

Mục lục

I. State of the Art OCR.....	2
1. OCR là gì và OCR làm được gì?	2
2. Công nghệ OCR tốt nhất hiện nay và bộ phận cấu thành cấu trúc đó	2
2.1. Transformer	5
2.2. DTrOCR: Decoder-only Transformer for Optical Character Recognition	17
II . Quy Trình Xây Dựng Một Ứng Dụng OCR	21
Bước 1: Thu Thập Dữ Liệu	21
Bước 2: Tiền Xử Lý Hình Ảnh (Image Preprocessing)	22
Bước 3: Text Detection	22
Bước 4: Text Recognition	23
Bước 5: Trích xuất thông tin (optional)	23
Bước 6: Hậu xử lý	24
III. Fine-Tune Mô Hình OCR	25
IV. OCR tiếng việt.....	26
4.1. Giới thiệu.....	26
4.2. Cấu trúc	27
a) Đầu tiên ảnh đi qua một module Dynamic High Resolution	27
b) Tiếp theo 5 ảnh sẽ đi qua một lớp mã hóa hình ảnh (Vision Encoder): Sử dụng mô hình InternViT-300M-448px.	28
c) Tiếp theo khi đã được vector chứa mọi thông tin cần có của ảnh , vector sẽ đi qua Pixel Shuffle và MLP Projector	31
d) Tiếp theo ta đến với phần mô hình ngôn ngữ lớn (LLM) Qwen2-0.5B-Instruct.	31
4.3. Dataset	31
4.4. Chi tiết quá trình thực hiện.....	32
4.5. Đánh giá	34

I. State of the Art OCR

1. OCR là gì và OCR làm được gì?

Nhận dạng ký tự quang học là quá trình trích xuất và chuyển đổi văn bản viết tay hoặc đánh máy từ hình ảnh, video hoặc tài liệu được quét như PDF sang định dạng có thể sửa đổi kỹ thuật số (txt, docx, v.v.). Đây là lĩnh vực nghiên cứu về trí tuệ nhân tạo gắn liền với thị giác máy tính và nhận diện mẫu

Với công nghệ OCR, chúng ta có thể thực hiện nhiều nhiệm vụ hữu ích. Đầu tiên, OCR giúp số hóa tài liệu bằng cách biến các tài liệu giấy thành tài liệu điện tử, từ đó giúp bảo quản và quản lý thông tin hiệu quả hơn. Thứ hai, văn bản được số hóa cho phép người dùng dễ dàng tìm kiếm thông tin cụ thể trong tài liệu mà không cần phải quét từng trang. Trong các lĩnh vực như kế toán, ngân hàng, và bảo hiểm, OCR có thể tự động hóa việc nhập dữ liệu từ các biểu mẫu giấy tờ, giảm thiểu sai sót do con người gây ra. Hơn nữa, OCR còn có khả năng nhận diện văn bản trong các bức ảnh hoặc video, chẳng hạn như văn bản trên biển hiệu, quảng cáo, hoặc trong các cảnh quay. Cuối cùng, OCR trong ngân hàng có thể thay thế một số tác vụ cần nhân viên như scan hồ sơ. Hiện nay quá trình chuyển tài liệu giấy, các hợp đồng viết tay sang dữ liệu lưu trữ dễ quản lý (dạng json) đang gặp khó khăn. Hầu hết tác vụ đang làm tay và tốn nhân công, việc để con người xử lý việc này vẫn có thể dẫn đến chậm trễ và sai sót. Việc chuyển đổi số quá trình scan tài liệu tiếng việt hiện nay đang được quan tâm và đạt được mức độ đáng kinh ngạc. OCR giúp quá trình đạt tốc độ nhanh, xử lý ít xảy ra sai sót hơn so với con người, ngoài ra còn tiết kiệm chi phí nhân công cho doanh nghiệp.

2. Công nghệ OCR tốt nhất hiện nay và bộ phận cấu thành cấu trúc đó

Công nghệ OCR có thể được sử dụng trên nhiều loại hình ảnh, từ các tài liệu được in đến các tài liệu viết tay. Một số công nghệ OCR phổ biến hiện nay bao gồm DTrOCR, Tesseract OCR, AttentionOCR với Inception-resnet-v2_Location, I2L-NOPOOL và I2L-STRIPS.

Về các mô hình OCR tốt nhất hiện nay, mỗi mô hình đều đạt tỷ lệ chính xác cao trên mỗi bộ dataset khác nhau. Bảng xếp hạng lấy ở <https://paperswithcode.com/task/optical-character-recognition> Trang **Papers with Code** (<https://paperswithcode.com/>) là một nguồn đáng tin cậy và được sử dụng rộng rãi trong cộng đồng nghiên cứu trí tuệ nhân tạo (AI) và học máy (Machine Learning). Trang web này liên kết các bài báo khoa học với mã nguồn mở, giúp các nhà nghiên cứu và kỹ sư có thể theo dõi tiến độ của các mô hình hiện đại nhất trên các tác vụ cụ thể.
















Đánh giá trên từng bộ data:

- Benchmarking Chinese Text Recognition: Datasets, Baselines, and an Empirical Study : Mô hình tốt nhất: DTrOCR
- FSNS – Test : Mô hình tốt nhất: AttentionOCR_Inception-resnet-v2_Location
- I2L-140K: Mô hình tốt nhất: I2L-NOPOOL
- SUT: Mô hình tốt nhất: Tesseract
- im2latex-100k: Mô hình tốt nhất: I2L-STRIPS

Benchmarks

[Add a Result](#)

These leaderboards are used to track progress in Optical Character Recognition (OCR)

Trend	Dataset	Best Model	Paper	Code	Compare
	Benchmarking Chinese Text Recognition: Datasets, Baselines, and an Empirical Study	DTrOCR			See all
	FSNS - Test	AttentionOCR_Inception-resnet-v2_Location			See all
	I2L-140K	I2L-NOPOOL			See all
	SUT	Tesseract			See all
	im2latex-100k	I2L-STRIPS			See all

1) DTrOCR:

- Đứng đầu trên bộ dữ liệu *Benchmarking Chinese Text Recognition*.
- DTrOCR là một mô hình OCR tiên tiến dựa trên kiến trúc Transformer chỉ với bộ giải mã. Mô hình này nổi bật trong việc nhận dạng văn bản phức tạp, đặc biệt là với tiếng Trung, nhờ vào khả năng sử dụng mô hình ngôn ngữ mạnh mẽ và bỏ qua bộ mã hóa (encoder).
- Phù hợp cho các tác vụ OCR yêu cầu độ chính xác cao và có khả năng hiểu ngữ cảnh ngôn ngữ tốt.

2) AttentionOCR với Inception-resnet-v2_Location:

- Đứng đầu trên bộ dữ liệu *FSNS - Test*.
- Đây là một mô hình OCR hiện đại sử dụng cơ chế Attention và kiến trúc Inception-ResNet-v2. AttentionOCR phù hợp cho việc nhận dạng văn bản trong ảnh tự nhiên

(scene text), đặc biệt là khi văn bản nằm trong các môi trường có nền phức tạp như biển hiệu và các ký tự trong cảnh ngoài trời.

- Sử dụng bộ mã hóa hình ảnh (vision encoder) để nắm bắt đặc trưng hình ảnh và vị trí, giúp mô hình đạt độ chính xác cao trên văn bản trong cảnh tự nhiên.

3) I2L-NOPOOL:

- Đứng đầu trên bộ dữ liệu *I2L-140K*.
- Mô hình I2L-NOPOOL được phát triển để tối ưu hóa cho việc nhận dạng ký tự trong các tập dữ liệu lớn và phức tạp. Mô hình này được tối ưu để xử lý các văn bản với nhiều loại phong chữ và định dạng, chủ yếu trong văn bản in.
- Đây là một lựa chọn tốt cho các tác vụ OCR yêu cầu nhận dạng chính xác với các loại văn bản in phức tạp và nhiều biến thể.

4) Tesseract:

- Đứng đầu trên bộ dữ liệu *SUT*.
- Tesseract là một trong những mô hình OCR phổ biến nhất và là mã nguồn mở do Google phát triển. Tesseract được sử dụng rộng rãi trong các tác vụ OCR cơ bản và hiệu quả với các văn bản in thông thường. Tuy không có khả năng nhận dạng cảnh phức tạp như các mô hình hiện đại khác, nhưng Tesseract là lựa chọn nhẹ và có hiệu quả cho các nhu cầu OCR thông thường.
- Phù hợp cho các tác vụ OCR với văn bản đơn giản và không yêu cầu khả năng phân tích ngữ cảnh mạnh mẽ.

5) I2L-STRIPS:

- Đứng đầu trên bộ dữ liệu *im2latex-100k*.
- Mô hình này được tối ưu hóa cho việc nhận dạng các ký hiệu phức tạp và các công thức, thường gặp trong văn bản toán học. Nó được sử dụng phổ biến trong các tác vụ chuyển đổi công thức toán học từ hình ảnh sang định dạng mã máy (LaTeX).
- Phù hợp cho OCR trong tài liệu khoa học, toán học và các lĩnh vực yêu cầu nhận dạng ký hiệu đặc biệt.

Tuy nhiên, việc sử dụng công nghệ OCR cũng có những hạn chế. Ví dụ, các hình ảnh có độ phân giải thấp hoặc các hình ảnh với định dạng không chuẩn có thể không được nhận dạng chính xác. Do đó, việc sử dụng công nghệ OCR đòi hỏi sự chính xác trong việc chọn các

công nghệ OCR phù hợp với yêu cầu của từng ứng dụng cụ thể. Ngoài ra, việc sử dụng các công nghệ xử lý hình ảnh khác như các thuật toán tiền xử lý, nhận dạng ký tự và phân tích cú pháp cũng là những yếu tố quan trọng giúp cho quá trình OCR trở nên hiệu quả và chính xác hơn. Các mô hình mới sinh ra có đạt được hiệu quả cao hay không đều phải giải quyết được những khó khăn này.

2.1. Transformer

Tại sao chúng tôi lại đề cập đến Transformer trong bài viết này, vì đây là cấu trúc tốt được ứng dụng không những với NLP mà còn với Computer Vision. Cấu trúc của Transformer được ứng dụng nhiều trong các cấu trúc của các ứng dụng CV hay rõ hơn là OCR (ViT/Transformer). Tìm hiểu Transformer giúp chúng ta có nhiều lợi thế trong tìm hiểu các ứng dụng về OCR.

2.1.1. Sự ra đời và ưu thế so với RNN

Tại sao lại có transformer? Transformer là một kiến trúc mạng nơ-ron được giới thiệu lần đầu trong bài báo "Attention is All You Need" vào năm 2017, và nó đã cách mạng hóa lĩnh vực xử lý ngôn ngữ tự nhiên (NLP) và nhiều lĩnh vực khác, bao gồm cả nhận dạng văn bản. Hiện nay mô hình ViT/Transformer dựa trên Transformer cũng đạt được nhiều thành tựu. Lợi thế của Transformer so với các cấu trúc khác cũng như so với cấu trúc tốt nhất thời đó là RNN:

a) **Cơ chế chú ý (Attention Mechanism):**

Transformer sử dụng cơ chế chú ý để cho phép mô hình tập trung vào các phần quan trọng của dữ liệu đầu vào, bất kể vị trí của chúng trong chuỗi. Điều này cực kỳ hữu ích trong việc xử lý các chuỗi dài, như văn bản, vì nó cho phép mô hình học cách liên kết các từ hoặc ký tự có liên quan, ngay cả khi chúng cách xa nhau.

b) **Khả năng song song hóa:**

Khác với các mô hình RNN truyền thống, nơi mà dữ liệu phải được xử lý tuần tự, Transformer có thể xử lý tất cả các phần của dữ liệu đầu vào đồng thời. Điều này giúp giảm thiểu thời gian huấn luyện và cho phép khai thác hiệu quả hơn về tài nguyên tính toán.

c) **Khả năng mở rộng:**

Transformer có khả năng mở rộng tốt với kích thước mô hình và dữ liệu huấn luyện lớn. Nhiều mô hình Transformer lớn như BERT, GPT và T5 đã được chứng minh là đạt được kết quả xuất sắc trên nhiều tác vụ NLP, cho thấy rằng chúng có thể học các đại diện ngữ nghĩa phức tạp từ dữ liệu.

d) **Tính linh hoạt:**

Transformer có thể áp dụng cho nhiều loại dữ liệu khác nhau, không chỉ giới hạn ở văn bản mà còn có thể được sử dụng cho hình ảnh, âm thanh, và các loại dữ liệu khác. Điều này làm cho Transformer trở thành một kiến trúc đa dụng trong các bài toán học sâu.

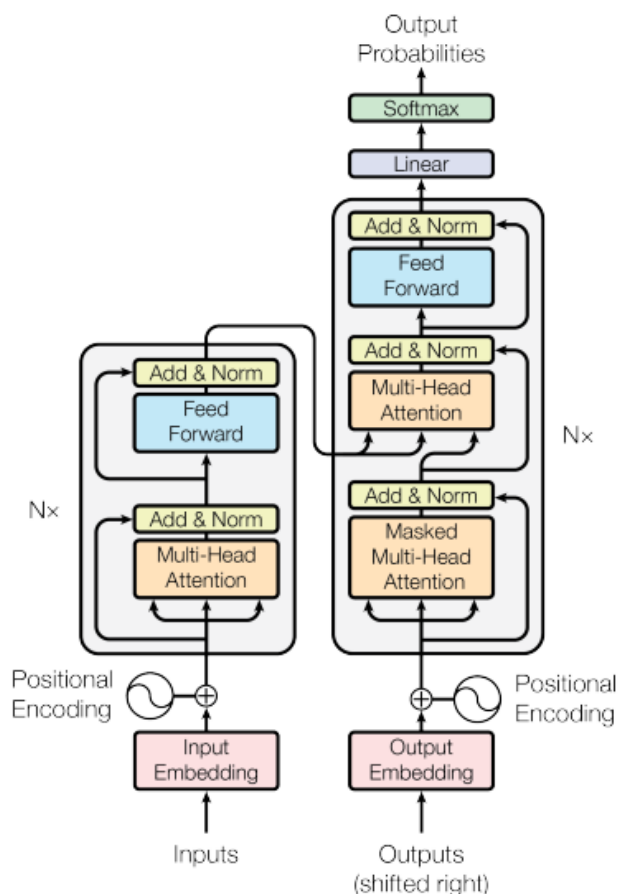
e) **Hiệu suất vượt trội:**

Nhiều nghiên cứu đã chỉ ra rằng các mô hình dựa trên Transformer thường đạt hiệu suất cao hơn so với các mô hình dựa trên RNN hoặc CNN trong các tác vụ như dịch máy, phân tích cảm xúc, và nhận dạng văn bản.

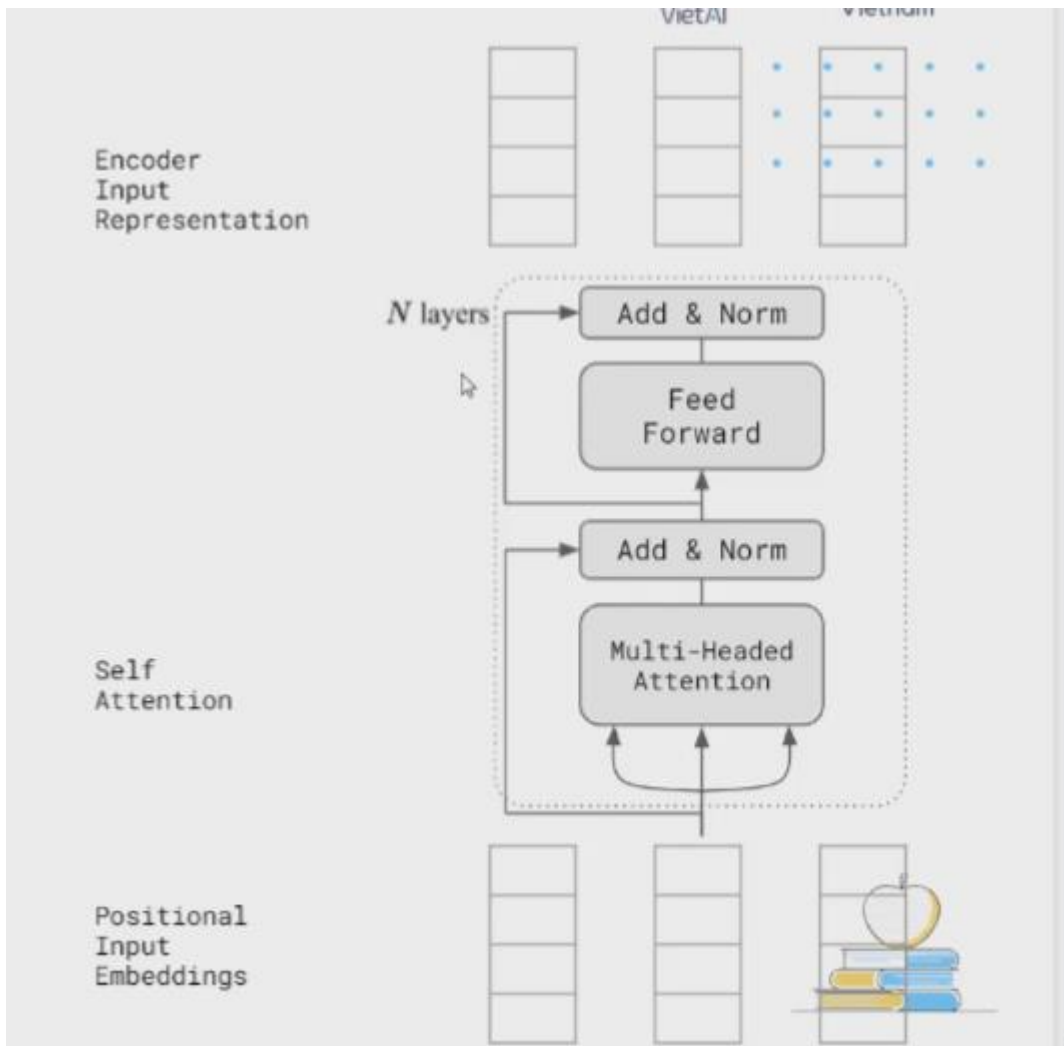
f) **Dễ dàng điều chỉnh và cải thiện:**

Transformer có thể dễ dàng được điều chỉnh cho các tác vụ cụ thể thông qua việc tiền huấn luyện trên các tập dữ liệu lớn và sau đó tinh chỉnh trên các tập dữ liệu nhỏ hơn, giúp cải thiện hiệu suất cho các ứng dụng cụ thể.

2.1.2. Cấu trúc



a) Encoder



Phần encoder có nhiệm vụ : Học mối tương quan giữa các từ trong câu với các từ còn lại, sau đó bổ sung mối quan hệ đó vào embedding của từng từ đầu vào.

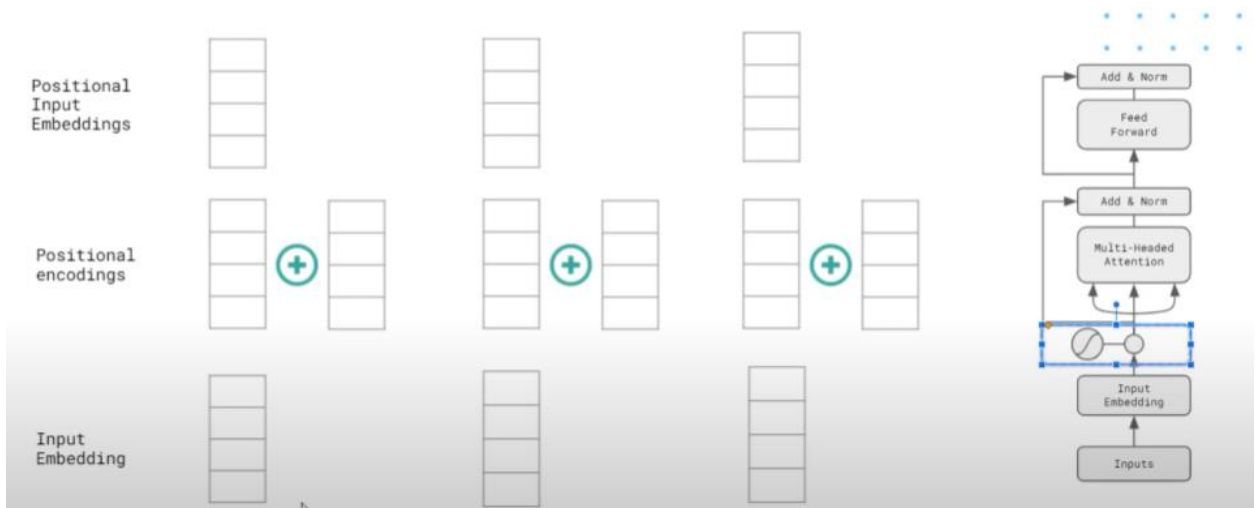
Phần encoder bao gồm 6 layer giống nhau: Mỗi layer có 2 phần nhỏ là multi head attention và position-wise fully connected feed forward network. Thông tin của các từ sau khi đã được biến đổi nhiều lần qua encoder được đưa vào decoder để thực hiện các nhiệm vụ. Ở đây đề cập đến nhiệm vụ dịch máy.

Đầu tiên từ đầu vào sẽ được đưa qua lớp embedding. Từ đầu vào sẽ được biến đổi thành các vector vì máy tính không hiểu câu chữ mà chỉ đọc được số, vector, ma trận; vì vậy ta phải biểu diễn câu chữ dưới dạng vector, gọi là input embedding. Điều này đảm bảo các từ gần nghĩa có vector gần giống nhau. Hiện đã có khá nhiều pretrained word embeddings như GloVe, Fasttext, gensim Word2Vec,... cho bạn lựa chọn.

Input Embedding

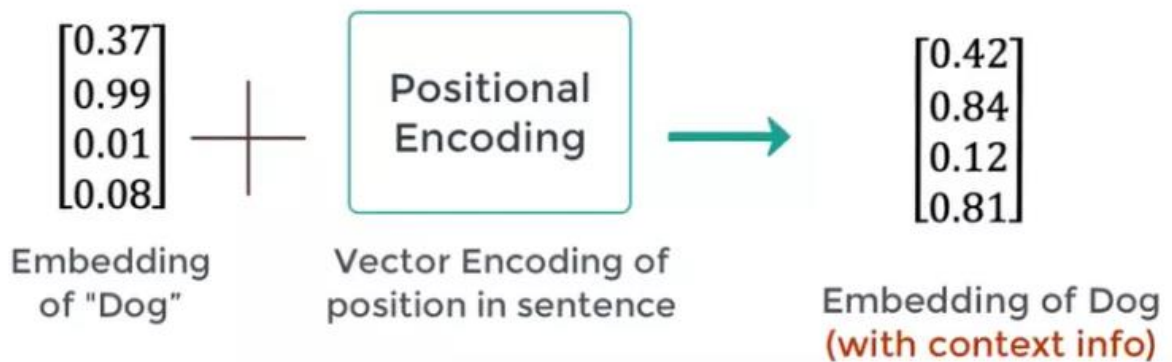
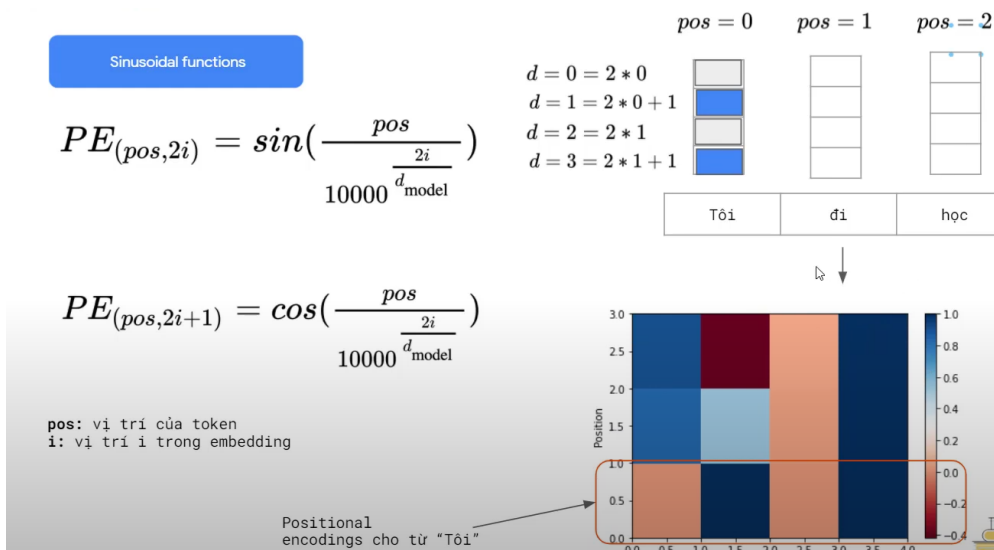


Word embeddings phần nào cho giúp ta biểu diễn ngữ nghĩa của một từ, tuy nhiên cùng một từ ở vị trí khác nhau của câu lại mang ý nghĩa khác nhau. Đó là lý do Transformers có thêm một phần Positional Encoding để inject thêm thông tin về vị trí của một



Có 2 cách bổ sung thông tin vị trí này đó là tạo 1 mạng nơ ron để học vị trí hoặc có thể tạo 1 hàm tự định nghĩa. Ở paper về transformer cơ bản, người ta ưu tiên sử dụng hàm tự định nghĩa. Vậy hàm này là gì?

Sau đây là biểu diễn vector vị trí cho 3 từ tôi đi học thông qua hàm sau(d model là chiều dài của vector) :



$$PE_{(pos, 2i)} = \sin(pos/10000^{2i/d_{model}})$$

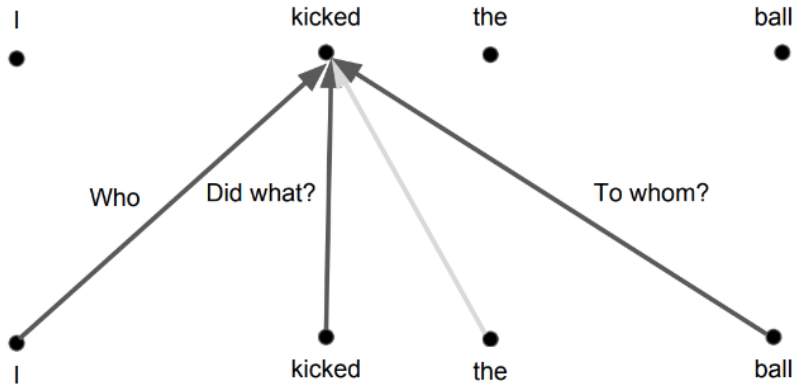
$$PE_{(pos, 2i+1)} = \cos(pos/10000^{2i/d_{model}+1})$$

Sau khi vector input embedding được cộng với vector vị trí tạo thành vector chứa thông tin vị trí. Sau khi đã được bổ sung thông tin vị trí, các từ cần được bổ sung thêm thông tin về attention- tức là bổ sung thêm thông tin về mối quan hệ của từ đó với các từ khác trong câu. Vậy việc này làm như thế nào?

b) Self-Attention

Self-Attention là cơ chế giúp Transformers "hiểu" được sự liên quan giữa các từ trong một câu. Ví dụ như từ "kicked" trong câu "I kicked the ball" (tôi đã đá quả bóng) liên quan như thế nào đến các từ khác? Rõ ràng nó liên quan mật thiết đến từ "I" (chủ ngữ), "kicked" là chính nó lên sẽ luôn "liên quan mạnh" và "ball" (vị ngữ). Ngoài ra từ "the" là giới từ nên sự liên kết với từ "kicked" gần như không có. Vậy Self-Attention trích xuất những sự "liên quan" này như thế nào.

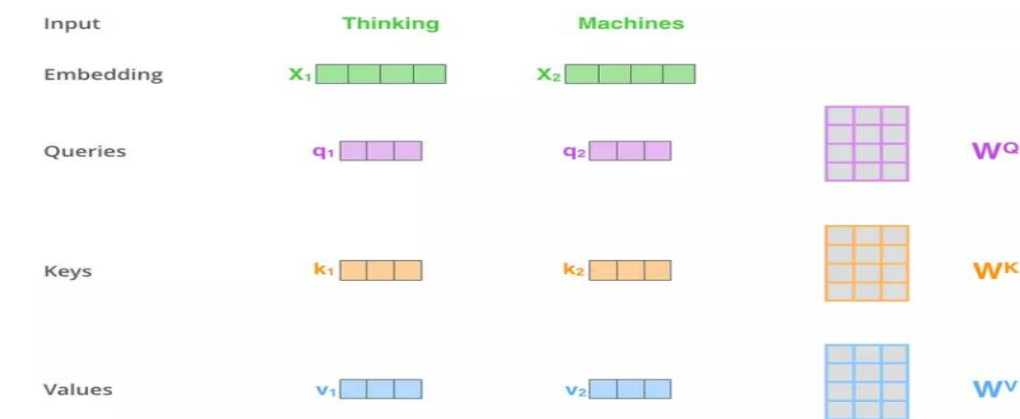
Self-Attention



Quay trở lại với kiến trúc tổng thể ở trên, các bạn có thể thấy đầu vào của các module Multi-head Attention (bản chất là Self-Attention) có 3 mũi tên, đó chính là 3 vectors Querys (Q), Keys (K) và Values (V). Từ 3 vectors này, ta sẽ tính vector attention Z cho một từ theo công thức sau:

$$Z = \text{softmax} \left(\frac{Q \cdot K^T}{\sqrt{\text{Dimension of vector } Q, K \text{ or } V}}} \right) \cdot V$$

Công thức này khá đơn giản, nó được thực hiện như sau. Đầu tiên, để có được 3 vectors Q, K, V, input embeddings được nhân với 3 ma trận trọng số tương ứng (được tune trong quá trình huấn luyện) W^Q , W^K , W^V .



Lúc này, Vector K đóng vai trò như một khóa đại diện cho từ, và Q sẽ truy vấn đến các vector K của các từ trong câu bằng cách nhân chập với những vector này. Mục đích của phép nhân chập để tính toán độ liên quan giữa các từ với nhau. Theo đó, 2 từ liên quan đến nhau sẽ có "Score" lớn và ngược lại.

Bước thứ 2 là bước "Scale", đơn giản chỉ là chia "Score" cho căn bậc hai của số chiều của Q/K/V (trong hình chia 8 vì Q/K/V là 64-D vectors). Việc này giúp cho giá trị "Score" không phụ thuộc vào độ dài của vector Q/K/V

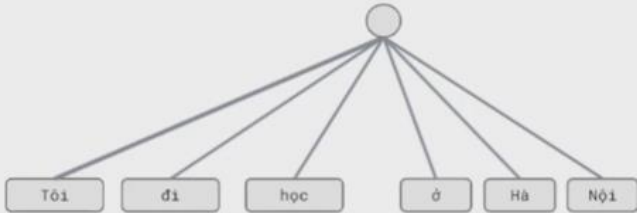
Bước thứ 3 là softmax các kết quả vừa rồi để đạt được một phân bố xác suất trên các từ.

Bước thứ 4 ta nhân phân bố xác suất đó với vector V để loại bỏ những từ không cần thiết (xác suất nhỏ) và giữ lại những từ quan trọng (xác suất lớn).

Ở bước cuối cùng, các vectors V (đã được nhân với softmax output) cộng lại với nhau, tạo ra vector attention Z cho một từ. Lặp lại quá trình trên cho tất cả các từ ta được ma trận attention cho 1 câu.

Nhưng trong bài không chỉ là attention mà là dùng muti head attention, vậy muti head attention có gì ưu việt hơn attention?

Self-Attention

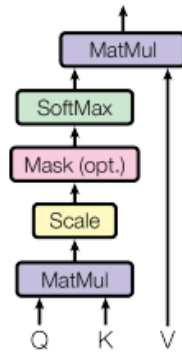


Khi sử dụng nhất duy nhất một Self-Attention, model thực hiện biến đổi tuyến tính trên toàn bộ embedding của các từ, cho nên không có khả năng tương tự như Convolution - bóc tách từng phần nhỏ thông tin tại các khu vực nhất định.

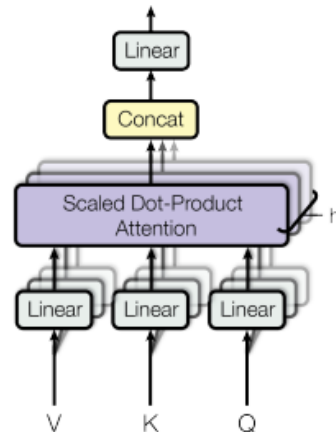
Cách giải quyết???

Cách giải quyết sử dụng nhiều self attention phụ trách học một phần thông tin của câu

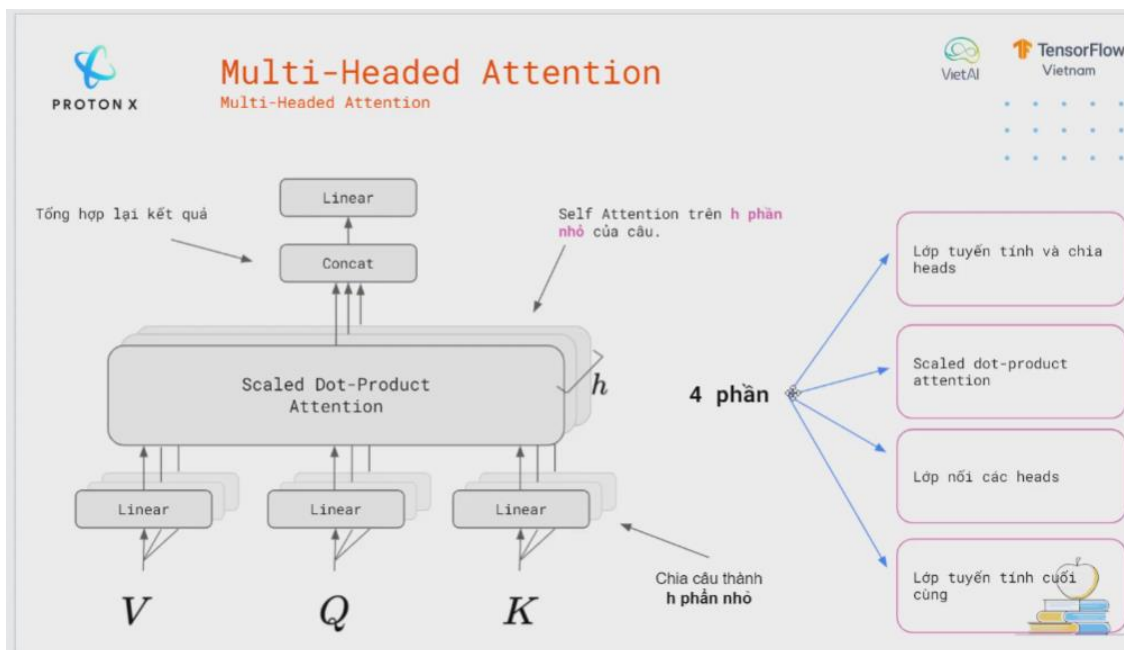
Scaled Dot-Product Attention



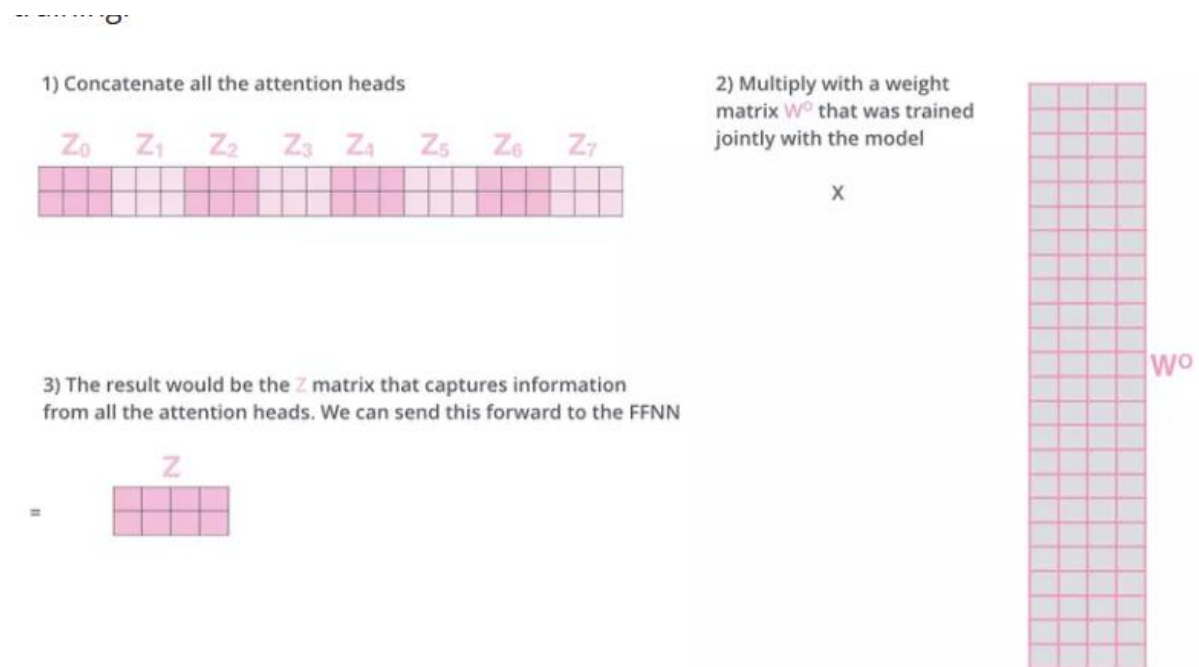
Multi-Head Attention



Multi-head attention cũng vậy. Multi-Head Attention là một kỹ thuật cho phép mô hình học và kết hợp thông tin từ nhiều "đầu" (head) khác nhau, mỗi đầu thực hiện một phép attention riêng biệt. Điều này giúp mô hình nắm bắt được nhiều khía cạnh khác nhau của đầu vào, từ đó cải thiện khả năng hiểu ngữ nghĩa và mối quan hệ giữa các từ trong văn bản. Trong transformer có 8 attention-head học 8 loại quan hệ khác nhau giữa các từ. Mối quan hệ này là mối quan hệ trừu tượng như mạng nơ ron chứ ko phải một mối quan hệ cụ thể. Ở paper ngta chỉ đưa ra 8 head chứ ko giải thích cụ thể tại sao có số 8 này.



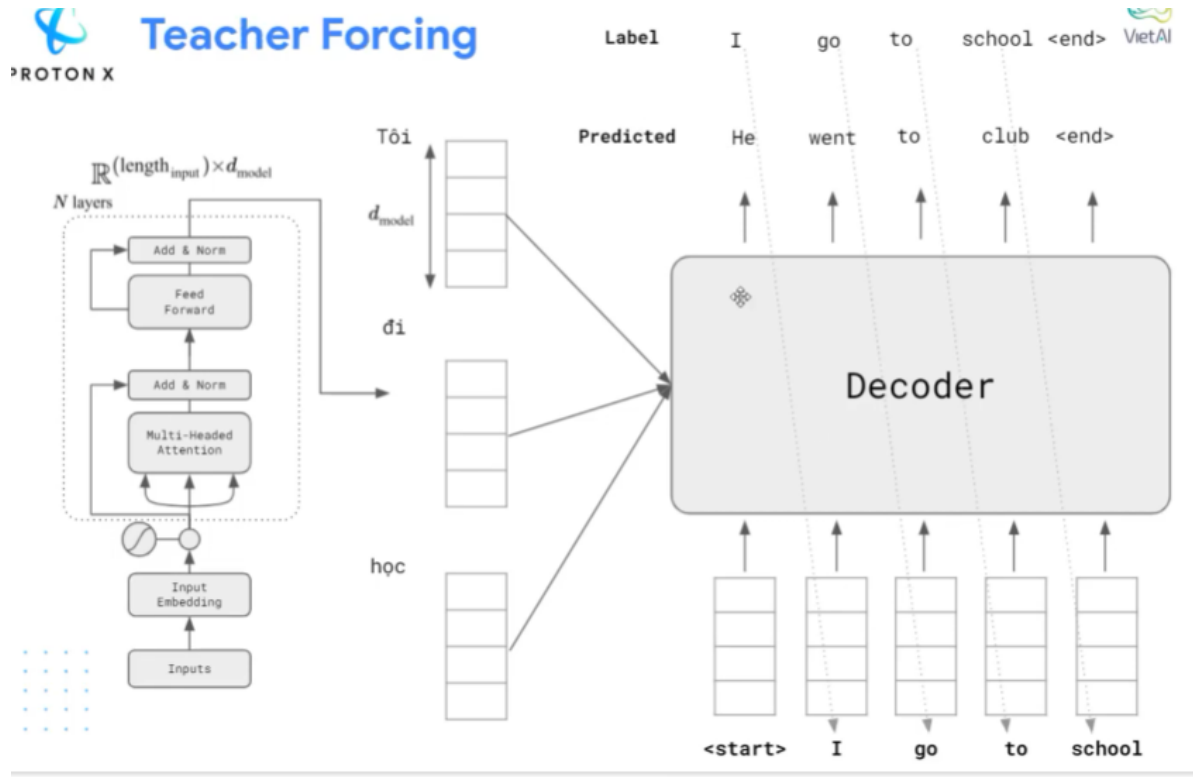
Sau khi lấy được 8 ma trận ta tiếp tục biến đổi lấy ma trận đặc trưng duy nhất có cả 8 đặc trưng bằng cách nhân với ma trận W^O



Sau khi qua multihead attention, vector z tiếp tục được cộng với vector đã chứa thông tin vị trí đầu vào vì các vector sau khi đi qua nhiều phép biến đổi bị mất thông tin vị trí, việc bổ sung thông tin vị trí liên tục giúp giảm thiểu sự mất thông tin dài hạn. Các bạn có thể thấy trong mô hình tổng quan ở trên, mỗi sub-layer đều là một residual block. Cũng giống như residual blocks trong Computer Vision, skip connections trong Transformers cho phép thông tin đi qua sub-layer trực tiếp. Thông tin này (x) được cộng với attention (z) của nó và thực hiện Layer Normalization.

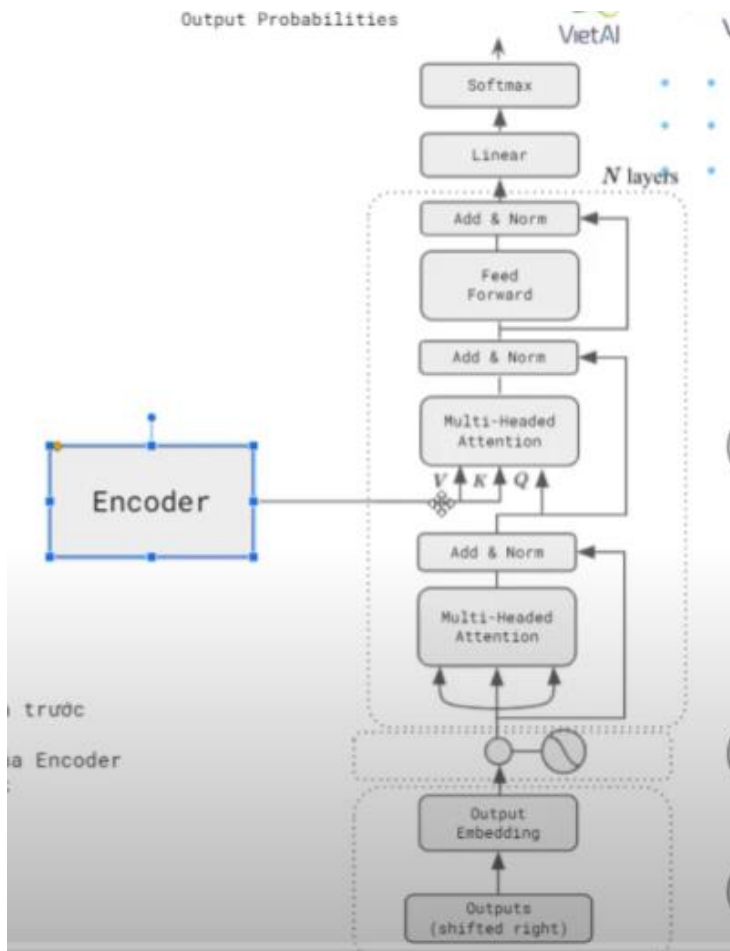
Sau các lớp add-norm thì đến một mạng ffnn, mạng này có vai trò mất sự tuyến tính của mô hình

b) Decoder



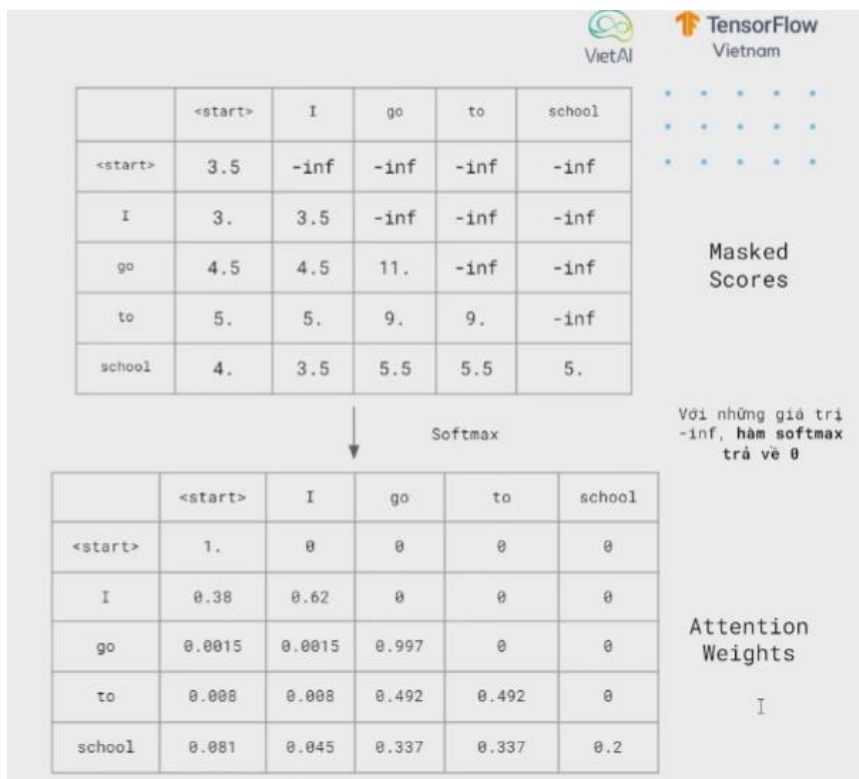
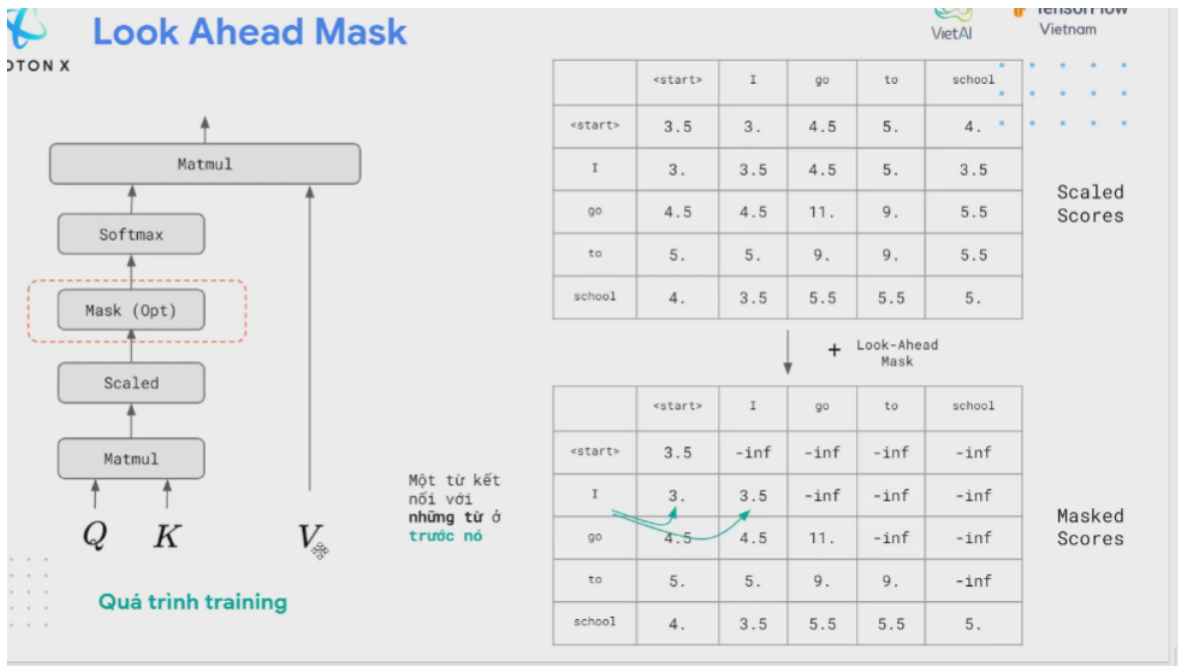
Traning quá trình này là từ từ đầu ra chính xác ta đưa vào mô hình + phần đầu ra của input tạo ra từ mới (Trong hình là tạo ra he went to club). Từ mới này đem so sánh với từ chính xác ta đưa vào trước đó để tính hàm mất mát và làm sao cho hàm này nhỏ nhất

Còn quá trình Inference là quá trình sau khi training, ta đưa tuần tự từng từ của câu dự đoán ví dụ ta đưa từ <start> mô hình dự đoán ra từ I sau đó lại lấy từ đã được dự đoán đưa vào input truyền tuần tự đến khi hết câu. Sau đó ta được câu hoàn chỉnh đó là quá trình Inference. Tiếp theo ta sẽ đi sâu vào mô hình để xem từng từ đưa qua mô hình sẽ như thế nào?



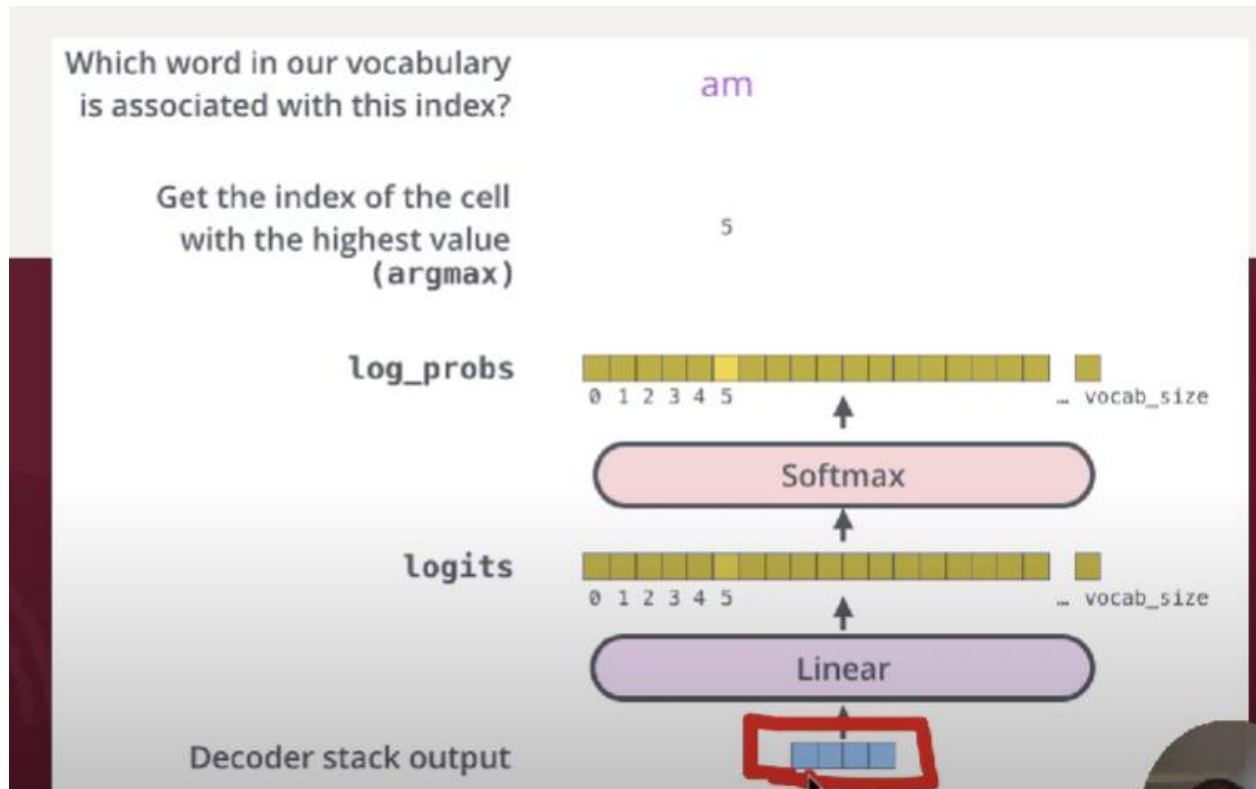
➤ Quá trình decode

Trong RNN khi thực hiện bài toán dịch máy, khi training các từ output được đưa tuần tự vào tuy nhiên ở transformer decoder các từ output sẽ được shifted right (thêm phần đầu và cuối câu vào chuỗi output) sau đó cũng được nhúng vị trí như encoder. Tiếp theo có một vấn đề như thế này: Khi chúng ta Inference thì output sẽ ko biết được từ tiếp theo để mà attention, vì vậy khi huấn luyện, chúng ta cũng phải có cách để từ trước có thể đoán được từ sau. Khi đó người ta đề xuất ra một cách đó là khi huấn luyện ta sẽ che đi lần lượt những từ đằng sau:



Ta được một ma trận có các từ đằng sau chỉ attention với các từ đằng trước. Tiếp theo ta đưa ma trận nhân với W_q sinh ra một tập vector Q (Query), đưa vào lớp multihead attention. Chúng ta đưa từng vector Q query với tập vector K và V từ đầu ra của encoder để tìm ra mối quan hệ giữa các từ trong decoder và encoder. Sau đó lại đưa vào mạng

FFN. Sau 6 lớp decoder như vậy(vẫn bổ sung thông tin vị trí bằng các lớp add- norm như encoder) ta đến với phần cuối cùng.



Sau khi đi qua phần decoder ra đưa các vector đầu ra qua lớp linear chiếu lên ko gian có chiều dài vocab_size-> đưa qua hàm soft max được ma trận nhiều xác suất. Ta lấy từ có xác suất lớn nhất làm đầu ra. Thực hiện như vậy cho đến khi thấy ký tự end.

2.2. DTrOCR: Decoder-only Transformer for Optical Character Recognition

Dựa trên đánh giá, ta có thể thấy không có mô hình nào tốt nhất trên tất cả dataset, chỉ có mô hình tốt nhất trên một bộ dataset cụ thể và đạt tỷ lệ tốt trong các bộ dataset lớn khác. Chúng ta sẽ đi vào một mô hình mới nhất hiện nay có cấu trúc đơn giản nhưng lại đạt độ chính xác đáng kinh ngạc đó là DTrOCR (Decoder-only Transformer for Optical Character Recognition).

2.2.1. Tổng quan

Mô hình DTrOCR được công bố lần đầu tiên vào ngày 30 tháng 8 năm 2023, trong bài báo với tiêu đề "DTrOCR: Decoder-only Transformer for Optical Character Recognition" của tác giả Masato Fujitake. Mô hình này được thiết kế để cải thiện khả năng nhận diện văn bản bằng cách sử dụng một Transformer chỉ có phần decoder, khác với các phương pháp truyền

thống sử dụng cấu trúc encoder-decoder. Trái ngược với các phương pháp truyền thống thường dựa vào cấu trúc encoder-decoder, trong đó encoder trích xuất đặc điểm từ hình ảnh và decoder chuyển đổi chúng thành văn bản đã nhận diện, DTrOCR đơn giản hóa quy trình bằng cách chỉ sử dụng một mô hình Transformer với phần decoder. Phương pháp này tận dụng một mô hình ngôn ngữ sinh đã được huấn luyện trước trên một tập dữ liệu lớn, nhằm khai thác sức mạnh của ngôn ngữ để nâng cao khả năng nhận diện văn bản. Các thí nghiệm đã chứng minh rằng DTrOCR vượt trội hơn hẳn các phương pháp hiện tại trong việc nhận diện văn bản in, chữ viết tay và văn bản trong cảnh, cả trong tiếng Anh và tiếng Trung. Điều này không chỉ cho thấy tiềm năng của mô hình trong lĩnh vực thị giác máy tính mà còn mở ra những cơ hội mới cho việc áp dụng các công nghệ NLP vào các nhiệm vụ nhận diện văn bản.

2.2.2. Đánh giá

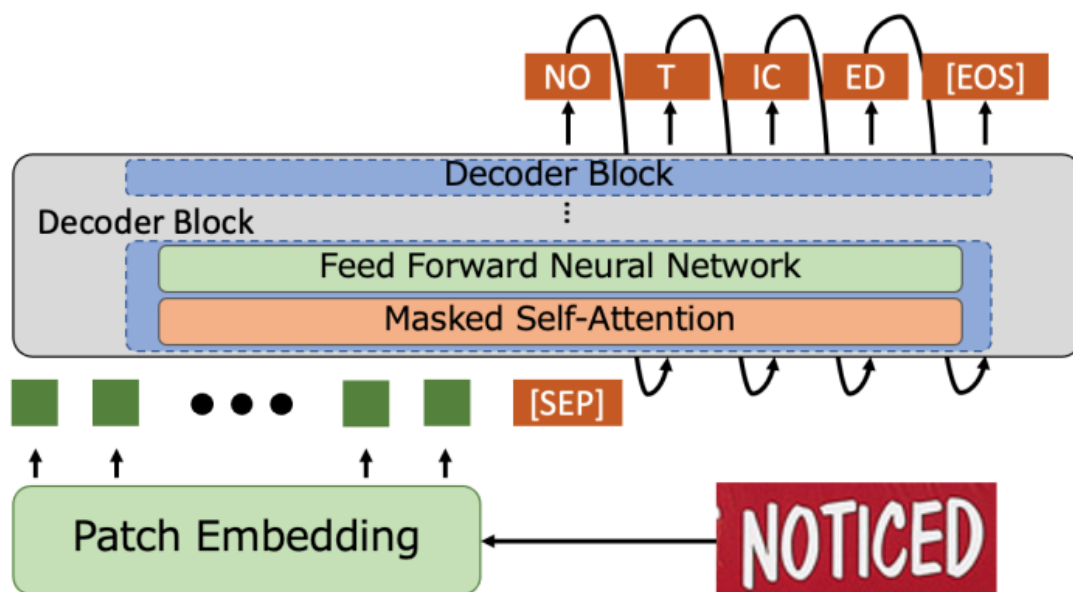
			
Ground truth	CHRISTMAS	PIONEER	Advertisement Space
ABINet	CHRISMMAS	LINEER	Advertisement
PARSeq	CHRUSTMAS	PIONEER	Advertisement
Ours	CHRISTMAS	PIONEER	Advertisement Space

Figure 3. Comparison of recognition results of state-of-the-art methods and proposed method [3, 13]. The result corresponding to an image is shown on each line, with the ground truth at the top. The proposed method is robust to occlusion and irregularly arranged scenes and is accurate even for two lines.

Mô hình DTrOCR đã chứng minh được hiệu suất vượt trội trong lĩnh vực nhận diện văn bản, thể hiện qua các kết quả ấn tượng trên nhiều bộ dữ liệu khác nhau. Với độ chính xác đạt 89.6% cho nhận diện ký tự quang học (OCR) tiếng Trung, DTrOCR không chỉ đứng đầu bảng xếp hạng mà còn thể hiện khả năng áp dụng hiệu quả vào cả văn bản in và viết tay. Đặc biệt, mô hình ghi nhận tỷ lệ lỗi ký tự (CER) chỉ 2.38% trên bộ dữ liệu IAM, khẳng định vị trí số 1 trong danh sách các phương pháp hiện có. Ngoài ra, DTrOCR cũng thể hiện sự cạnh tranh mạnh mẽ trong các bài kiểm tra nhận diện văn bản cảnh, với độ chính xác 99.1% trên bộ dữ liệu CUTE80. Những thành công này không chỉ cho thấy sự linh hoạt của DTrOCR trong việc xử lý đa dạng các định dạng văn bản mà còn mở ra triển vọng lớn cho việc ứng dụng trong các lĩnh vực cần độ chính xác cao và khả năng nhận diện văn bản trong điều kiện thực tế.

Task	Dataset	Model	Metric Name	Metric Value	Global Rank	Result	Benchmark
Optical Character Recognition (OCR)	Benchmarking Chinese Text Recognition: Datasets, Baselines, and an Empirical Study	DTrOCR 105M	Accuracy (%)	89.6	# 1	🔗	Compare
Optical Character Recognition (OCR)	Benchmarking Chinese Text Recognition: Datasets, Baselines, and an Empirical Study	DTrOCR 105M	Accuracy (%)	89.6	# 1	🔗	Compare
Scene Text Recognition	CUTE80	DTrOCR 105M	Accuracy	99.1	# 6	🔗	Compare
Handwritten Text Recognition	IAM	DTrOCR 105M	CER	2.38	# 1	🔗	Compare
Scene Text Recognition	ICDAR2013	DTrOCR 105M	Accuracy	99.4	# 2	🔗	Compare
Scene Text Recognition	ICDAR2015	DTrOCR 105M	Accuracy	93.5	# 1	🔗	Compare
Scene Text Recognition	IIIT5k	DTrOCR 105M	Accuracy	99.6	# 1	🔗	Compare

2.2.3. Cấu trúc



Cấu trúc gồm 2 phần : The patch embedding và Transformer decoder

a) The patch embedding

Vì đầu vào của Transformer là một chuỗi các token, nên module nhúng bản vá (patch embedding) sẽ chuyển đổi các token để hình ảnh có thể đưa vào bộ giải mã (decoder). Hình

ảnh đầu vào được điều chỉnh kích thước thành một kích thước cố định $I \in R^{W \times H \times C}$, trong đó W, H, và C lần lượt là chiều rộng, chiều cao và kênh của hình ảnh. Hình ảnh đầu vào sau đó được chia thành các bản vá có kích thước cố định $p_w \times p_h$, trong đó p_w và p_h là chiều rộng và chiều cao của mỗi bản vá. Các bản vá này được chuyển thành các vector và điều chỉnh để phù hợp với kích thước đầu vào của Transformer. Mã hóa vị trí (position encoding) được thêm vào để duy trì thông tin vị trí của từng bản vá. Sau đó, chuỗi kết quả, chứa cả các bản vá đã được chuyển đổi và thông tin vị trí, được gửi đến bộ giải mã.

b) Transformer decoder

Bộ giải mã thực hiện nhận dạng văn bản dựa trên chuỗi hình ảnh đầu vào. Bộ giải mã ban đầu sử dụng chuỗi hình ảnh đầu vào và tạo ra token đầu tiên bằng cách bắt đầu với một token đặc biệt gọi là [SEP], đánh dấu sự phân chia giữa chuỗi hình ảnh và chuỗi văn bản. Sau đó, mô hình sử dụng chuỗi hình ảnh và token đã dự đoán để tạo văn bản theo cách tự hồi quy (autoregressive) cho đến khi gặp token [EOS]. Kết quả đầu ra của bộ giải mã được chiếu qua một lớp tuyến tính để chuyển từ kích thước của mô hình sang kích thước của từ vựng V. Sau đó, xác suất được tính toán trên từ vựng bằng hàm softmax. Cuối cùng, thuật toán tìm kiếm chùm (beam search) được sử dụng để lấy kết quả cuối cùng. Hàm mất mát cross-entropy được sử dụng trong quá trình này.

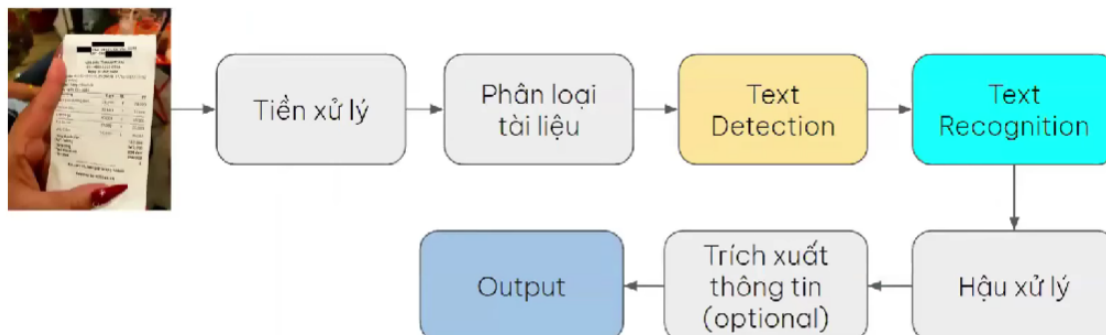
Bộ giải mã sử dụng GPT [43, 44] để nhận dạng văn bản một cách chính xác nhờ vào kiến thức ngôn ngữ. Nó dự đoán từ tiếp theo trong câu bằng cách tối đa hóa entropy. Các mô hình đã được huấn luyện trước (pre-trained models) được cung cấp công khai, giúp giảm nhu cầu tài nguyên tính toán để thu thập kiến thức ngôn ngữ.

Bộ giải mã bao gồm nhiều ngăn xếp, với lớp Transformer [51] tạo thành một khối. Khối này bao gồm một cơ chế tự chú ý (multi-head mask self-attention) và mạng truyền thẳng (feed-forward network). Khác với các cấu trúc encoder-decoder trước đây, phương pháp này chỉ sử dụng bộ giải mã để dự đoán, loại bỏ nhu cầu chú ý chéo (cross-attention) giữa các đặc trưng hình ảnh và văn bản, từ đó đơn giản hóa đáng kể thiết kế.

2.2.4. Dataset

II . Quy Trình Xây Dựng Một Ứng Dụng OCR

Các bước cơ bản để tạo một hệ thống OCR



Xây dựng một ứng dụng OCR bao gồm nhiều bước, từ tiền xử lý hình ảnh đến triển khai mô hình và xử lý kết quả. Hình trên là các bước chung để xây dựng mô hình OCR tuy nhiên trong thực tế có thể lược bỏ một vài phần xử lý hoặc một số phần trong ứng dụng được xử lý khác nhau tùy vào nhiệm vụ của mình.

Bước 1: Thu Thập Dữ Liệu

a) Vấn đề

Đây là giai đoạn đầu tiên và quan trọng nhất khi làm việc trên bất kỳ trường hợp sử dụng ML hoặc DL nào. Hầu hết các giải pháp OCR được các tổ chức tài chính như ngân hàng, công ty bảo hiểm, công ty môi giới, v.v. áp dụng. Vì các tổ chức này có rất nhiều tài liệu khó xử lý thủ công. Vì họ là các tổ chức tài chính nên có các quy tắc và quy định của chính phủ mà các tổ chức tài chính này phải tuân theo.

Vì lý do này, nếu bạn đang làm việc trên bất kỳ **POC (Proof of Concept)** nào cho các công ty tài chính này, có thể có khả năng họ sẽ không chia sẻ toàn bộ dữ liệu để bạn đào tạo các mô hình phát hiện và nhận dạng văn bản của mình. Vì các giải pháp học sâu đều liên quan đến dữ liệu nên bạn có thể nhận được các mô hình có hiệu suất kém. Tất nhiên, điều này liên quan đến việc tuân thủ quy định rằng họ có thể vi phạm quyền riêng tư của người dùng, có thể gây ra tổn thất tài chính và các loại tổn thất khác cho khách hàng nếu họ chia sẻ dữ liệu.

b) Hướng giải quyết

Vấn đề này có giải pháp nào không? Có chứ. Giả sử bạn muốn làm việc trên một số loại Biểu mẫu hoặc Thẻ căn cước để trích xuất văn bản. Đối với biểu mẫu, bạn có thể yêu cầu khách hàng cung cấp các mẫu trống và điền dữ liệu ngẫu nhiên của bạn vào đó (tốn thời

gian nhưng hiệu quả) và đối với thẻ căn cước, bạn có thể tìm thấy rất nhiều mẫu trên internet mà bạn có thể sử dụng để bắt đầu. Ngoài ra, bạn chỉ có thể có một vài mẫu biểu mẫu và thẻ căn cước này và sử dụng các kỹ thuật tăng cường hình ảnh để tạo ra các hình ảnh tương tự mới cho quá trình đào tạo mô hình của bạn để tránh overfit trong quá trình đào tạo.

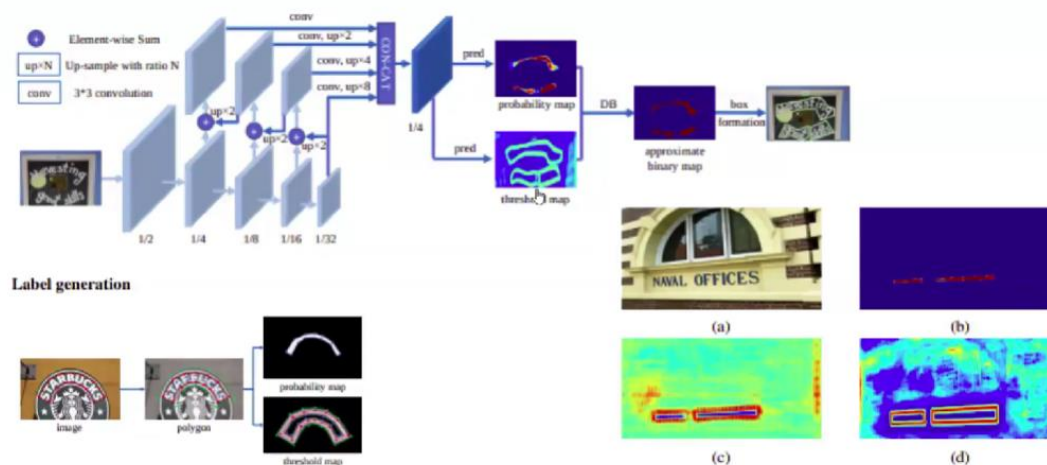
Bước 2: Tiền Xử Lý Hình Ảnh (Image Preprocessing)

Tại sao phải tiền xử lý ảnh? Vì các dữ liệu đầu vào là dữ liệu thực tế, đó là các ảnh, hóa đơn, giấy tờ thường được con người chụp lại. Mà không phải ai cũng có khả năng chụp ảnh đẹp như tôi hoặc ai cũng có iphone 16promax màu vàng sa mạc nên ảnh xuất ra chắc chắn sẽ có ảnh mờ, ảnh tối, ảnh xoay nghiêng xoay dọc, ảnh chụp không sát,.... Mục tiêu của tiền xử lý là tạo ra hình ảnh có chất lượng tốt nhất để đảm bảo hệ thống OCR có thể nhận dạng văn bản chính xác.



Bước 3: Text Detection

Text Detection



Bây giờ bạn đã có dữ liệu và cũng đã tạo các mẫu mới bằng các kỹ thuật tăng cường hình ảnh, việc tiếp theo cần làm là nhận diện được text nó nằm ở đâu sau đó ghi nhãn dữ liệu. Text Detection là quá trình tạo các hộp giới hạn trên các đối tượng mà bạn muốn mô hình phát hiện đối tượng của mình tìm thấy trong hình ảnh. Trong trường hợp này, đối tượng của chúng ta là văn bản nên bạn cần tạo các hộp giới hạn trên vùng văn bản mà bạn muốn mô hình của mình xác định.

Ngoài ra, các hộp giới hạn quá chung chung khi chúng ta nói về chú thích, đối với các loại trường hợp sử dụng khác nhau, các loại chú thích khác nhau được sử dụng. Ví dụ, đối với các trường hợp sử dụng mà bạn muốn có tọa độ chính xác nhất của một đối tượng, bạn không thể sử dụng các hộp giới hạn hình vuông hoặc hình chữ nhật, Ở đó, bạn cần sử dụng các hộp giới hạn bằng các đa thức như hình starbucks trên kia. Quá trình phát hiện văn bản thường dựa trên các thuật toán như YOLO (You Only Look Once), EAST, hoặc các mô hình deep learning khác để xác định các khu vực văn bản.

Bước 4: Text Recognition



Sau khi đã phát hiện được vị trí văn bản, hệ thống cắt các khu vực đó ra và tiến hành nhận dạng nội dung của văn bản trong từng khu vực .

Đây là bước mà các mô hình OCR, như Tesseract, DTrOCR, hoặc AttentionOCR, sẽ chuyển các hình ảnh văn bản thành ký tự.

Kết quả của bước này là chuỗi văn bản đã được nhận dạng từ hình ảnh.

Bước 5: Trích xuất thông tin (optional)

Đây là bước tùy chọn, trong đó hệ thống có thể trích xuất các thông tin cụ thể từ văn bản đã được nhận dạng, chẳng hạn như ngày tháng, số tiền, tên công ty trong hóa đơn.

Trích xuất thông tin thường sử dụng các kỹ thuật xử lý ngôn ngữ tự nhiên (NLP) để phân tích và lấy ra các thông tin quan trọng từ văn bản.

Bước 6: Hậu xử lý

Hậu xử lý OCR đóng vai trò quyết định trong việc đảm bảo độ chính xác và chất lượng của kết quả nhận dạng. Từ việc sửa lỗi chính tả, định dạng lại văn bản, loại bỏ các ký tự không mong muốn đến việc sử dụng mô hình ngôn ngữ để cải thiện ngữ cảnh, tất cả đều giúp văn bản OCR trở nên đáng tin cậy và dễ sử dụng hơn. Các kỹ thuật hậu xử lý có thể kết hợp các phương pháp truyền thống với các mô hình học máy và học sâu để đạt được kết quả tối ưu. Trong thực tế thường sử dụng regex hoặc mô hình ngôn ngữ lớn. Các lỗi hậu xử lý thường phải xử lý:

1. Sửa lỗi chính tả và lỗi nhận dạng ký tự

- Kiểm tra chính tả: Sau khi nhận dạng, văn bản thường chứa các lỗi chính tả do hệ thống OCR nhận dạng sai. Sử dụng bộ kiểm tra chính tả hoặc từ điển ngôn ngữ để phát hiện và sửa lỗi giúp cải thiện độ chính xác.
- Sửa lỗi dựa trên ngữ cảnh: Sử dụng mô hình ngôn ngữ (Language Model) như BERT hoặc GPT để xác định từ nào có khả năng xuất hiện nhất trong ngữ cảnh đó, thay thế các từ nhận dạng sai bằng từ có ý nghĩa phù hợp trong ngữ cảnh.
- Sửa lỗi ký tự thường gặp: Các lỗi phổ biến như nhầm lẫn giữa chữ "O" và số "0", chữ "l" và số "1", hoặc chữ "S" và số "5" thường xảy ra. Một hệ thống hậu xử lý có thể chứa các quy tắc hoặc mô hình để phát hiện và sửa các lỗi này.

2. Sắp xếp lại cấu trúc văn bản

- Định dạng văn bản: Khi văn bản được nhận dạng từ các tài liệu có cấu trúc như bảng biểu, hóa đơn, hoặc biểu mẫu, việc sắp xếp lại cấu trúc là rất quan trọng. Hệ thống có thể tái tạo lại bảng, định dạng đúng các cột và hàng, để giữ nguyên bố cục của tài liệu gốc.
- Ghép nối các dòng văn bản bị chia cắt: Trong một số trường hợp, OCR có thể chia cắt các dòng văn bản liên tục thành các đoạn nhỏ. Hệ thống hậu xử lý có thể ghép nối lại các đoạn này để tạo thành câu hoàn chỉnh, đảm bảo tính liên mạch cho văn bản.

3. Trích xuất và chuẩn hóa dữ liệu

- Trích xuất thông tin quan trọng: Trong các tài liệu có cấu trúc như hóa đơn, hợp đồng, chứng từ tài chính, hậu xử lý bao gồm bước trích xuất các thông tin cụ thể như

ngày tháng, số tiền, tên công ty, địa chỉ, và chuẩn hóa chúng vào các trường dữ liệu cố định.

- Chuẩn hóa định dạng: Các thông tin như ngày tháng, số điện thoại, mã số có thể được chuẩn hóa theo định dạng cụ thể để phù hợp với yêu cầu của ứng dụng đích. Ví dụ, ngày tháng có thể chuyển thành định dạng chuẩn như “YYYY-MM-DD” để dễ dàng truy vấn.

4. Loại bỏ các ký tự không mong muốn

- Xử lý ký tự không hợp lệ: Văn bản OCR có thể chứa các ký tự lạ hoặc không mong muốn do nhầm lẫn trong nhận dạng, đặc biệt là từ các hình ảnh nhiễu. Hệ thống hậu xử lý có thể sử dụng quy tắc hoặc mô hình để phát hiện và loại bỏ các ký tự này.
- Loại các từ vô nghĩa: Đôi khi, OCR nhận dạng các ký tự không có ý nghĩa như dấu chấm, dấu phẩy, ký hiệu đặc biệt. Các từ hoặc ký tự này có thể được loại bỏ để văn bản trở nên sạch sẽ và dễ hiểu hơn.

III. Fine-Tune Mô Hình OCR

Trước khi đi vào khái niệm fine-tune mô hình, chúng ta đến với khái niệm pre-train mô hình.

Pre-training là bước đầu tiên, trong đó mô hình OCR được huấn luyện trên các tập dữ liệu lớn, thường là dữ liệu tổng hợp, để học các đặc trưng cơ bản của văn bản và ngôn ngữ. Pre-training (tiền huấn luyện) cung cấp cho mô hình một nền tảng kiến thức chung về ngôn ngữ. Ví dụ: Mô hình GPT-3 của OpenAI được pre-trained trên một lượng dữ liệu văn bản khổng lồ. Tuy nhiên để nó có thể thực hiện tốt từng tác vụ cụ thể chúng ta phải fine-tune lại nó.

Thế Fine-tune mô hình là gì và fine tune trong một ứng dụng OCR là như thế nào?

Fine-tuning là bước tiếp theo sau pre-training, trong đó mô hình được huấn luyện thêm trên các tập dữ liệu thực hoặc các tập dữ liệu chuyên biệt để cải thiện độ chính xác trên một tác vụ hoặc ngữ cảnh cụ thể. Cho dễ hiểu thì pre-train nó như kiểu giáo dục phổ thông cấp 1, cấp 2, dạy cho model những cái gọi là kiến thức tổng quát (general). Còn fine-tune thì nó như là đại học chuyên ngành, dạy cho model chuyên sâu về 1 mảng gì đó (domain specific). Ví dụ như mình lên mạng lấy 1 mô hình pre-train BERT về thì nó chỉ có thể hiểu được những nội dung phổ thông, nếu đưa 1 văn bản y khoa vào thì nó sẽ hoàn toàn ko hiểu gì hết. Nếu bạn fine-tune model đó bằng 1 tập data về y học thì sau đó nó sẽ hiểu được và thực hiện được cái bài toán về NLP như là NER,...

Trong ứng dụng OCR, chúng ta có thể lấy một mô hình OCR đã tốt ở hiện tại như 5 mô hình ở đầu đã giới thiệu, đóng băng một số tham số của một số layer đầu và sử dụng bộ dữ liệu của chúng ta để điều chỉnh cho tham số của layer tiếp theo. Tinh chỉnh sao cho phù hợp với nhiệm vụ cần làm và bộ data hiện có.

IV. OCR tiếng việt

Vintern-1B (Vietnamese-InternVL2-1B) là mô hình ngôn ngữ lớn đa phương thức dành cho các tác vụ ngôn ngữ tiếng Việt, được phát triển bởi nhóm tác giả gồm Khang T. Doan, Bao G. Huynh, Dung T. Hoang, Thuc D. Pham, Nhat H. Pham, Quan T.M. Nguyen, và Bang Q. Vo. Ra mắt năm 2024, Vintern-1B có 1 tỷ tham số và kết hợp mô hình ngôn ngữ Qwen2-0.5B-Instruct với mô hình thị giác InternViT-300M-448px, tối ưu hóa cho các tác vụ nhận dạng ký tự quang học, trích xuất thông tin tài liệu và trả lời câu hỏi từ hình ảnh trong ngữ cảnh tiếng Việt. Link bài viết: <https://arxiv.org/pdf/2408.12480>

4.1. Giới thiệu

Thách thức trong việc phát triển MLLM cho tiếng Việt :

Mặc dù đã có nhiều tiến bộ về MLLM trên toàn cầu, việc phát triển các mô hình MLLM tiếng Việt vẫn bị hạn chế do thiếu hụt các tập dữ liệu đa phương thức chất lượng cao. Các mô hình như Qwen-VL, MiniCPM-Llama3-V-2.5, MoE-LLaVA-Qwen, và InternVL đã thể hiện tiềm năng trong việc xử lý văn bản tiếng Việt, nhưng vẫn còn thiếu nguồn tài nguyên và các mô hình hỗ trợ đầy đủ các tác vụ đa phương thức tiếng Việt.

Những sáng kiến tại Việt Nam :

Ở Việt Nam, các mô hình và tập dữ liệu như V-Vistral, Vista, và LaVy đã đóng góp quan trọng cho nghiên cứu MLLM. Tuy nhiên, các mô hình này hiện vẫn chưa hỗ trợ tốt các tác vụ như OCR và xử lý tài liệu tiếng Việt, vốn rất quan trọng trong nhiều ứng dụng thực tiễn.

Đóng góp của Vintern-1B :

Để khắc phục những hạn chế này, nhóm nghiên cứu đã tạo ra các tập dữ liệu mới, tập trung vào xử lý văn bản và tài liệu, cũng như các tập dữ liệu hình ảnh có yếu tố địa phương tiếng Việt. Dựa trên các tập dữ liệu này, nhóm đã tinh chỉnh mô hình InternVL2-1B, đạt được kết quả ấn tượng trong xử lý văn bản.

Nhóm cũng cung cấp mô tả chi tiết về kiến trúc của Vintern-1B, quy trình tạo dữ liệu, cách huấn luyện mô hình và phương pháp đánh giá hiệu quả của MLLM tiếng Việt. Mục tiêu của họ là thúc đẩy nghiên cứu và cung cấp công cụ cần thiết để khám phá và đổi mới tại giao điểm giữa ngôn ngữ và hình ảnh trong ngữ cảnh tiếng Việt.

4.2. Cấu trúc

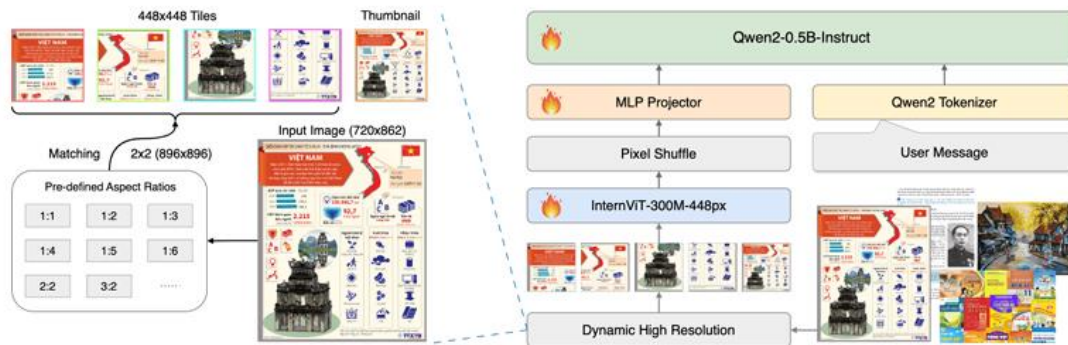


Figure 1. Overall Architecture. Vintern-1B is built upon the ViT-MLP-LLM framework, following the structure of well-known MLLMs ([2] [16] [17] [6]). It inherits from InternVL 1.5 [2], integrating a pre-trained InternViT-300M-448px [2] with Qwen2-0.5B-Instruct [1] via an MLP projector. The input image is processed by the Dynamic High Resolution module, which splits it into smaller 448x448 pixel images along with a thumbnail. These images are then passed through InternViT-300M-448px to extract visual features. A Pixel Shuffle step is also applied before feeding the data into the MLP projector to align it with the embeddings of the large language model Qwen2-0.5B-Instruct which takes the aligned visual tokens and the related question as inputs, and generates the corresponding answer.

Hình ảnh mô tả kiến trúc tổng thể của **Vintern-1B**. Mô hình này được xây dựng trên cấu trúc **ViT-MLP-LLM**, theo cấu trúc của các mô hình ngôn ngữ lớn đa phương thức (MLLMs) nổi tiếng. Vintern-1B kế thừa từ **InternVL 1.5**, kết hợp mô hình **InternViT-300M-448px** đã được huấn luyện trước với **Qwen2-0.5B-Instruct** thông qua một bộ chiếu **MLP**. Chúng ta hãy đi tìm hiểu 1 ảnh đi qua cấu trúc này sẽ được training như thế nào?

a) Đầu tiên ảnh đi qua một module Dynamic High Resolution

Đầu ra là ảnh với nhiều phần nhỏ với kích thước 448*448px. Dynamic High Resolution là một module trong kiến trúc của Vintern-1B, được thiết kế để xử lý hình ảnh đầu vào với độ phân giải cao một cách linh hoạt. Vì ảnh đầu vào có rất nhiều kích thước khác nhau, mục đích chính của module này là đầu tiên ảnh sẽ được đưa về kích thước 896*896 chia nhỏ hình ảnh lớn thành các phần tử nhỏ hơn với kích thước cố định (trong trường hợp này là a ảnh 448x448 pixel), giúp mô hình dễ dàng xử lý thông tin hình ảnh mà không bị quá tải tài nguyên, đồng thời đảm bảo khả năng giữ lại chi tiết hình ảnh ở độ phân giải cao.

Cấu trúc và Cách Hoạt Động

- **Nhận Hình Ảnh Đầu Vào:** Dynamic High Resolution nhận hình ảnh đầu vào có kích thước lớn (ví dụ: 720x862).

- **Tinh chỉnh tỷ Lệ Khung Hình:** Ở bước này module sẽ biến đổi ảnh bằng các phương pháp biến đổi ảnh về 896*896 pixel
- **Chia Nhỏ Hình Ảnh:** Module này chia hình ảnh thành 4 ô nhỏ hơn với kích thước cố định 448x448 pixel. Điều này cho phép mô hình xử lý các vùng khác nhau của hình ảnh một cách chi tiết hơn mà không bị giảm chất lượng do co lại hoặc bị giới hạn kích thước. Tuy nhiên trong thực tế, quá trình test được chia thành 12 mảnh.
- **Xuất Ảnh Thu Nhỏ (Thumbnail):** Sau khi chia nhỏ hình ảnh, module tạo ra một ảnh thu nhỏ của toàn bộ hình ảnh đầu vào để làm tham chiếu, giúp mô hình có cái nhìn tổng quan về bối cảnh của toàn bộ hình ảnh.
- **Gửi Dữ Liệu Đến Module Thị Giác:** Các ô hình ảnh nhỏ sau đó được gửi đến mô hình InternViT-300M-448px để trích xuất các đặc trưng thị giác. Các đặc trưng này sẽ được sử dụng làm đầu vào cho phần xử lý ngôn ngữ và trả lời câu hỏi của mô hình.

b) Tiếp theo 5 ảnh sẽ đi qua một lớp mã hóa hình ảnh (Vision Encoder): Sử dụng mô hình InternViT-300M-448px.

Tức là đã có 5 ảnh rồi giờ làm sao máy tính hiểu được 5 ảnh và mối quan hệ giữa 5 ảnh này. Chúng ta phải dùng bộ mã hóa này để làm điều đó, chúng ta đi bóc từng từ để phân tích:

- **InternViT:** Đây là phiên bản của ViT được tối ưu hóa cho việc huấn luyện trên dữ liệu lớn với quy mô lớn và hiệu suất tốt hơn. "Intern" có thể ám chỉ đến sự phát triển và tinh chỉnh mô hình trong các dự án nghiên cứu hoặc thương mại.
- **300M:** Số lượng tham số của mô hình là khoảng 300 triệu. Điều này cho thấy mô hình có độ phức tạp và khả năng học cao, giúp nó có thể xử lý và nhận diện nhiều loại đặc trưng trong dữ liệu hình ảnh.
- **448px:** Kích thước đầu vào của hình ảnh là 448 x 448 pixel. Kích thước này thường được sử dụng để đảm bảo rằng mô hình có đủ thông tin để phân tích mà không bị giảm chất lượng quá nhiều.

Ok thế thì cái quan trọng nhất của phần này là ViT(Vision Transformer), thế vision transformer là gì? Để nói ngắn gọn và đơn giản thì trong việc chuyển ảnh thành số như thế nào cho máy tính hiểu thì trước giờ chỉ có CNN, transformer đang làm rất tốt ở NLP nhưng có vẻ ở CV còn rất hạn chế. Bài báo <https://arxiv.org/pdf/2010.11929v2> đã đề cập đến việc sử dụng transformer vào CV(Computer Vision) đạt được hiệu quả tốt hơn CNN. Thế thì ViT hoạt động như thế nào?



Source – <https://github.com/lucidrains/vit-pytorch/blob/main/vit.gif>

1. Chia Ảnh Thành Các Phần Nhỏ và Biến Đổi Thành Chuỗi

- **Chia Ảnh Thành Các Gói (Patches):** Hình ảnh đầu vào (có kích thước 2D) được chia thành các gói (patch) có kích thước cố định, tương tự như cách chia văn bản thành các token trong NLP.
- **Trích Xuất Đặc Trưng Bằng CNN:** Trong thực tế, ViT thường không trực tiếp xử lý ảnh mà sẽ sử dụng một mạng CNN để trích xuất đặc trưng từ ảnh. Sau đó, bản đồ đặc trưng cuối cùng từ CNN sẽ được chia thành các gói và đưa vào ViT.
- **Patch Embedding:** Sau khi chia ảnh thành các gói, mỗi gói sẽ được chuyển thành vector kích thước D thông qua một phép biến đổi tuyến tính gọi là patch embedding để phù hợp với kích thước của Transformer.

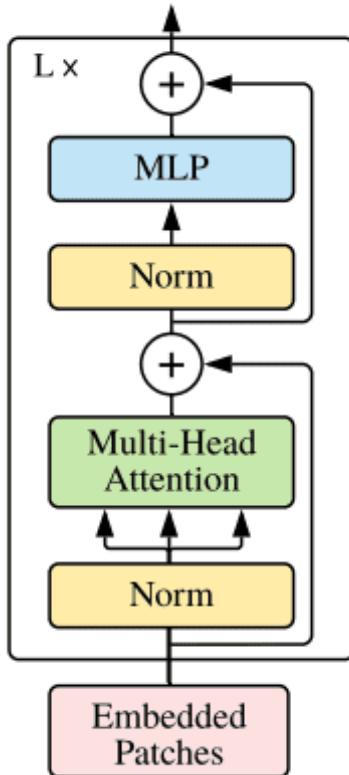
2. Nhúng Vị Trí (Position Embedding)

- Để giữ lại thông tin vị trí của các gói, **position embedding** (vector vị trí) được thêm vào mỗi patch embedding. Vector vị trí này là 1D (thay vì 2D) để giảm thiểu kích thước lưu trữ và chi phí tính toán.

- Mặc dù có ý kiến cho rằng việc học thứ tự của các gói trong ảnh là dễ dàng, việc giữ lại thông tin vị trí vẫn rất quan trọng vì giúp mô hình hiểu rõ hơn thứ tự và ngữ nghĩa của các đối tượng trong ảnh.

3. Transformer Encoder

Transformer Encoder



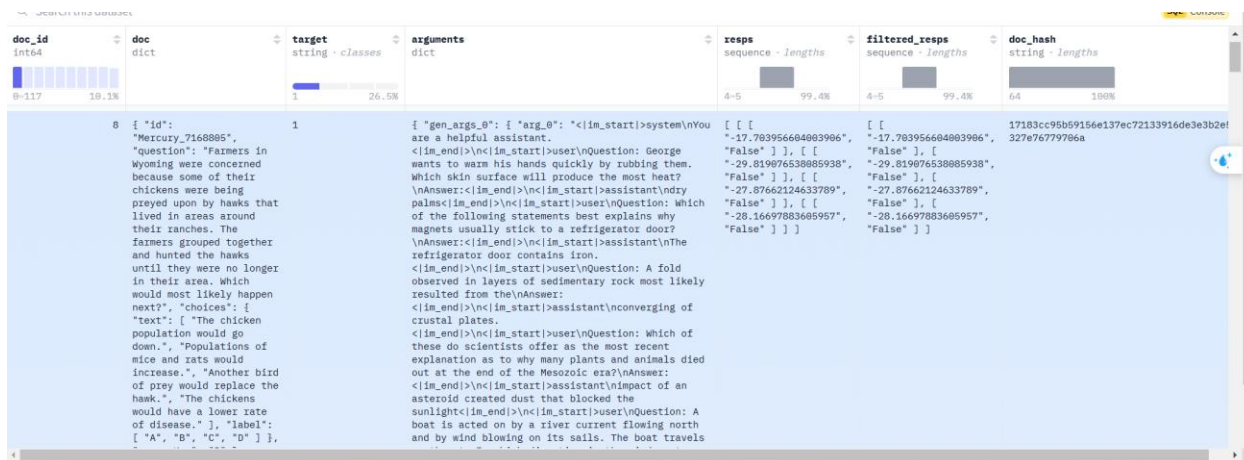
- **Encoder Của Transformer** nhận vào tổng hợp các thông tin từ patch embedding và position embedding, bao gồm các lớp **Multihead Attention**, **MLP** và **Layer Norm**.
- **Inductive Bias**: ViT có ít inductive bias hơn so với CNN, điều này có nghĩa là nó không tích hợp sẵn những quy luật liên quan đến cấu trúc không gian trong ảnh như CNN. Trong CNN, mỗi lớp đều có các đặc tính như kiến trúc hàng xóm 2 chiều và dịch chuyển tương đương, trong khi ViT chỉ có lớp MLP là mang tính địa phương hóa và dịch chuyển tương đương.

c) Tiếp theo khi đã được vector chữ mọi thông tin cần có của ảnh , vector sẽ đi qua Pixel Shuffle và MLP Projector

- Pixel Shuffle được sử dụng để sắp xếp lại các đặc trưng từ các ô nhỏ trước khi chúng được đưa vào MLP Projector.
- MLP Projector có nhiệm vụ chuyển đổi các đặc trưng hình ảnh để căn chỉnh chúng với các embedding ngôn ngữ của mô hình ngôn ngữ lớn (LLM) Qwen2-0.5B-Instruct.

d) Tiếp theo ta đến với phần mô hình ngôn ngữ lớn (LLM) Qwen2-0.5B-Instruct.

Thế tại sao lại không lấy mô hình ngôn ngữ khác lại lấy Qwen, vì **Mô hình Qwen2-0.5B-Instruct** là một phần trong dòng mô hình Qwen, được thiết kế như một mô hình ngôn ngữ lớn (LLM) dành cho các tác vụ đa phương thức (hỗ trợ cả văn bản và hình ảnh) với khả năng xử lý yêu cầu và phản hồi theo hướng dẫn (instruction-following). Tập data trên huggingface



doc_id	doc	target	arguments	resps	filtered_resps	doc_hash
int64 0-117 10.1%	dict	string · classes 1 26.5%	dict	sequence · lengths 4-5 99.4%	sequence · lengths 4-5 99.4%	string · lengths 64 100%
8	{ "id": "Mercury_7168895", "question": "Farmers in Wyoming were concerned because some of their chickens were being preyed upon by hawks that lived in areas around their ranches. The farmers grouped together and hunted the hawks until they were no longer in their area. Which would most likely happen next?", "choices": ["text": ["The chicken population would go down.", "Populations of mice and rats would increase.", "Another bird of prey would replace the hawk.", "The chickens would have a lower rate of disease."], "label": ["A", "B", "C", "D"] },	1	{ "gen_args_0": { "arg_0": "< im_start >system\nYou are a helpful assistant. < im_end >\n< im_start >user\nQuestion: George wants to warm his hands quickly by rubbing them. Which skin surface will produce the most heat? \nAnswer:< im_end >\n< im_start >assistant\nDry palms< im_end >\n< im_start >user\nQuestion: which of the following statements best explains why magnets usually stick to a refrigerator door? \nAnswer:< im_end >\n< im_start >assistant\nThe refrigerator door contains iron. < im_end >\n< im_start >user\nQuestion: A fold observed in layers of sedimentary rock most likely resulted from the\nAnswer: < im_end >\n< im_start >assistant\nnonconverging of crustal plates. < im_end >\n< im_start >user\nQuestion: which of these do scientists offer as the most recent explanation as to why many plants and animals died out at the end of the Mesozoic era?\nAnswer: < im_end >\n< im_start >assistant\nImpact of an asteroid created dust that blocked the sunlight< im_end >\n< im_start >user\nQuestion: A boat is acted on by a river current flowing north and by wind blowing on its sails. The boat travels	[[["-17.783956604083906", "False"]], [["-29.819876538085938", "False"]], [["-27.87662124633789", "False"]], [["-28.16697883605957", "False"]]]	[[["-17.783956604083906", "False"]], [["-29.819876538085938", "False"]], [["-27.87662124633789", "False"]], [["-28.16697883605957", "False"]]]	17183cc95b59156e137ec72133916de3e3b2ef327e76779706a

Sau khi máy tính đã hiểu được hình ảnh, ngữ cảnh thì sẽ đưa qua mô hình này nhằm có thể giải quyết được các nhiệm vụ như: đề cập nội dung ảnh, hỏi đáp, nhận dạng chữ trong biển báo, nhận dạng chữ viết tay.v.v

4.3. Dataset

Phần cấu trúc được xem như 1 phần nằm giữa , để phần này chuẩn thì phần đầu vào và đầu ra của nó cũng phải chuẩn. Vậy dataset của mô hình này sẽ như thế nào? Vì nó mà mô hình đa phương thức nên dataset cho mỗi phương thức cũng là khác nhau. Ta sẽ đi qua các dataset cho từng nhiệm vụ khác nhau, cách thu thập hình ảnh từ nhiều nguồn khác nhau, bao gồm thu thập dữ liệu trên web từ các chủ đề đa dạng liên quan đến văn hóa và môi trường của Việt Nam. Mục tiêu là tạo ra một tập dữ liệu phong phú với nội dung phản ánh các đặc trưng riêng biệt của xã hội Việt Nam. Các loại hình ảnh được thu thập bao gồm:

- **Cảnh đường phố:** Hình ảnh về cuộc sống đường phố tại Việt Nam.
- **Sự kiện truyền thống:** Hình ảnh về các lễ hội và sự kiện văn hóa truyền thống.
- **Tài liệu giáo dục:** Hình ảnh từ các tài liệu giáo dục.
- **Vật dụng hàng ngày:** Các hình ảnh của những đồ vật quen thuộc trong cuộc sống hàng ngày.

Việc mở rộng dữ liệu này rất quan trọng để huấn luyện mô hình có thể nhận diện và xử lý thông tin hình ảnh liên quan đến văn hóa Việt Nam, giúp mô hình hiểu biết sâu hơn về các yếu tố đặc trưng của văn hóa và môi trường Việt Nam.

Table 1. Datasets used in the fine-tuning stage.

Task	Datasets
General QA	Vista [12], Viet-OpenViVQA-gemini-VQA, Viet-Localization-VQA
OCR	Viet-OCR-VQA, Viet-ViTextVQA-gemini-VQA, Viet-Vintext-gemini-VQA
Document	Viet-Doc-VQA, Viet-Doc-VQA-II, Viet-Geometry-VQA, Viet-ComputerScience-VQA, Viet-Sketches-VQA
Handwriting	Viet-Handwriting-VQA, Viet-Vintext-gemini-VQA
Extraction	Viet-Receipt-VQA, Viet-Menu-Gemini-VQA

Để xem được chi tiết trực quan từng dataset cho từng nhiệm vụ : tham khảo mục 3.2 bài báo <https://arxiv.org/pdf/2408.12480>

4.4. Chi tiết quá trình thực hiện

1. Phương pháp phân giải cao động:

- **Chia nhỏ hình ảnh:** Hình ảnh được chia thành các ô có kích thước 448×448 pixel, và số lượng ô có thể thay đổi tùy theo tỷ lệ khung hình và độ phân giải của hình ảnh đầu vào trong quá trình huấn luyện. Số lượng ô có thể lên đến 12 ô.
- **Giai đoạn kiểm tra:** Trong giai đoạn kiểm tra, mô hình có thể xử lý tối đa 12 ô, giúp nó tiếp nhận thông tin từ các hình ảnh có độ phân giải cao hoặc kích thước lớn.

2. Tinh chỉnh tham số đầy đủ:

- **Vision Encoder và MLP Projector:** Để tối ưu hóa hiệu suất trong các tác vụ xử lý hình ảnh, bộ mã hóa hình ảnh (Vision Encoder) và bộ chiếu MLP (MLP Projector) được tinh chỉnh toàn bộ tham số. Điều này có nghĩa là mọi tham số trong các phần này đều được điều chỉnh để đạt hiệu suất tốt nhất.

- **LoRA cho Mô hình Ngôn ngữ Lớn (LLM):** Đối với phần ngôn ngữ, kỹ thuật LoRA (Low-Rank Adaptation) được sử dụng để giảm số lượng tham số cần tinh chỉnh. LoRA chỉ điều chỉnh một phần nhỏ các tham số của mô hình, giúp tiết kiệm tài nguyên tính toán mà vẫn giữ được hiệu suất cao.

3. Tận dụng mô hình InternVL2-1B:

Huấn luyện trước: Mô hình InternVL2-1B, đã được huấn luyện trước trên các tác vụ hướng dẫn hình ảnh (visual instruction), được sử dụng để làm nền tảng cho Vintern-1B. Điều này giúp tận dụng tối đa khả năng của mô hình ngôn ngữ lớn đa phương thức trong các tác vụ phức tạp liên quan đến hình ảnh và ngôn ngữ.

4. Dự đoán từ tiếp theo:

Hàm mất mát cross-entropy: Mô hình sử dụng hàm mất mát cross-entropy cho dự đoán từ tiếp theo, giống như cách tiếp cận trong giai đoạn tiền huấn luyện. Điều này giúp mô hình có thể tối ưu hóa dự đoán và cải thiện hiệu suất trong các tác vụ yêu cầu tạo văn bản liên tục.

5. Các phiên bản của Vintern-1B:

- **Vintern-1B-v1:** Phiên bản đầu tiên, được tinh chỉnh từ InternVL2-1B, sử dụng các bộ dữ liệu Viet-Doc-VQA, Viet-Doc-VQA-II OCR và Viet-OCR-VQA trong 1 epoch. Các bộ dữ liệu này chủ yếu tập trung vào các tác vụ hỏi-đáp và nhận dạng ký tự quang học (OCR) trong ngữ cảnh tiếng Việt.
- **Vintern-1B-v2:** Phiên bản thứ hai tiếp tục tinh chỉnh từ phiên bản v1, sử dụng các bộ dữ liệu giống như v1, cùng với tất cả các bộ dữ liệu khác được liệt kê trong phần 3.2, trong 1 epoch nữa. Điều này giúp mô hình học thêm các kiến thức mới từ nhiều nguồn dữ liệu khác nhau, cải thiện độ chính xác và khả năng xử lý đa dạng.

6. Thông số huấn luyện:

- **Phần cứng:** Cả hai phiên bản được huấn luyện trên 4 GPU Nvidia RTX-3090, một loại GPU mạnh mẽ cho các tác vụ học sâu.
- **Batch size:** Sử dụng kích thước batch toàn cục là 128, giúp mô hình xử lý dữ liệu hiệu quả mà không gặp vấn đề về bộ nhớ.
- **Tốc độ học:** Tốc độ học được thiết lập ở mức $4e-5$, giúp mô hình học chậm rãi và ổn định, tránh quá khớp hoặc học quá nhanh.

- **Độ dài ngữ cảnh:** Sử dụng độ dài ngữ cảnh là 4096, cho phép mô hình có thể tiếp nhận và xử lý thông tin dài hạn trong một chuỗi văn bản dài. Điều này đặc biệt hữu ích cho các tác vụ yêu cầu hiểu biết sâu về ngữ cảnh và chuỗi sự kiện.

7. Định dạng phản hồi:

Định dạng của LLaVA 1.5: Trong quá trình huấn luyện và đánh giá, mô hình tuân theo định dạng phản hồi được định nghĩa trong LLaVA 1.5, giúp chuẩn hóa cách trả lời và đảm bảo tính nhất quán trong các phản hồi của mô hình.

4.5. Đánh giá

Chỉ số đánh giá của chúng tôi được lấy cảm hứng từ Lavy [13], sử dụng phương pháp MLLM-as-a-Judge để kiểm tra độ chính xác của các phản hồi được tạo ra cho các cặp câu hỏi-trả lời. Cụ thể, chúng tôi sử dụng GPT-4o để đánh giá chất lượng các câu trả lời trên hai bộ dữ liệu: OpenViVQA [3] và ViTextVQA [18]. Chúng tôi đánh giá hiệu suất VQA (Visual Question Answering) không cần huấn luyện (zero-shot) của các mô hình trên bộ dữ liệu OpenViVQA-dev [3], bao gồm 3.505 mẫu. Bộ dữ liệu này là một thách thức đối với các mô hình, yêu cầu chúng phải hiểu mối quan hệ giữa hình ảnh tiếng Việt và ngôn ngữ tự nhiên.

Ngoài ra, chúng tôi sử dụng bộ dữ liệu ViTextVQA-dev để đánh giá khả năng của các mô hình VQA trong xử lý các tác vụ liên quan đến OCR, đặc biệt là trong ngữ cảnh tiếng Việt. Bộ dữ liệu này chủ yếu tập trung vào việc trích xuất và diễn giải thông tin từ văn bản và văn bản cảnh xuất hiện trong hình ảnh.

Quá trình đánh giá bao gồm việc nhập các hình ảnh, câu hỏi, nhãn và câu trả lời dự đoán vào GPT-4o, sau đó GPT-4o sẽ gán một điểm chất lượng từ 0 đến 10 dựa trên độ chính xác của các câu trả lời. Kết quả tương ứng được tóm tắt trong bảng sau.

Table 7: Zero-shot VQA on OpenViVQA-dev and ViTextVQA-dev. Models' output accuracy are evaluated by GPT-4o

Model	GPT-4o-score	
	OpenVivQA-dev	ViTextVQA-dev
Vintern-1B-v1	7.1/10	7.6/10
Vintern-1B-v2	7.7/10	7.7/10



Thêm vào đó, chúng tôi đánh giá Vintern-1B-v2 bằng cách sử dụng bài kiểm tra riêng VLSP 2023, báo cáo điểm F1 và BLEU như được thể hiện trong Bảng sau. Lưu ý rằng điểm F1 và BLEU có thể không cao, do mô hình Vintern-1B có xu hướng tạo ra các câu trả lời dài hơn so với các câu trả lời tham chiếu, điều này ảnh hưởng tiêu cực đến các chỉ số đánh giá.

Table 8: VLSP 2023 Benchmarks

Team Name	F1	avg. BLEU
ICNLP	3.6384 (1)	0.4663 (4)
Vintern-1B-v2	3.4616 (2)	0.4422 (7)
linh	3.4293 (3)	0.4609 (5)
DS@ViVRC	3.4121 (4)	0.4457 (6)
DS@UIT Dynasty	3.3172 (5)	0.4742 (3)
NTQ Solution	3.2926 (6)	0.4876 (1)
I, Me & Myself	3.2396 (7)	0.4795 (2)
AVQA_AIO	2.9018 (8)	0.4380 (8)
Vintern-1B-v1	2.7256 (9)	0.3082 (10)
NguyenLe	2.7053 (10)	0.3997 (9)
nowj2	1.6808 (11)	0.2412 (11)

Thử nghiệm một vài trường hợp:

Image	Question	Model Answer
	<p>Mô tả hình ảnh một cách chi tiết.</p> <p>(Detailed description of the image.)</p>	<p>Bức ảnh là một thông báo tuyển dụng của công ty Mobifone. Bức ảnh được chia thành 3 phần chính:</p> <ul style="list-style-type: none"> - Phần đầu tiên là tiêu đề "TUYỂN DỤNG" được in chữ màu đỏ đậm, cỡ chữ lớn, nằm ở vị trí trung tâm phía trên cùng của bức ảnh. - Phần thứ hai là phần giới thiệu về vị trí và yêu cầu của ứng viên. Phần này được trình bày theo dạng liệt kê các mục: - Vị trí: Nhân viên kinh doanh - Địa điểm: Thành phố và các huyện Yên Sơn, Sơn Dương, Hàm Yên, Chiêm Hóa - Yêu cầu: Trình độ: Tốt nghiệp Trung cấp trở lên, Độ tuổi: Không quá 35 tuổi - Phần thứ ba là phần giới thiệu về chế độ đãi ngộ. Phần này được trình bày theo dạng liệt kê các mục: - Chế độ đãi ngộ: - Mức lương: 7.000.000đ - 17.000.000đ - Thưởng thi đua hàng tháng, quý: Thưởng theo doanh số - Chế độ bảo hiểm theo quy định - Được trau dồi các kỹ năng: Giao tiếp, thuyết phục, chăm sóc khách hàng... <p>Phần cuối cùng là phần "NỘP HỒ SƠ" với thông tin liên lạc của người nộp hồ sơ. Phần này được trình bày theo dạng liệt kê các mục: - Người nhận: Trần Thị Thủy - Địa chỉ: 547 đường Quang Trung, Tổ 25, Phường Phan Thiết, TP Tuyên Quang - Điện thoại liên hệ: 0901.452.888 - Email nhận hồ sơ: thuy.tranthi.kd@mobifone.vn - Thời hạn nộp hồ sơ: đến hết ngày 28/5/2021</p> <p>(The picture is a job announcement from Mobifone. The layout is divided into three main sections:</p> <ul style="list-style-type: none"> - The first section is the title 'RECRUITMENT,' printed in bold red letters, large font size, centered at the top of the image. - The second section introduces the position and requirements for the candidates. This section is presented in a bullet point format: - Position: Sales Staff - Location: The city and districts of Yên Sơn, Sơn Dương, Hàm Yên, Chiêm Hóa - Requirements: Education: At least a vocational diploma, Age: Not over 35 years old. - The third section introduces the benefits package. This section is also presented in a bullet point format: - Benefits: - Salary: 7,000,000 VND - 17,000,000 VND - Monthly and quarterly performance bonuses: Sales-based bonuses - Insurance benefits as per regulations - Opportunities to develop skills such as communication, persuasion, customer service, etc. The final section is the 'SUBMIT APPLICATION' part with the contact information for the applicant. This section is presented in a bullet point format: - Recipient: Trần Thị Thủy - Address: 547 Quang Trung Street Group 25 Phan Thiết Ward Tuyên Quang City

Image	Question	Model Answer
	<p>Mô tả hình ảnh một cách chi tiết.</p> <p>{Detailed description of the image.}</p>	<p>Bức ảnh là một trang giấy kẻ ô vuông với dòng chữ viết tay bằng mực đen. Bỏ cục đơn giản, chỉ có nội dung văn bản được viết theo chiều dọc từ trên xuống dưới. Các dòng chữ được sắp xếp theo thứ tự từ trên xuống dưới như sau:</p> <ul style="list-style-type: none"> - Nghe thấy đọc thơ - Em nghe thấy đọc bao ngày - Tiếng thơ đỏ nắng xanh cây quanh nhà - Mái chèo nghe vọng sông xa - Em êm nghe tiếng cửa bà năm xưa - Nghe trăng thổ động tàu dừa - Rào rào nghe chuyển cơn mưa giữa trời - Thêm yêu tiếng hát nụ cười - Nghe thơ em thấy đất trời đẹp ra. - Trần Đăng Khoa <p>{The image is of a grid-lined page with handwritten text in black ink. The layout is simple, containing only the text, which is written vertically from top to bottom. The lines of text are arranged in the following order:</p> <ul style="list-style-type: none"> - Nghe thấy đọc thơ - Em nghe thấy đọc bao ngày - Tiếng thơ đỏ nắng xanh cây quanh nhà - Mái chèo nghe vọng sông xa - Em êm nghe tiếng cửa bà năm xưa - Nghe trăng thổ động tàu dừa - Rào rào nghe chuyển cơn mưa giữa trời - Thêm yêu tiếng hát nụ cười - Nghe thơ em thấy đất trời đẹp ra. - Trần Đăng Khoa
	<p>Tam giác ABC là gì của đường tròn tâm O ?</p> <p>{What is triangle ABC with respect to the circle centered at O?}</p>	<p>Tam giác ABC là tam giác nội tiếp của đường tròn tâm O.</p> <p>{Triangle ABC is the inscribed triangle of the circle centered at O.}</p>

4.6. Kết luận

Vintern-1B đánh dấu một bước tiến tiên phong trong việc phát triển các mô hình đa phương thức (multimodal) cho tiếng Việt. Bằng cách tích hợp các mô hình ngôn ngữ và hình ảnh tiên tiến với các bộ dữ liệu chất lượng cao được thiết kế riêng cho tiếng Việt, Vintern-1B đã đạt hiệu suất ấn tượng trên nhiều tác vụ trong ngữ cảnh ngôn ngữ này.

Mô hình đặc biệt nổi trội trong các tác vụ liên quan đến nhận dạng ký tự quang học (OCR) và cho thấy sự cải thiện đáng kể trong khả năng hiểu bối cảnh và suy luận phức tạp về các hình ảnh liên quan đến văn hóa và đời sống Việt Nam. Với thiết kế tối ưu, Vintern-1B đảm bảo tính gọn nhẹ, đủ khả năng triển khai trên các thiết bị biên (edge devices) như điện thoại di động, từ đó mở rộng khả năng ứng dụng của mô hình trong nhiều lĩnh vực khác nhau. Được tạo ra và tinh chỉnh trên các bộ dữ liệu đa phương thức quy mô lớn dành riêng cho tiếng Việt, Vintern-1B không chỉ hiệu quả mà còn rất phù hợp với bối cảnh địa phương, thiết lập một chuẩn mực mới cho các mô hình đa phương thức tiếng Việt.

Việc phát hành mã nguồn mở của Vintern-1B cũng khuyến khích sự hợp tác và đổi mới, góp phần thu hẹp khoảng cách về tài nguyên và năng lực trong công nghệ ngôn ngữ tiếng Việt. Mô hình này mở ra những cơ hội phát triển và nghiên cứu sâu rộng hơn trong tương lai, tạo nền tảng vững chắc cho công nghệ ngôn ngữ Việt.