi văn bản S.
  - w_i: Từ thứ i trong chuỗi, được lấy từ một tập từ vựng V cố định.

  Tập từ vựng V được tạo tùy theo tokenizer của mô hình.

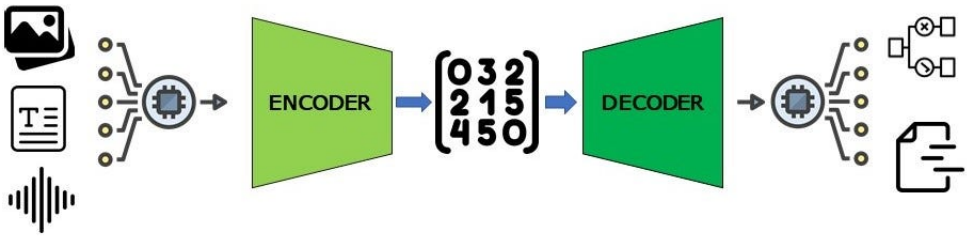
4.2 Phạm vi bài toán

Bài toán được thực hiện trên UIT-ViIC [2], bộ dữ liệu dành cho Image Captioning trong bối cảnh Tiếng Việt trên mảng thể thao có sử dụng bóng.

5 Phương pháp

5.1 Kiến trúc tổng quát: Encoder + Decoder

Hầu hết các mô hình Image Captioning đều bao gồm hai thành phần chính: **Encoder** (bộ mã hóa) và **Decoder** (bộ giải mã).



Hình 3: Kiến trúc Encoder+Decoder

- **Encoder** đóng vai trò trích xuất đặc trưng từ hình ảnh đầu vào, chuyển đổi thông tin từ dạng trực quan thành các vector đặc trưng giàu thông tin, biểu diễn ý nghĩa tổng quát của hình ảnh.
- **Decoder** sử dụng các vector đặc trưng này để phân tích và tạo ra câu mô tả, biến những đặc điểm trích xuất được thành chuỗi văn bản phù hợp, phản ánh nội dung hình ảnh một cách chính xác và tự nhiên.

Ngoài ra, giữa Encoder và Decoder, có thể bổ sung các thành phần để tối ưu hóa quá trình truyền tải thông tin. Chẳng hạn, **lớp chú ý (Attention)** giúp mô hình tập trung vào các vùng quan trọng trong ảnh, tăng cường khả năng nắm bắt chi tiết cần thiết cho từng bước sinh từ. Bên cạnh đó, một **lớp chuyển đổi (Projection Layer)** có thể được sử dụng để điều chỉnh vector đặc trưng từ Encoder, đảm bảo tương thích và hiệu quả khi làm đầu vào cho Decoder. Tùy thuộc vào phương pháp và bài toán cụ thể, việc thêm các lớp này sẽ mở ra những cách tiếp cận độc đáo và hiệu quả hơn.

5.2 CNN+LSTM

Mô hình CNN + LSTM là một trong những phương pháp kinh điển cho bài toán "Image Captioning", đặc biệt là trong giai đoạn trước khi mô hình Transformer trở nên phổ biến. Mô hình kết hợp mạng CNN để xử lý thông tin hình ảnh và mạng LSTM để xử lý chuỗi văn bản.

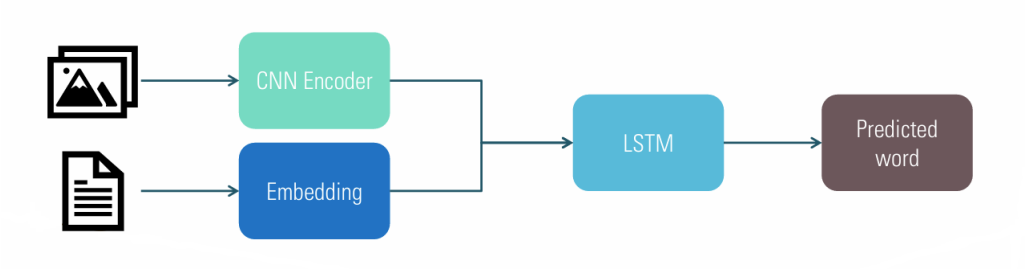
Trong phần Encoder, mạng **CNN (Convolutional neural network)** được sử dụng để trích xuất đặc trưng của ảnh đầu vào. Nhóm sử dụng các mạng CNN phổ biến như DenseNet201, VGG16, hoặc EfficientNetV2 vì khả năng trích xuất tính năng đáng tin cậy của chúng.

- **DenseNet201:** Một biến thể của DenseNet, được thiết kế để giải quyết các vấn đề trong huấn luyện mạng sâu, như sự biến mất gradient và mô hình hóa hiệu quả hơn.
- **VGG16:** Một mạng CNN nổi tiếng do nhóm nghiên cứu tại Đại học Oxford phát triển, được sử dụng trong các bài toán phân loại, phát hiện đối tượng và các ứng dụng liên quan đến thị giác máy tính.
- **EfficientNetV2:** Phiên bản cải tiến của EfficientNet, được thiết kế bởi Google, nhằm tối ưu hóa hiệu suất và tốc độ tính toán. Hiệu quả về tốc độ và tài nguyên tính toán, nhanh hơn và nhỏ hơn so với EfficientNet ban đầu. Tuy nhiên, mô hình cần các kỹ thuật tối ưu hóa hiện đại và phần cứng GPU mạnh để đạt hiệu quả tốt nhất.

Đối với phần Decoder, nhóm sử dụng **LSTM (Long Short Term Memory)** cho việc dự đoán câu mô tả dựa trên đặc trưng ảnh đầu vào và ngữ cảnh. Câu ngữ cảnh sẽ được mã hóa thành các vector embedding, sau đó kết hợp với đặc trưng ảnh, đóng vai trò là bước thời gian đầu tiên, để tạo thành input cho LSTM Decoder. Output của LSTM layer sau đó được đưa qua các lớp fully connected layer để ánh xạ các đặc trưng tuần tự sang không gian đầu ra.

Ngoài ra, nhóm còn sử dụng một số layer sau để tăng hiệu suất của mô hình:

- **Dropout layer:** Có vai trò làm giảm thiểu tình trạng overfitting bằng cách ngẫu nhiên "tắt"một số đơn vị (neurons) trong quá trình huấn luyện.
- **Add layer:** Kết hợp output của LSTM với CNN features để tránh mất mát đặc trưng về ảnh.



Hình 4: Sơ đồ mô hình CNN + LSTM

5.3 CNN+LSTM with Attention Mechanism

Một cách cải thiện hiệu suất cho mô hình CNN + LSTM truyền thống đó là sử dụng Attention Mechanism. Theo bài báo Show, Attend and Tell [5], phương pháp này cho kết quả tốt hơn các mô hình tiền nhiệm trên các tập dữ liệu chuẩn như MSCOCO, Flickr8k.

Trong đồ án này, nhóm sử dụng phương pháp Additive Attention (Bahdanau Attention) giúp mô hình tập trung vào các vùng quan trọng trong hình ảnh khi sinh từng từ trong chú thích. Phương pháp này kết hợp vector truy vấn q (query) và vector khóa k (key) thông qua phép cộng để tính trọng số chú ý α, sau đó dùng α để tính vector ngữ cảnh c. Cụ thể, đặc trưng ảnh (CNN features) sẽ đóng vai trò là khóa k, trong khi LSTM output sẽ được sử dụng như vector truy vấn q. Công thức cụ thể:

$$e_{ij} = \mathbf{v}^\top \tanh\left(\mathbf{W}_q \mathbf{q}_i + \mathbf{W}_k \mathbf{k}_j + \mathbf{b}\right)$$

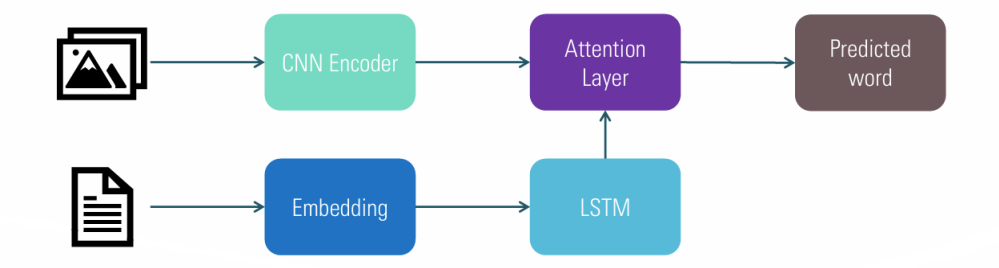
$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{j'} \exp(e_{ij'})}$$

$$\mathbf{c}_i = \sum_j \alpha_{ij} \mathbf{v}_j$$

Trong đó:

- e_{ij}: Điểm tương quan giữa query $\mathbf{q}_i$ và key $\mathbf{k}_j$.
- v: Vector trọng số.

- $\mathbf{W}_q, \mathbf{W}_k$: Ma trận trọng số áp dụng lên query và key.
- $\mathbf{q}_i, \mathbf{k}_j$: Query và key tại bước thời gian tương ứng.
- α_{ij}: Trọng số attention được tính từ điểm tương quan e_{ij}
- $\mathbf{c}_i$: Context vector tại bước thời gian i.



Hình 5: Sơ đồ mô hình CNN + LSTM kết hợp Attention Mechanism

5.4 ClipCap

Mô hình ClipCap sử dụng ViT (Vision Transformer) cho việc trích xuất đặc trưng hình ảnh + GPT-2 cho việc đọc hiểu embedding ảnh và chuyển đổi sang caption tiếng Việt. Cầu nối giữa encoder và decoder là một Mapping Network (mạng lưới chuyển đổi đầu ra của ViT sao cho phù hợp với GPT-2).

Encoder: Vision Transformer (ViT)

- ViT chia hình ảnh đầu vào thành các patch (mảnh nhỏ).
- Biến đổi các patch này thành các vector đặc trưng thông qua các lớp transformer.
- Kết quả của ViT là một vector đặc trưng biểu diễn toàn bộ hình ảnh, và sẽ được dùng làm "prefix"(tiền tố) cho GPT-2 trong quá trình sinh caption.

Mapping

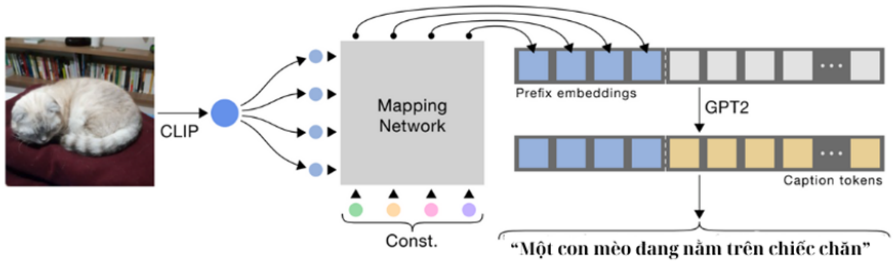
Embedding từ ViT và GPT-2 không có cùng không gian vector. Do đó, cần ánh xạ (mapping) embedding của ViT sang không gian GPT-2.

Hai phương pháp chính để ánh xạ:

- MLP (Multi-Layer Perceptron): Một mạng nơ-ron feed-forward đơn giản để ánh xạ embedding.
- Transformer Mapper: Một transformer để ánh xạ embedding, phù hợp khi cần mô hình hóa các mối quan hệ phức tạp giữa các phần embedding.

Decoder: GPT-2-vietnamesee

- GPT-2-vietnamese là một mô hình ngôn ngữ đã được pretrained dành cho tiếng việt dựa trên transformer, được tối ưu để sinh văn bản tự nhiên.
- GPT-2-vietnamese nhận các đặc trưng từ ViT (qua các vector đầu ra của encoder) và sinh một chuỗi văn bản (caption) mô tả hình ảnh.



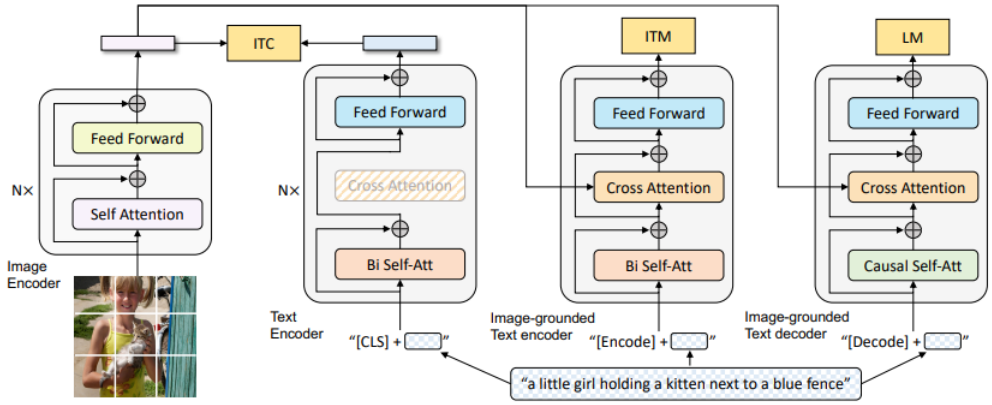
Hình 6: Sơ đồ kiến trúc ClipCap

5.5 BLIP

BLIP (Bootstrapping Language-Image Pre-training) là một Vision-Language Pre-training (VLP) xử lý linh hoạt đa tác vụ trong việc hiểu và sinh ngữ liệu từ hình ảnh. Các tác vụ bao gồm chú thích hình ảnh, truy vấn thông tin giữa hình ảnh và văn bản, VQA (Vision Question Answering). BLIP sử dụng một kỹ thuật "bootstrapping", nghĩa là sử dụng một mô hình ban đầu để tạo ra dữ liệu đào tạo tốt hơn, sau đó sử dụng dữ liệu đó để huấn luyện mô hình. Quá trình này lặp đi lặp lại, cải thiện chất lượng của mô hình theo từng bước.

Về mặt kiến trúc, BLIP kết hợp 3 khối Image Encoder, Text Encoder và Text Decoder nhằm xử lý đồng thời hình ảnh và ngôn ngữ. Các khối này không hoạt

động độc lập mà chia sẻ "kiến thức"với nhau, tạo nên sự đồng bộ trong việc hiểu và tạo ra thông tin. Trong đó Image Encoder và Text Encoder đảm nhiệm phần hiểu được dữ liệu và Text Decoder trả lại đầu ra tương ứng.



Hình 7: Sơ đồ kiến trúc BLIP

- **Image Encoder**: BLIP sử dụng ViT-B/ViT-L (Vision Transformer) làm bộ mã hóa hình ảnh. ViT chia hình ảnh thành các patches và xử lý chúng như các token, tương tự như cách xử lý các từ trong văn bản.
- **Text Encoder**: BLIP sử dụng BERT (Bidirectional Encoder Representations from Transformers) làm bộ mã hóa văn bản để xử lý đầu vào văn bản.
- **Image-Text Encoder-Decoder**: Nhận đầu vào từ cả Image Encoder và Text Encoder. Sử dụng cơ self-attention để chú ý các phần quan trọng của hình ảnh và cross-attention để kết hợp thông tin từ hình ảnh và văn bản. Phần Encoder nhằm học các biểu diễn chung giữa hình ảnh và văn bản còn Decoder chịu trách nhiệm tạo ra văn bản.

Với kiến trúc trên, BLIP được huấn luyện với ba mục tiêu:

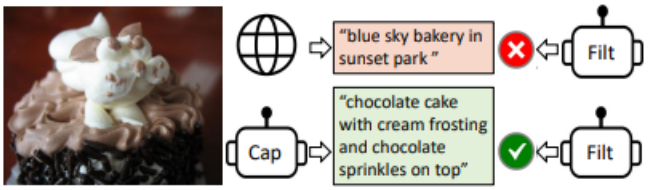
- **Image-Text Contrastive Learning (ITC)**: Mục tiêu này khuyến khích các biểu diễn của hình ảnh và văn bản tương ứng của nó gần nhau trong không gian nhúng, nếu các biểu diễn của các hình ảnh và văn bản không khớp nhau thì xa nhau.

- **Image-Text Matching (ITM)**: Mục tiêu này đào tạo mô hình phân biệt xem một cặp hình ảnh-văn bản có khớp nhau hay không, kết quả là hệ nhị phân 0 (không khớp) hoặc 1 (khớp),
- **Language Modeling (LM)**: Mục tiêu này đào tạo bộ giải mã để tạo ra các chuỗi văn bản mạch lạc, trôi chảy dựa trên biểu diễn hình ảnh.

Về mặt dữ liệu, dựa trên các khối chính trong kiến trúc để tạo ra hai mô-đun chính nhằm xử lý nhiễu trong bộ dữ liệu (Image-Text):

- **Filter**: Đây là Image-Text Encoder, kiểm tra xem ảnh và văn bản có liên quan không.
- **Captioner**: Đây là Image-Text Decoder, tạo ra các văn bản cho hình ảnh.

Ví dụ như một chuỗi văn bản trong bộ dữ liệu huấn luyện không hề liên quan tới bức ảnh tương ứng, Filter sẽ từ chối chuỗi văn bản này và Captioner sẽ tạo ra một chuỗi văn bản thay thế khác phù hợp hơn.



Hình 8: Cách thức hoạt động Captioner-Filter

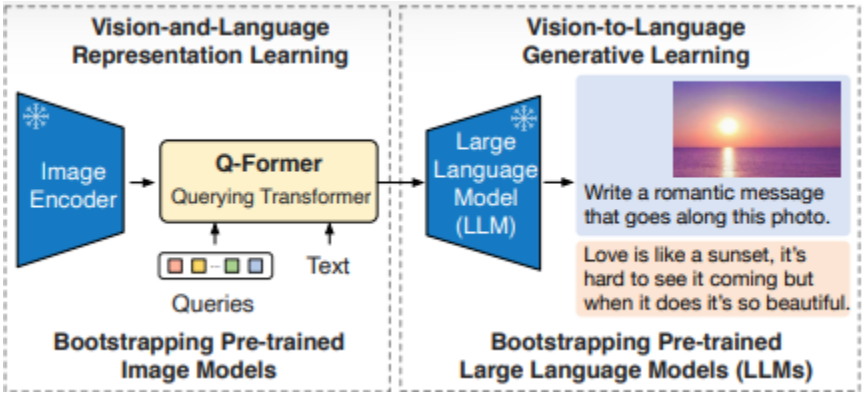
5.6 BLIP-2 + PEFT

5.6.1 BLIP-2

BLIP-2 là một phiên bản cải tiến của BLIP, tận dụng các mô hình chuyên xử lý hình ảnh (ViT) và các mô hình chuyên xử lý văn bản (LLM) vô cùng mạnh mẽ ở thời điểm hiện tại và tạo ra một kiến trúc mới "Query Transformer"là cầu nối giữa Image Encoder và LLM, giúp Image Encoder và LLM hiểu nhau.

BLIP-2 bao gồm ba thành phần chính:

- **Image Encoder**: sử dụng ViT (Vision Transformer) đã được đào tạo trước để trích xuất các đặc trưng từ hình ảnh. Các tham số được giữ nguyên trong quá trình đào tạo.
- **LLM Decoder**: sử dụng các mô hình ngôn ngữ lớn đã được đào tạo trước (OPT, FlanT5).
- **Q-Former (Querying Transformer)**: Cấu tạo gồm 2 khối Transformer: Image Transformer và Text Transformer.
  - **Image Transformer** xử lý các đặc trưng từ hình ảnh đầu vào dựa vào Learned Queries (các truy vấn học được), các query đóng vai trò như "đại diện"cho thông tin cần trích xuất từ hình ảnh, Transformer này được cấu tạo bởi các lớp self-attention, cross-attention và feed-forward network (FFN).
  - **Text Transformer** xử lý các thông tin đầu ra từ Image Transformer kết hợp với thông tin văn bản để huấn luyện qua nhiều lớp self-attention và feed forward network, từ đó đưa ra một vector đặc trưng hình ảnh đã được tinh chỉnh để đưa vào LLM Decoder.

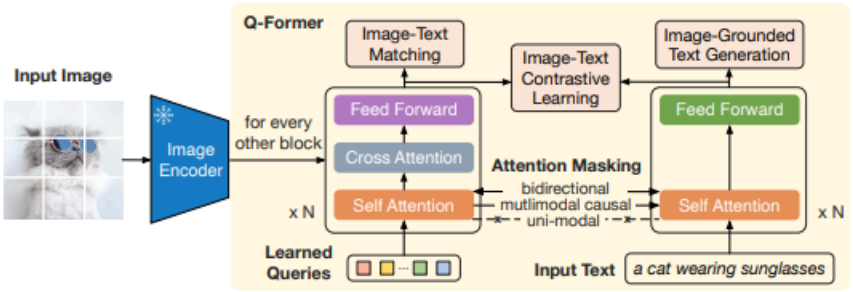


Hình 9: Sơ đồ kiến trúc BLIP-2

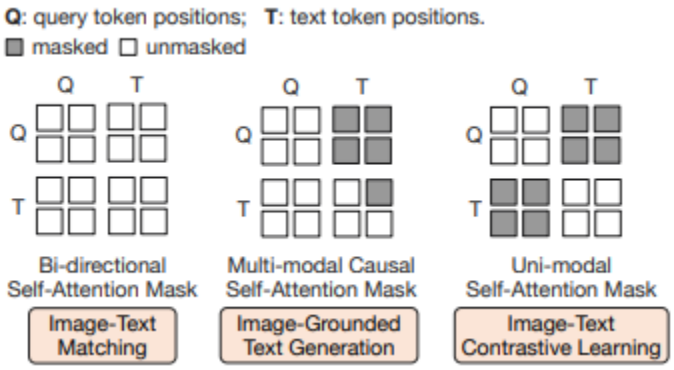
Q-Former được huấn luyện trong hai giai đoạn:

- **Representation Learning**: Q-Former học liên kết giữa đặc trưng hình ảnh và ngôn ngữ sao cho phần Query hiểu được đặc trưng nào của ảnh tương đương với đặc trưng của văn bản. Được đào tạo trên ba mục tiêu sau:

- **Image-Text Contrastive Learning (ITC)**: Tương tự như BLIP, mục tiêu này khuyến khích các biểu diễn đặc trưng của hình ảnh và văn bản tương ứng gần nhau. Sử dụng uni-modal self-attention mask nhằm không cho phép tương tác giữa các query và văn bản, tránh rò rỉ thông tin.
- **Image-Grounded Text Generation (ITG)**: Mục tiêu này tạo văn bản dựa trên ảnh cho trước. Ở khối này Q-Former không dùng trực tiếp thông tin của ảnh mà phải thông qua các query, thế nên query chỉ nhặt thông tin hữu ích nhất cho Text Transformer. Nhóm nghiên cứu áp dụng multimodal casual self-attention mask, cho phép văn bản được tương tác một chiều với các query để lấy thông tin.
- **Image-Text Matching (ITM)**: Tương tự như BLIP, mục tiêu này là làm sao để học được sự liên kết giữa ảnh và văn bản tương ứng xem có khớp với nhau hay không. Đây là bài toán nhị phân: 0 (không khớp) và 1 (khớp). Sử dụng Bi-directional self-attention mask, cho phép các query và text tương tác với nhau hoàn toàn.

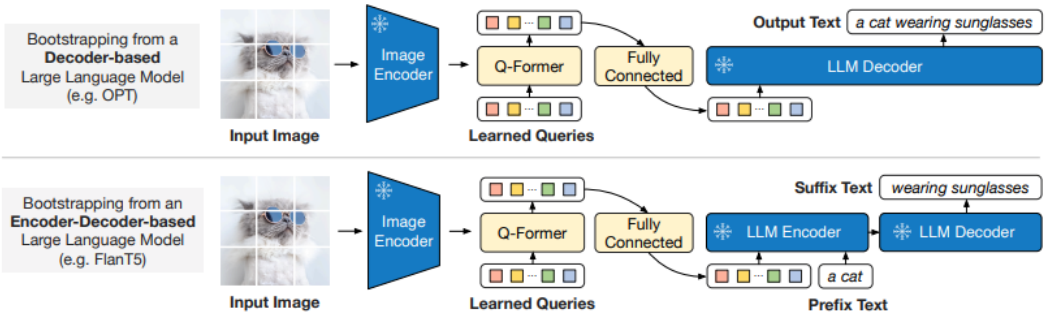


Hình 10: Mô hình Q-Former và BLIP-2 ở giai đoạn một



Hình 11: Chiến lược self-attention masking ở từng mục tiêu để kiểm soát sự tương tác query-text

- **Generative Learning**: sau giai đoạn 1, Q-Former đã có khả năng "hiểu"và "dịch"cho LLM dùng. Nhưng vẫn không thể dùng trực tiếp do khác biệt về chiều, thế nên đầu ra của Q-Former phải chạy qua một khối Fully Connected (FC) để biến đổi tuyến tính sao cho phù hợp làm đầu vào của LLM Decoder. Trọng tâm giai đoạn này giúp khối FC học làm sao từ queries embedding để có một đầu ra từ LLM Decoder mong muốn.



Hình 12: Mô hình BLIP-2 ở giai đoạn 2 (Trên: OPT, Dưới: FlanT5)

5.6.2 PEFT

PEFT (Parameter-Efficient Fine-Tuning): là một thư viện để điều chỉnh một cách hiệu quả các mô hình được đào tạo trước mà không cần tinh chỉnh tất cả các tham số vì nó quá tốn kém. Các phương pháp PEFT chỉ tinh chỉnh số lượng nhỏ tham số - giảm đáng kể chi phí tính toán và lưu trữ trong khi gần như

giữ được hiệu suất tương đương với mô hình được tinh chỉnh hoàn toàn. Điều này dễ dàng huấn luyện và tiếp cận nhiều người dùng hơn. [11, 12]

Cụ thể trong thư viện này, LoRA được sử dụng để tăng tốc quá trình huấn luyện trên mô hình lớn và sử dụng ít bộ nhớ hơn, giảm thiểu số lượng tham số đáng kể.

6 Thực nghiệm

6.1 Phương pháp đánh giá

6.1.1 BLEU

BLEU (Bilingual Evaluation Understudy) [9] là một thuật toán để đánh giá chất lượng văn bản, ban đầu được thiết kế để đo lường chất lượng dịch máy giữa các ngôn ngữ tự nhiên, trong **Image Captioning**, BLEU được áp dụng để đánh giá mức độ tương đồng giữa câu mô tả tự động sinh ra từ mô hình và các câu mô tả tham chiếu.

BLEU so sánh các **n-gram** (cụm từ gồn n từ liên tiếp) của câu dự đoán với các câu tham chiếu. Điểm số được tính toán dựa trên số lượng n-gram khớp chính xác giữa hai loại câu, với trọng số giảm dần cho các n-gram dài hơn.

BLEU không đánh giá ngữ pháp hay sự mạch lạc của câu mà chỉ đo lường mức độ trùng khớp về mặt từ vựng. Cách tính BLEU như sau:

$$BLEU = min(1, \frac{output\_length}{reference\_length}) \cdot \exp(\sum_{n=1}^{N} w_n \log p_n)$$

Trong đó:

- N: n-gram
- output_length: chiều dài của câu dự đoán.
- reference_length: chiều dài của câu tham chiếu.
- w_n: Trọng số cho các n-gram từ 1 tới N (thường là $\frac{1}{N}$).
- p_n: Precision của từng n-gram từ 1 tới N được tính như sau:

1 Tính **Count** = số lần khớp lớn nhất giữa n-grams câu dự đoán và các câu tham chiếu.

2 Tính **Ref** = số lần xuất hiện của n-grams câu dự đoán trong các câu tham chiếu

3 **Max Ref Count** = $max(Ref_1, Ref_2, ...Ref_m)$ với m= số lượng câu tham chiếu

4 **Count-Clip**=$min(Count, Max - Ref - Count)$

5 Cộng tất cả Count-Clip của các từ trong câu dự đoán

6 $p_n = \frac{Count-Clip}{w_t}$ với w_t là số n_gram trong câu dự đoán.

6.1.2 ROUGE-L

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [10] là một tập hợp các chỉ số được sử dụng để đánh giá chất lượng của văn bản sinh ra bởi các mô hình xử lý ngôn ngữ tự nhiên. Nó đo lường mức độ tương đồng giữa văn bản sinh ra và một hoặc nhiều văn bản tham chiếu. Các biến thể: ROUGE-N, ROUGE-L, ROUGE-W,...

Trong phạm vi đồ án này, nhóm sử dụng biến thể **ROUGE-L** để đánh giá khả năng sinh văn bản của các mô hình. ROUGE-L tập trung vào việc đo lường **Longest Common Subsequence (LCS)** giữa văn bản tham chiếu và văn bản được tạo ra. Lý do sử dụng LCS là vì nó bảo tồn thứ tự từ, một yếu tố quan trọng trong ngôn ngữ tự nhiên.

ROUGE-L được định nghĩa dựa trên 3 chỉ số chính: Precision (P), Recall (R) và F-Measure (F). Công thức cụ thể như sau:

$$P = \frac{\text{LCS}(X, Y)}{\text{len}(Y)}$$

$$R = \frac{\text{LCS}(X, Y)}{\text{len}(X)}$$

$$F = \frac{(1 + \beta^2) \cdot P \cdot R}{\beta^2 \cdot P + R}$$

Trong đó LCS(X, Y) là độ dài của subsequence chung dài nhất giữa hai chuỗi X (chuỗi tham chiếu) và Y (chuỗi sinh ra).

6.2 Dataset

UIT-ViIC được sử dụng trong quá trình huấn luyện và đánh giá kết quả của các mô hình. Bộ dữ liệu chứa các câu mô tả được tạo thủ công cho các hình ảnh liên quan đến thể thao chơi với bóng từ MS-COCO 2017.

UIT-ViIC chứa 19,250 câu mô tả Tiếng Việt cho 3,850 bức ảnh, mỗi bức ảnh chứa 5 câu mô tả tham chiếu tương ứng.

Tập train, validation, test được chia theo dataset gốc từ kaggle như sau:

- Train dataset: gồm 2695 ảnh và 13481 câu mô tả.
- Val dataset: gồm 924 ảnh và 4620 câu mô tả.
- Test dataset: gồm 231 ảnh và 1155 câu mô tả.

6.3 Tiền xử lý dữ liệu

Với từng hình ảnh trong dữ liệu, chúng tôi chuyển đổi kích thước hình ảnh về 224x224x3 nhằm đồng nhất kích thước đầu vào của các mô hình chúng tôi vừa nêu ở phần ... cho việc huấn luyện cũng như đánh giá mô hình.

Đồng thời với các câu mô tả tương ứng với từng hình ảnh, chúng tôi xử lý như sau:

- Lowercasing (Bóng đá → bóng đá)
- Sử dụng pyvi cho Word segmentation (bóng đá → bóng_đá)
- Xóa bỏ các dữ liệu rỗng
- Loại bỏ các ký tự đặc biệt
- Chuyển đổi sang tiếng Việt không dấu (chỉ áp dụng cho BLIP) nhằm tương thích với mô hình ngôn ngữ BERT, và sau đó sử dụng api Gemini để thêm dấu cho câu được tạo bởi mô hình BLIP

6.4 Cài đặt thực nghiệm

6.4.1 Môi trường thực nghiệm

Môi trường thực nghiệm của nhóm chúng tôi sử dụng **GPU T4x2 Kaggle** để huấn luyện, đánh giá các mô hình.

6.4.2 Cài đặt phương pháp

CNN+LSTM

Ở phần Encoder, sử dụng một trong 3 mô hình CNN DenseNet201, VGG16, EfficientNetV2 để trích xuất đặc trưng ảnh. Optimizer được sử dụng là **Adam**, với **learning rate** $\in$ (1e-3, 1e-4, 5e-4). Mô hình sử dụng loss function **categorical crossentropy**, phù hợp với yêu cầu dự đoán từ trong từ điển có sẵn. Quá trình huấn luyện được thực hiện với **epoch** = 50, **batch_size** = 64.

CNN+LSTM with Attention Mechanism

Tương tự với phương pháp CNN+LSTM, phần Encoder vẫn sử dụng một trong 3 mô hình CNN DenseNet201, VGG16, EfficientNetV2 để trích xuất đặc trưng ảnh. Optimizer được sử dụng là **Adam**, với **learning rate** $\in$ (1e-3, 1e-4, 5e-4). Mô hình sử dụng loss function **categorical crossentropy**, phù hợp với yêu cầu dự đoán từ trong từ điển có sẵn. Quá trình huấn luyện được thực hiện với **epoch** = 30, **batch_size** = 100.

ClipCap

Sử dụng clip-ViT-B-32 cho việc trích xuất đặc trưng của hình ảnh, tokenizer được lấy từ NlpHUST/gpt2-vietnamese, **Optimizer (AdamW)**: Sử dụng AdamW cho việc tối ưu hóa mô hình, mô hình sử dụng **loss function chuẩn** của GPT-2 để tối ưu hóa quá trình huấn luyện, với **learning rate=** 1e-4. Quá trình được thực hiện trong 5 epoch, **bath size** = 64.

BLIP

Sử dụng transfer learning cho BLIP base và tinh chỉnh với tham số sau: Bộ **tokenizer** được sử dụng là pyvi, **optimizer** được lựa chọn là **Adam**, với **learning rate=**$5e-5$. Quá trình huấn luyện được thực hiện trong **30 epoch**, áp dụng **early stopping** để dừng sớm nếu độ chính xác trên tập validation không cải thiện trong **2 epoch** liên tiếp. **batch size=** 3 và xáo trộn dữ liệu trước khi huấn luyện.

BLIP-2 (int8)

Sử dụng transfer learning cho BLIP-2 OPT-2.7b và tinh chỉnh với tham số sau: **LoraConfig** với $r = 16$, $lora\_alpha = 32$, $bias =$ 'none' và áp dụng $target\_modules$ bao gồm "q_proj"và "k_proj". Bộ **tokenizer** được sử dụng là pyvi, **optimier** được lựa chọn là **Adam**, với **learning rate**=$5e-4$. Quá trình huấn luyện được thực hiện trong **30 epoch**, áp dụng **early stopping** để dừng sớm nếu độ chính xác trên tập validation không cải thiện trong **2 epoch** liên tiếp. **batch size**= 3 và xáo trộn dữ liệu trước khi huấn luyện.

6.5 Kết quả thu được

	Type	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**	**ROUGE-L**
CNN + LSTM	DenseNet201	67.21	52.49	42.76	35.39	47.21
	VGG16	61.67	45.3	35.53	28.17	45.13
	EfficientNetV2	56.84	40.06	31.68	24.68	41.60
CNN + LSTM (attention)	DenseNet201	65.98	52.58	44.92	39.04	48.00
	VGG16	56.84	44.93	38.22	32.69	41.60
	EfficientNetV2	56.84	44.93	38.22	32.69	41.60
ClipCap	ViT-g + GPT2	43.42	40.25	37.24	35.63	42.12
BLIP	ViT-b + BERT-b	63.50	52.06	43.50	36.06	49.01
BLIP-2	ViT-g + OPT 2.7B + PEFT (int8)	53.78	43.20	35.90	29.47	48.06

Bảng 2: Kết quả thực nghiệm trên tập dataset ViIC (learning rate default)

	Type	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L
CNN + LSTM	DenseNet201	69.85	56.03	45.97	37.93	48.73
	VGG16	63.11	47.11	37.68	30.10	45.64
	EfficientNetV2	56.84	40.06	31.68	24.68	41.60
CNN + LSTM (attention)	DenseNet201	69.24	57.44	49.47	42.99	48.03
	VGG16	62.31	49.93	42.74	36.97	45.08
	EfficientNetV2	56.84	44.93	38.22	32.69	41.60
ClipCap	ViT-g + GPT2	43.42	40.25	37.24	35.63	42.12
BLIP	ViT-b + BERT-b	0	0	0	0	0
BLIP-2	ViT-g + OPT 2.7B + PEFT (int8)	23.77	20.6	17.78	14.75	39.88

Bảng 3: Kết quả thực nghiệm trên tập dataset ViIC (learning rate = 1e-4)

	Type	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L
CNN + LSTM	DenseNet201	68.00	54.82	46.04	38.77	48.13
	VGG16	63.13	47.11	37.45	29.76	45.22
	EfficientNetV2	56.84	40.06	31.68	24.68	41.60
CNN + LSTM (attention)	DenseNet201	65.68	52.62	45.19	39.29	48.18
	VGG16	54.75	42.66	36.06	30.85	41.43
	EfficientNetV2	56.84	44.93	38.22	32.69	41.60
ClipCap	ViT-g + GPT2	43.42	40.25	37.24	35.63	42.12
BLIP	ViT-b + BERT-b	0.00	0.00	0.00	0.00	0.00
BLIP-2	ViT-g + OPT 2.7B + PEFT (int8)	53.78	43.20	35.90	29.47	48.06

Bảng 4: Kết quả thực nghiệm trên tập dataset ViIC (learning rate = 5e-4)

	Type	Training time	Trainable Params
CNN + LSTM	DenseNet201	38m	1,570,119
	VGG16	38m	1,570,119
	EfficientNetV2	36m	1,570,119
CNN + LSTM (attention)	DenseNet201	18m40s	1,603,143
	VGG16	17m47s	1,603,143
	EfficientNetV2	23m42s	1,603,143
ClipCap	ViT-g + GPT2	6h44m	124,439,808
BLIP	ViT-b + BERT-b	4h17m	161,323,580
BLIP-2	ViT-g + OPT 2.7B + PEFT (int8)	2h55m	5,242,880

Bảng 5: Thời gian thực nghiệm trên tập dataset ViIC

→ **Nhận xét:** Qua kết quả thực nghiệm của các mô hình với các learning rate khác nhau, nhóm đã rút ra được những kết luận sau:

Hiệu quả mô hình trên ViIC:

- Các mô hình sử dụng DenseNet201 kết hợp với LSTM hoặc LSTM (attention) đạt hiệu suất cao nhất, đặc biệt khi tinh chỉnh lại learning rate, cho thấy khả năng trích xuất đặc trưng hình ảnh hiệu quả của DenseNet201 trong Image Captioning.

- Mô hình sử dụng VGG16 và EfficientNet cho kết quả thấp hơn, đặc biệt với chỉ số BLEU-4, phản ánh rằng chúng không phù hợp bằng DenseNet201 trong việc biểu diễn dữ liệu hình ảnh cho nhiệm vụ này.

- Các mô hình dựa trên ViT (Vision Transformer) như BLIP và ClipCap đạt hiệu suất tốt hơn ở một số khía cạnh nhưng chưa vượt qua CNN+LSTM trong tác vụ tạo chú thích chính xác.

Ảnh hưởng của Attention: Việc bổ sung Attention vào kiến trúc LSTM giúp cải thiện các chỉ số BLEU và ROUGE-L, nhấn mạnh tầm quan trọng của việc tập trung vào các vùng quan trọng trong hình ảnh khi sinh chú thích.

Ảnh hưởng của Learning rate: Việc thay đổi learning rate ảnh hưởng lớn đến hiệu suất của mô hình, đặc biệt trong trường hợp của phương pháp

CNN+LSTM, cho thấy tầm quan trọng của việc lựa chọn learning rate để cải thiện hiệu suất mô hình. Với BLIP, có thể learning rate lớn hơn mặc định (5e-5) và sử dụng transfer learning, mô hình này không thể sinh ra được chú thích, từ đó không thể đánh giá được mô hình này. Với BLIP-2, việc giảm learning rate làm mô hình đi về điểm hội tụ hoặc bị rơi vào điểm cực tiểu địa phương, độ chính xác giảm đi so với learning rate mặc định (5e-4).

Các mô hình tiên tiến: Mặc dù BLIP-2 (sử dụng ViT + OPT) cho thấy cải tiến trong một số chỉ số, cho thấy tiềm năng của các kiến trúc Transformer. Tuy nhiên, hiệu suất tổng thể vẫn chưa vượt qua CNN+LSTM khi không có tối ưu hóa cụ thể.

Khó khăn gặp phải:

- Tối ưu hóa mô hình: Việc lựa chọn kiến trúc, tham số phù hợp để tối ưu hóa mô hình là một thử thách, đặc biệt với các mô hình phức tạp như ViT.
- Giới hạn phần cứng: Khi huấn luyện các mô hình lớn sử dụng Transformer, thời gian huấn luyện và dự đoán lâu hơn so với mô hình thông thường.
- Tài nguyên thực nghiệm giới hạn, chưa khai phá hết mô hình đầy đủ của BLIP-2.

Hướng phát triển: Tối ưu hóa tham số phù hợp với mô hình sử dụng Transformer như ClipCap, BLIP và BLIP-2. Thay đổi mô hình ngôn ngữ cụ thể phù hợp với ngôn ngữ tiếng Việt như PhoBERT, ViSoBERT và tối ưu hóa cụ thể phù hợp với bài toán.

7 Bảng phân công công việc

Thành viên	**Phân công nhiệm vụ**	**Mức độ hoàn thành**
Phạm Quốc Việt	- Quản lý, phân công nhiệm vụ cho các thành viên nhóm. - Huấn luyện mô hình BLIP, BLIP-2 và triển khai trên demo. - Lên ý tưởng đề tài và tìm kiếm các bài báo khoa học phù hợp để tham khảo cho đề tài. - Xây dựng ứng dụng trực quan Image Captioning trên nền tảng web sử dụng Flask và Bootstrap5	100%
Trần Hoài An	- Xây dựng và huấn luyện mô hình CNN+LSTM, CNN+LSTM with attention. - Thực hiện slide thuyết trình của nhóm. - Triển khai mô hình CNN+LSTM, CNN+LSTM with attention trên demo	95%
Lường Đại Phát	Huấn luyện mô hình CLipCap (kết hợp ViT-g + Gpt2-Vietnamese) - Triển khai mô hình ClipCap trên demo	90%

8 Tài liệu tham khảo

[1] Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Silvia Cascianelli, Giuseppe Fiameni, Rita Cucchiara: From Show to Tell: A Survey on Deep Learning-Based Image Captioning. IEEE Trans. Pattern Anal. Mach. Intell. 45(1): 539-559 (2023).

[2] Taraneh Ghandi, Hamidreza Pourreza, Hamidreza Mahyar: Deep Learning Approaches on Image Captioning: A Review. ACM Comput. Surv. 56(3): 62:1-62:39 (2024)

[3] Quan Hoang Lam, Quang-Duy Le, Kiet Van Nguyen, Ngan Luu-Thuy Nguyen: UIT-ViIC: A Dataset for the First Evaluation on Vietnamese Image Captioning. ICCCI 2020: 730-742.

[4] Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan: Show and tell: A neural image caption generator. CVPR 2015: 3156-3164

[5] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, Yoshua Bengio: Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. ICML 2015: 2048-2057

[6] Ron Mokady, Amir Hertz, Amit H. Bermano: ClipCap: CLIP Prefix for Image Captioning. CoRR abs/2111.09734 (2021)

[7] Junnan Li, Dongxu Li, Caiming Xiong, Steven C. H. Hoi: BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. ICML 2022: 12888-12900

[8] Junnan Li, Dongxu Li, Silvio Savarese, Steven C. H. Hoi: BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. ICML 2023: 19730-19742

[9] Kishore Papineni, Salim Roukos, Todd Ward, Wei-Jing Zhu: Bleu: a Method for Automatic Evaluation of Machine Translation. ACL 2002: 311-318

[10] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In Text Summarization Branches Out, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

[11] Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, Sai Qian Zhang: Parameter-Efficient Fine-Tuning for Large Models: A Comprehensive Survey. CoRR abs/2403.14608 (2024)

[12] Hugging Face PEFT Documentation.