

ĐẠI HỌC QUỐC GIA VIỆT NAM THÀNH PHỐ HỒ CHÍ MINH
ĐẠI HỌC BÁCH KHOA THÀNH PHỐ HỒ CHÍ MINH
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



XỬ LÝ NGÔN NGỮ TỰ NHIÊN

Chú thích nội dung, hoạt động từ hình ảnh

Giảng viên hướng dẫn: Giảng viên
Sinh viên thực hiện: Đoàn Trần Cao Trí - 2010733
Bùi Hoàng Minh - 2010410
Bùi Quốc Khánh - 2010325

Thành phố Hồ Chí Minh, 11/2023

Danh sách hình ảnh

2.1	Pokemon dataset	5
2.2	Flickr8k dataset	6
3.1	Kiến trúc Transformer.	8
3.2	Sơ đồ để tính các biểu diễn (z_1, z_2, \dots, z_n) từ (x_1, x_2, \dots, x_n) . Trong đó Q, K, V là các ma trận được tính tuyến tính từ ma trận của chuỗi (x_1, x_2, \dots, x_n)	9
3.3	Công thức tính self attention	9
3.4	Ví dụ về ma trận attention là giá trị của QK^T	10
3.5	Sơ đồ quá trình thực hiện multi head attention	10
3.6	Công thức tính multi head attention	10
3.7	Ví dụ về các ma trận attention trong quá trình tính multi-head attention	11
3.8	Kiến trúc BERT.	12
3.9	Kiến trúc GIT.	14
4.1	Train progress với Pokemon dataset	16
4.2	Train progress với Flickr8k dataset	17
5.1	Kết quả dự đoán với Pokemon dataset	18
5.2	Kết quả dự đoán với Flickr8k dataset	19

Danh sách chỉ mục

1	Giới thiệu	3
2	Tổng quan về tập dữ liệu	4
2.1	Pokemon [5]	4
2.2	Flickr8k [1]	4
3	Mô hình	7
3.1	Kiến trúc tổng quát	7
3.1.1	Tiền xử lý	7
3.1.2	Mô hình ngôn ngữ dựa trên Transformer	7
3.2	Transformer: Attention is all you need [2]	8
3.2.1	Giới thiệu	8
3.2.2	Kiến trúc	8
3.3	BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [3] .	11
3.3.1	Giới thiệu	11
3.3.2	Kiến trúc	12
3.3.3	Ứng dụng	13
3.3.4	Lợi ích và thử thách	13
3.4	GIT: A Generative Image-to-text Transformer for Vision and Language [4]	13
3.4.1	Giới thiệu	13
3.4.2	Kiến trúc	14
3.4.3	Ứng dụng	14
3.4.4	Lợi ích và thử thách	14
4	Huấn luyện mô hình	16
4.1	Pokemon	16
4.2	Flickr8k	17
5	Kết quả	18
5.1	Kết quả dự đoán của mô hình	18
5.2	Phương pháp đánh giá: WER	19
5.2.1	Giới thiệu	19
5.2.2	Công thức	20
5.2.3	Hạn chế	20
5.3	Kết luận	20
6	Tổng kết	21

Chương 1

Giới thiệu

Image captioning là quá trình tạo ra mô tả văn bản tự động cho hình ảnh nhờ các mô hình máy học được huấn luyện để nhận diện và mô tả nội dung của hình ảnh một cách tự động.

Mục tiêu của image captioning là kết hợp khả năng nhận diện hình ảnh và hiểu ngữ cảnh để tạo ra các mô tả tự nhiên và chính xác cho hình ảnh.

Các ứng dụng:

- kiểm soát nội dung trên các nền tảng mạng xã hội
- nhận diện vấn đề từ hình ảnh trong một số lĩnh vực cụ thể như ảnh y tế, giao thông, ...
- tăng cường trải nghiệm người dùng trong các ứng dụng như truyền thông xã hội, thương mại điện tử và nhiều lĩnh vực khác.

Chương 2

Tổng quan về tập dữ liệu

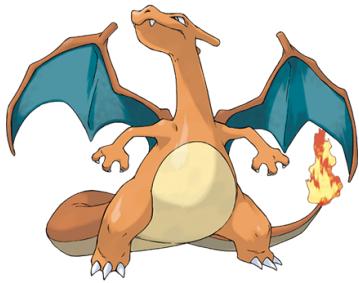
Trong phạm vi đề tài, nhóm sẽ nghiên cứu mô hình với 2 tập dataset vừa và nhỏ. Với dataset nhỏ, nhóm chọn tập dataset Pokemon và với dataset vừa, nhóm chọn tập dataset Flickr8k

2.1 Pokemon [5]

- Description: A dataset of Pokémon characters.
- Size: 833 images
- Train-test ratio: 70% - 30%
- Sample:

2.2 Flickr8k [1]

- Description: A dataset of everyday scenes with captions.
- Size: 8,092 images
- Train-test ratio: 6,000 - 2,092
- Sample:



(a) A large, orange dragon Pokémon with a long tail and wings.



(b) A small, blue turtle Pokémon with a long tail.



(c) A small, green dinosaur Pokémon with a plant on its back.



(d) A small, yellow cat Pokémon with a coin-shaped forehead.

Hình 2.1: Pokemon dataset



(a) An older woman and a young girl standing in front of a large bush covered in pink flowers.



(b) A woman wearing a straw hat and floral skirt is sitting on cement stairs with her head in her hands.



(c) A big grey dog wearing a chain collar has a smaller brown dog pinned down.



(d) A girl does a handstand on a trampoline.

Hình 2.2: Flickr8k dataset

Chương 3

Mô hình

3.1 Kiến trúc tổng quát

3.1.1 Tiền xử lý

- Hình ảnh: sử dụng CLIPImageProcessor - 1 lớp của thư viện Hugging Face Transformers bao gồm:
 - Cắt: Hình ảnh được cắt thành hình vuông có kích thước được chỉ định ($224x224$).
 - Chuyển đổi sang RGB: Hình ảnh được chuyển đổi sang không gian màu RGB.
 - Chuẩn hóa: Hình ảnh được chuẩn hóa để có giá trị trung bình là ($0.48145466, 0.4578275, 0.40821073$) và độ lệch chuẩn là ($0.26862954, 0.26130258, 0.27577711$).
- Văn bản: BertTokenizerFast - một lớp từ thư viện Hugging Face Transformers được sử dụng để phân chia văn bản thành các token trước khi đưa vào mô hình BERT, bao gồm các thao tác:
 - Chia văn bản thành các từ: Văn bản được chia thành các từ bằng cách sử dụng bộ chia tách khoảng trắng.
 - Chuyển đổi các từ thành token: Các từ được chuyển đổi thành token bằng cách sử dụng từ điển BERT.
 - Thêm các token đặc biệt: Các token đặc biệt, chẳng hạn như token CLS và token SEP, được thêm vào đầu và cuối của chuỗi token.
 - Dệm hoặc cắt chuỗi token: Chuỗi token được dệm hoặc cắt thành độ dài được chỉ định.

3.1.2 Mô hình ngôn ngữ dựa trên Transformer

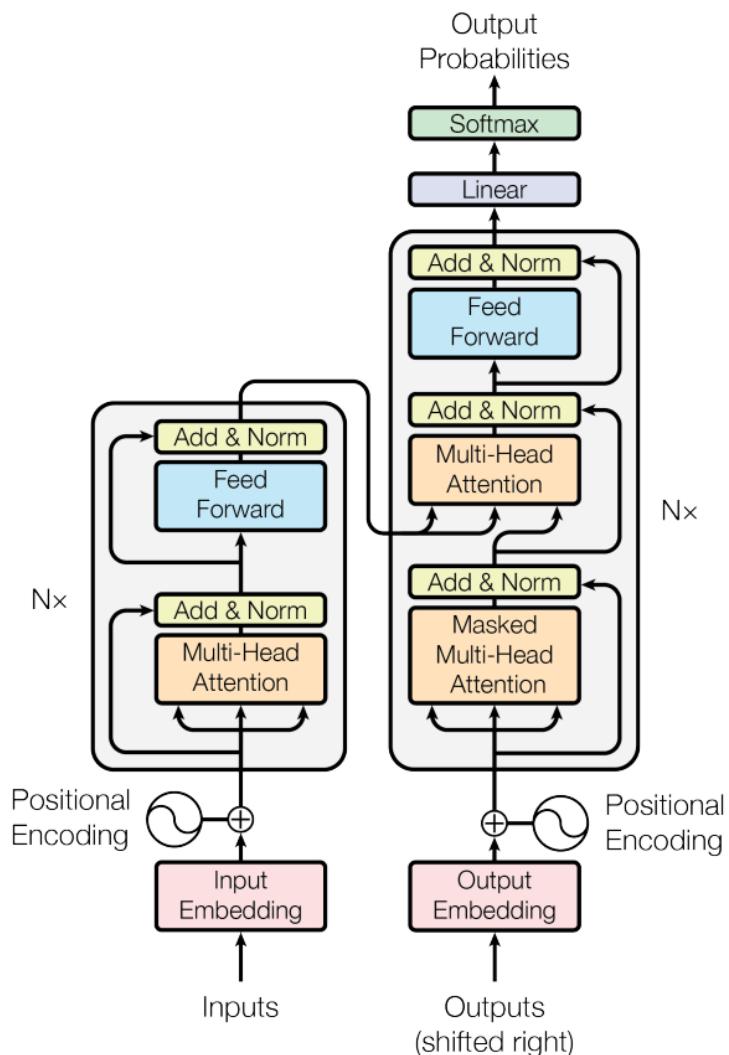
- Khối GitModel
 - 1. Tầng GitEmbeddings: nhúng các từ và vị trí thành các vector có kích thước 768.
 - 2. Tầng GitVisionModel: mã hóa hình ảnh thành các vector có kích thước 768 bằng cơ chế self-attention.
 - 3. Tầng GitEncoder: mã hóa đầu ra của GitEmbeddings và GitVisionModel thành một không gian đa chiều duy nhất. GitEncoder bao gồm một 6 lớp GitLayer. Mỗi lớp GitLayer bao gồm một lớp self-attention và một lớp trung gian. Lớp self-attention cho phép mô hình học các phụ thuộc tầm xa trong chuỗi đầu vào, lớp trung gian cho phép mô hình học các biến đổi phi tuyến của chuỗi đầu vào.
 - 4. Tầng GitProjection: chiếu đầu ra GitEncoder thành một không gian đa chiều duy nhất, bao gồm một lớp tuyến tính và một lớp chuẩn hóa. Lớp tuyến tính chiếu các đầu ra của GitEncoder vào một chiều mong muốn, trong khi lớp chuẩn hóa chuẩn hóa phân phối của kết quả sau khi chiếu.
- Tầng xuất: tạo ra các token văn bản đầu ra bằng cách chiếu kết quả tầng GitProjection vào không gian từ vựng.

3.2 Transformer: Attention is all you need [2]

3.2.1 Giới thiệu

Transformer là một kiến trúc mạng thần kinh được giới thiệu trong bài báo "Attention is All You Need" của Vaswani et al. (2017). Transformer đã đạt được kết quả tiên tiến trên nhiều tác vụ xử lý ngôn ngữ tự nhiên (NLP), bao gồm dịch máy, tóm tắt văn bản và câu trả lời câu hỏi.

3.2.2 Kiến trúc



Hình 3.1: Kiến trúc Transformer.

- Encoder - Decoder

- Encoder:

- * là thành phần đầu tiên của mô hình Transformer
 - * có chức năng mã hóa chuỗi đầu vào thành một biểu diễn mà kiểu biểu diễn này chứa nhiều thông tin hữu ích được trích xuất từ chuỗi đầu vào
 - * bao gồm một gồm N lớp tương tự nhau, mỗi lớp bao gồm hai thành phần con: multi-head self-attention (MHSA) và feed forward network (FFN). MHSA và FFN được trình bày ở phần tiếp theo.

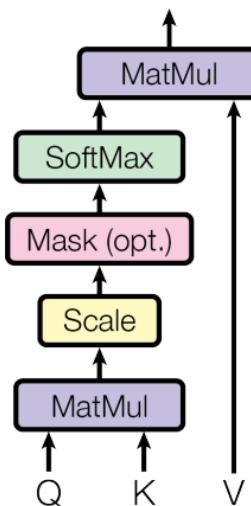
- Decoder:

- * là thành phần thứ hai của Transformer
- * có chức năng giải mã biểu diễn nội bộ được tạo ra bởi encoder thành chuỗi đầu ra
- * Decoder cũng bao gồm n lớp tương tự nhau. Tuy nhiên ở mỗi lớp được thêm vào layer thứ 3 là multi-head attention dùng để thực hiện attention trên đầu ra của n lớp encoder phía trên
- * Đồng thời layer MHSA của decoder có sự khác biệt so với encoder là vì mỗi lần output ra 1 token của chuỗi đầu ra, decoder chỉ được nhìn thấy những đầu ra đã được sinh ra phía trước mà không được nhìn thấy các token chưa được sinh ra. Vì vậy layer MHSA của decoder có thêm vào mặt nạ (Masked multi-head self-attention) để đảm bảo khi dự đoán đầu ra ở vị trí i , decoder chỉ phụ thuộc vào các đầu ra đã xuất hiện trước đó (ở vị trí nhỏ hơn i).

- Attention

- Self attention: input là 1 chuỗi (x_1, x_2, \dots, x_n) và output là 1 chuỗi (z_1, z_2, \dots, z_n) với z_i là biểu diễn tương ứng của x_i khi đã xem xét đã sự liên kết giữa x_i với $x_j, j = 1..n$.

Scaled Dot-Product Attention

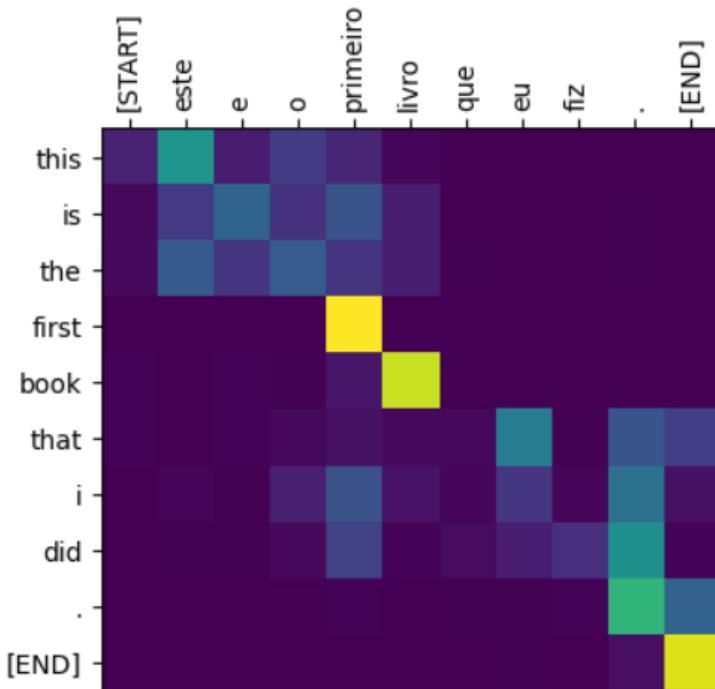


Hình 3.2: Sơ đồ để tính các biểu diễn (z_1, z_2, \dots, z_n) từ (x_1, x_2, \dots, x_n) . Trong đó Q, K, V là các ma trận được tính tuyến tính từ ma trận của chuỗi (x_1, x_2, \dots, x_n)

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

Hình 3.3: Công thức tính self attention

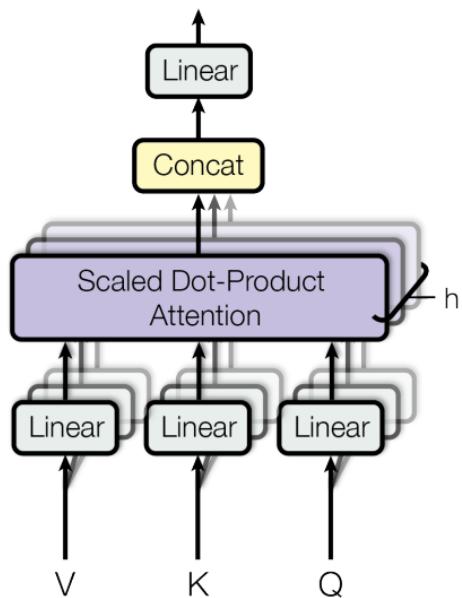
- * QK^T là phép nhân 2 ma trận có kết quả là ma trận attention thể hiện sự chú ý của mỗi token lên các token khác trong chuỗi.
- * Giá trị QK^T được scale bởi căn bậc 2 số chiều của mỗi token $\sqrt{d_k}$. Sau khi có được ma trận attention, thực hiện nhân lại vào V chính là giá trị các token chuỗi đầu vào.



Hình 3.4: Ví dụ về ma trận attention là giá trị của QK^T

- Multi head self attention: ta thực hiện nhiều lần self-attention phía trên để thu được nhiều phiên bản (z_1, z_2, \dots, z_n) từ chuỗi input (x_1, x_2, \dots, x_n) .

Multi-Head Attention

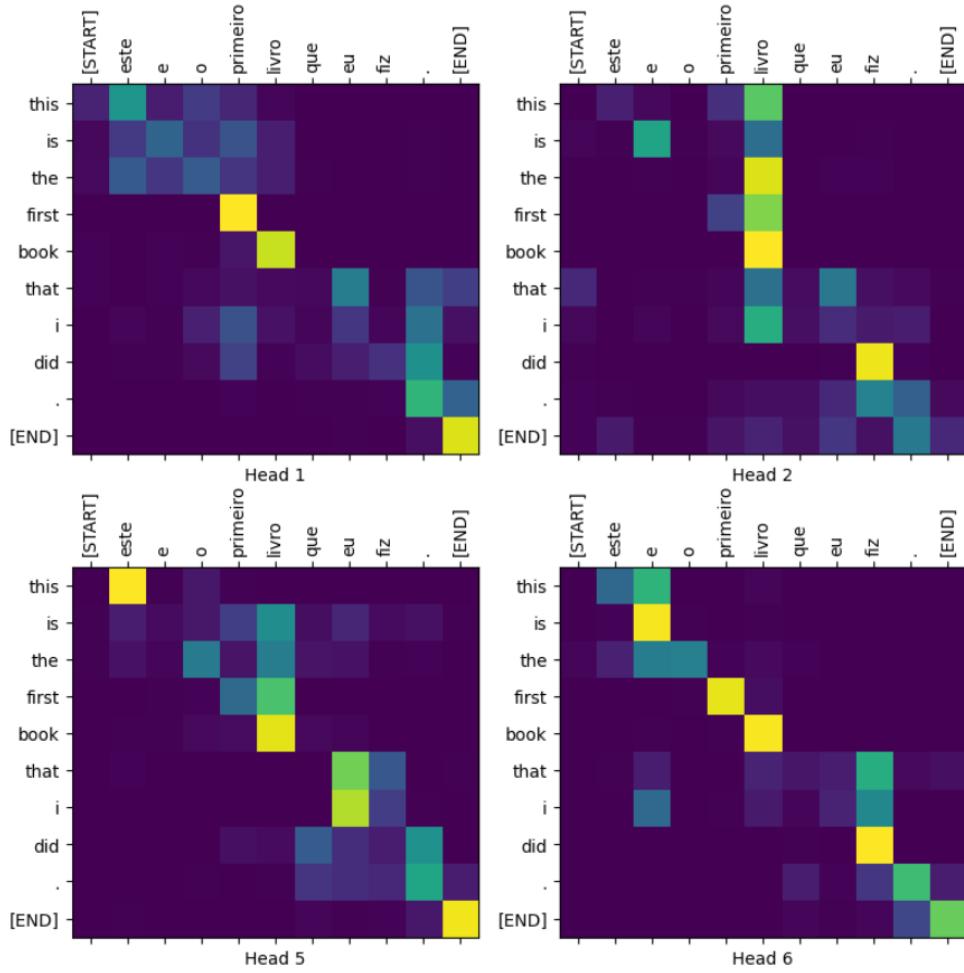


Hình 3.5: Sơ đồ quá trình thực hiện multi head attention

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

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

Hình 3.6: Công thức tính multi head attention



Hình 3.7: Ví dụ về các ma trận attention trong quá trình tính multi-head attention

- Feed Forward Network (FFN): gồm 2 layer biến đổi tuyến tính với hàm kích hoạt ReLU ở giữa:

$$FFN(x) = \max(0, x * W_1 + b_1) * W_2 + b_2 \quad (3.1)$$

- Positional Encoding: Multi-head self-attention phía trên bao gồm xem xét sự liên kết giữa các x_i mà chưa tính đến vị trí của x_i như recurrent network và convolution network. Vì vậy x_i sẽ được cộng thêm một lượng giá trị (giá trị này phụ thuộc vào vị trí của nó trong dãy) thông qua positional encoding - giá trị này là vector có chiều tương tự x_i .

Trong paper Attention is all you need, d_{model} là số chiều của x_i . Positional encoding được thực hiện như sau:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right) \quad (3.2)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right) \quad (3.3)$$

với pos là vị trí của x_i trong chuỗi input, i là chiều của x_i , $i = [1; d_{model}]$.

3.3 BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [3]

3.3.1 Giới thiệu

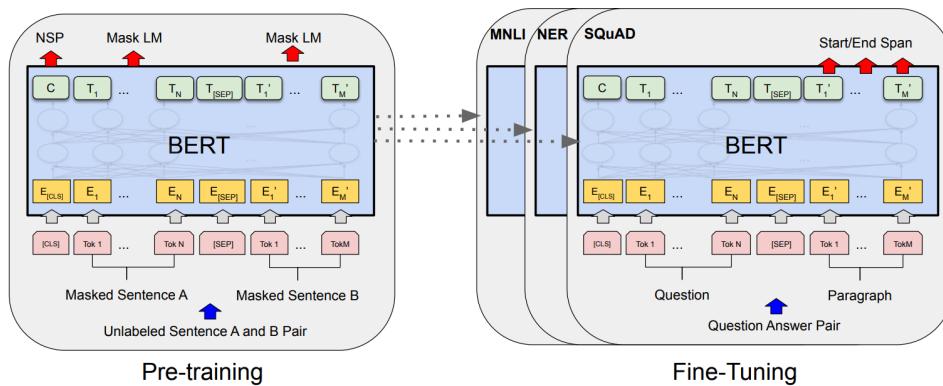
BERT, viết tắt của Bidirectional Encoder Representations from Transformers, là một mô hình ngôn ngữ (Language Model) được phát triển bởi Google AI. Nó được giới thiệu lần đầu tiên vào năm 2018 và

kể từ đó đã trở thành một trong những mô hình ngôn ngữ phổ biến và được sử dụng rộng rãi nhất trên thế giới.

BERT là một mô hình hai chiều, nghĩa là nó có thể học ngữ cảnh của một từ từ cả các từ đứng trước và sau nó. Điều này trái ngược với các mô hình ngôn ngữ truyền thống, thường là một chiều và chỉ có thể học ngữ cảnh của một từ từ các từ đứng trước nó.

Kiến trúc hai chiều của BERT được hỗ trợ bởi một kỹ thuật gọi là self-attention. Self-attention cho phép BERT học các phụ thuộc dài hạn trong văn bản, điều này là cần thiết để hiểu ý nghĩa của các câu phức tạp.

3.3.2 Kiến trúc



Hình 3.8: Kiến trúc BERT.

- Pre-training: mô hình được huấn luyện trên tập dữ liệu không có nhãn trên các tác vụ khác nhau:
 - Masked Language Model (Mask LM): dự đoán các từ bị che lấp trong một câu, dựa trên ngữ cảnh của các từ xung quanh. Ví dụ, với câu "Con mèo ngồi trên tấm thảm _", mô hình sẽ dự đoán từ "trải sàn".
 - Next Sentence Prediction (NSP): dự đoán liệu câu thứ hai trong một cặp câu có phải là phần tiếp theo logic của câu thứ nhất hay không. Ví dụ, với cặp câu "Con mèo ngồi trên tấm thảm. Đó là một ngày ấm áp.", mô hình sẽ dự đoán rằng câu thứ hai là phần tiếp theo logic của câu thứ nhất.

Để có thể đưa ra dự đoán chính xác trong 2 tác vụ trên, trong quá trình huấn luyện, BERT dần cải thiện khả năng biểu diễn chuỗi đầu vào thành một dạng biểu diễn chứa nhiều thông tin hữu ích ở mức cục bộ (ở mỗi từ) và mức toàn cục (ngữ cảnh của cả câu hoặc cả đoạn văn).

- Fine-tuning: được khởi tạo với các tham số học được ở Pre-training BERT và dùng để bắt đầu học tiếp với tác vụ:
 - Multi-Genre Natural Language Inference (MNLI): xác định liệu một câu giả thuyết có được kéo theo, trung lập hay mâu thuẫn với một câu tiền đề hay không. Ví dụ, với câu tiền đề "Con mèo ngồi trên tấm thảm." và câu giả thuyết "Con mèo ở trên tấm thảm.", mô hình sẽ dự đoán rằng câu giả thuyết được kéo theo bởi câu tiền đề.
 - Name Entity Recognition (NER): xác định và phân loại các thực thể có tên trong văn bản, chẳng hạn như người, địa điểm và tổ chức. Ví dụ, với câu "Barack Obama là tổng thống thứ 44 của Hoa Kỳ.", mô hình sẽ xác định và phân loại các thực thể có tên "Barack Obama", "44" và "Hoa Kỳ" thành người, số và quốc gia, tương ứng.
 - Stanford Question Answering Dataset (SQuAD): trả lời các câu hỏi về một đoạn văn bản nhất định. Ví dụ, với đoạn văn "Barack Obama là tổng thống thứ 44 của Hoa Kỳ. Ông sinh ra ở Honolulu, Hawaii vào ngày 4 tháng 8 năm 1961." và câu hỏi "Barack Obama sinh ra khi nào?", mô hình sẽ trả lời "4 tháng 8 năm 1961". (SQuAD)



3.3.3 Ứng dụng

BERT có thể được sử dụng cho nhiều loại tác vụ xử lý ngôn ngữ tự nhiên (NLP), bao gồm:

- **Phân loại văn bản:** BERT có thể được sử dụng để phân loại văn bản thành các danh mục khác nhau, chẳng hạn như spam hoặc không phải spam, hoặc tích cực hoặc tiêu cực.
- **Trả lời câu hỏi:** BERT có thể được sử dụng để trả lời các câu hỏi về văn bản, chẳng hạn như "Thủ đô của Pháp là gì?" hoặc "Ý chính của đoạn này là gì?"
- **Kết luận ngôn ngữ tự nhiên:** BERT có thể được sử dụng để xác định liệu một giả thuyết có đúng hay sai dựa trên một tập hợp các tiền đề.
- **Dịch máy:** BERT có thể được sử dụng để dịch văn bản từ một ngôn ngữ sang ngôn ngữ khác.

3.3.4 Lợi ích và thử thách

- BERT mang lại một số lợi ích so với các mô hình ngôn ngữ truyền thống, bao gồm:
 - **Độ chính xác:** BERT đã được chứng minh là đạt được kết quả tiên tiến trên nhiều loại tác vụ NLP.
 - **Tính linh hoạt:** BERT có thể được sử dụng cho nhiều loại tác vụ NLP, từ phân loại văn bản đến trả lời câu hỏi.
 - **Nguồn mở:** BERT là một mô hình mã nguồn mở, có nghĩa là nó có sẵn miễn phí cho bất kỳ ai sử dụng và sửa đổi.
- BERT là một mô hình lớn và phức tạp, có thể khiến nó khó đào tạo và sử dụng. BERT cũng yêu cầu nhiều tài nguyên tính toán, có thể khiến nó tốn kém để triển khai.

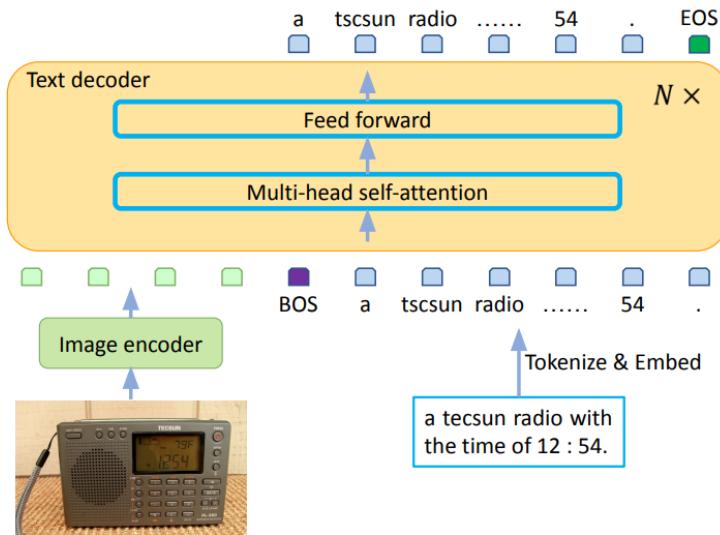
3.4 GIT: A Generative Image-to-text Transformer for Vision and Language [4]

3.4.1 Giới thiệu

GenerativeImage2Text (GIT) là một mô hình ngôn ngữ (Language Model) được phát triển bởi Microsoft AI. Nó được giới thiệu lần đầu tiên vào năm 2022 và đã đạt được kết quả tiên tiến trên các tác vụ mô tả hình ảnh và trả lời câu hỏi về hình ảnh.

GIT là một mô hình chuyển đổi (Transformer) hai chiều, được đào tạo trên một tập dữ liệu khổng lồ gồm hình ảnh và văn bản mô tả hình ảnh. Tập dữ liệu này bao gồm hình ảnh từ nhiều nguồn khác nhau, chẳng hạn như ImageNet và Visual Genome.

3.4.2 Kiến trúc



Hình 3.9: Kiến trúc GIT.

- **Image encoder:** được đào tạo trên một bộ dữ liệu lớn các hình ảnh và văn bản mô tả hình ảnh, và học cách phân biệt giữa các cặp hình ảnh tương tự và khác nhau (contrastive pre-trained model).
 - Sử dụng mô hình dựa trên contrastive pre-training để biểu diễn hóa ảnh thành một danh sách đặc trưng 2D.
 - Áp dụng lớp linear và layernorm để chuyển đổi các đặc trưng thành không gian D chiều.
- **Text decoder:**
 - Sử dụng một module transformer với nhiều khối transformer, bao gồm self-attention và feed-forward layers.
 - Mã hóa văn bản thành các token và nhúng chúng vào không gian D chiều, kết hợp với mã hóa vị trí và layernorm.
 - Bắt đầu với token [BOS] và tiếp tục giải mã theo cách tự học giữa các token đến khi gặp token [EOS] hoặc đạt đến số bước tối đa.

3.4.3 Ứng dụng

GIT có thể được sử dụng cho nhiều loại tác vụ xử lý ngôn ngữ tự nhiên (NLP) liên quan đến hình ảnh, bao gồm:

- **Mô tả hình ảnh:** GIT có thể được sử dụng để tạo văn bản mô tả hình ảnh.
- **Trả lời câu hỏi về hình ảnh:** GIT có thể được sử dụng để trả lời các câu hỏi về hình ảnh, chẳng hạn như "Có ai trong ảnh không?" hoặc "Cánh trong ảnh là gì?"
- **Tạo hình ảnh từ văn bản:** GIT có thể được sử dụng để tạo hình ảnh từ văn bản mô tả hình ảnh.

3.4.4 Lợi ích và thử thách

- GIT mang lại một số lợi ích so với các mô hình mô tả hình ảnh truyền thống, bao gồm:
 - **Độ chính xác:** GIT đã được chứng minh là đạt được kết quả tiên tiến trên nhiều loại tác vụ mô tả hình ảnh.
 - **Tính linh hoạt:** GIT có thể được sử dụng cho nhiều loại tác vụ mô tả hình ảnh, từ mô tả hình ảnh đơn giản đến trả lời câu hỏi về hình ảnh phức tạp.



- **Nguồn mở:** GIT là một mô hình mã nguồn mở, có nghĩa là nó có sẵn miễn phí cho bất kỳ ai sử dụng và sửa đổi.
- GIT là một mô hình lớn và phức tạp, có thể khiến nó khó đào tạo và sử dụng. GIT cũng yêu cầu nhiều tài nguyên tính toán, có thể khiến nó tốn kém để triển khai.

Chương 4

Huấn luyện mô hình

Quá trình huấn luyện trên cả 2 tập dataset được hoàn thành trên Google Colab, với T4 GPU được sử dụng

Colab cung cấp truy cập miễn phí đến GPU T4 của Google. GPU này được phát triển bởi NVIDIA và được tối ưu hóa cho công việc học máy và tính toán thông qua mạng. Dưới đây là một số thông số cơ bản của GPU T4:

- Kiến trúc: Turing
- GPU RAM: 15GB
- Hỗ trợ cho Tensor Cores để tăng tốc tính toán tensor

GPU T4 cung cấp một giải pháp mạnh mẽ và hiệu quả cho các tác vụ học máy và tính toán phức tạp trên nền tảng Colab, cung cấp sức mạnh tính toán đáng kể để xử lý các tác vụ lớn.

Với giới hạn về GPU RAM và sự khác biệt về kích thước và độ phức tạp giữa 2 dataset, nhóm sẽ thay đổi config training cho phù hợp.

4.1 Pokemon

- learning rate: $5e - 5$
- train batch size: 8
- eval batch size: 16
- eval by epoch

Epoch	Training Loss	Validation Loss	Wer Score
0	9.158200	7.557196	74.681159
1	6.862600	5.745167	21.407076
2	4.977800	3.866543	11.232310
4	1.366500	0.636763	4.313299
5	0.377500	0.142961	1.605712
6	0.088700	0.047337	0.849531
8	0.022900	0.030367	0.506394
9	0.017700	0.029864	0.462063
10	0.013300	0.029311	1.227195
12	0.009100	0.030604	1.867860

Hình 4.1: Train progress với Pokemon dataset

- time training: 18 minutes
- early stopping: True

4.2 Flickr8k

- learning rate: $5e - 5$
- train batch size: 8
- eval batch size: 16
- eval by steps
- eval steps: 50

Step	Training Loss	Validation Loss	Wer Score
50	7.114700	4.257147	1.803492
100	2.012600	0.283577	6.577424
150	0.113500	0.069973	8.001378
200	0.050100	0.067654	10.451218
250	0.033500	0.068245	10.812529
300	0.022200	0.071887	12.210752
350	0.014100	0.074315	9.185633
400	0.008400	0.077360	10.007352
450	0.005900	0.079654	10.657528
500	0.004200	0.081749	10.687548
550	0.003100	0.083451	10.299740
600	0.002400	0.083748	10.847144

Hình 4.2: Train progress với Flickr8k dataset

- time training: 40 minutes
- early stopping: True

Chương 5

Kết quả

5.1 Kết quả dự đoán của mô hình



(a) a drawing of a fire breathing dragon



(b) a candle with a blue flame on top of it

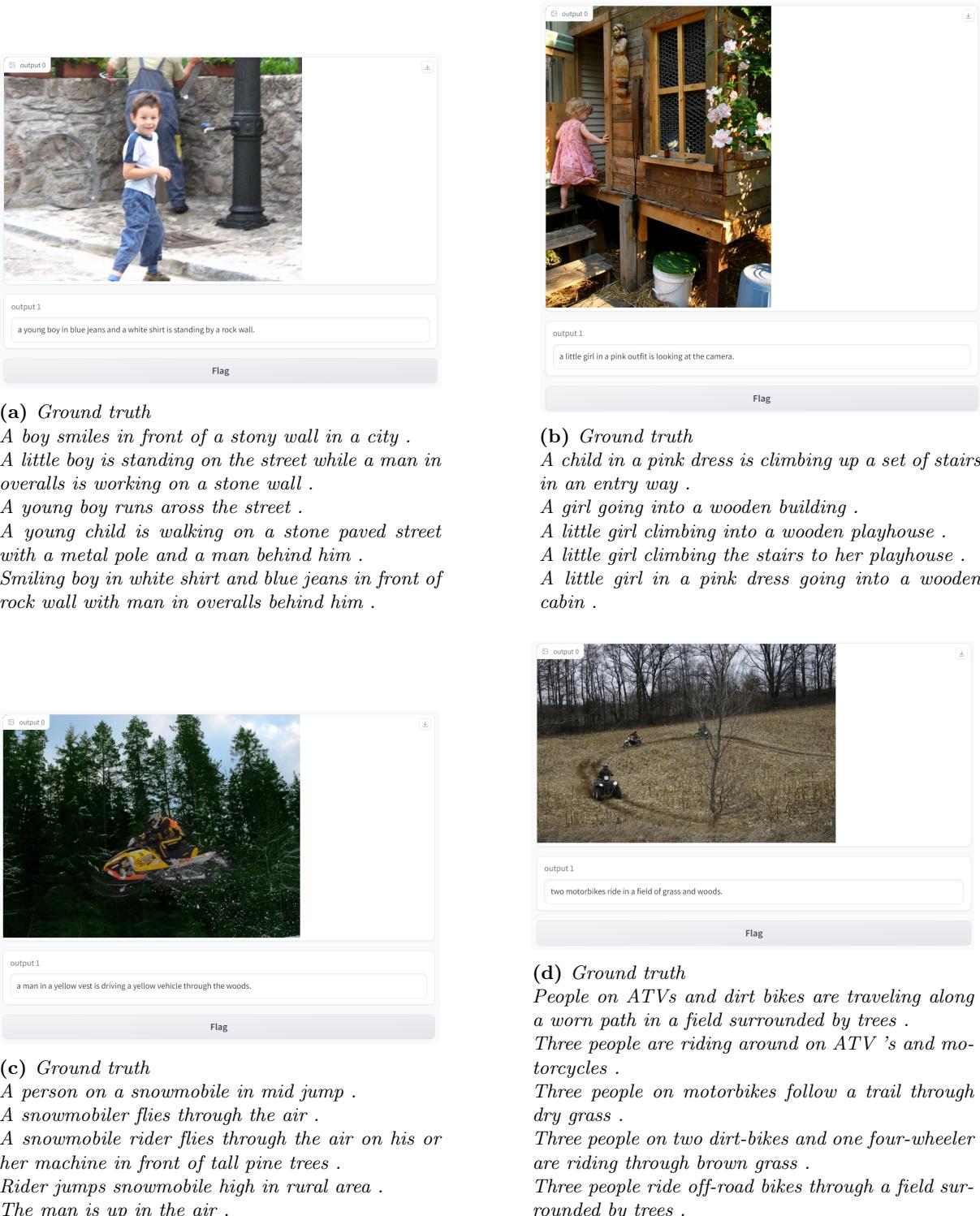


(c) a red and blue dragon with its mouth open



(d) a cartoon character with a fork and spoon in his hand

Hình 5.1: Kết quả dự đoán với Pokemon dataset



Hình 5.2: Kết quả dự đoán với Flickr8k dataset

5.2 Phương pháp đánh giá: WER

5.2.1 Giới thiệu

- Chỉ số phổ biến được sử dụng để đánh giá hiệu suất của các hệ thống nhận dạng giọng nói tự động (ASR)
- được tính bằng tỷ lệ phần trăm lỗi của kết quả dự đoán so với số lượng từ trong kết quả tham



chiếu.

5.2.2 Công thức

$$WER = \frac{I + D + S}{N} \quad (5.1)$$

Trong đó:

- I: Số lần chèn (các từ trong kết quả dự đoán bị thiếu so với bản ghi tham chiếu)
- D: Số lần xóa (các từ trong kết quả dự đoán bị dư so với bản ghi tham chiếu)
- S: Số lần thay thế
- N: Số lượng từ trong bản ghi tham chiếu

5.2.3 Hạn chế

- WER không tính đến mức độ nghiêm trọng của lỗi. Một lỗi thay thế đơn lẻ có thể ít nghiêm trọng hơn lỗi xóa, có thể ít nghiêm trọng hơn lỗi chèn.
- WER không tính đến ngữ cảnh của các lỗi. Lỗi thay thế trong một từ khóa có thể nghiêm trọng hơn lỗi thay thế trong một từ không phải là từ khóa.

5.3 Kết luận

Flickr8k Dataset

- Dựa vào kiến trúc của mô hình, đó là trích xuất đặc trưng và mapping với nhãn caption đã chuyển thành token, từ đó mô hình dự đoán đúng các đặc điểm về tính chất, màu sắc, vật thể trong bức hình.
- Tuy nhiên về mặt logic như số lượng thì mô hình còn gặp hạn chế.

Pokemon Dataset

- Dataset được label chưa thực sự chính chu, câu mô tả tương đối ít thông tin dẫn đến mô hình cố gắng học thì vẫn không thể diễn tả phong phú hơn được.

Chương 6

Tổng kết

Các đóng góp của nhóm

- Template training model với framework của Huggingface
- Chuẩn bị dữ liệu và huấn luyện: pokemon và flickr8k
- Dánh giá chất lượng mô hình và demo

Hướng phát triển thêm

- So sánh với một số mô hình Image Captioning khác
- Thử nghiệm với tập dữ liệu lớn và phong phú hơn
- Thử nghiệm với tập dữ liệu là video.

Tài liệu tham khảo

- [1] adityajn105. Flickr 8k dataset. <https://www.kaggle.com/datasets/adityajn105/flickr8k>, 2019.
- [2] Niki Parmar Jakob Uszkoreit Llion Jones Aidan N. Gomez Lukasz Kaiser Illia Polosukhin Ashish Vaswani, Noam Shazeer. Attention is all you need. 2017.
- [3] Kenton Lee Kristina Toutanova Jacob Devlin, Ming-Wei Chang. Bert: Pre-training of deep bidirectional transformers for language understanding. 2018.
- [4] Xiaowei Hu Linjie Li Kevin Lin Zhe Gan Zicheng Liu Ce Liu Lijuan Wang Jianfeng Wang, Zhengyuan Yang. Git: A generative image-to-text transformer for vision and language. 2022.
- [5] Justin N. M. Pinkney. Pokemon blip captions. <https://huggingface.co/datasets/lambdalabs/pokemon-blip-captions/>, 2022.