



ĐỒ ÁN CUỐI KỲ
CS331.M21.KHCL
GVHD: TS. Nguyễn Vinh Tiệp

ClipCap
Fast and Simple Image Captioning model using
CLIP & GPT-2

Ngày 3 tháng 7 năm 2022

Nhóm thực hiện:
Lương Phạm Bảo 19521242
Nguyễn Gia Thông 19520993
Phạm Ngọc Dương 19521412

MỤC LỤC

1 Giới thiệu	2
1.1 Bài toán Image Captioning	2
1.1.1 Dataflow	2
1.2 Ứng dụng thực tế	2
1.3 Thách thức	3
1.4 Một số cách tiếp cận trước	3
1.4.1 Các cách tiếp cận liên quan	3
1.5 Hướng cải tiến Visual Encoder	4
1.5.1 Non-Attentive	4
1.5.2 Additive Attention	4
1.5.3 Graph based Attention	4
1.5.4 Self Attention	5
1.6 Hướng cải tiến Language Model	5
1.6.1 LSTM based	6
1.6.2 CNN based	6
1.6.3 Transformer và Bert	6
1.7 Hướng cải tiến Training Strategy	6
1.7.1 Các model SOTA	7
1.7.2 Model sử dụng Visual Encoder và Textual Encoder	7
1.8 Tổng quát về mô hình ClipCap	7
1.8.1 Abstract	7
1.8.2 Method	8
2 Kiến trúc	8
2.1 CLIP	9
2.1.1 Giới thiệu	9
2.1.2 Cơ chế hoạt động của CLIP	10
2.1.3 CLIP trong ClipCap	11
2.2 GPT-2	11
2.3 Language model fine-tuning	12
2.3.1 Apdater-tuning	12
2.3.2 Prompting	12
2.3.3 Prefix-tuning	13
2.4 Mapping Network	14
2.4.1 Transformer	14
3 Hạn chế	15
4 Kết quả thực nghiệm	15
4.1 Dataset	15
4.2 Demo	16
4.3 Training	18
4.4 Kết quả	18

1 Giới thiệu

1.1 Bài toán Image Captioning

- Image Captioning là quá trình tạo ra câu mô tả (caption) cho một hình ảnh dựa trên ngôn ngữ tự nhiên.
- Bài toán yêu cầu xác định được các đối tượng quan trọng, thuộc tính và mối quan hệ của chúng trên tổng thể hình ảnh.
- Câu mô tả cần chính xác về mặt ngữ pháp/ngữ nghĩa và mạch lạc về nội dung.

1.1.1 Dataflow

Input: một bức ảnh về các sự vật, đối tượng.

Output: caption đầy đủ và hợp lệ mô tả bức ảnh.



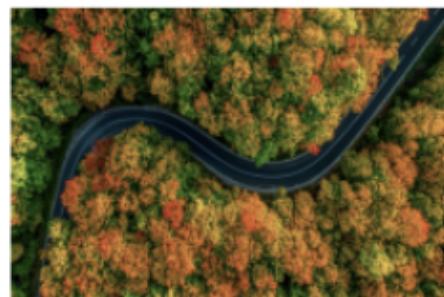
A politician receives a gift from politician.



A collage of different colored ties on a white background.



Silhouette of a woman practicing yoga on the beach at sunset.



Aerial view of a road in autumn.

Hình 1: Dataflow

1.2 Ứng dụng thực tế



"người đàn ông mặc áo đen đang chơi guitar."



"công nhân xây dựng mặc áo bảo hộ màu cam đang làm việc trên đường."



"hai cô gái đang chơi đồ chơi lego."

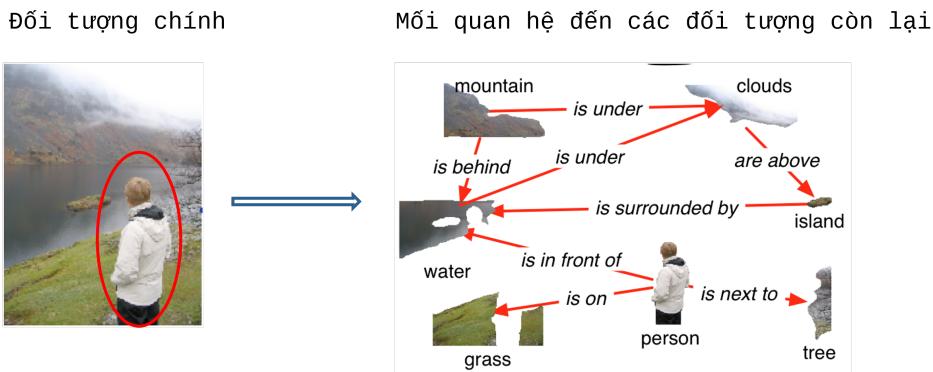
Hình 2: Image captioning và những ứng dụng đời thực

- Tạo mô tả cho hình ảnh.
- Tích hợp vào các công cụ tìm kiếm hình ảnh dựa trên mô tả.

- Hỗ trợ người khuyết tật.
- Xây dựng hệ thống truy vấn hình ảnh dựa trên nội dung (CBIR).

1.3 Thách thức

- **Semantic Understanding:** Là sự hiểu biết về mặt ý nghĩa của bức ảnh. Khía cạnh đơn giản ta có thể giải quyết bằng các bài toán như object detection, classification, ngoài ra ta cần phải hiểu nhiều hơn về các mối quan hệ giữa các mối quan hệ giữa các object trong ảnh.



Hình 3: Semantic Understanding

- **Nhiều caption thay thế:** Với cùng một hình ảnh, ta có thể có rất nhiều cách mô tả khác nhau.



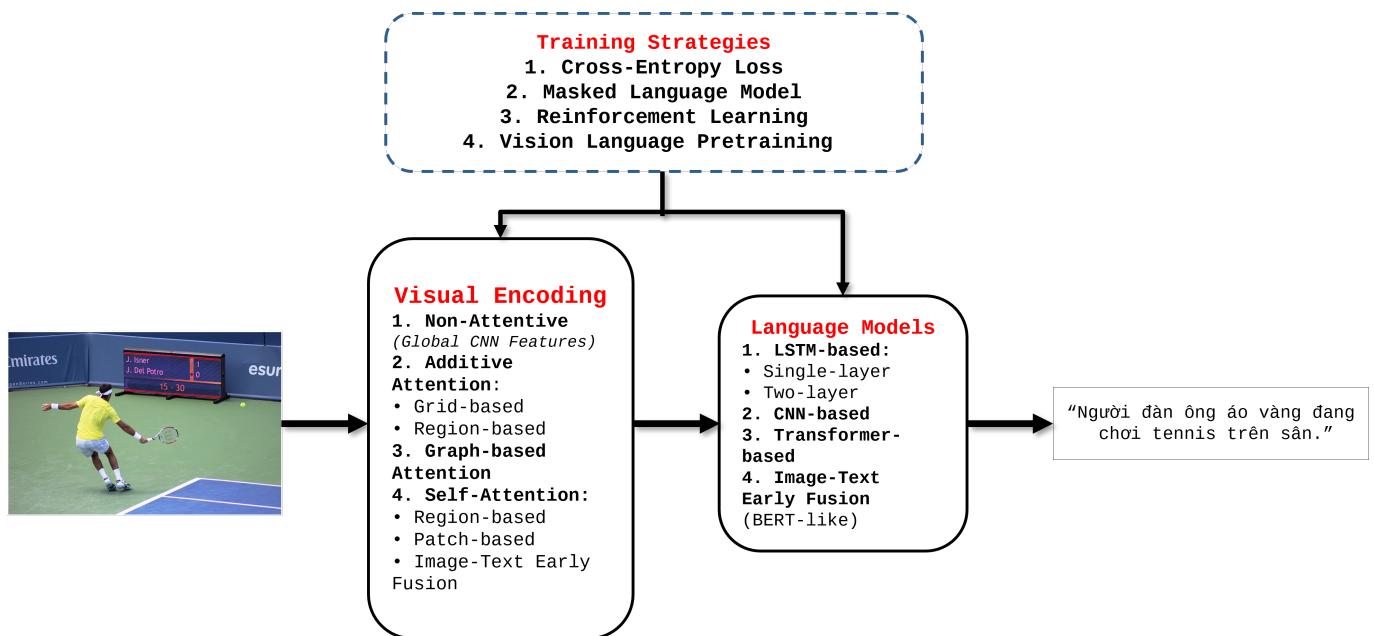
Caption 1: Người đàn ông mặc áo khoác đang nhìn xuống hồ.

Caption 2: Một người áo trắng, tóc vàng đứng trước mặt hồ.

Hình 4: Alternative captions

1.4 Một số cách tiếp cận trước

1.4.1 Các cách tiếp cận liên quan



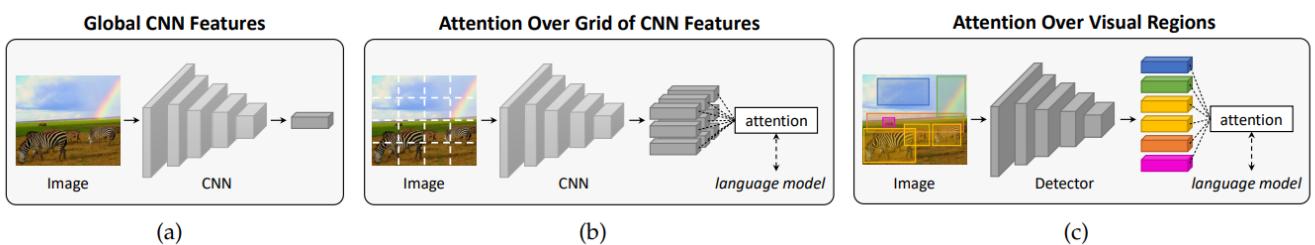
Hình 5: Relevant approaches

Với bài toán Image captioning ta thấy rằng các cách giải quyết cho bài toán trên sẽ gồm 3 phần: Visual Encoder dùng để biểu diễn hình ảnh input đầu vào thành các feature vector, Language model dùng để dự đoán output là caption dựa trên các mô hình mạng hồi tiếp (ta xem mỗi từ là mỗi time step), Và chiến lược huấn luyện tương tự như các bài toán classification ta cũng sử dụng mô hình Cross -Entropy Loss. Vì thế để cải thiện các mô hình cho bài toán image captioning, ta có thể lựa chọn cải tiến một hoặc nhiều module cho bài toán trên

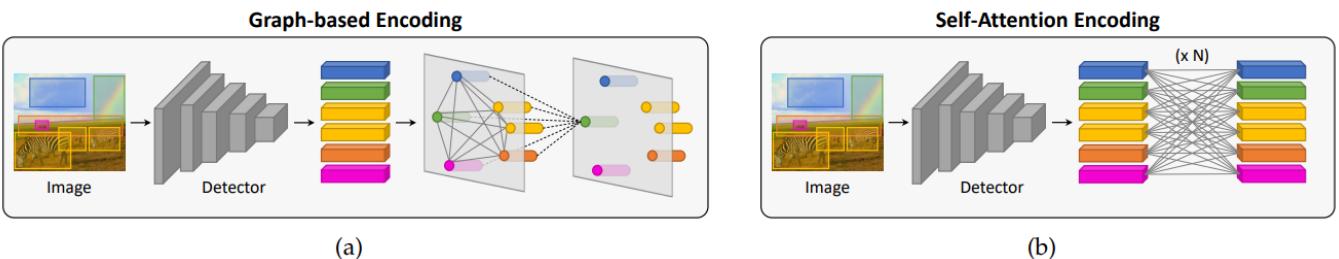
1.5 Hướng cải tiến Visual Encoder

Dù các mô hình CNN, deep learning cho các lĩnh vực deep learning đã phát triển rất nhiều, việc tạo ra cách biểu diễn hiệu quả cho các hình ảnh trực quan thật sự là một câu hỏi vẫn không dễ giải quyết. Các cách tiếp cận hiện tại để mã hóa trực quan có thể được phân loại là thuộc bốn loại chính:

- Non-Attentive
- Additive Attention
- Graph based Attention
- Self Attention



Hình 6: (a) Global CNN features; (b) fine-grained features extracted from the activation of a convolutional layer, together with an attention mechanism guided by the language model; (c) image region features coming from a detector, together with an attention mechanism



Hình 7: Summary of the two most recent visual encoding strategies for image captioning

1.5.1 Non-Attentive

Ta thấy rằng giống với các bài toán Infomation Retrieval hoặc CBIR, ta có thể sử dụng các mô hình pretrained như VGG, Resnet, Alexnet để trích xuất các tính năng cấp cao của ảnh Input sau khi loại bỏ các lớp FC. Lợi thế chính của việc sử dụng các global feature CNN này nằm ở sự đơn giản và gọn nhẹ của chúng trong cách biểu diễn, bao hàm khả năng trích xuất và cô đọng thông tin từ toàn bộ đầu vào và để xem xét bối cảnh tổng thể của hình ảnh. Tuy nhiên, mô hình này cũng dẫn đến nén quá nhiều thông tin và thiếu chi tiết, khiến cho một mô hình image caption khó có thể tạo ra các mô tả chi tiết

1.5.2 Additive Attention

Được thúc đẩy bởi những hạn chế của các global representation rằng thiếu khả năng biểu diễn chi tiết, phương pháp cho hướng huống cận này làm tăng mức độ chi tiết mức độ mã hóa trực quan đã sử dụng 2D activation map thay cho 1D activation map vector đặc trưng để mang lại cấu trúc không gian trực tiếp trong mô hình ngôn ngữ. Dựa trên ý tưởng từ bài toán machine translation, cộng đồng cho bài toán image captioning đã áp dụng cơ chế attention cho kiến trúc image captioning và cụ thể là giai đoạn mã hóa các tính năng hình ảnh thay đổi theo thời gian, cho phép linh hoạt hơn và độ chi tiết tốt hơn.

1.5.3 Graph based Attention

Để cải thiện quá trình việc encoding các region image và biểu diễn mối quan hệ giữa các object, một số nghiên cứu xem xét việc sử dụng graph được xây dựng trên region image để làm phong phú và đa dạng thêm sự biểu diễn bằng cách sử dụng các semantic và spatial connect. Việc sử dụng encoding dựa graph mang lại một cơ chế để tận dụng mối quan hệ giữa các đối tượng được phát

hiện, cho phép trao đổi thông tin trong các nút liền kề. Hơn nữa, nó cho phép tích hợp một cách liền mạch thông tin ngữ nghĩa từ bên ngoài.

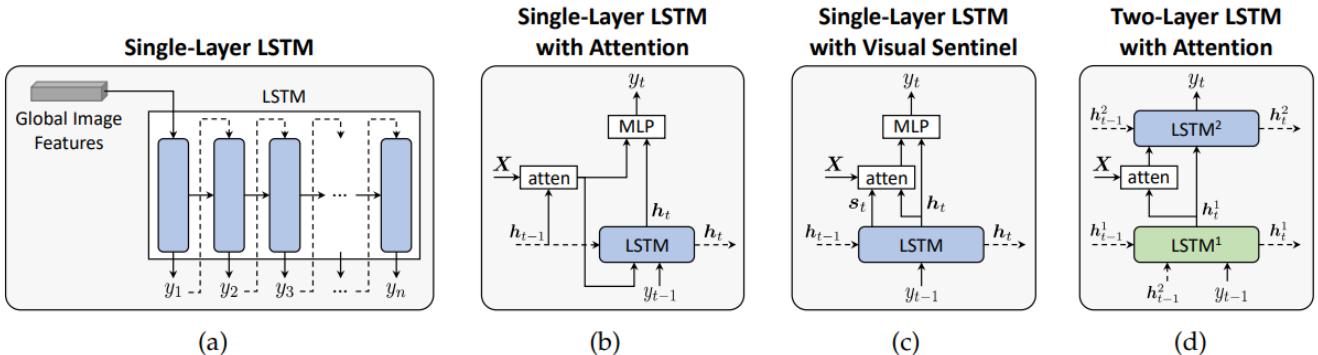
1.5.4 Self Attention

Self-attention là một cơ chế chú ý trong đó mỗi yếu tố của một tập hợp được kết nối với tất cả các tập hợp khác và điều đó có thể được chấp nhận để tính toán một biểu diễn tinh chỉnh của cùng một tập hợp các phần tử thông qua các residual connection. Nó được giới thiệu lần đầu tiên bởi Vaswani và các cộng sự cho machine translation và machine understanding kiến trúc Transformer và các biến thể của nó, có thống trị lĩnh vực NLP và sau đó là Thị giác máy tính.

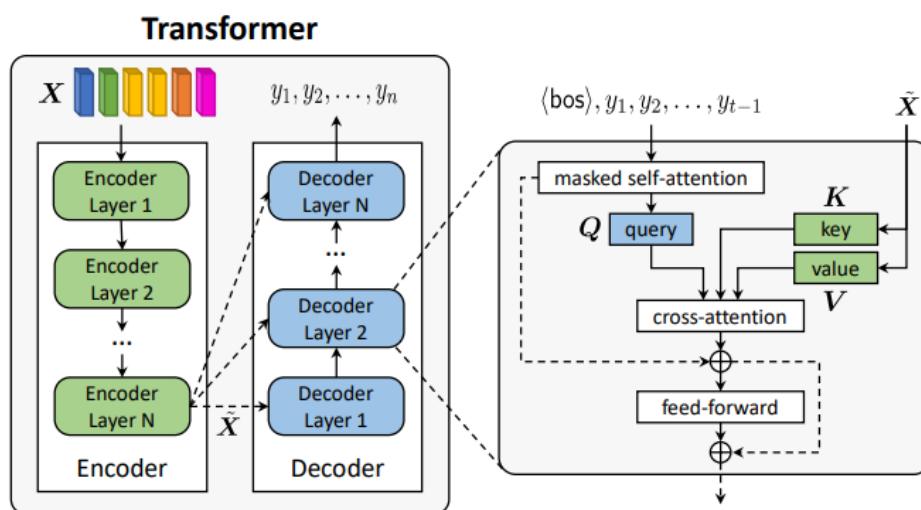
1.6 Hướng cải tiến Language Model

Mục tiêu của mô hình ngôn ngữ là dự đoán xác suất của một chuỗi các từ nhất định xảy ra trong một câu. Như vậy, nó là một thành phần tối quan trọng trong bài toán image captioning, vì nó mang lại khả năng hiểu và biểu diễn ngôn ngữ như một bài toán NLP thông thường. Các chiến lược mô hình hóa ngôn ngữ chính được áp dụng cho bài toán gồm 4 nhóm chính:

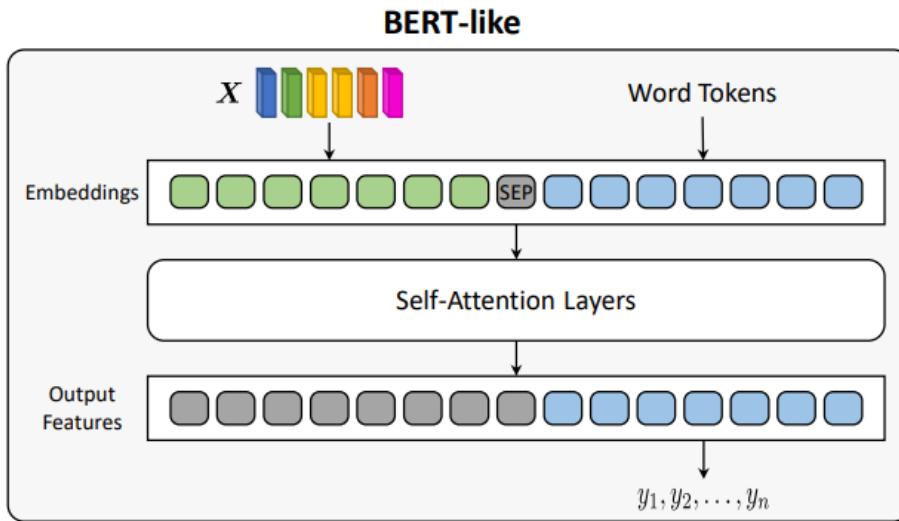
- LSTM-based approach
- CNN-based approach
- Transformer-based
- Image-text early-fusion (Bert)



Hình 8: LSTM-based language modeling strategies



Hình 9: Schema of the Transformer-based language model



Hình 10: Schema of a BERT-like language model

1.6.1 LSTM based

Do output của bài toán là một câu caption, nên dễ thấy rằng ta có thể sử dụng các mô hình mạng hồi tiếp như RNN, LSTM, GRU,... Các mạng LSTM có input là các visual encoder output được sử dụng làm hidden state ban đầu của LSTM, sau đó tạo caption đầu ra. Tại mỗi bước thời gian, một word được dự đoán bằng cách áp dụng activation softmax trên phép chiếu hidden state vào một vectơ có cùng kích thước với bộ vocab sử dụng. Trong quá trình training, các từ input được lấy từ ground truth sentence, trong khi suy luận, từ input là những từ được tạo ở bước trước đó. Ngoài ra để chất lượng các câu được nâng cao về tính mạch lạc cũng như chính xác ngữ pháp, ta có thể sử dụng nhiều lớp LSTM để có thể lấy được nhiều thông tin hơn từ biểu diễn visual encoder.

1.6.2 CNN based

Một cách tiếp cận đáng được đề cập cho image captioning được đề xuất bởi Aneya và các công sự, sử dụng Convolutional làm mô hình ngôn ngữ. Ý tưởng chính sử dụng global feature vectors CNN được kết hợp với Transformer. Việc tạo caption được thực hiện thông qua cơ chế self-attention masked đối với các generated token và cross attention với các feature image được encode với tính năng word embedding và được cung cấp cho CNN, hoạt động trên tất cả các từ song song trong quá trình đào tạo và tuân tự trong sự suy luận.

Mặc dù lợi thế rõ ràng của việc đào tạo song song, việc sử dụng toán tử tích chập trong các mô hình ngôn ngữ không trở nên phổ biến do hiệu suất kém và sự ra đời của kiến trúc Transformer.

1.6.3 Transformer và Bert

Chúng ta có thể xem các bài toán language model như các Seq2Seq. Trước khi Google công bố bài báo về Transformers (Attention Is All You Need), hầu hết các tác vụ xử lý ngôn ngữ tự nhiên, đặc biệt là dịch máy (Machine Translation) sử dụng kiến trúc Recurrent Neural Networks (RNNs). Điểm yếu của phương pháp này là rất khó bắt được sự phụ thuộc xa giữa các từ trong câu và tốc độ huấn luyện chậm do phải xử lý input tuần tự. Vì vậy Transformer với việc đưa các input vào cùng một lúc sử dụng một ý tưởng đột phá, self-attention. Và từ đó các mô hình ngôn ngữ lớn GPT, Bert ra đời. BERT được viết tắt của Bidirectional Encoder Representations from Transformers, một kiến trúc mới cho lớp bài toán Language Representation được Google công bố vào đầu tháng 11 năm 2018. Không giống như các mô hình trước đó, BERT được thiết kế để đào tạo ra các vector đại diện cho ngôn ngữ văn bản thông qua ngữ cảnh 2 chiều của chúng. Kết quả là, vector đại diện được sinh ra từ mô hình BERT được tính chỉnh với các lớp đầu ra bổ sung đã tạo ra nhiều kiến trúc cải tiến đáng kể cho các nhiệm vụ xử lý ngôn ngữ tự nhiên như Question Answering, Language Inference,... mà không cần thay đổi quá nhiều từ các kiến trúc cũ. Vì thế các mô hình image captioning sử dụng Bert đã đạt được hiệu suất vượt trội và là SOTA hiện nay.

1.7 Hướng cải tiến Training Strategy

Một mô hình image captioning thường được tạo caption dựa trên từng từ một bằng cách tính các thông tin dựa trên từ ở phía trước và ảnh. Ở mỗi bước, output từ được lấy mẫu từ một phân phối đã học trên bộ từ vựng. Trong giải pháp đơn giản nhất cho giai đoạn decode, ta lựa chọn từ có xác suất cao nhất là output. Hạn chế chính của cách này là có thể tích lũy loss function rất nhanh qua các bước suy luận các từ. Để giảm bớt nhược điểm này, một chiến lược hiệu quả là sử dụng beamsearch (tìm kiếm chùm) thay vì xuất ra từ với xác suất cao nhất tại mỗi time step, duy trì k candidates (những người có xác suất cao nhất ở mỗi time step) và cuối cùng là kết quả có thể xảy ra nhất. Trong quá trình training, mô hình tạo caption phải học cách dự đoán đúng xác suất của các từ xuất hiện trong câu caption. Để đạt được điều này, cách đào tạo phổ biến nhất là các chiến lược dựa trên:

Cross-entropy loss, Masked language model, Reinforcement learning, Vison-language pretraining.

Trong đó Masked language model (Bert) và VLP đang đem lại hiệu suất cao nhất vì các mô hình trên, đặc biệt là VLP đã được huấn luyện trên các tập dữ liệu lớn và đa tác vụ cũng như kết hợp tốt hơn thông tin từ image và caption. Đây là hướng tiếp cận mới nhất cũng như hiệu quả nhất cho các bài image captioning và có nhiều phương pháp mới đạt SOTA như VINVL, OCSAR, VLP,...

1.7.1 Các model SOTA

Model	Visual Encoding				Language Model			Training Strategies				Main Results			
	Global	Grid	Regions	Graph	Self-Attention	RNN/LSTM	Transformer	BERT	XE	MLM	Reinforce	VL Pre-Training	BLEU-4	METEOR	CIDEr
VinVL [83]		✓			✓				✓	✓	✓	✓	41.0	31.1	140.9
Oscar [80]		✓			✓				✓	✓	✓	✓	41.7	30.6	140.0
Unified VLP [81]		✓			✓				✓	✓	✓	✓	39.5	29.3	129.3
AutoCaption [87]		✓			✓	✓			✓	✓	✓	✓	40.2	29.9	135.8
RSTNet [74]		✓			✓				✓	✓	✓	✓	40.1	29.8	135.6
DLCT [73]		✓	✓		✓				✓	✓	✓	✓	39.8	29.5	133.8
DPA [70]		✓			✓		✓			✓	✓	✓	40.5	29.6	133.4
X-Transformer [68]		✓			✓				✓	✓	✓	✓	39.7	29.5	132.8
NG-SAN [66]		✓			✓				✓	✓	✓	✓	39.9	29.3	132.1
X-LAN [68]		✓			✓				✓	✓	✓	✓	39.5	29.5	132.0
GET [72]		✓			✓				✓	✓	✓	✓	39.5	29.3	131.6
M^2 Transformer [69]		✓			✓				✓	✓	✓	✓	39.1	29.2	131.2
AoANet [67]		✓			✓				✓	✓	✓	✓	38.9	29.2	129.8
CPTR [77]					✓				✓	✓	✓	✓	40.0	29.1	129.4
ORT [65]		✓			✓				✓	✓	✓	✓	38.6	28.7	128.3
CNM [63]		✓			✓				✓	✓	✓	✓	38.9	28.4	127.9
ETA [64]		✓			✓				✓	✓	✓	✓	39.9	28.9	127.6
GCN-LSTM+HIP [60]		✓	✓			✓			✓	✓	✓	✓	39.1	28.9	130.6
MT [59]		✓	✓			✓			✓	✓	✓	✓	38.9	28.8	129.6
SGAE [58]		✓	✓			✓			✓	✓	✓	✓	39.0	28.4	129.1
GCN-LSTM [55]		✓	✓			✓			✓	✓	✓	✓	38.3	28.6	128.7
VSUA [56]		✓	✓			✓			✓	✓	✓	✓	38.4	28.5	128.6
SG-RWS [52]		✓				✓			✓	✓	✓	✓	38.5	28.7	129.1
LBPF [50]		✓				✓			✓	✓	✓	✓	38.3	28.5	127.6
AAT [51]		✓				✓			✓	✓	✓	✓	38.2	28.3	126.7
CAVP [53]		✓				✓			✓	✓	✓	✓	38.6	28.3	126.3
Up-Down [45]		✓				✓			✓	✓	✓	✓	36.3	27.7	120.1
RDN [49]		✓				✓			✓	✓	✓	✓	36.8	27.2	115.3
Neural Baby Talk [85]		✓				✓			✓	✓	✓	✓	34.7	27.1	107.2

Hình 11: SOTA methods

1.7.2 Model sử dụng Visual Encoder và Textual Encoder

Thách thức đặt ra: Làm thế nào để các thông tin ngữ nghĩa trích xuất từ hình ảnh (from the Visual Encoder) được truyền đạt một cách đầy đủ (feed into Textual Encoder) để những mô tả tạo ra đúng với những gì model trích xuất từ hình ảnh?

→ Có thể được giải quyết bằng việc sử dụng các model có khả năng chuyển đổi đầy đủ các biểu diễn ngữ nghĩa từ Visual Encoder sang Language Model. Thế nhưng những model này thường yêu cầu rất nhiều tài nguyên về:

- Thời gian train
- Trainable parameters
- Datasets
- Annotation bổ sung (e.g. Kết quả detect)

và khó áp dụng cho thực thế, cụ thể khi:

- Train model với nhiều bộ dataset khác nhau giúp tạo ra nhiều caption cho cùng một hình ảnh trong nhiều trường hợp.
- ...

→ **Ta cần một model lightweight và linh hoạt hơn.**

1.8 Tổng quát về mô hình ClipCap

1.8.1 Abstract

- **Ý tưởng:** Sử dụng CLIP Encoder như là một prefix embedding cho caption kết hợp với một mapping network Multi Layer Perceptron (MLP) đơn giản và sau đó fine tune language model (prefix fine tune) để tạo ra caption.
- CLIP được sử dụng vì có các feature semantic phong phú và đa dạng ngữ cảnh, có khả năng nhận thức tồn cho các vấn đề Vision-Language.
- Sử dụng model GPT được đào tạo trước để có được kiến thức đa dạng về visual lẩn text.

- Nếu không có annotation bổ sung hoặc pretrained, CLIP rất hiệu quả trong việc tạo ra các chú thích có ý nghĩa cho các datasets quy mô lớn và đa dạng.
- Chỉ cần huấn luyện mapping network là đã có kết quả đủ tốt, và có thời gian train ngắn hơn rất nhiều.

Sử dụng cách tiếp cận linh hoạt không cần **Additional Supervision** (e.g. object annotation) và có thể áp dụng cho nhiều loại data khác nhau, CLipCap là một mô hình linh hoạt với khả năng thích ứng cao trên dữ liệu của nhiều domain khác nhau. Không những vậy, hướng tiếp cận tận dụng những mô hình pre-train mạnh mẽ, có khả năng tổng hợp cao còn giúp tốn ít thời gian huấn luyện so với các cách tiếp cận tương tự nhưng đạt kết quả vượt trội không thua kém các model SOTA.

(A) **Conceptual Captions**

Model	ROUGE-L ↑	CIDEr ↑	SPICE ↑	#Params (M) ↓	Training Time ↓
VLP	24.35	77.57	16.59	115	1200h (V100)
Ours; MLP + GPT2 tuning	26.71	87.26	18.5	156	80h (GTX1080)
Ours; Transformer	25.12	71.82	16.07	43	72h (GTX1080)

Hình 12: SOTAs vs ClipCap

1.8.2 Method

Cho một tập dataset với $\{x^i, c^i\}_{i=1}^N$, x^i là hình ảnh còn c^i là caption tương ứng.

Mục tiêu: tạo ra một caption có ý nghĩa, mạch lạc và chính xác về ngữ pháp với một hình ảnh đầu vào chưa từng gặp (unseen input image). Chúng ta có thể xem caption là một chuỗi tokens $c^i = c_1^i, \dots, c_\ell^i$ (caption có độ dài tối đa ℓ). Vì vậy mục tiêu của việc huấn luyện là cực đại hóa hàm sau:

$$\max_{\theta} \sum_{i=1}^N \log p_{\theta}(c_1^i, \dots, c_\ell^i | x^i) \quad (1)$$

Trong đó θ biểu thị số lượng parameter cần train của model.

Ý tưởng chính của nhóm tác giả là sự dụng feature giàu semantic của CLIP, hầu hết chứa dữ liệu, đối tượng trực quan cần thiết. Qua các công trình nghiên cứu trước, nhóm tác giả xem điều kiện cần thiết là prefix cho caption. Vì các thông tin ngữ nghĩa bắt buộc được gói gọn trong prefix, có thể sử dụng language model tự động để dự đoán các token tiếp theo mà không cần tính đến các token trong tương lai. Mục tiêu này có thể được mô tả bằng công thức sau:

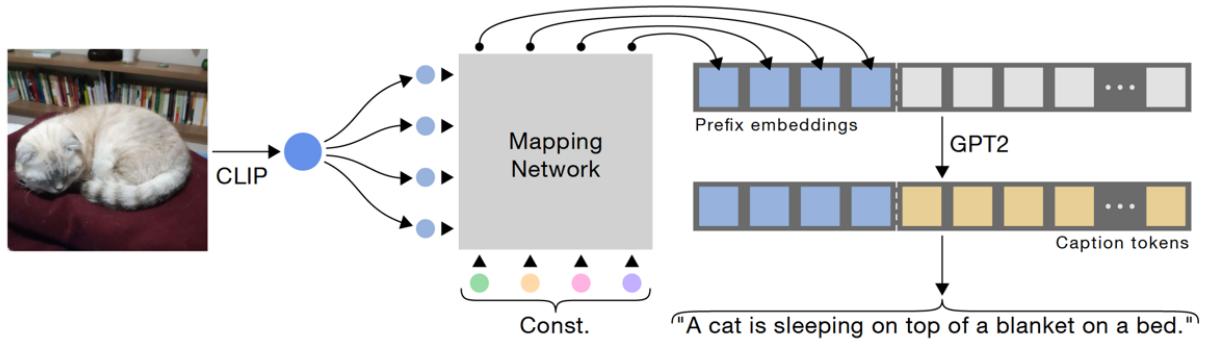
$$\max_{\theta} \sum_{i=1}^N \sum_{j=1}^{\ell} \log p_{\theta}(c_j^i | x^i, c_1^i, \dots, c_{j-1}^i) \quad (2)$$

2 Kiến trúc

ClipCap có 3 thành phần chính: **CLIP**, **Mapping Network** và **GPT-2**:

- Dùng pre-train model CLIP để trích xuất thông tin ngữ nghĩa của hình ảnh.
- Pretrain GPT-2 để tạo ra caption từ thông tin ngữ nghĩa đó.
- Sử dụng Mapping Network (key component của paper/model) để biến đổi encoding (output của CLIP) về word embedding (input của GPT-2).

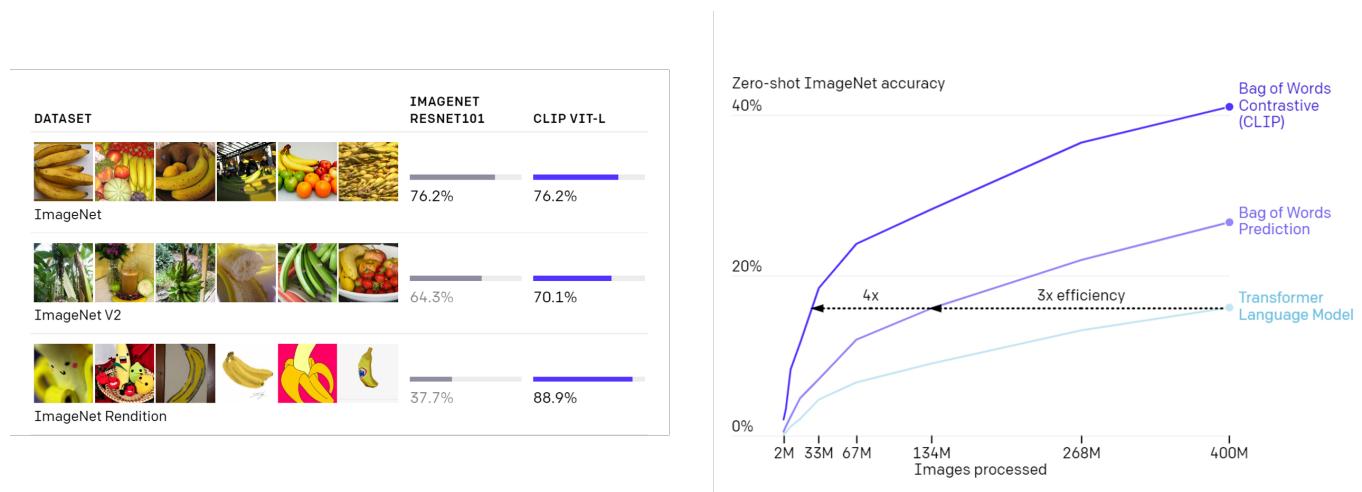
Quá trình train của model này chính là quá trình train Mapping Network, do CLIP và GPT-2 là các model pretrained có kích thước rất lớn không cần train lại mà vẫn đạt được kết quả cao.



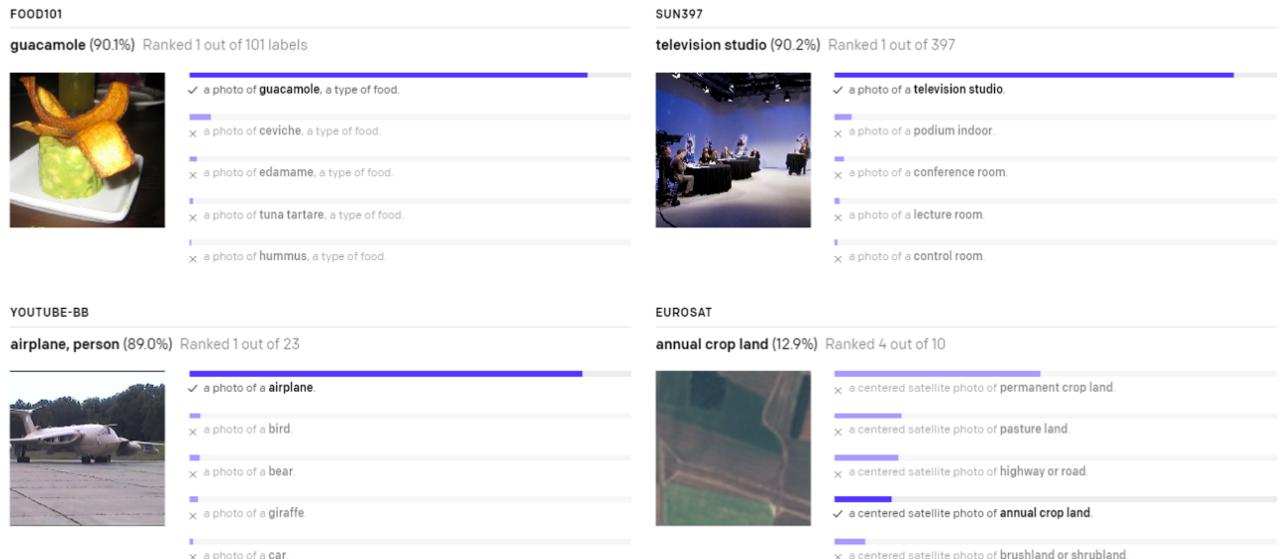
Hình 13: ClipCap Architecture Overview

2.1 CLIP

2.1.1 Giới thiệu



Hình 14



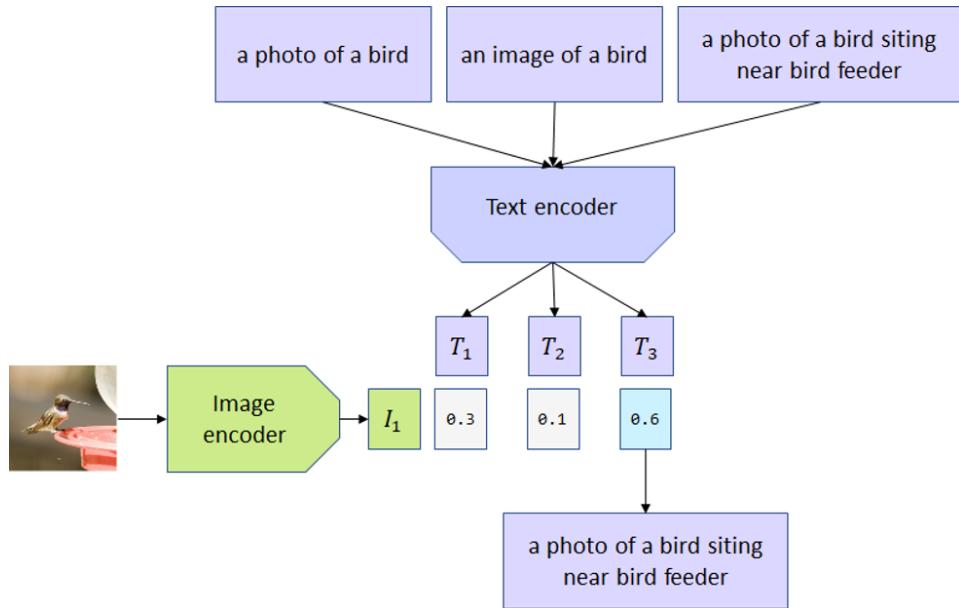
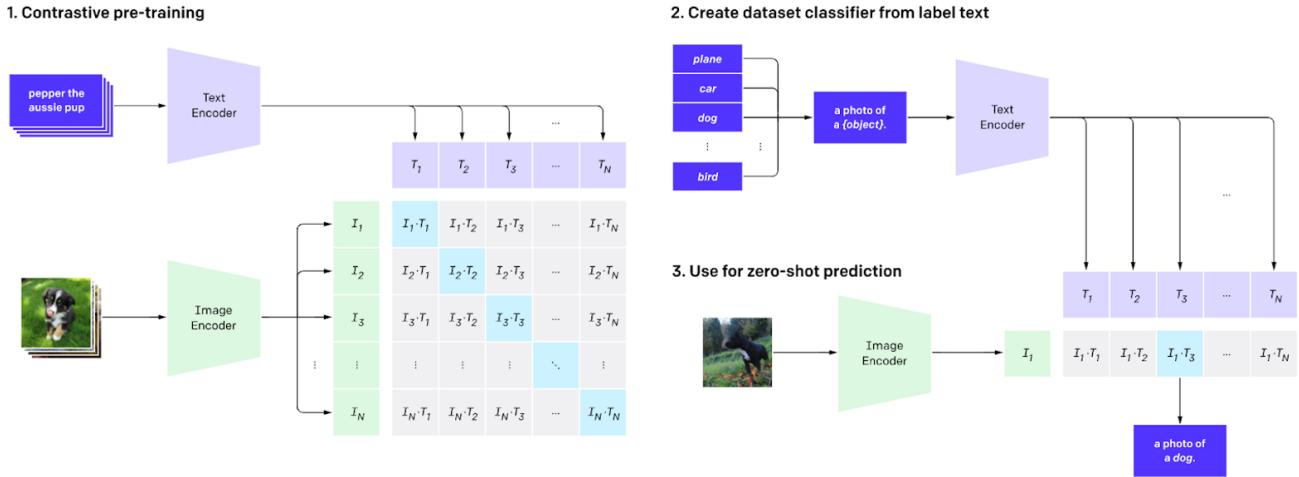
Hình 15

CLIP (Contrastive Language–Image Pre-training) là một model deep learning đa tác vụ (multimodal) vision-language mạnh mẽ của OpenAI vào năm 2021, được huấn luyện dựa trên 400,000,000 cặp (image, text).

Model CLIP có khả năng học được nhiều loại khái niệm (object, action, ...) một cách trực quan từ hình ảnh và liên kết chúng với nhau. Vì thế chúng ta không chỉ có thể sử dụng được CLIP cho nhiều bài toán image classification với sự đa dạng về class, mà còn có khả năng Zero-shot learning, có thể áp dụng trên các bài toán nhận dạng với dữ liệu mà CLIP chưa được huấn luyện và tối ưu, tương tự như các mô hình GPT-2 và GPT-3 cũng từ OpenAI.

Ý tưởng nền tảng của CLIP chính là việc encode các cặp image và text bằng cùng một dạng biểu diễn, để từ đó có thể so sánh các khái niệm này, như việc chúng ta có thể so sánh nội dung của một quyển sách, tồn tại dưới dạng text, và một bộ phim, là sự kết hợp giữa hình ảnh và âm thanh, bằng việc tóm tắt nội dung của cả hai rồi so sánh chúng với nhau, chính là việc đưa chúng về một biểu diễn chung là từ ngữ. CLIP ở đây đã tối ưu sự liên kết giữa image và text, và là cầu nối thu hẹp khoảng cách giữa Computer Vision và Natural Language Processing.

2.1.2 Cơ chế hoạt động của CLIP



CLIP model bao gồm 2 sub-models là image và text encoders để encode input thành các vector đặc trưng để xây dựng similarity matrix ($I \times T$ is an inner product).

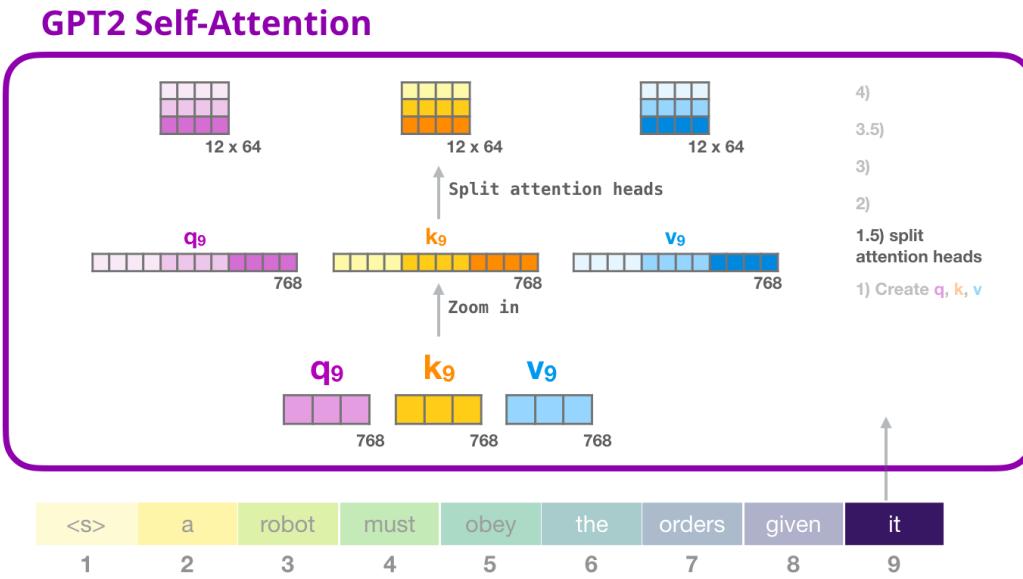
Mỗi hàng là một classification task: với input là image I , thì output là text tương ứng với image. Tương tự mỗi cột cũng là classification task: với input là text T , thì output là image tương ứng.

Trong quá trình inference chúng ta sẽ lấy một tập hợp label, tạo text dựa trên label và đưa các text này qua bộ text encoder để tạo ra text embeddings. Text embeddings sau đó được so khớp với image representation.

2.1.3 CLIP trong ClipCap

Trong mô hình ClipCap, CLIP được sử dụng như một visual encoder giúp trích xuất những thông tin ngữ nghĩa (semantic encoding) từ hình ảnh đầu vào. Nhờ vào khả năng tổng quát hoá vượt trội hơn các model phân loại thông thường bằng việc hiểu thêm ý nghĩa của các class labels bức ảnh, CLIP là một mô hình mang lại sự linh hoạt và tổng quát hơn đáng kể so với các model ImageNet hiện có, có thể thực hiện nhiều tác vụ khác nhau. CLIP còn đem lại lợi ích rất lớn về thời gian và tài nguyên, khi đã được huấn luyện trên một bộ dữ liệu cực lớn, nhờ đó Có khả năng encode ngữ nghĩa cho bất kỳ image nào mà không cần "additional supervision", giảm thiểu yêu cầu annotations bổ sung.

2.2 GPT-2



Hình 16: GPT-2

GPT-2 là một mô hình ngôn ngữ dựa trên mô hình transformer được OpenAI tạo ra vào tháng 2 năm 2019 với mục đích duy nhất là dự đoán (các) từ tiếp theo trong một câu. GPT-2 là từ viết tắt của ‘Generative Pretrained Transformer 2’. Mô hình là mã nguồn mở và được đào tạo trên 1,5 tỷ tham số để tạo chuỗi văn bản tiếp theo cho một câu nhất định. Nhờ sự đa dạng của tập dữ liệu được sử dụng trong quá trình đào tạo, chúng ta có thể tạo ra văn bản đầy đủ cho văn bản từ nhiều lĩnh vực khác nhau. GPT-2 gấp 10 lần thông số và 10 lần dữ liệu của GPT tiền nhiệm. Hiện tại, GPT-2 có 4 phiên bản được phân loại theo kích thước như sau:

- GPT-2 Small: Mô hình có 117 triệu tham số
- GPT-2 Medium: Mô hình có 345 triệu tham số
- GPT-2 Large: Mô hình có 762 triệu tham số
- GPT-2 Extra Large: Mô hình có 1.542 tỉ tham số

GPT-2 có thể học các tác vụ ngôn ngữ như đọc, tóm tắt và dịch từ văn bản thô mà không cần sử dụng dữ liệu đào tạo dành riêng cho lĩnh vực đó.

Mô hình GPT-2 được huấn luyện với tác vụ dự đoán từ tiếp theo khi đã biết tập hợp những từ trước đó. Định dạng của dữ liệu đầu vào yêu cầu một token đặc biệt đánh dấu vị trí bắt đầu của một chuỗi. Sau mỗi lần mô hình sinh ra một token mới, token này sẽ được thêm vào chuỗi đầu vào và chuỗi mới này sẽ trở thành đầu vào của mô hình trong bước lặp tiếp theo. Quá trình này sẽ được lặp lại liên tục cho đến khi token được sinh ra, đánh dấu sự kết thúc của caption.

GPT-2 được huấn luyện trên một nguồn dữ liệu khổng lồ (WebText), với 40GB dữ liệu được thu thập bằng cách crawl từ các trang web được liên kết với các bài đăng trên Reddit có ít nhất ba phiếu tán thành trước tháng 12 năm 2017. Dữ liệu này được các tác giả đánh giá là tốt hơn so với Common Crawl11, một tập dữ liệu khác được sử dụng khá thường xuyên trong quá trình huấn luyện mô hình xử lý ngôn ngữ.

Do thừa hưởng những ưu điểm của Transformer và được huấn luyện trên một tập dữ liệu khổng lồ và phong phú, GPT-2 đã cho thấy khả năng thực hiện tốt nhiều tác vụ khác nhau ngoài sinh văn bản như: Trả lời câu hỏi, Tóm tắt văn bản hay thậm chí là dịch máy,... Trong bài báo, các tác giả cũng đã chứng minh hiệu suất đáng kinh ngạc của mô hình này trên một số tập dữ liệu phổ biến như WMT-14-Fr-En cho dịch máy, CoQA cho đọc hiểu văn bản, CNN and Daily Mail cho tóm tắt văn bản,... Đây là một kết quả ngoài mong đợi khi GPT-2 chỉ được thiết kế cho bài toán dự đoán từ tiếp theo trong chuỗi.

2.3 Language model fine-tuning

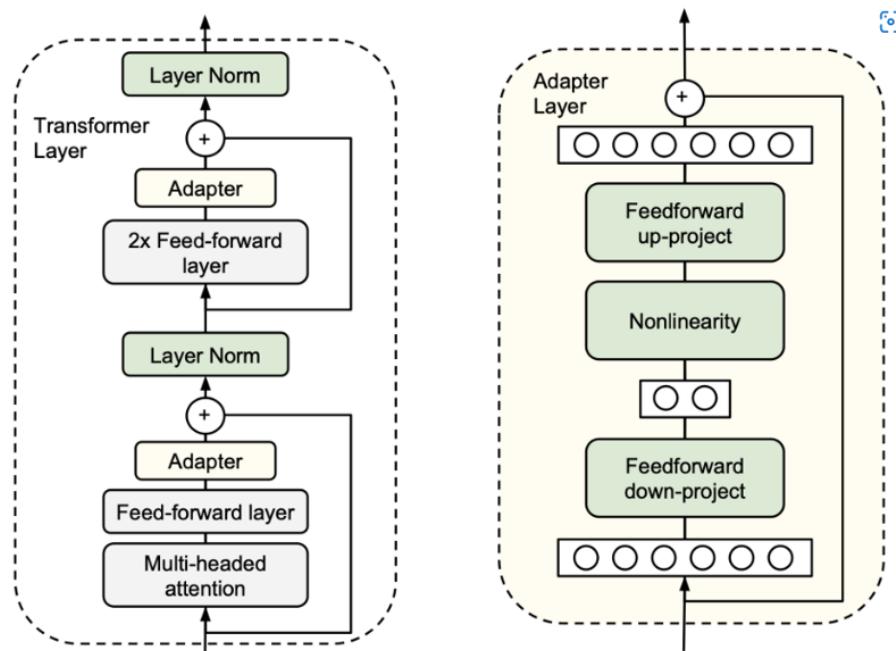
Khi train model ClipCap, Một thách thức đặt ra nằm ở khả năng chuyển đổi một cách đầy đủ thông tin ngữ nghĩa (semantic representation) giữa CLIP (output CLIP visual encoder) và GPT-2 (input GPT-2 model). Tuy GPT-2 và CLIP model đều là những model có khả năng phát triển một lượng văn bản phản hồi đa dạng và phong phú, nhưng sự chênh lệch giữa các biến diễn ngữ nghĩa giữa 2 mô hình này sẽ dẫn đến khác biệt về concept caption dự đoán do quá trình train của 2 mô hình này là độc lập.

Do đó, nhóm tác giả đã đề xuất phương pháp tiếp cận thứ nhất: fine-tuning language model (GPT-2) trong quá trình train Mapping Network. Ưu điểm của cách tiếp cận này chính là language model sẽ cho kết quả tốt hơn sau quá trình fine-tuning. Tuy nhiên, fine-tuning sẽ dẫn đến số lượng tham số (trainable parameters) của model tăng lên đáng kể, ảnh hưởng đến kích thước cũng như là hiệu năng của mô hình.

Nhược điểm trên đã dẫn đến phương pháp tiếp cận thứ hai: chỉ train Mapping Network, không fine-tuning GPT-2 mà áp dụng kĩ thuật prefix-tuning, đóng băng model, chỉ học prefix khi train, từ đó giúp cho model nhẹ hơn, giảm số tham số cũng như thời gian huấn luyện đáng kể, nhưng vẫn đạt được kết quả tương đương, và trong một số trường hợp đạt được kết quả vượt trội hơn so với các mô hình SOTA.

Bên cạnh đó, nhóm tác giả cũng đã thử nghiệm với việc fine-tune model CLIP, nhưng không cho thấy cải thiện về kết quả, mà còn làm tăng thời gian train và độ phức tạp của mô hình, cho rằng không gian biểu diễn của CLIP đã bao hàm đầy đủ các thông tin ngữ nghĩa của hình ảnh mà không cần phải adapt theo kiểu cách captioning bất kì nào.

2.3.1 Adapter-tuning



Adapter illustration from [Parameter-Efficient Transfer Learning for NLP](#)

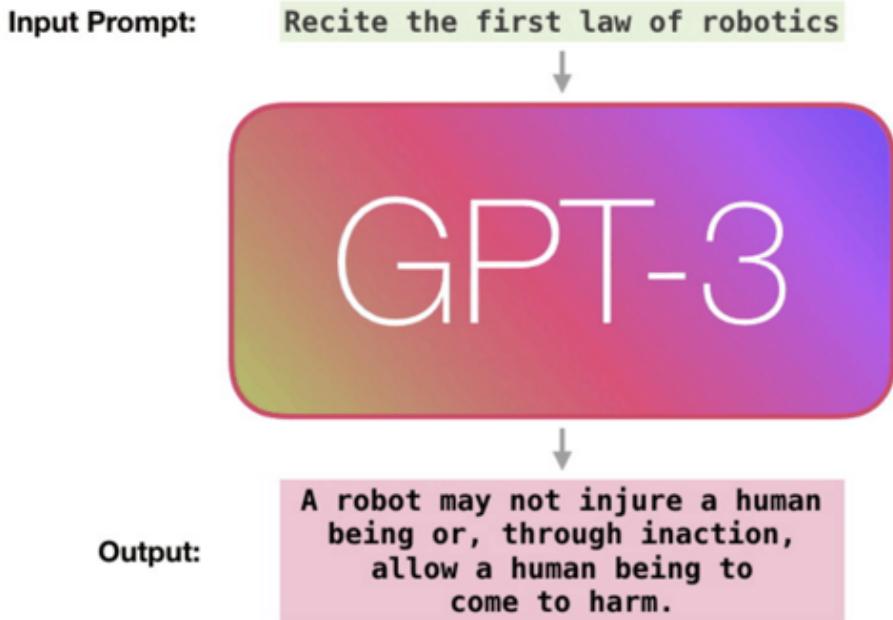
Hình 17: Adapter-tuning

Adapter-tuning: Là một kỹ thuật trong quá trình fine-tune language model bằng cách thêm các task-specific layers nhưng vẫn giữ được các tham số của frozen language model, giúp tiết kiệm không gian lưu trữ và đạt vẫn đạt được hiệu xuất tương đương fine-tuning.

Việc giữ lại các tham số ban đầu của LM giúp giải quyết vấn đề lãng phí dung lượng lưu trữ cho các tác vụ downstream khác nhau. Tuy nhiên, quá trình fine-tuning vẫn cần có đủ dữ liệu để đạt được kết quả tốt, đặt ra thách thức trong các tình huống hạn hẹp về tài nguyên phần cứng và lưu trữ.

2.3.2 Prompting

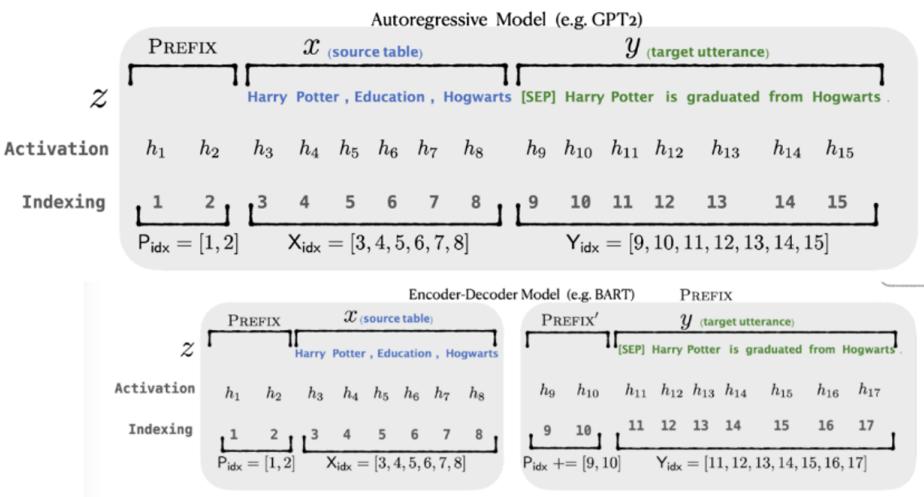
Prompting: GPT-3 là language model được train trên một lượng lớn corpus. Khác với fine-tuning, chỉ với lượng tài nguyên ít ỏi nhưng vẫn có thể tạo ra output mạch lạc về nội dung.



Hình 18: Prompting

2.3.3 Prefix-tuning

Nếu chúng ta muốn điều khiển language model để tạo ra một từ đích cụ thể, việc "prepending"/chuẩn bị trước các cụm từ có liên quan sẽ giúp tăng xác suất có điều kiện của language model cho đầu ra mong muốn.



Hình 19: Prefix-tuning

Nhóm tác giả đề xuất phương pháp tối ưu ngữ cảnh thích hợp bằng cách định nghĩa các prompt như một vector liên tục thay vì bằng các từ rời rạc hoặc word embeddings. Tiếp theo đó, mục tiêu chính là tìm cách tối ưu những prompt liên tục này xuống những tác vụ downstream.

$$h_i = \begin{cases} P_\theta[i, :], & \text{if } i \in \mathbf{P}_{\text{idx}}, \\ \text{LM}_\phi(z_i, h_{<i}), & \text{otherwise.} \end{cases}$$

Với mô hình tự hồi quy tuyến tính như GPT-2, phương pháp prefix-tuning sẽ thêm các prefix học được vào trước x , y và thu được activations $h1$ và $h2$, với chiều dài prefix = 2 như trong hình minh họa. Từ đó chúng ta sẽ thu được ma trận P_θ có thể huấn luyện

chứa các tham số với số chiều của prefix nhân với số chiều của activation vectors. Và đối với các chỉ số khác, các activations sẽ giống như mô hình fine-tuning thông thường khác với các tham số ϕ .

2.4 Mapping Network

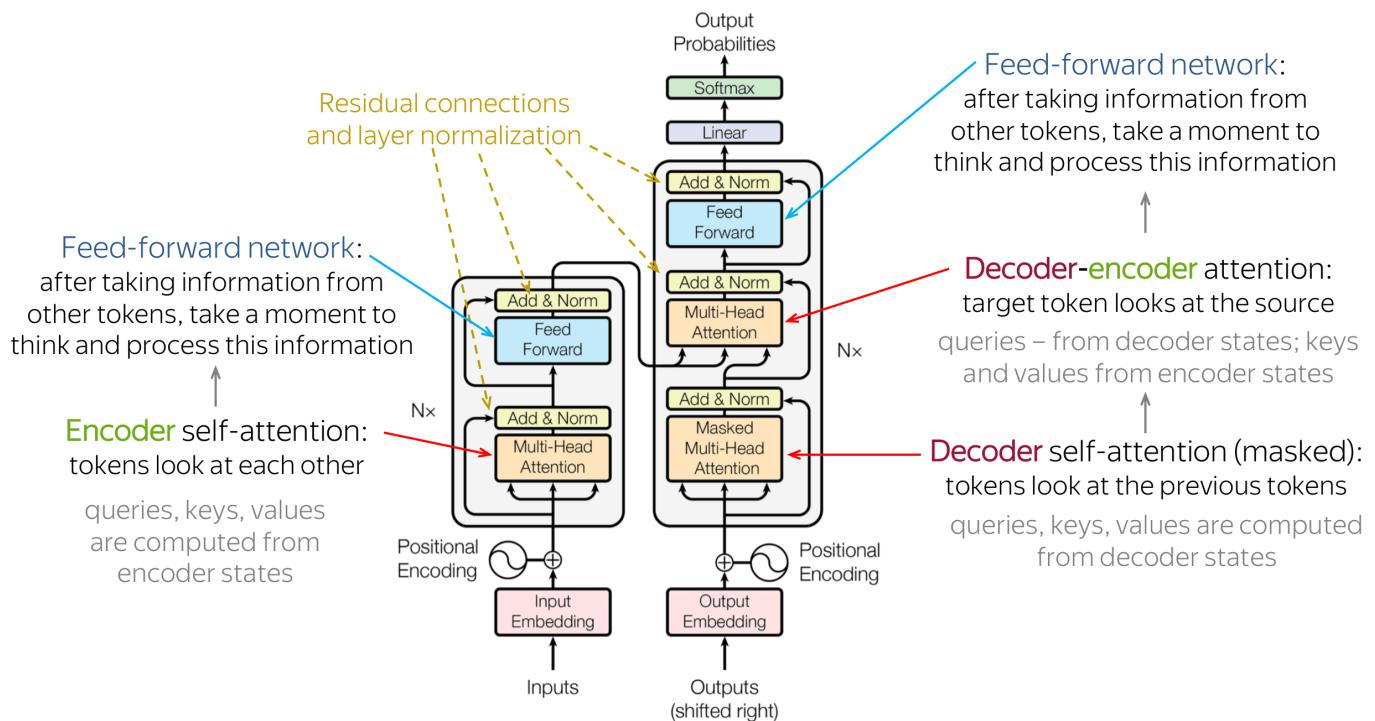
Mapping Network là thành phần chủ chốt của mô hình ClipCap để biến đổi/match CLIP encoding (output của CLIP) về word embedding (input của GPT-2).

Khi tiến hành fine-tune language model như approach ban đầu trong 2.3, quá trình mapping sẽ dễ hơn do ta có thể dễ dàng kiểm soát 2 mô hình/mạng này, từ đó nhóm tác giả đã đề xuất kiến trúc đầu tiên cho Mapping Network là một single Multi-Layers Perceptron (MLP) với kết quả tốt dù chỉ có 1 lớp ẩn do pre-train CLIP cho kết quả tối ưu với các tác vụ vision-language.

Bên cạnh đó, với hướng tiếp cận thứ hai (prefix-tuning, đóng băng model), thì nhóm tác giả đã đề xuất sử dụng kiến trúc Transformer cho Mapping Network. Mạng Transformer sử dụng cơ chế global attention trên các token input sẽ giúp giảm số tham số yêu cầu với cái chuỗi dài và làm tăng kích thước prefix, từ đó cải thiện kết quả của mô hình.

2.4.1 Transformer

Transformer lần đầu được giới thiệu trong bài báo Attention is all you need vào năm 2017. Kiến trúc của Transformer được mô tả như sau, với bên trái là encoder, thông thường có $N_x = 6$ layers chồng lên nhau. Mỗi layer sẽ có multi-head attention như đã tìm hiểu và khôi feed-forward. Ngoài ra còn các kết nối residual giống như trong mạng Resnet. Ở bên phải là decoder, tương tự cũng có $N_x = 6$ layers chồng lên nhau. Kiến trúc thì khá giống encoder những chỉ có thêm khôi masked multi-head attention ở vị trí đầu tiên.



Hình 20: Kiến trúc của mạng Transformer

- Positional encoding:** Bởi vì transformer không có các mạng hồi tiếp hay mạng tích chập nên nó sẽ không biết được thứ tự của các token đâu vào. Vì vậy, cần phải có cách nào đó để mô hình biết được thông tin này. Đó chính là nhiệm vụ của positional encoding. Như vậy, sau bước nhúng từ (embedding layers) để thu được các tokens thì ta sẽ cộng nó với các vector thể hiện vị trí của từ trong câu.
- Lớp Normalization:** Trong hình ảnh cấu trúc, có lớp "Add Norm" thì từ Norm thể hiện cho lớp Normalization. Lớp này đơn giản là sẽ chuẩn hóa lại đầu ra của multi-head attention, mang lại hiệu quả cho việc nâng cao khả năng hội tụ.
- Kết nối Residual:** Kết nối residual bản chất rất đơn giản: thêm đầu vào của một khôi vào đầu ra của nó. Với kết nối này giúp mạng có thể chồng được nhiều layers. Như trên hình, kết nối residual sẽ được sử dụng sau các khôi FFN và khôi attention. Như trên hình từ "Add" trong "Add Norm" sẽ thể hiện cho kết nối residual.
- Khôi Feed-Forward:** Đây là khôi cơ bản, sau khi thực hiện tính toán ở khôi attention ở mỗi lớp thì khôi tiếp theo là FFN. Có thể hiểu là cơ chế attention giúp thu thập thông tin từ những tokens đầu vào thì FFN là khôi xử lý những thông tin đó.

Trong mô hình ClipCap, Mapping Network với kiến trúc Transformer sẽ gồm 2 input: CLIP visual encoding và learned constant input. Learned constant input sẽ chứa các thông tin học được từ CLIP encoding thông qua cơ chế multi-head attention, từ đó giúp

hiệu chỉnh language model với dữ liệu mới. Quá trình thực nghiệm của nhóm tác giả đã chứng minh tính hiệu quả của kiến trúc Transformer với Approach sử dụng prefix-tuning, với learned constant học được nhiều các embed quan trọng và chi tiết về CLIP encoding, từ đó tăng khả năng diễn giải generalized prefix.

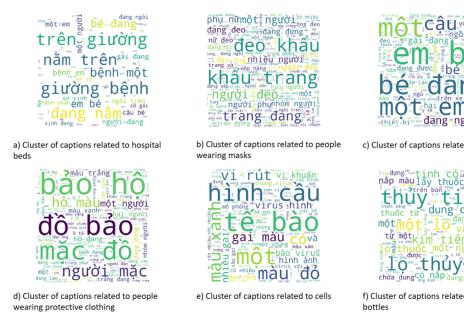
3 Hạn chế

- ClipCap thừa hưởng không chỉ những ưu điểm của CLIP, mà còn mang trong mình hạn chế cơ bản của model này, và cụ thể là ở khả năng Zero-Shot.
- CLIP thường sẽ cho ra kết quả tốt với các task nhận dạng trên khá nhiều dataset về các đối tượng phổ biến, nhưng vẫn gặp trớ ngai trước những task với đối tượng trừu tượng, phức tạp, hay chuyên môn hoá.
E.g: Task classification trên dataset y khoa, đếm số lượng đối tượng trong ảnh, phân biệt cụ thể các dòng xe khác nhau, ước lượng khoảng cách tương đối giữa các đối tượng.

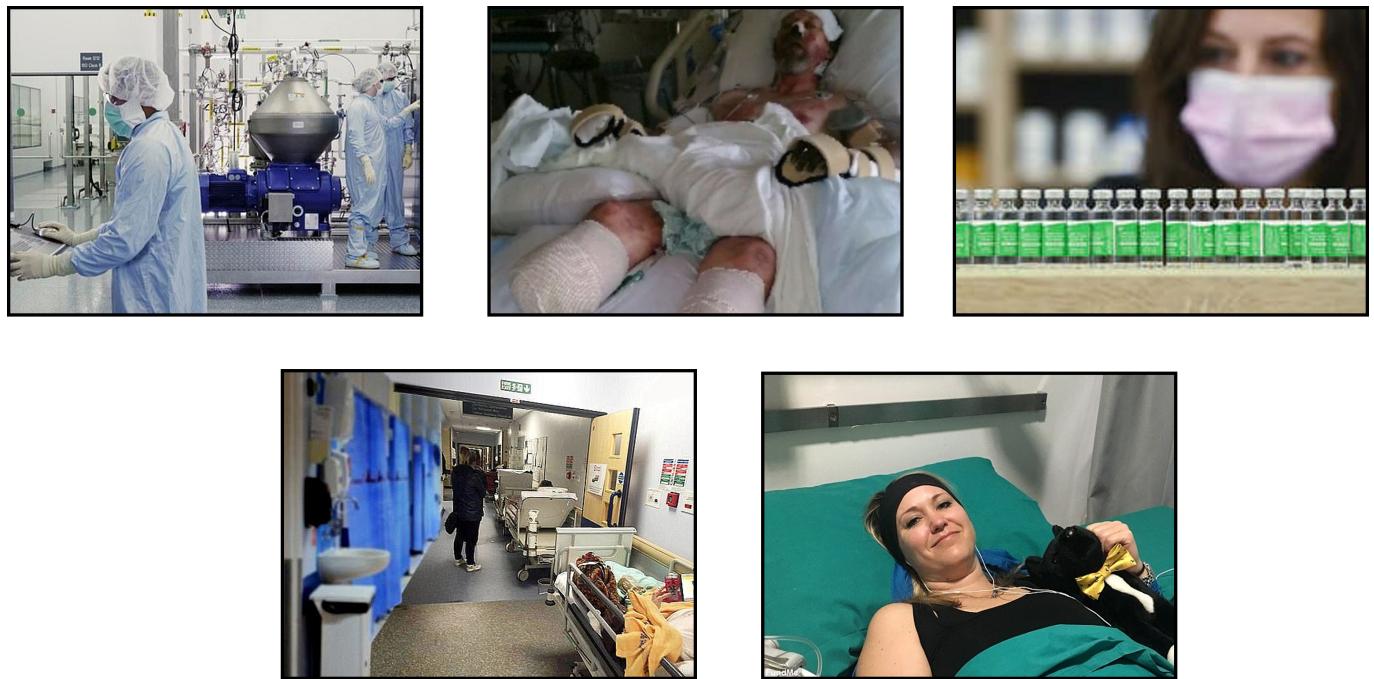
4 Kết quả thực nghiệm

4.1 Dataset

Ở bài toán trên nhóm sẽ thực nghiệm phương pháp ClipCap cho 2 bộ dữ liệu dành cho Tiếng Việt là VietCap(2021) và UIT VIC(2017). Đây là 2 bộ dữ liệu có chất lượng tốt có GT sẵn để đánh giá hiệu suất mô hình. Một số thống kê về 2 bộ dữ liệu:

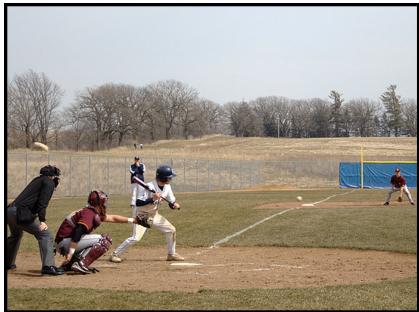
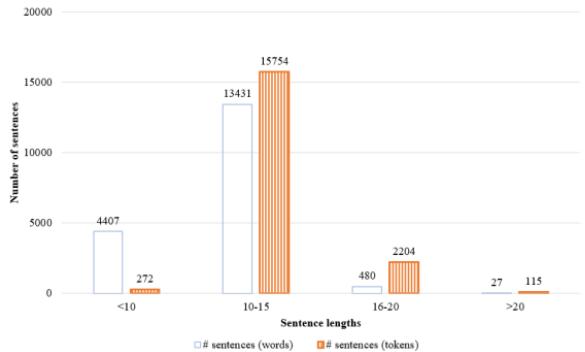


	Images	Captions	Avg. captions per image	Avg. caption's length (tokens)
Train	8032	9429	1.17	11.88
Public test	1002	1039	1.04	11.86
Private test	1034	1095	1.05	11.97
All	10068	11563	1.15	11.89



Hình 21: VLSP2021_VIETCAP

Verbs	Nouns	Adjectives
cầm (hold): 3,344	bóng (ball): 7,686	tennis: 3,005
chơi (play): 2,760	sân (pitch): 6,725	bóng chày (baseball): 880
dánh bóng (hit): 2,581	cầu thủ (player): 2,635	cao (tall): 687



Hình 22: UIT_VIC

4.2 Demo

VLSP2021_Vietcap:

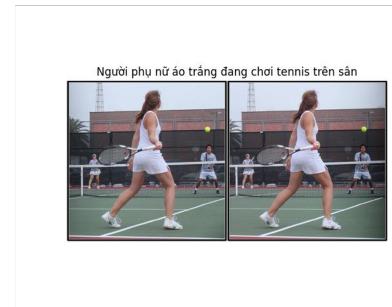
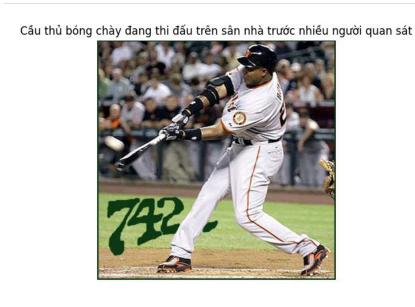


Hình 23: VLSP2021_Vietcap correct predictions



Hình 24: VLSP2021_Vietcap incorrect predictions

UIT_VIC:



Hình 25: UIT_VIC correct predictions

Một con gấu đang chơi với một quả bóng bàn ở trong phòng khách



Khán giả đang theo dõi một trận thi đấu bóng rổ



Người phụ nữ áo đỏ đang chuẩn bị đánh quả bóng tennis



Hình 26: UIT_VIC incorrect predictions

4.3 Training

Nhóm chúng em sử dụng pretrained model của CLIP, còn ở mô hình ngôn ngữ sử dụng model pretrain của GPT2-News cho Tiếng Việt, là mô hình được huấn luyện trên các dữ liệu wiki, các bài báo,... Mô hình mapping network là mô hình MLP cho dễ thực hiện và có độ chính xác cao hơn. Thông số quá trình train:

- Epochs: 20
- Batch-size: 32

Ngoài ra để giảm thời gian huấn luyện nhóm còn train trên TPU High Ram Colab (28GB) với khoảng 2h train.

4.4 Kết quả

Ưu điểm:

- Mô hình có khả năng xác định được các đối tượng, hành động trong ảnh.
- Tạo ra mô tả tự nhiên, đầy đủ, đúng ngữ pháp và chính tả.
- Thời gian huấn luyện nhanh (2h, TPU Colab Pro).

Model gặp các trở ngại như:

- Nhạy cảm về màu sắc và chi tiết của đối tượng.
- Nhạy cảm với số lượng đối tượng trong ảnh.

Model	public_test	private_test
VLP	30.6	32.7
ClipCap	29.8	27.8
AOA net	28.75	27.5
CNN+LSTM	27.5	26.7

Bảng 1: VLSP2021_VIETCAP

Model	Bleu1	Bleu2	Bleu3	Bleu4
Pytorch-tutorial	0.71	0.575	0.476	0.394
Show and Tell	0.682	0.561	0.411	0.327
ClipCap	0.802	0.671	0.542	0.445

Bảng 2: UIT_VIC

TÀI LIỆU THAM KHẢO

- [1] ClipCap: CLIP Prefix for Image Captioning.
<https://arxiv.org/pdf/2111.09734v1.pdf>
- [2] CLIP prefix captioning. Github.
https://github.com/rmokady/CLIP_prefix_caption
- [3] GPTTeam solution - VLSP 2021 viecap4h Challenge. Github.
https://github.com/Luvata/vlsp_viecap4h_gptteam_code
- [4] Better Language Models and Their Implications.
<https://openai.com/blog/better-language-models/>
- [5] CLIP: Connecting Text and Images.
<https://openai.com/blog/clip/>
- [6] UIT-ViIC: A Dataset for the First Evaluation on Vietnamese Image Captioning.
<https://arxiv.org/pdf/2002.00175.pdf>
- [7] From Show to Tell: A Survey on Deep Learning-based Image Captioning.
<https://arxiv.org/pdf/2107.06912.pdf>
- [8] VLSP 2021 - VieCap4H Challenge: Automatic Image Caption Generation for Healthcare Domain in Vietnamese.
https://people.cs.umu.se/sonvx/files/VieCap4H_VLSP21.pdf
- [9] Alternative for fine-tuning? Prefix-tuning may be your answer!
<https://medium.com/@vincentchen0110/alternative-for-fine-tuning-prefix-tuning-may-be-your-answer-a1ce05e95464>
- [10] Prefix-Tuning: Optimizing Continuous Prompts for Generation
<https://arxiv.org/pdf/2101.00190.pdf>
- [11] Tìm hiểu về kiến trúc Transformer
<https://viblo.asia/p/tim-hieu-ve-kien-truc-transformer-Az45byM6lxY>
- [12] A Beginner's Guide to the CLIP Model
<https://www.kdnuggets.com/2021/03/beginners-guide-clip-model.html>
- [13] CLIP from OpenAI: what is it and how you can try it out yourself
<https://habr.com/en/post/537334/>
- [14] Why did OpenAI introduce CLIP?
<https://kakaobrain.com/contents/?contentId=ba1484fd-3b39-4747-9f0c-07199dd78b7c>
- [15] New SOTA Image Captioning: ClipCap
<https://www.louisbouchard.ai/clipcap/>
- [16] Tìm hiểu về cơ chế Attention
<https://viblo.asia/p/tim-hieu-ve-co-che-attention-924lJjbmlPM>