ĐẠI HỌC HUẾ
TRƯỜNG ĐẠI HỌC KHOA HỌC
KHOA CÔNG NGHỆ THÔNG TIN

KHÓA LUẬN
TỐT NGHIỆP ĐẠI HỌC

Đề tài:

TÌM HIỂU MÔ HÌNH PHOBERT VÀ ỨNG DỤNG TRONG BÀI TOÁN HỎI ĐÁP TIẾNG VIỆT

Sinh viên thực hiện: **NGUYỄN LUÔN MONG ĐỔ**

Khóa: **K44-HỆ CHÍNH QUY**

Huế, 05 - 2024

ĐẠI HỌC HUẾ
TRƯỜNG ĐẠI HỌC KHOA HỌC
KHOA CÔNG NGHỆ THÔNG TIN

KHÓA LUẬN
TỐT NGHIỆP ĐẠI HỌC
NGÀNH CÔNG NGHỆ THÔNG TIN

Đề tài:

TÌM HIỂU MÔ HÌNH PHOBERT VÀ ỨNG DỤNG TRONG BÀI TOÁN HỎI ĐÁP TIẾNG VIỆT

Sinh viên thực hiện: NGUYỄN LUÔN MONG ĐỔ

Khóa: K44-HỆ CHÍNH QUY

Giáo viên hướng dẫn: TS. ĐOÀN THỊ HỒNG PHƯỚC

Huế, 05 - 2024

LỜI CAM ĐOAN

Tôi xin cam đoan đồ án này là công trình nghiên cứu riêng của tôi, không sao chép ở bất kỳ công trình khoa học nào trước đây. Những phần có sử dụng tài liệu trong đồ án sẽ ghi rõ tên tài liệu trong phần tài liệu tham khảo. Các kết quả nêu trong đồ án có nguồn gốc rõ ràng và được trích dẫn đầy đủ.

Tôi xin hoàn toàn chịu trách nhiệm về đồ án này.

Thừa Thiên Huế, ngày....tháng 05 năm 2024

Sinh Viên

Nguyễn Luôn Mong Đổ

LỜI CẢM ƠN

Tôi xin chân thành cảm ơn Khoa Công Nghệ Thông Tin, Trường Đại Học Khoa Học Huế đã tạo điều kiện thuận lợi cho tôi học tập và thực hiện đề tài tốt nghiệp này. Trong suốt 4 năm học tập tại Trường Đài Học Khoa Học Huế, nhận được sự hướng dẫn, giảng dạy và giúp đỡ tận tình, nhiệt huyết của các thầy, cô giáo là niềm vinh dự lớn lao của cá nhân tôi và gia đình. Ngày hôm nay, tôi muốn gửi lời cảm ơn chân thành nhất đến tất cả các thầy, các cô trong nhà trường, họ là những giảng viên mà tôi vô cùng kính trọng.

Khóa luận này được hoàn thành nhờ sự hướng dẫn, tận tình chỉ bảo của cô giáo - TS. Đoàn Thị Hồng Phước. Cô luôn tạo điều kiện và có những góp ý thiết thực giúp tôi hoàn thành tốt nhiệm vụ. Tôi muốn gửi lời tri ân chân thành và sâu sắc nhất đến cô Đoàn Thị Hồng Phước: Kính chúc cô luôn mạnh khỏe, thành công và tận hưởng công việc đáng kính mà cô đã và đang làm.

Tôi muốn gửi lời cảm ơn đến gia đình và người thân đã nuôi dưỡng, luôn bên cạnh, thấu hiểu và tin tưởng, giúp tôi yên tâm học tập và làm việc.

Mặc dù đã cố gắng hoàn thành khóa luận trong phạm vi và khả năng cho phép của mình nhưng chắc chắn sẽ không tránh khỏi những thiếu sót. Tôi kính mong nhận được sự cảm thông, chỉ bảo, góp ý tận tình của quý thầy cô và các bạn.

Xin chân thành cám ơn!

DANH MỤC KÝ HIỆU VÀ CÁC CHỮ VIẾT TẮT

Từ viết tắt	Diễn giải	Dịch nghĩa
GPU	Graphics Processing Unit	Đơn vị xử lý đồ họa
CPU	Central Processing Unit	Đơn vị xử lý trung tâm
Q&A	Question and Answering	Hệ thống trả lời câu hỏi
POS TAGGING	Part Of Speech tagging	Gán nhãn thẻ từ loại

Danh sách bảng

Danh sách hình vẽ

Mục lục

MỞ ĐẦU

1. Lý do chọn đề tài

Ngôn ngữ của con người là một hệ thống các tín hiệu/ký hiệu được xây dựng một cách đặc biệt để truyền đạt được thông tin có chủ đích của người viết/người nói. Các tín hiệu/ký hiệu này được con người sử dụng để giao tiếp với nhau. Xử lý ngôn ngữ tự nhiên là một lĩnh vực đặc biệt, đó là sự kết hợp giữa các ngành khoa học máy tính, trí tuệ nhân tạo và ngôn ngữ học. Mục tiêu của việc xử lý ngôn ngữ tự nhiên là làm cho máy tính hiểu và xử lý được ngôn ngữ tự nhiên của con người.

Nghiên cứu về hệ thống hỏi đáp tự động(Q&A) đã thu hút sự quan tâm lớn trên thế giới từ rất lâu. Vào đầu những năm 1960, các hệ thống hỏi đáp đầu tiên sử dụng cơ sở dữ liệu đã được tạo ra, các hệ thống này được xây dựng dựa trên hai mô hình chính bao gồm: mô hình dựa trên trích xuất thông tin và mô hình dựa trên kiến thức. Các hệ thống này được xây dựng nhằm mục đích trả lời các câu hỏi về số liệu thống kê các trận đấu bóng chày và trả lời các sự kiện khoa học [8].

Vào cuối những năm 1990, World Wide Web ra đời và phát triển nhanh chóng tạo ra kho dữ liệu khổng lồ. Cho đến này nay, khi internet đã phát triển vượt trội, lượng thông tin mà con người được cung cấp và tiêu thụ ngày càng lớn. Hệ thống trả lời câu hỏi tự động ra đời nhằm cung cấp cho con người giải pháp giúp tiết kiệm thời gian đọc và làm giảm khối lượng kiến thức mà con người phải tiếp thu.

Với sự phát triển của các mô hình học sâu đã tạo ra bước độ phá của trí tuệ nhân tạo trong lĩnh vực xử lý ngôn ngữ tự nhiên. Cùng với sự thành công của

Chat GPT trong những năm gần đây, việc hiểu ngôn ngữ tự nhiên của máy tính đã trở nên dễ dàng hơn so với trước. Thay vì phải đọc toàn bộ bài viết để hiểu được nội dung, với hệ thống trả lời câu hỏi, người dùng chỉ cần cung cấp nội dung của bài viết và đặt ra các câu hỏi thắc mắc liên quan đến bài viết. Hệ thống giúp người dùng tiết kiệm thời gian đọc và loại bỏ các thông tin gây nhiễu cho người đọc. Tuy nhiên, việc xử lý ngôn ngữ tự nhiên đối với tiếng Việt vẫn còn nhiều hạn chế. Nguyên nhân đến từ việc tiếng Việt là một ngôn ngữ đơn lập với hệ thống từ ghép và từ láy đa dạng. Các yếu tố đó tạo ra sự "nhập nhằng"trong quá trình xử lý ngôn ngữ tự nhiên.

Với những lợi ích và thách thức mà hệ thống trả lời câu hỏi cho tiếng Việt mang lại, trong khóa luận tốt nghiệp này, tôi xin trình bày về đề tài "Tìm hiểu mô hình PhoBERT và ứng dụng trong bài toán hỏi đáp tiếng Việt".

2. Mục tiêu của khóa luận

Các mục tiêu chính của khóa luận bao gồm:

- Tìm hiểu mô hình Transformer, BERT
- Ứng dụng xây dựng hệ thống hỏi đáp tiếng Việt

Để đạt được các mục tiêu chính mà khóa luận đã đưa ra, các mục tiêu cụ thể bao gồm:

- Tìm hiểu mô hình Transformer, BERT
- Tìm hiểu về pretrain-model PhoBERT, mô hình được huấn luyện dựa trên tập dữ liệu tiếng Việt
- Xây dựng mô hình trả lời câu hỏi bằng việc tinh chỉnh lại PhoBERT
- Triển khai mô hình thành ứng dụng

3. Đối tượng và phạm vi của khóa luận

Đối tượng nghiên cứu của khóa luận bao gồm các mô hình Transformer, BERT và mô hình PhoBERT

Phạm vi nghiên cứu của khóa luận tập trung vào nghiên cứu quá trình xử lý và hiệu xuất của mô hình PhoBERT trong các tác vụ xử lý ngôn ngữ tự nhiên của tiếng Việt, đặc biệt là tác vụ trả lời câu hỏi.

4. Cấu trúc chính của khóa luận

Bài luận tốt nghiệp gồm các phần như sau:

Chương 1 : Giới thiệu về bài toán hệ thống hỏi đáp, các cách tiếp cận cho hệ thống hỏi đáp

Chương 2 : Tổng quan về mô hình Transformer, BERT, đề cập đến ưu điểm, cấu trúc của mô hình Transformer, BERT

Chương 3 : Xây dựng mô hình trả lời câu hỏi thông qua việc tinh chỉnh lại mô hình PhoBERT, đề cập đến cách tinh chỉnh mô hình PhoBERT cho nhiệm vụ cụ thể - trả lời câu hỏi, thiết lập thí nghiệm và đánh giá kết quả thu được, hiệu suất của mô hình đã xây dựng.

Chương 1

GIỚI THIỆU

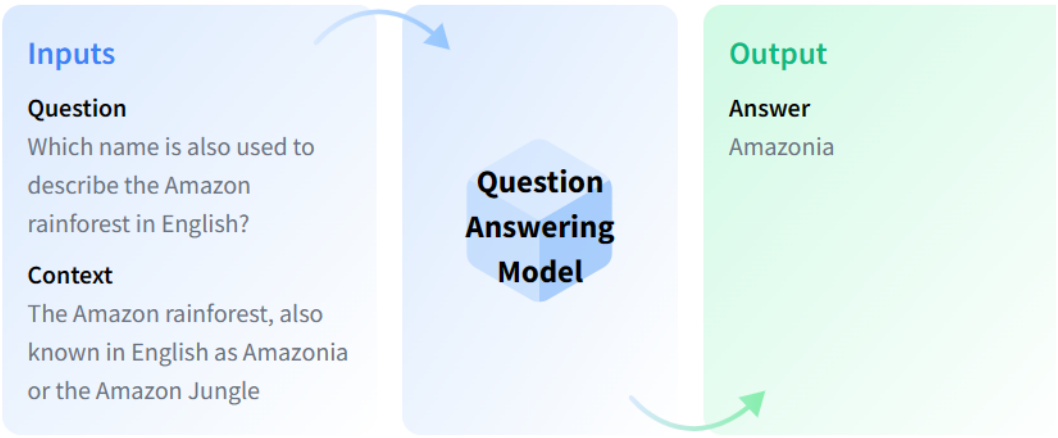
1.1 Mô tả bài toán

Nhiệm vụ của bài toán trả lời câu hỏi nhằm mục đích cho phép máy tính hiểu và trả lời câu hỏi của con người một cách tự nhiên, mở ra tiềm năng về xây dựng trợ lý ảo và xây dựng các công cụ tìm kiếm, quản lý hệ thống tri thức. Hệ thống trả lời câu hỏi có nhiệm vụ trả về một đoạn văn bản là câu trả lời cho câu hỏi được cung cấp, có hai loại tác vụ phổ biến cho hệ thống trả lời câu hỏi, bao gồm:

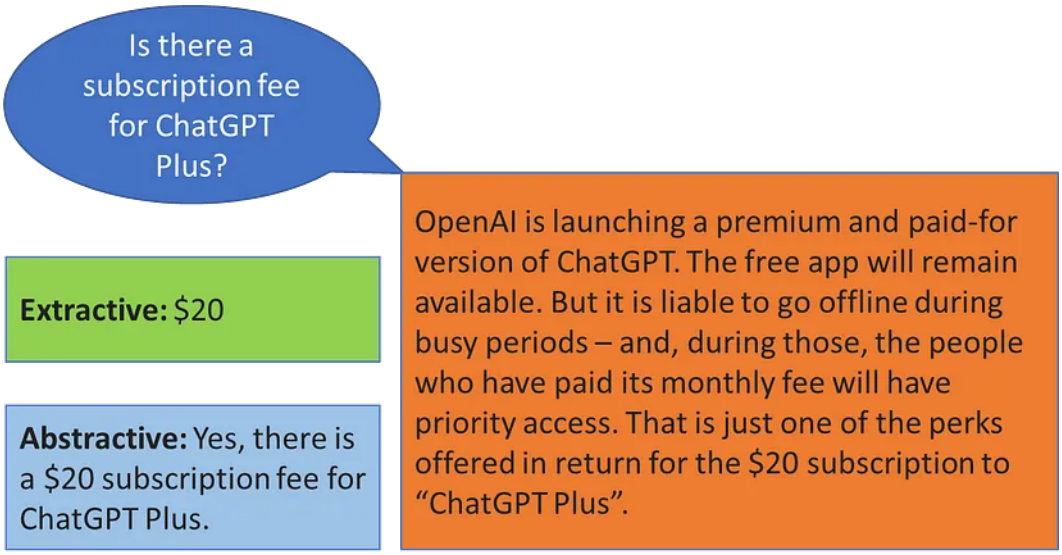
- Extractive: Trích xuất câu trả lời từ nội dung đoạn văn bản được cung cấp (mô tả trong hình 1.1)
- Abstractive: Tạo ra câu trả lời từ nội dung đoạn văn bản được cung cấp (mô tả trong hình 1.2)

Với hai tác vụ cho hệ thống trả lời câu hỏi bao gồm Extractive và Abstractive được nêu ở trên, các biến thể của hệ thống trả lời câu hỏi được ra đời, các biến thể đó bao gồm:

- Extractive QA: Mô hình trích xuất thông tin câu trả lời thông qua đoạn nội dung được cung cấp.



Hình 1.1: Yêu cầu đầu vào, đầu ra của nhiệm vụ Extractive đối với hệ thống trả lời câu hỏi



Hình 1.2: Sự khác nhau giữa hai tác vụ Extractive và Abstractive

- Open Generative QA: Mô hình sinh văn bản trực tiếp từ nội dung đoạn văn bản dược cung cấp.
- Closed Generative QA: Trong trường hợp này, không yêu cầu cung cấp đoạn văn bản chứa nội dung của câu hỏi. Câu trả lời hoàn toàn được mô hình sinh ra dựa trên tri thức đã học.

Với những tác vụ và biến thể của mô hình giải quyết nhiệm vụ trả lời câu hỏi đã nêu trên, hệ thống trả lời câu hỏi mà báo cáo này xây dựng thuộc mô hình Extractive QA - mô hình trích xuất câu trả lời từ nội dung đoạn văn bản và câu hỏi được cung cấp

Hệ thống trả lời câu hỏi mà báo cáo này tạo ra nhằm mục đích tóm tắt lại cho người dùng nội dung của đoạn văn bản, trả lời các câu hỏi mà người dùng đặt ra liên quan đến đoạn văn bản đã cung cấp.

Hệ thống trả lời câu hỏi yêu cầu người dùng cung cấp tri thức liên quan đến vấn đề thắc mắc (được gọi là Context) và câu hỏi cho hệ thống (được gọi là Question).

Thông qua tri thức được người dùng cung cấp (context), hệ thống sẽ thực hiện truy xuất dữ liệu phù hợp với câu hỏi và đưa ra câu trả lời xuất hiện trong tri thức được cung cấp trước đó đến người dùng. Thông tin về đầu vào và đầu ra của hệ thống trả lời câu hỏi bài báo cáo này sẽ xây dựng được mô trả trong hình 1.3

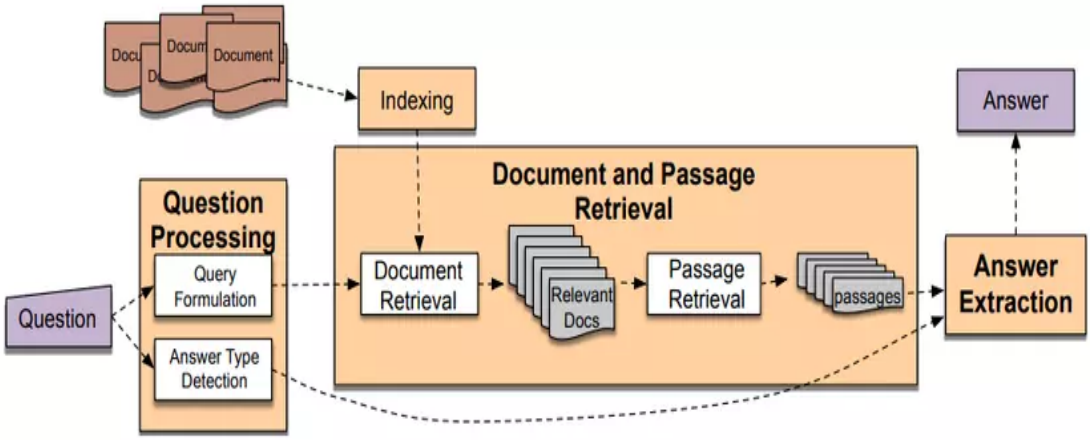
```
Question: Tên khoa học của động vật dưới nước là gì?

Context: Động vật lưỡng cư (danh pháp khoa học: Amphibia) là một lớp động
vật có xương sống máu lạnh. Tất cả các loài lưỡng cư hiện đại đều là phân
nhánh Lissamphibia của nhóm lớn Amphibia này. Động vật lưỡng cư phải trải
qua quá trình biến thái từ ấu trùng sống dưới nước tới dạng trưởng thành
có phổi thở không khí, mặc dù vài loài đã phát triển qua nhiều giai đoạn
khác nhau để bảo vệ hoặc bỏ qua giai đoạn ấu trùng ở trong nước dễ gặp
nguy hiểm. Da được dùng như cơ quan hô hấp phụ, một số loài kỳ giông và
ếch thiếu phổi phụ thuộc hoàn toàn vào da. Động vật lưỡng cư có hình dáng
giống bò sát, nhưng bò sát, cùng với chim và động vật có vú, là các loài
động vật có màng ối và không cần có nước để sinh sản. Trong những thập kỷ
gần đây, đã có sự suy giảm số lượng của nhiều loài lưỡng cư trên toàn cầu.

Answer: Amphibia
```

Hình 1.3: Hình ảnh mô tả đầu vào và đầu ra của bài toán. Đầu vào của bài toán bao gồm câu hỏi (question) và nội dung có liên quan (context). Đầu ra của bài toán là câu trả lời cho câu hỏi đó.

1.2 Ứng dụng của hệ thống trả lời câu hỏi

Hệ thống trả lời câu hỏi (Q&A) được ứng dụng trong nhiều lĩnh vực của đời sống, bao gồm:

Tìm kiếm thông tin nhanh chóng bằng cách trả lời các câu hỏi dựa trên nội dung có trong các cuốn sách, các bài báo liên quan.

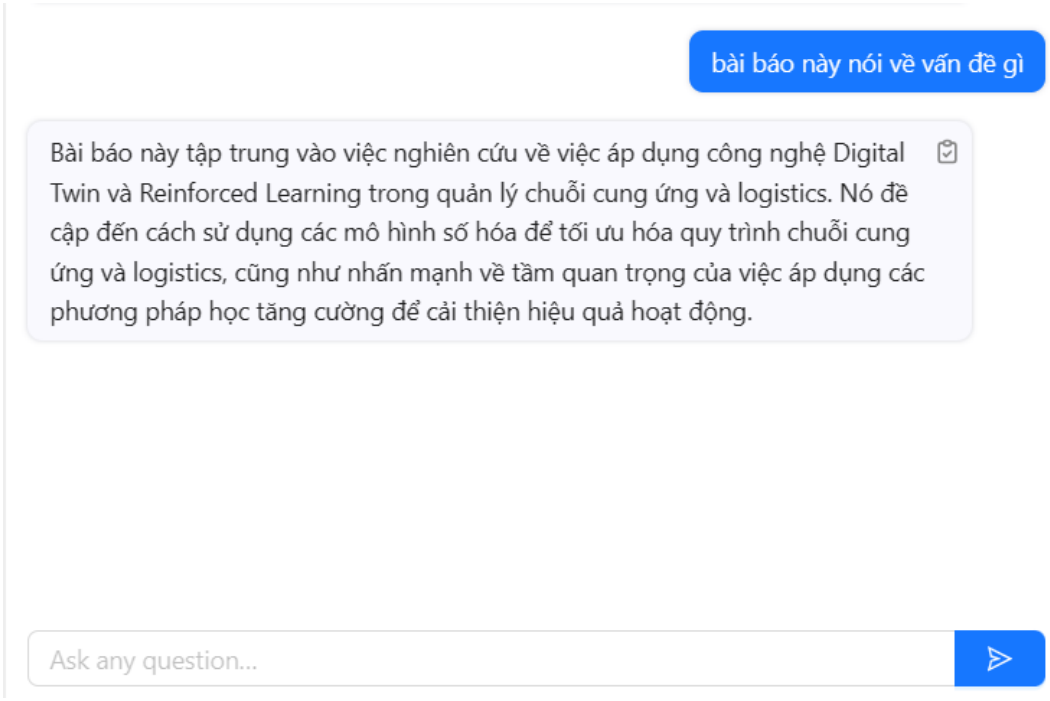


Hình 1.4: Mô hình mô tả ứng dụng của hệ thống trả lời câu hỏi trong việc tìm kiếm thông tin

Hỗ trợ việc học tập và nghiên cứu: người dùng có thể sử dụng hệ thống trả lời câu hỏi để tìm hiểu về một chủ đề cụ thể, đọc hiểu các bài báo, tóm tắt các ý chính... Chat PDF mô tả tại hình 1.5 và hình 1.6 được xây dựng nhằm đáp ứng nhu cầu này.

Ngoài ra, hệ thống trả lời câu hỏi còn được ứng dụng trong các lĩnh vực như y tế, phân tích dữ liệu. Trong lĩnh vực y tế, hệ thống trả lời câu hỏi có thể giúp các bệnh nhân tìm hiểu về các triệu chứng bệnh tật và các phương pháp điều trị. Trong lĩnh vực phân tích dữ liệu, hệ thống trả lời câu hỏi có thể giúp các doanh nghiệp trích xuất các thông tin có trong hợp đồng, các số liệu báo cáo...

Hình 1.5: Chat PDF yêu cầu người dung cung cấp tập tin chứa thông tin



Hình 1.6: Sau khi đã cung cấp thông tin, người dùng có thể đặt bất kỳ câu hỏi nào liên quan đến tập tin đó

1.3 Các thách thức của việc xây dựng ứng dụng trả lời câu hỏi cho tiếng Việt

Việc xây dựng hệ thống trả lời câu hỏi được sử dụng cho ngôn ngữ tiếng Việt gặp phải những thách thức sau:

Tiếng Việt là một ngôn ngữ đơn lập cùng với hệ thống từ ghép, từ láy, các từ đồng âm tạo nên vấn đề "nhập nhằng"trong việc hiểu ngữ nghĩa của câu.Ví dụ, trong câu sau: "hổ mang bò lên núi", ở đây có thể hiểu theo hai nghĩa khác nhau bao gồm: hổ (danh từ), mang (động từ), bò (danh từ), hoặc cũng có thể hiểu theo nghĩa: hổ mang (danh từ), bò (động từ).

Để huấn luyện mô hình trả lời câu hỏi cho tiếng Việt cần lượng dữ liệu lớn và tài nguyên GPU để đảm bảo hiệu suất và chất lượng của mô hình.

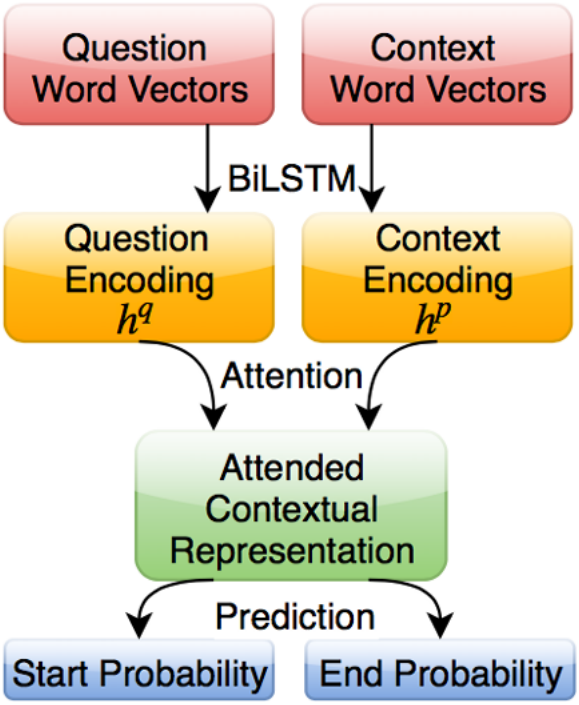
1.4 Các cách tiếp cận cho bài toán

1.4.1 Hệ thống trả lời câu hỏi với cách tiếp cận của mạng Neural hồi quy (RNN) và cơ chế Attention

Mô hình này được giới thiệu trong tài liệu "Attention-based Recurrent Neural Networks for Question Answering"của nhóm tác giả Dapeng Hong và Billy Wan.

Với cách tiếp cận này, nhóm tác giả sử dụng bộ vector GloVe [9] đã được tiền huấn luyện trước với số chiều là 100, phương pháp tối ưu hóa Adam [10], sử dụng hàm sofmax cross-entropy loss để tính toán sai lệch giữa vị trí bắt đầu và kết thúc của câu trả lời có trong tri thức cung cấp (gọi là context) và vị trí thực sự của đáp án. Kết quả thu được trong bảng 1.1

Với mô hình này, việc sử dụng bộ vector GloVe đã được tiền huấn luyện



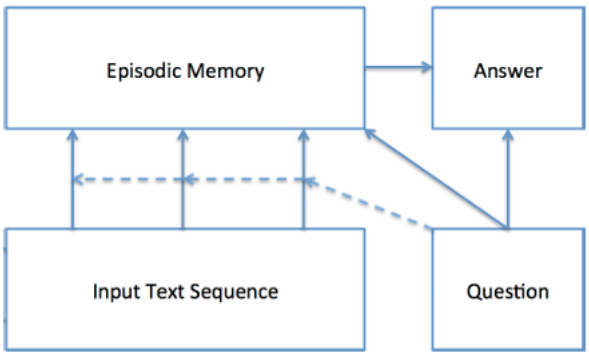
Hình 1.7: Cấu trúc chung của mô hình trả lời câu hỏi sử dụng mạng Neural (Nguồn: [3])

Bảng 1.1: Kết quả dựa trên tập dữ liệu SQuAD dev

Model	F1	EM	Score
Baseline	48.7	35.2	40.8
Match-LSTM	58.8	44.6	50.7
BiDAF	62.8	48.6	54.8
Ensemble	58.4	43.6	49.9

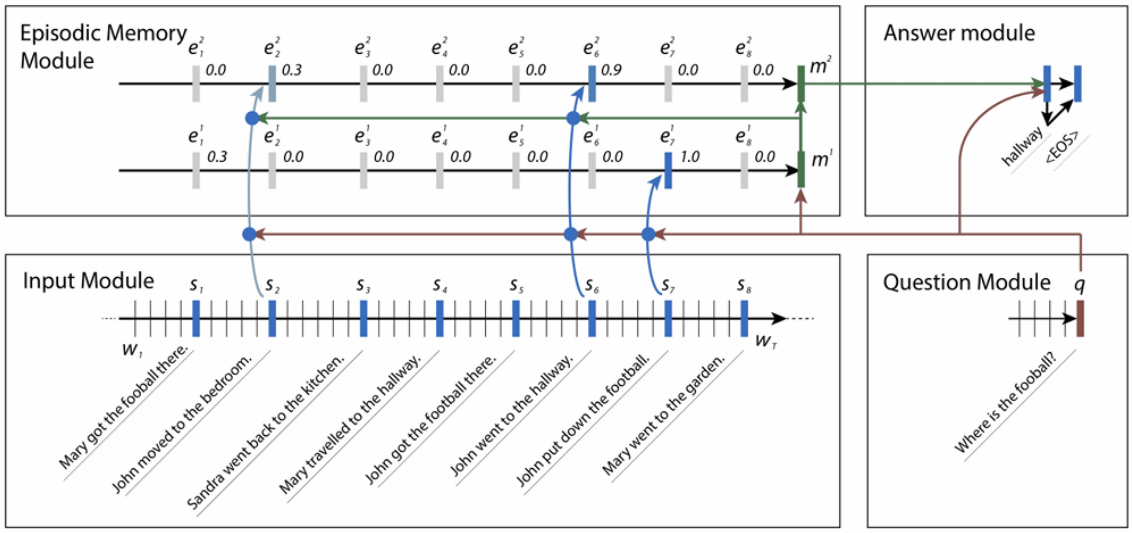
làm cho việc truyền tải ngữ cảnh của từ mất đi tính đa dạng của chúng. Cụ thể, bộ vector Glove tìm ra 1 vector đại diện cho mỗi từ dựa trên tập ngữ liệu lớn nên không thể hiện được hết sự đa dạng của ngữ cảnh.

1.4.2 Hệ thống trả lời câu hỏi với cách tiếp cận của mô hình Dynamic Memory Networks



Hình 1.8: Cấu trúc của mô hình trả lời câu hỏi sử dụng Dynamic Memory Networks (Nguồn: [4]

Mô hình này được nhắc đến trong tài liệu "Ask MeAnything: Dynamic Memory Networks for Natural Language Processing" [4] của nhóm tác giả gồm 9 người thực hiện. Tập dữ liệu nhóm tác giả sử dụng là The Facebook bAbI, tập dữ liệu gồm 20 nhiệm vụ liên quan đến hệ thống trả lời câu hỏi.



Hình 1.9: Ví dụ mô tả cách thức hoạt động của Dynamic Memory Networks (Nguồn: [4])

Mô hình sử dụng Word2vec của bộ vector được tiền huấn luyện Glove. Kết quả của mô hình thu được với tập dữ liệu The Facebook bAbI được biểu diễn tại bảng 1.2

Bảng 1.2: Kết quả dựa trên tập dữ liệu bABI dev

Task	DMN	Task	DMN
1: Single Supporting Fact	100.0	11: Basic Coreference	99.9
2: Two Supporting Facts	98.2	12: Conjunction	100
3: Three Supporting Facts	95.2	13: Compound Coreference	99.8
4: Two Argument Relations	100	14: Time Reasoning	100
5: Three Argument Relations	99.3	15: Basic Deduction	100
6: Yes/No Questions	100	16: Basic Induction	99.4
7: Counting	96.9	17: Positional Reasoning	59.6
8: Lists/Sets	96.5	18: Size Reasoning	95.3
9: Simple Negation	100	19: Path Finding	34.5
10: Indefinite Knowledge	97.5	20: Agent's Motivations	100
		Mean Accuracy (%)	93.6

1.5 Tiểu kết chương 1

Nội dung chương 1 đã giới thiệu bài toán trả lời câu hỏi và ứng dụng rộng rãi của bài toán này trong nhiều lĩnh vực: từ việc tìm kiếm thông tin trên các

sách, báo, phương tiện truyền thông đến việc hỗ trợ học tập, nghiên cứu, ứng dụng trong lĩnh vực y tế, doanh nghiệp... Đồng thời chương 1 cũng nêu ra thách thức mà bài toán trả lời câu hỏi gặp phải đối với việc xây dựng với tập dữ liệu tiếng Việt. Bên cạnh đó, chương 1 cũng nên ra các cách tiếp cận trong việc giải quyết bài toán trả lời câu hỏi này.

Chương 2

TỔNG QUAN VỀ MÔ HÌNH TRANSFORMER VÀ MÔ HÌNH BERT

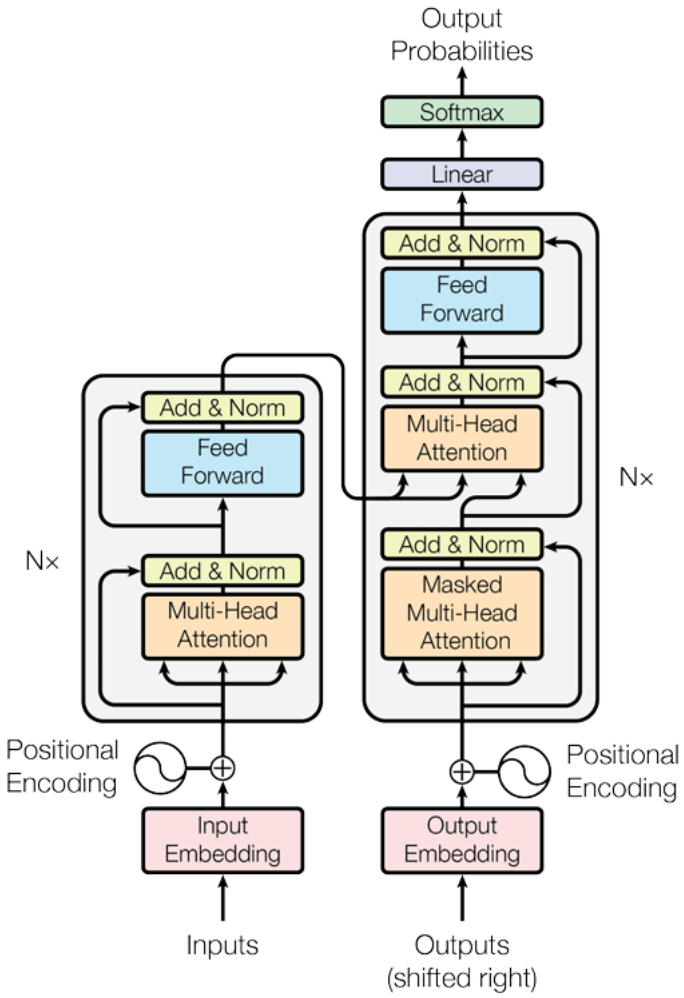
2.1 Mô hình Transformer

Transformer là một mô hình học sâu được giới thiệu vào năm 2017 trong paper "Attention is all you need" [5] bởi nhóm tác giả đến từ Google. Khác với các mô hình Neural hồi quy (RNN, LSTM...) được sử dụng trong bài toán seq2seq, mô hình Transformer hoàn toàn hoạt động dựa trên cơ chế Attention. Mô hình Transformer có 2 phần bao gồm Bộ mã hóa (Encoder) và Bộ giải mã (Decoder).

Bộ mã hóa (Encoder) được tạo thành từ một chuỗi gồm N = 6 lớp có cấu tạo giống nhau xếp chồng lên. Mỗi lớp bao gồm 2 lớp con. Lớp đầu tiên được gọi là Multi-head-self-attention, lớp thứ hai là mạng Neural Feed-FForward. Đầu ra của mỗi lớp con trong Bộ mã hóa có số chiều là $d_{\text{model}} = 512$

Bộ giải mã (Decoder) được tạo thành từ một chuỗi gồm N = 6 lớp có cấu tạo giống nhau xếp chồng lên. Ngoài hai lớp con tương tự như Bộ mã hóa (Encoder), Bộ giải mã chèn thêm một lớp con thứ ba, thực hiện cơ chế tự chú ý (Self-multi-head-attention) trên đầu ra của Bộ mã hóa. Bộ mã hóa điều chỉnh

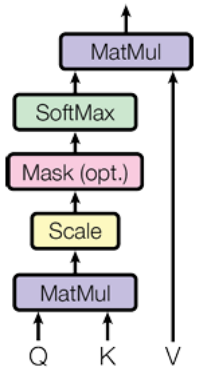
lớp tự chú ý (self-attention) trong mỗi lớp của Bộ giải mã để ngăn các từ bị ảnh hưởng bởi cơ chế attention của các từ xuất hiện sau đó thông qua lớp Masked Multi-head Attention.



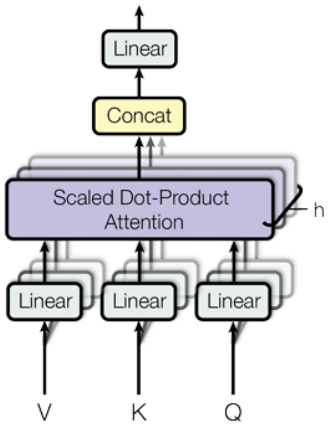
Hình 2.1: Cấu trúc của mô hình Transformer (Nguồn: [5])

2.1.1 Cơ chế chú ý (Attention)

Cơ chế chú ý (Attention) được mô tả như việc ánh xạ một truy vấn (query) vầ một cặp khóa - giá trị (key - value) với đầu ra. Trong đó truy vấn (query), khóa (key), giá trị (value) và đầu ra đều là các vector. Đầu ra được tính toán bằng cách tính tổng có trọng số của các vector giá trị (value), với trọng số được gán cho mỗi vector giá trị (value) được tính bằng một hàm liên quan giữa vector truy vấn (query) và khóa (key) tương ứng.



Hình 2.2: Scaled Dot-Product (Nguồn: [5])



Hình 2.3: Multi-Head Attention (Nguồn: [5])

2.1.1.1 Scaled Dot-Product Attention

Cơ chế chú ý (Attention) được giới thiệu trong bài báo "Attention is all you need"được gọi là "Scaled Dot-Product Attention"Hình 3.2. Đầu vào bao gồm các vector truy vấn và các khóa có số chiều là d_{k}, các vector giá trị có số chiều là d_{v}. Sau đó, tiến hành tính tích vô hướng của của các vector truy vấn và các vector khóa, chia mỗi giá trị đó cho $\sqrt{d_{\mathrm{k}}}$ và áp dụng hàm softmax để thu được trọng số của các vector giá trị.

Trong thực tế, các vector truy vấn, vector khóa và các vector giá trị sẽ được tập hợp lại thành các ma trận được gọi là Q, K, V. Khi đó, việc tính toán dựa trên ma trận sẽ được thực hiện dựa trên công thức:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.1}$$

2.1.1.2 Multi-Head Attention

Thay vì thực hiện cơ chế chú ý (Attention) duy nhất trên ma trận truy vấn, ma trận khóa và ma trận giá trị. Multi-Head Attention cho phép tạo ra nhiều phiên bản của các ma trận đó, được gọi là các "đầu"(heads). Mỗi đầu sẽ được học để chú ý vào các mối quan hệ khác nhau trong dữ liệu đầu vào bằng cách thực hiện cơ chế chú ý (Attention). Quá trình chú ý (Attention) sẽ được thực hiện

độc lập trên các "đầu"(heads) đó. Các kết quả của quá trình chú ý (Attention) tại các "đầu"(heads) sẽ được nối với nhau thông qua lớp Concat Hình 3.3

$$\text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}_1,...,\text{head}_h) \cdot W^O \tag{2.2}$$

trong đó:

$$\text{head}_i = \text{Attention}(Q \cdot W_i^Q, K \cdot W_i^K, V \cdot W_i^V) \tag{2.3}$$

Với kích thước của các ma trận như sau: $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, and $W^O \in \mathbb{R}^{h_d \times d_{\text{model}}}$.

2.1.2 Position-wise Feed-Forward Networks(FFN)

Các vector sau khi thực hiện bước Add & Normalize sẽ làm đầu vào cho mạng Neural Feed-Forward. Mạng Neural Feed-Forward bao gồm hai tầng biến đổi thông tin và một hàm kích hoạt ReLU ở giữa hai tầng biến đổi tuyến tính đó. Tham số Dropout với tỉ lệ 0.1 được áp dụng ở lần biến đổi thứ nhất sau khi các vector qua hàm ReLU.

$$FFN(x) = \max(0, XW_1 + b_1)W_2 + b_2$$

Trong đó:

- X là đầu vào.

- W_1 và b_1 là trọng số và sai số của lớp ẩn.

- W_2 và b_2 là trọng số và sai số của lớp đầu ra.

- $\max(0,...)$ biểu thị hàm ReLU (Rectified Linear Activation Function).

2.1.3 Positional Encoding (PE)

Với mô hình Transformer, tất cả các vector của từ sẽ được chuyển vào lớp đầu vào cùng một lúc, song song với nhau dẫn đến việc mô hình không thể nhận diện được thứ tự của chuỗi đầu vào như các mô hình mạng Neural hồi quy. Khi đó, mô hình yêu cầu cần phải chèn thông tin về vị trí của các token trong chuỗi hiện tại để mô hình có thể biết được trong quá trình tính toán. Để làm được điều này, Positional Encoding (PE) đã ra đời. PE sẽ có cùng kích thước với các vector nhúng từ (kích thước = d_{model})

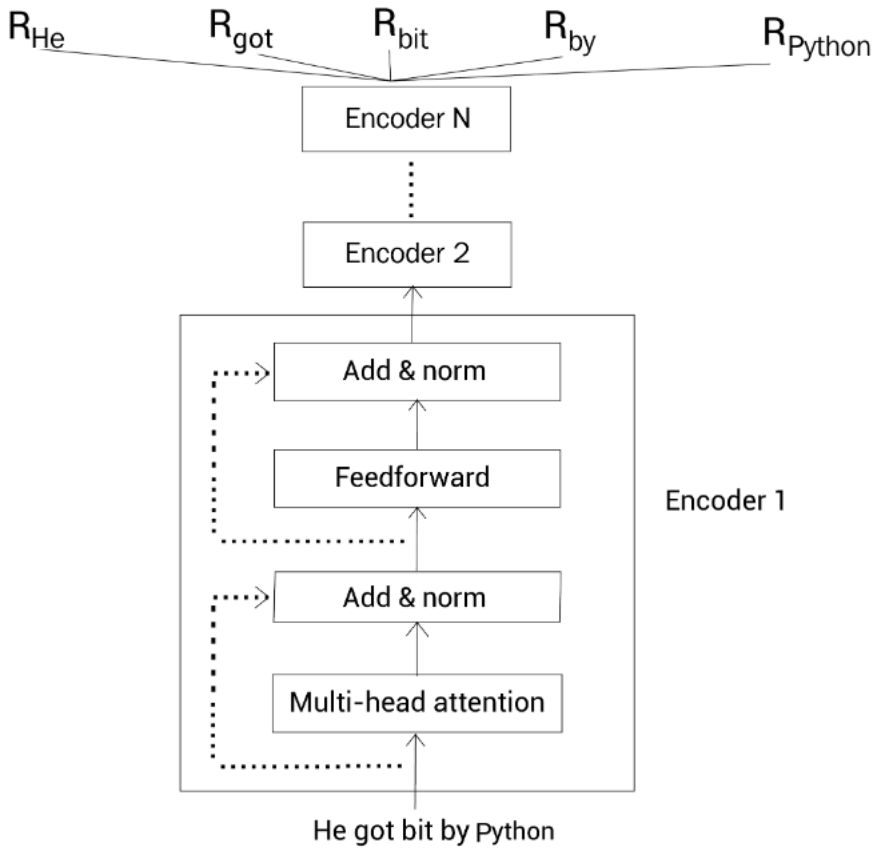
$$\text{PE}(\text{pos}, 2i) = \sin\left(\frac{\text{pos}}{10000^{2i/d_{\text{model}}}}\right)$$

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

Trong đó, pos là vị trí của từ trong câu và i là chiều của embedding.

2.2 Mô hình BERT

BERT, viết tắt của Bidirectional Encoder Representations from Transformers. BERT là một kiến trúc mô hình mới cho lớp bài toán Language Representation được công bố vào tháng 11 năm 2018 bởi nhóm tác giả đến từ Google, bao gồm Jacob Devlin, Ming-Wei Chang, Kenton Lee và Kristina Toutanova [6]. Khác với các mô hình trước đó, BERT được thiết kế để tiền huấn luyện các biểu diễn hai chiều từ dữ liệu văn bản chưa được gán nhãn dựa trên ngữ cảnh hai chiều của chúng. Từ đó, mô hình BERT sau khi được tiền huấn luyện có thể tinh chỉnh với một lớp đầu ra bổ sung phù hợp với từng nhiệm vụ của mô hình, có thể là như trả lời câu hỏi (question and answering) và suy luận ngôn ngữ (language inference), mà không cần thay đổi cấu trúc của cả mô hình cho từng nhiệm vụ cụ thể.

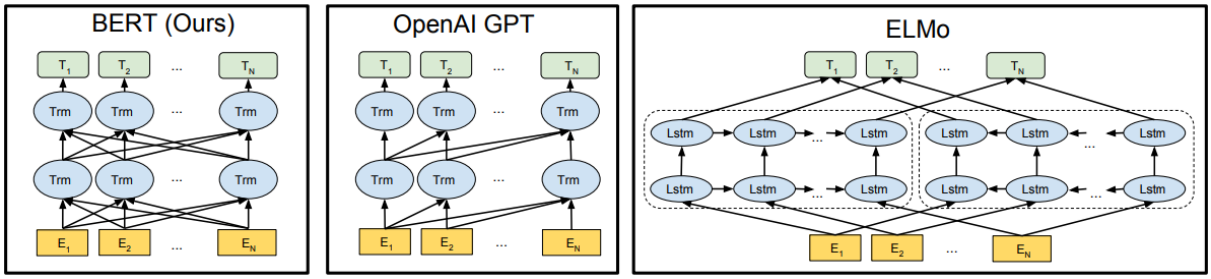


Hình 2.4: BERT tạo ra vector đại diện cho từng từ dựa trên ngữ cảnh của câu

Có hai chiến lược hiện áp dụng cho các biểu diễn ngôn ngữ đã được huấn luyện trước là: feature-based và fine-tuning. Phương pháp feature-based, như ELMo [11], sử dụng các kiến trúc cụ thể cho từng nhiệm vụ bao gồm các biểu diễn được tiền huấn luyện làm đặc trưng bổ sung. Phương pháp fine-tuning, như Generative Pre-trained Transformer (OpenAI GPT) [12], giới thiệu các tham số cụ thể cho từng nhiệm vụ tối thiểu, được huấn luyện trên các nhiệm vụ bằng cách tinh chỉnh các tham số được tiền huấn luyện. Hai phương pháp này sử dụng chung một hàm mục tiêu trong quá trình tiền huấn luyện.

Tuy nhiên các kỹ thuật trên gặp phải vấn đề trong việc thể hiện khả năng của các mô hình vector đại diện, đặc biệt với hướng tiếp cận fine-tuning. Với cách tiếp cận fine-tuning, mô hình được huấn luyện dựa trên ngữ cảnh một chiều của văn bản. Trong OpenAI GPT, nhóm tác giả sử dụng kiến trúc left-to-right, nghĩa là các tokens chỉ phụ thuộc vào các token ở trước đó. Với cách xây dựng

của mô hình BERT, nhóm tác giả đã khắc phục được hạn chế này bằng cách tiếp cận fine-tuning theo hướng khác, gọi là Bidirectional Encoder Representations from Transformers.



Hình 2.5: Sự khác nhau giữa BERT, OpenAI GPT, ELMo (Nguồn: [6] figure 3)

Mô hình BERT gồm cấu trúc đa tầng với nhiều lớp Bidirectional Transformer encoder. Các phiên bản BERT được đào tạo với những thiết lập khác nhau được mô tả trong bảng 2.1

Bảng 2.1: Thông tin về các phiên bản mô hình BERT được đào tạo

	H=128	**H=256**	**H=512**	**H=768**
L=2	2/128 BERT-tiny	2/256	2/512	2/768
L=4	4/128	4/256 BERT-mini	4/512 BERT-small	4/768
L=6	6/128	6/256	6/512	6/768
L=8	8/128	8/256	8/512 BERT-medium	8/768
L=10	10/128	10/256	10/512	10/768
L=12	12/128	12/256	12/512	12/768 BERT-base

Có hai phiên bản BERT được giới thiệu trong tài liệu năm 2018 [6] bao gồm:

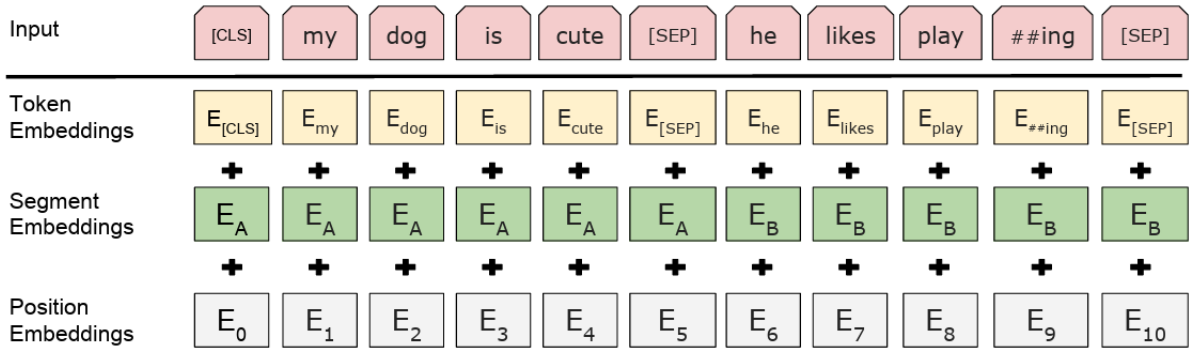
$$BERT_{\text{BASE}}(L = 12,\ H = 768,\ A = 12,\ \text{Tổng tham số} = 110M)$$

$$BERT_{\text{LARGE}}(L = 24,\ H = 1024,\ A = 16,\ \text{Tổng tham số} = 340M)$$

2.2.1 Input Representation

Để mô hình BERT có thể xử lý được một loạt các tác vụ khác nhau, biểu diễn đầu vào (input representation) của BERT có khả năng biểu diễn một cách rõ ràng cả một câu đơn và một cặp câu (trong trường hợp cặp: câu hỏi, câu trả

lời) trong một chuỗi token duy nhất.

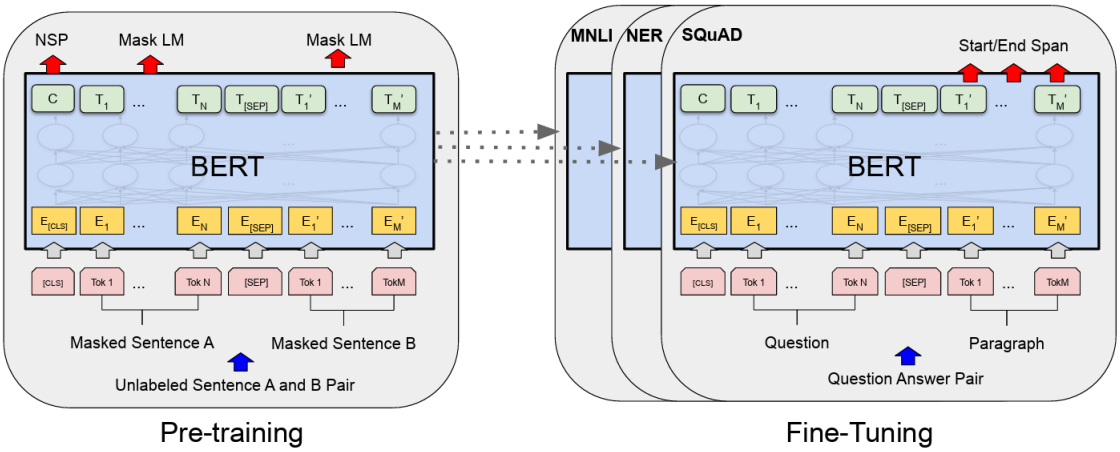


Hình 2.6: Biểu diễn đầu vào của BERT (BERT Input Representation), được tính bằng tổng của token embeddings, segmentation embeddings và position embeddings (PE) (Nguồn: [6] figure 2)

2.2.2 Pre-training Tasks

BERT được đào tạo đồng thời bởi hai nhiệm vụ không giám sát bao gồm: Mô hình ngôn ngữ mặt nạ (Masked language modeling) và dự đoán câu tiếp theo (Next sentence prediction).

Sau khi đã được huấn luyện dựa trên hai nhiệm vụ, mô hình BERT có thể được tinh chỉnh để phù hợp với từng nhiệm vụ cụ thể khác nhau



Hình 2.7: Quá trình tiền huấn luyện và fine-tuning của mô hình BERT (Nguồn: [6] figure 1)

2.2.3 Mô hình ngôn ngữ mặt nạ

Trước khi đi vào tìm hiểu nhiệm vụ xây dựng mô hình ngôn ngữ mặt nạ , hãy tìm hiểu về cách hoạt động của nhiệm vụ xây dựng mô hình ngôn ngữ (Language modeling)

Language modeling

Trong nhiệm vụ xây dựng mô hình ngôn ngữ, mô hình được huấn luyện để dự đoán từ tiếp theo với đầu vào là một chuỗi các từ. Mô hình ngôn ngữ gồm 2 loại:

- Mô hình ngôn ngữ tự động hồi quy (Auto-regressive language modeling): Với mô hình ngôn ngữ tự động hồi quy này có 2 cách tiếp cận
  - Dự đoán tiến (Forward prediction - dự đoán từ theo thứ tự từ trái phải)
  - Dự đoán lùi (Backward prediction - dự đoán từ theo thứ tự từ phải sang trái)
- Mô hình ngôn ngữ mã hóa tự động (Auto-encoding language modeling):
  - Mô hình ngôn ngữ mã hóa tự động tận dụng cả việc dự đoán tiến và dự đoán lùi. Mô hình sẽ đọc dữ liệu đầu vào theo cả hai hướng để đưa ra dự đoán.

BERT là mô hình ngôn ngữ mã hóa tự động, mô hình đọc dữ liệu đầu vào theo cả hai hướng để đưa ra dự đoán.Để huấn luyện mô hình tìm ra đại diện dựa vào ngữ cảnh 2 chiều, BERT sử dụng cách tiếp cận che giấu đi một cách ngẫu nhiên một số token đầu vào, sau đó mô hình chỉ dự đoán các token được giấu đi đó, được gọi là "masked LM"(MLM). Trong trường hợp này, các hidden vectors ở lớp cuối cùng tương ứng với các tokens được ẩn đi được đưa vào 1 lớp softmax trên toàn bộ từ vựng để dự đoán. Nhóm nghiên cứu của Google đã thử nghiệm che đi 15% tất cả các token lấy từ từ điển của WordPiece trong câu một cách ngẫu nhiên và chỉ dự đoán các từ được che đi đó.

Mặc dù điều này cho phép mô hình học được ngữ cảnh hai chiều của từ nhưng có một nhược điểm rằng các token được che đi sẽ không thể được nhìn

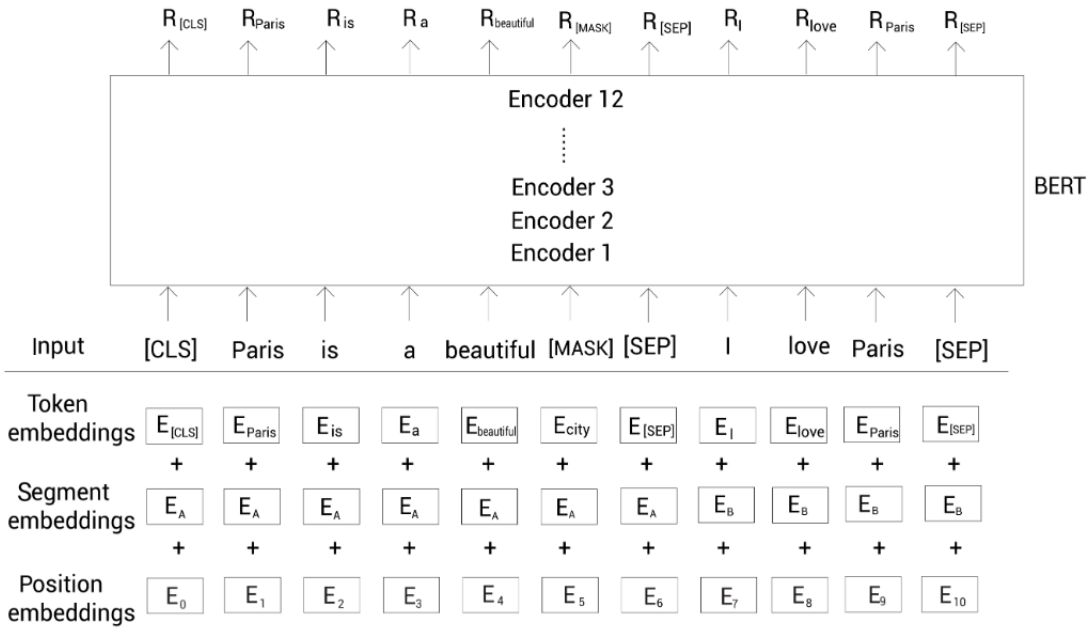
thấy trong quá trình fine-tuning. Để hạn chế nhược điểm đó, không phải lúc nào mô hình cũng thay thế các từ được che đi bằng token [MASK]. Thay vào đó, mô hình chọn 15% tokens một cách ngẫu nhiên và thực hiện các bước như sau:

(1) Thay thế 80% từ được chọn trong dữ liệu thành token [MASK].

(2) 10% các từ được chọn sẽ được thay thế bởi 1 từ ngẫu nhiên.

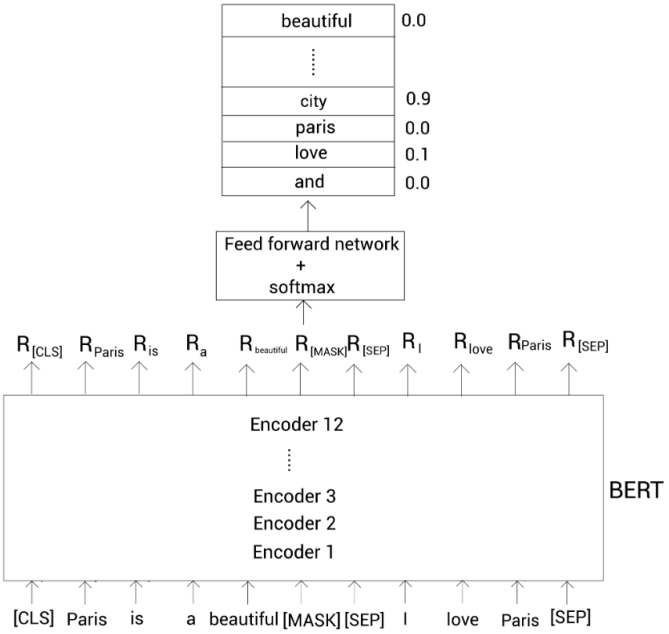
(3) 10% còn lại được giữ không thay đổi.

Quá trình xây dựng mô hình ngôn ngữ mặt nạ được mô tả tại hình 2.8 dưới đây:



Hình 2.8: Quá trình tiền huấn luyện mô hình BERT với nhiệm vụ xây dựng mô hình ngôn ngữ mặt nạ)

Sau đó, để dự đoán token được mask, mô hình sẽ đưa vector đại diện của token được mask do mô hình vừa đưa ra vào một mạng chuyển tiếp (feedfoward network) với hàm kích hoạt softmax. Khi đó mạng chuyển tiếp sẽ nhận $R_{\mathbf{MASK}}$ làm đầu vào và trả về giá trị xác suất có thể là token được mask đó của tất cả các từ có trong tập từ vựng của mô hình. Kết quả được mô tả tại hình 2.9

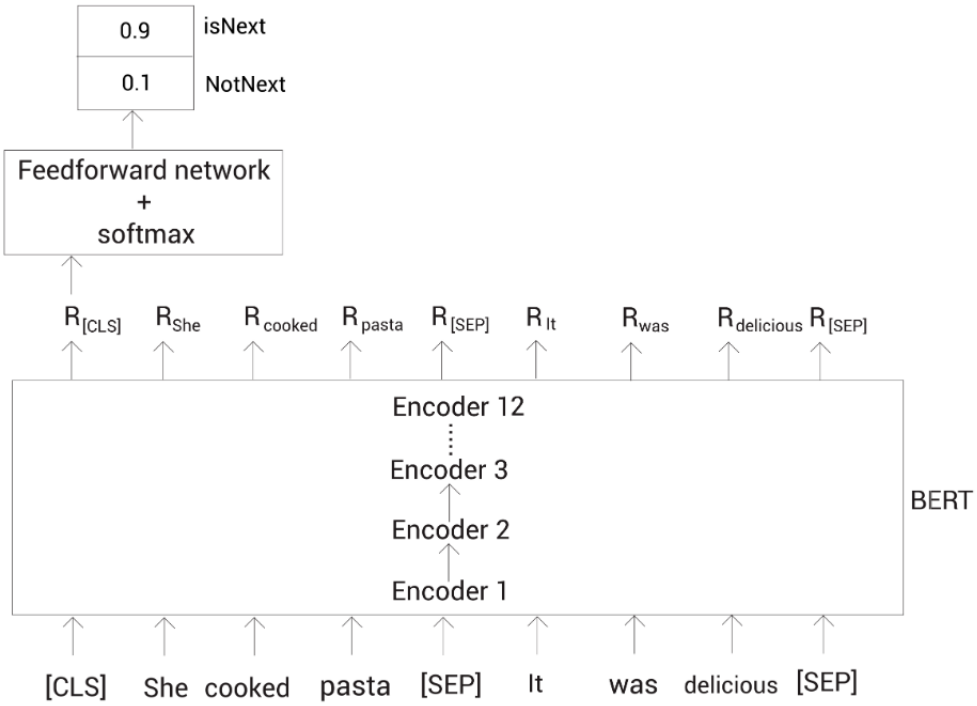


Hình 2.9: Quá trình đưa ra kết quả xác suất có thể là $R_{\mathbf{MASK}}$ của tất cả các từ có trong tập từ vựng

2.2.4 Mô hình dự đoán câu tiếp theo

Nhiều nhiệm vụ quan trọng trong xử lý ngôn ngữ tự nhiên như trả lời câu hỏi (Question Answering) dựa trên việc hiểu mối quan hệ giữa hai câu liên tiếp. Để huấn luyện mô hình hiểu mối liên hệ giữa 2 câu, BERT sử dụng tác vụ với nhiệm vụ xây dựng một mô hình dự đoán câu tiếp theo dựa vào câu hiện tại. Cụ thể, khi chọn câu A và câu B cho mỗi training sample, 50% khả năng câu B là câu tiếp theo sau câu A (được gán nhãn là isNext) và 50% còn lại là một câu ngẫu nhiên nào đó trong corpus (được gán nhãn là notNext).

Sau khi mô hình tính toán và đưa ra các vector đại diện, $R_{\mathbf{CLS}}$ chứa hầu hết tất cả thông tin tổng hợp của các câu văn bản, do đó chỉ cần đưa $R_{\mathbf{CLS}}$ làm đầu vào cho mạng chuyển tiếp (feedfoward network) với hàm kích hoạt softmax để lấy được xác suất của 2 lớp đầu ra isNext và notNext.



Hình 2.10: Quá trình đưa ra kết quả xác suất có thể là $R_{\mathbf{MASK}}$ của tất cả các từ có trong tập từ vựng

2.3 Tiểu kết chương 2

Nội dung chương 2 giới thiệu về mô hình Transformer và mô hình BERT. Mô hình Transformer có cơ chế chú ý (Attention) nổi bật lên so với các mô hình hồi quy được phát triển trước đó. Mô hình BERT lại có ưu điểm trong việc hiểu rõ ngữ cảnh của từ, kế thừa và phát triển từ mô hình Transformer, BERT hiểu được ngữ cảnh của câu đồng thời theo hai chiều. Đồng thời, trong chương 2 này cũng đã giới thiệu các tác vụ huấn luyện của mô hình BERT bao gồm mô hình ngôn ngữ mặt nạ và mô hình dự đoán câu tiếp theo, hai nhiệm vụ này sẽ được học đồng thời trong quá trình huấn luyện.

Chương 3

XÂY DỰNG MÔ HÌNH TRẢ LỜI CÂU HỎI

3.1 Thông tin tập dữ liệu

Tập dữ liệu được sử dụng có tên là Vietnamese Question Answering Dataset (UIT-ViQuAD) [1] được giới thiệu bởi nhóm tác giả đến từ Trường Đại học Công nghệ Thông tin, Đại học Quốc gia Thành phố Hồ Chí Minh. Tập dữ liệu bao gồm tổng cộng 23074 câu hỏi đến từ nhiều lĩnh vực khác nhau được giới thiệu trong phiên bản 1.0. Bảng 4.1 mô tả chi tiết tập dữ liệu UIT-ViQuAD 1.0

Bảng 3.1: Thống kê tổng quan về tập dữ liệu UIT-ViQuAD (Nguồn: [1])

	Train	**Test**	**Dev**	**All**
Số lượng bài báo	138	18	18	174
Số lượng đoạn văn	4,101	515	493	5,109
Số lượng câu hỏi	18,579	2,285	2,210	23,074
Độ dài trung bình của đoạn văn	153.9	147.9	155.0	153.4
Độ dài trung bình của câu hỏi	12.2	11.9	12.2	12.2
Độ dài trung bình của câu trả lời	8.1	8.4	8.9	8.2
Kích thước từ vựng	36,174	9,184	9,792	41,773

Tập dữ liệu được lưu trữ dưới định dạng file .json, dưới đây là một mô tả về cấu trúc của tập dữ liệu:

```
{
  "context": "Khác với nhiều ngôn ngữ Ấn-Âu khác, tiếng Anh đã gần
  như loại bỏ hệ thống biến tố dựa trên cách để thay bằng cấu trúc
```

```
  phân tích. Đại từ nhân xưng duy trì hệ thống cách hoàn chỉnh hơn
  những lớp từ khác. Tiếng  Anh có bảy lớp từ chính: động từ, danh
  từ, tính từ, trạng từ, hạn định từ (tức mạo từ), giới từ, và
  liên từ. Có thể tách đại từ khỏi danh từ, và thêm vào thán từ.",
  "questions": [
    {
      "question": "Tiếng Anh có bao nhiêu loại từ?",
      "is_impossible": false,
      "answer": "bảy."
    },
    {
      "question": "Ngôn ngữ Ấn-Âu có bao nhiêu loại từ?",
      "is_impossible": true,
      "plausible_answer": "bảy."
    }
  ]
}
```


Tập dữ liệu được sử dụng để xây dựng mô hình PhoBERT trong bài báo cáo này là tập dữ liệu UIT-ViQuAD 2.0. Tập dữ liệu này mở rộng số lượng câu hỏi từ 23074 lên 35, 990. Chi tiết mô tả trong bảng 3.2

Bảng 3.2: Thống kê tổng quan về tập dữ liệu UIT-ViQuAD (Nguồn: [1])

	Train	**Test**	**Dev**	**All**
Số lượng bài báo	138	19	19	176
Số lượng đoạn văn	4,101	557	515	5,173
Số lượng câu hỏi	28,457	3,821	3,712	35,990
Độ dài trung bình của đoạn văn	179.0	167.6	177.3	177.6
Độ dài trung bình của câu hỏi	14.6	14.3	14.7	14.6

3.2 Tinh chỉnh mô hình BERT cho nhiệm vụ trả lời câu hỏi (Question and Answering)

Trong nhiệm vụ trả lời câu hỏi, mô hình sẽ được cung cấp một câu hỏi kèm theo đoạn văn bản chứa câu trả lời cho câu hỏi đó. Mục tiêu của mô hình là trích xuất câu trả lời từ đoạn văn bản chứa câu trả lời đó. Phần tiếp theo tôi sẽ trình bày cách tinh chỉnh mô hình BERT đã được đào tạo để thực hiện nhiệm vụ trả lời câu hỏi.

Đầu vào của mô hình BERT là một cặp câu hỏi - đoạn văn chứa câu trả lời. Mô hình BERT sẽ phải trích xuất câu trả lời từ đoạn văn bản đó sau đó trả về khoảng văn bản chứa câu trả lời trong đoạn văn. Hãy xem xét ví dụ bên dưới:

Câu hỏi: *"Hệ thống miễn dịch là gì?"*

Đoạn văn bản: *"Hệ thống miễn dịch là hệ thống gồm nhiều cấu trúc và quá trình sinh học bên trong cơ thể nhằm bảo vệ chống lại bệnh tật. Để hoạt động bình thường, hệ thống miễn dịch phải phát hiện nhiều loại tác nhân, được gọi là mầm bệnh, từ vi rút đến giun ký sinh và phân biệt chúng từ mô khỏe mạnh của chính sinh vật đó."*

Mô hình trích xuất câu trả lời từ đoạn văn, sau đó trả về khoảng văn bản chứa câu trả như sau:

Câu trả lời: *"hệ thống gồm nhiều cấu trúc và quá trình sinh học trong cơ thể giúp bảo vệ chống lại bệnh tật"*

Với câu trả lời đó, mô hình phải tìm được chỉ số bắt đầu và chỉ số kết thúc của câu trả lời nằm trong đoạn văn bản. Trong đoạn văn bản đã cung cấp, câu trả lời bắt đầu từ chỉ số 6 ("hệ") và kết thúc tại chỉ số 27 ("tật"). Để làm được

điều đó, mô hình sử dụng hai vector được gọi là: vector bắt đầu S và vector kết thúc E, giá trị của vector S và E sẽ được điều chỉnh trong quá trình học của mô hình.

Đầu tiên, mô hình tính toán xác suất mỗi token (từ) trong đoạn văn bản có thể là token bắt đầu của câu trả lời. Để tính toán xác suất này, với mỗi token mô hình tính toán tích giữa vector biểu diễn của token thứ i R_i và vector bắt đầu S

$$P_i = \frac{e^{S \cdot R_i}}{\sum_j e^{S \cdot R_j}}$$

Trong đó:

- P_i là giá trị xác suất có thể là token bắt đầu của R_i
- R_i là biểu diễn vector của token thứ i trong đoạn văn sau khi đi qua mô hình BERT được tiền huấn luyện.
- S là vector được tạo ra và được học thông qua mô hình để xác định token bắt đầu của câu trả lời.

Tương tự như việc xác định token bắt đầu của câu trả lời, mô hình tính toán xác suất mỗi token (từ) trong đoạn văn bản có thể là token kết thúc của câu trả lời. Để tính toán xác suất này, với mỗi token mô hình tính toán tích giữa vector biểu diễn của token thứ i R_i và vector kết thúc E

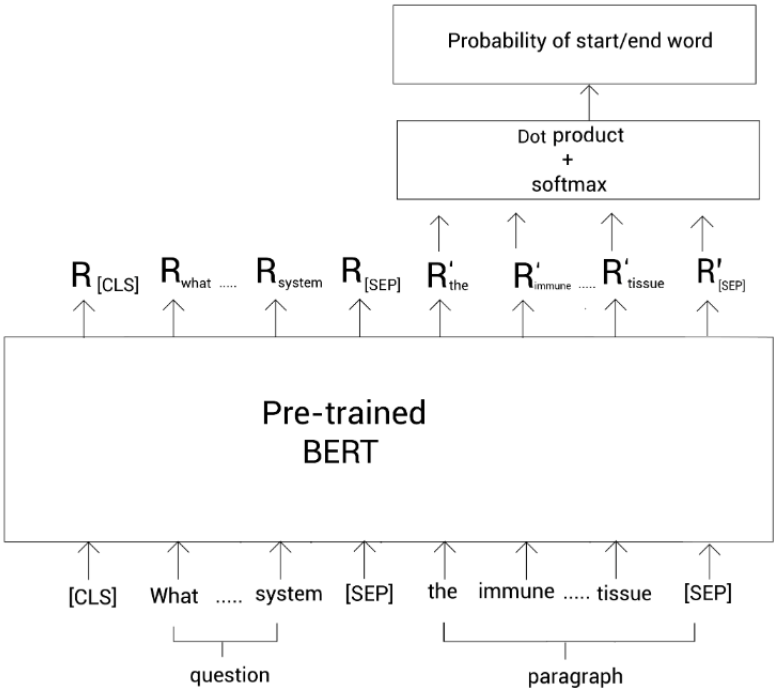
$$P_i = \frac{e^{E \cdot R_i}}{\sum_j e^{E \cdot R_j}}$$

Trong đó:

- P_i là giá trị xác suất có thể là token kết thúc của R_i
- R_i là biểu diễn vector của token thứ i trong đoạn văn sau khi đi qua mô hình BERT được tiền huấn luyện.

- *E* là vector được tạo ra và được học thông qua mô hình để xác định token kết thúc của câu trả lời.

Hình 3.1 mô tả quá trình đưa ra câu trả lời của mô hình BERT được tinh chỉnh cho nhiệm vụ trả lời câu hỏi.



Hình 3.1: Quá trình đưa ra kết quả của mô hình BERT được tinh chỉnh cho nhiệm vụ trả lời câu hỏi

3.3 Phương thức đánh giá mô hình trả lời câu hỏi

Để đánh giá mô hình trả lời câu hỏi, có ba độ đo phổ biến cho nhiệm vụ này là độ chính xác tuyệt đối (Exact Match - EM), F1-score và top-n-accuracy. Bài báo cáo này sẽ sử dụng hai trong ba độ đo đó là EM và F1-score. [13]

- Độ đo chính xác tuyệt đối (EM): : Đối với mỗi cặp câu hỏi - câu trả lời, nếu các ký tự của câu trả lời được dự đoán bởi hệ thống chính xác hoàn toàn với các ký tự của câu trả lời thực tế, EM = 1, nếu ngược lại EM = 0. EM là một chỉ số tuyệt đối, EM = 0 nếu sai lệch chỉ một ký tự.
- Độ đo F1-score được tính thông qua các từ trong câu trả lời của hệ thống so với câu trả lời thực tế (nhãn). F1-score dựa trên số lượng từ phù hợp giữa câu trả lời của hệ thống và câu trả lời thực tế.
  - Precision = (số lượng từ của hệ thống phù hợp với câu trả với thực tế/ (tổng số từ trong câu trả lời của hệ thống)
  - Recall = (số lượng từ phù hợp của hệ thống) / (tổng số từ trong câu trả lời thực tế)
  - F1-score = (2 * Precision * Recall) / (Precision + Recall)

Đối với các câu trả lời như tên riêng, địa chỉ, nơi chốn cần độ chính xác tuyệt đối, độ đo EM cho thấy độ hữu ích của mình. Tuy nhiên với các câu trả lời phức tạp, chứa độ dài lớn rất khó tránh khỏi thiếu sót trong câu trả lời của hệ thống. Để giảm thiểu vấn đề đó và có một điểm số liên tục từ 0 đến 1, F1-score là độ đo thích hợp để giải quyết vấn đề này. Với những lý do trên, trong bài báo cáo này, tôi sẽ sử dụng hai độ do là F1-score và Exact Match để đánh giá mô hình trả lời câu hỏi.

Vào cuối năm 2021, một cuộc thi do CLB Xử lí Ngôn ngữ và Tiếng nói tiếng Việt thuộc Chi hội của Hội Tin học Việt Nam có tên là VLSP 2021 - Vietnamese Machine Reading Comprehension đã được tổ chức. Nội dung của cuộc thi là xây dựng hệ thống để tìm câu trả lời chính xác cho các câu hỏi được đặt ra bằng ngôn ngữ tiếng Việt, từ các tài liệu được cung cấp. Mô hình được cho phép sử dụng trong cuộc thi là mô hình được xây dựng dựa trên cấu trúc của mô hình mBERT(Vietnamese, Base). Một số kết quả của cuộc thi được thể hiện trong hình 3.2

	Ensemble	F1 (%)	EM (%)
Retrospective Reader + XLM-R	x	81.013	71.316
BLANC + XLM-R/SemBERT	x	82.622	73.698
$XLM\text{-}R_{Large}$	x	80.578	70.662
$XLM\text{-}R_{Large}$		79.594	69.092
$PhoBERT_{Large}$+R3F+CS		75.842	63.544
mBERT – baseline		63.031	53.546

Hình 3.2: Một số kết quả từ cuộc thi VLSP 2021 - Vietnamese Machine Reading Comprehension (Nguồn: [7])

3.4 Giới thiệu về mô hình PhoBERT

PhoBERT là một mô hình SOTA (State-of-the-art) đã được huấn luyện trước sử dụng cho tiếng Việt. PhoBERT có hai phiên bản, bao gồm $PhoBERT_{base}$ và

$PhoBERT_{large}$ là mô hình ngôn ngữ lớn được tiền huấn luyện đầu tiên dựa trên nguồn dữ liệu tiếng Việt. Mô hình PHoBERT được nhóm tác giả bao gồm Nguyễn Quốc Đạt và Nguyễn Tuấn Anh, thuộc nhóm nghiên cứu VinAI giới thiệu vào năm 2020 [2]

PhoBERT được xây dựng dựa trên nền tảng cách tiếp cận của RoBERTa -

Bảng 3.3: Thông tin về các tham số ứng với từng phiên bản của mô hình PHOBERT (Nguồn: [2])

Model	Params	Arch	Max length	Pre-training data
vinai/phobert-base-v2	135M	Base	256	20GB của Wikipedia và các bài báo + 120GB từ dữ liệu của OSCAR-2301
vinai/phobert-base	135M	Base	256	20GB của Wikipedia và các bài báo
vinai/phobert-large	370M	Large	256	20GB của Wikipedia và các bài báo

là mô hình được tối ưu trên cơ sở nền tảng của mô hình BERT. RoBERTa là mô hình được giới thiệu vào năm 2019 bởi nhóm tác giả nghiên cứu của FaceBook AI [14]. Mô hình này được xây dựng dựa trên việc cải tiến lại cách tiếp cận của BERT dựa trên các yếu tố như: thời gian huấn luyện, kích thước tập dữ liệu, cải tiến cách tiếp cận của nhiệm vụ xây dựng mô hình mặt nạ (Masked Language Model)

Các thay đổi của mô hình RoBERTa so với BERT bao gồm:

- Huấn luyện mô hình dựa trên tập dữ liệu lớn, với kích thước mỗi batch lớn hơn, thời gian huấn luyện mô hình lâu hơn.
- Loại bỏ đi phần dự đoán câu tiếp theo (Next sentense prediction).
- Huấn luyện mô hình dựa trên những câu văn có độ dài lớn.
- Để tránh che đi cùng một token của một câu trong nhiều lần huấn luyện khác nhau, phương pháp static masking, mô hình RoBERTa sử dụng phương pháp dynamic masking.

PhoBERT sử dụng phương pháp BPE-encoding để phân tách từ. BPE là phương pháp xây dựng tập từ vựng với kích thước nhỏ, được giới thiệu vào năm 2015 bởi nhóm tác giả Rico Sennrich, Barry Haddow và Alexandra Birch [15] với ý tưởng hợp nhất các ký tự xuất hiện liên tiếp thường xuyên trong đoạn văn bản thành một đơn vị từ vựng cho đến khi đạt được kích thước nhất định

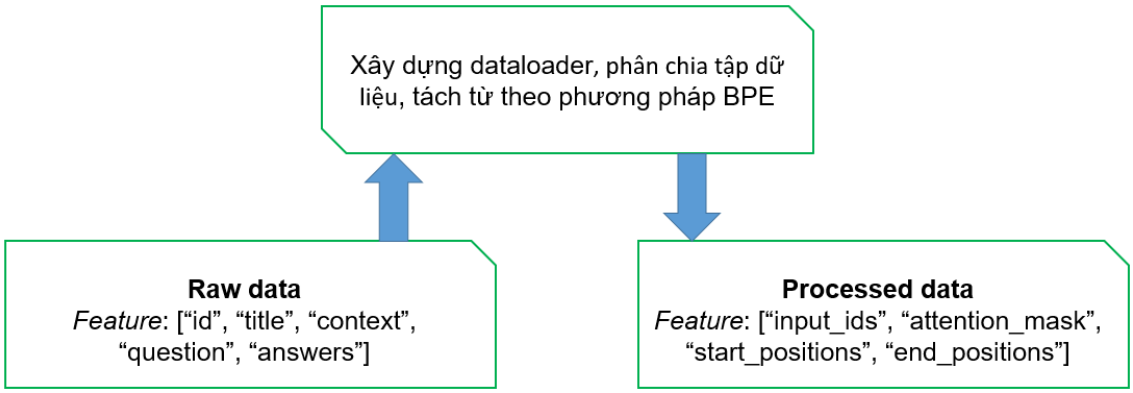
Algorithm 1 Mô tả giải thuật BPE

function BYTE-PAIR ENCODING(String C, number of merges k)
    $V \leftarrow$ all unique character in C
    for $i \leftarrow 1$ to k **do**
        $t_L, t_R \leftarrow$ Most frequent pair of adjacent tokens in C
        $t_{NEW} \leftarrow t_L + t_R$
        $V \leftarrow V + t_{NEW}$
        Replace each occurrence of t_L , t_R in C with t_{NEW}
    end for
    return V
end function

3.5 Xây dựng mô hình

3.5.1 Tiền xử lý dữ liệu

Với dữ liệu ban đầu là một tệp định dạng json, các trường dữ liệu bao gồm: "id", "title", "context", "question", "answers"... được mô tả như trong hình 3.5. Với dữ liệu thô đó, tiến hành xây dựng dataloader để xử lý dữ liệu, tách từ theo phương pháp BPE (được trình bày ở phần thuật toán BPE của mô hình PhoBERT 1), kết quả cuối cùng sẽ thu được một danh sách theo định dạng đầu vào của PhoBERT bao gồm các trường dữ liệu: input_ids, attention_mask, start_positions, end_positions.



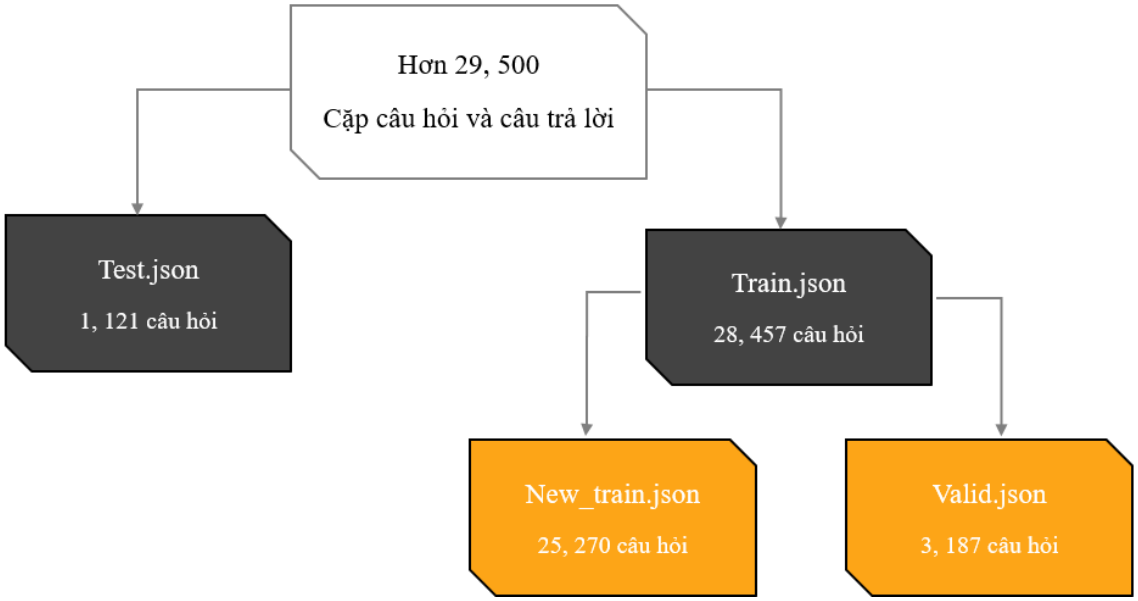
Hình 3.3: Dữ liệu trước khi đưa được xử lý

Với hình 3.3, đầu vào của mô hình PhoBERT yêu cầu bao gồm:

- **input_ids:** Dữ liệu vector sau khi đã chuyển đổi từ dạng text sang dạng

vector, với số chiều của vector là 258 (256 token trong câu văn bản và 2 token được gọi là bắt đầu và kết thúc câu)

- **attention_mask:** Định dạng dưới một vector có chiều dài bằng 258, được đánh số là 1 ở vị trí i nếu tại đó token được mã hóa không phải là padding, ngược lại nhận giá trị là 0.
- **start_positions:** Định dạng bằng một số nguyên, nhận giá trị là chỉ số bắt đầu của câu trả lời trong đoạn văn bản ban đầu (context).
- **end_positions:** Định dạng bằng một số nguyên, nhận giá trị là chỉ số kết thúc của câu trả lời trong đoạn văn bản ban đầu (context).



Hình 3.4: Phân chia dữ liệu cho tập huấn luyện, tập kiểm tra và tập đánh giá

```
{
    "data": [
        {
            "paragraphs": [
                {
                    "context": "Tin tức về chủ đề \" Thủ tướng Ấn Độ Narendra Modi \"",
                    "qas": [
                        {
                            "answers": [
                                {
                                    "answer_start": 36,
                                    "text": "Narendra Modi"
                                }
                            ],
                            "id": "4e03f08a78b744a48f0f380441a227f4",
                            "question": "Ai là thủ tướng Ấn Độ hiện tại",
                            "is_impossible": false,
                            "plausible_answers": []
                        }
                    ]
                }
            ],
            "title": "Ai là thủ tướng Ấn Độ hiện tại"
        }
    ]
}
```


Hình 3.5: Dữ liệu thô dưới dạng tệp json

3.5.2 Thiết lập thí nghiệm, kết quả thu được

Định nghĩa các tham số:

- **args_pretrained_model**: Mô hình PhoBERT pre-train được sử dụng
- **Epochs**: Số lần sử dụng toàn bộ dữ liệu trong quá trình huấn luyện mô hình.
- **Learning rate**: Tốc độ học của mô hình trong quá trình huấn luyện.
- **Max length** Độ dài tối đa của vector sau khi chuyển đổi cặp câu hỏi - đoạn văn bản thành vector.
- **Stride**: Số lượng từ trùng nhau giữa 2 đoạn văn bản liên tiếp sau khi phân đoạn.
- **Batch size**: Số lượng dữ liệu được đưa vào cùng 1 lúc trong quá trình huấn

luyện

Thử nghiệm xây dựng mô hình với pre-train model PhoBERT

Các tham số thiết lập thí nghiệm với mô hình phobert-base được mô tả trong bảng 3.4

Bảng 3.4: Thông tin tham số được thiết lập cho việc fine-tuning lại mô hình PhoBERT lần 1

Max length	Strike	Learning rate	Batch size	Epoch	Train Size	Valid Size
256	128	2.10^{-5}	10	20	25270	3187

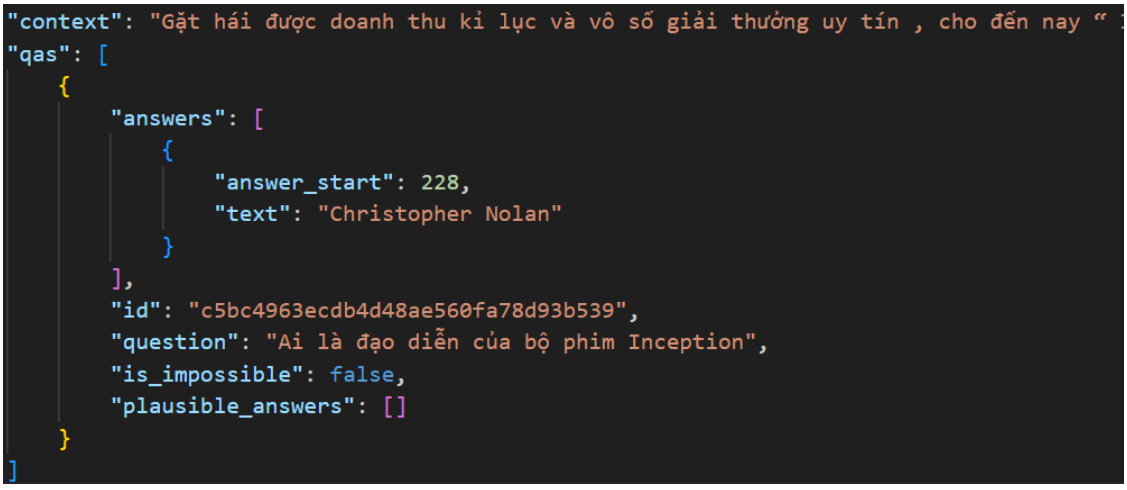
Bảng 3.5: Thông tin kết quả tốt nhất mà mô hình thu được với việc thiết lập các tham số tại bảng 3.4

Mô hình	EM	F1
PhoBERT-base	53.8473	77.9264

3.5.2.1 Kết quả thu được

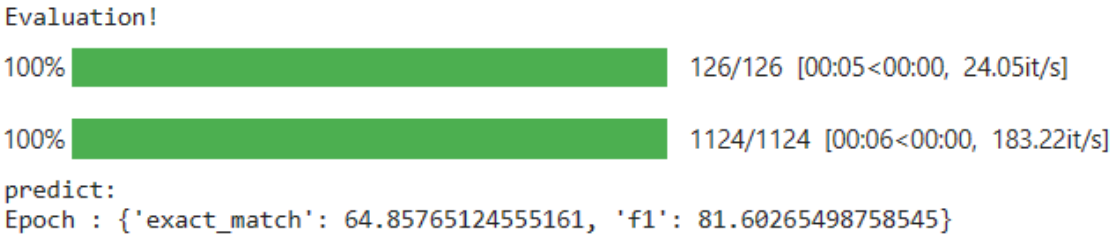
Kết quả của mô hình với tập dữ liệu thu thập được

Tập dữ liệu bao gồm 1123 câu hỏi liên quan đến các lĩnh vực khác nhau.



Hình 3.6: Hình ảnh mô tả một cặp question- context trong tập dữ liệu thô thu thập dưới dạng tệp json để sử dụng đánh giá mô hình

Kết quả mô hình thu được



Hình 3.7: Kết quả đánh giá mô hình huấn luyện dựa trên tập dữ liệu thu thập được

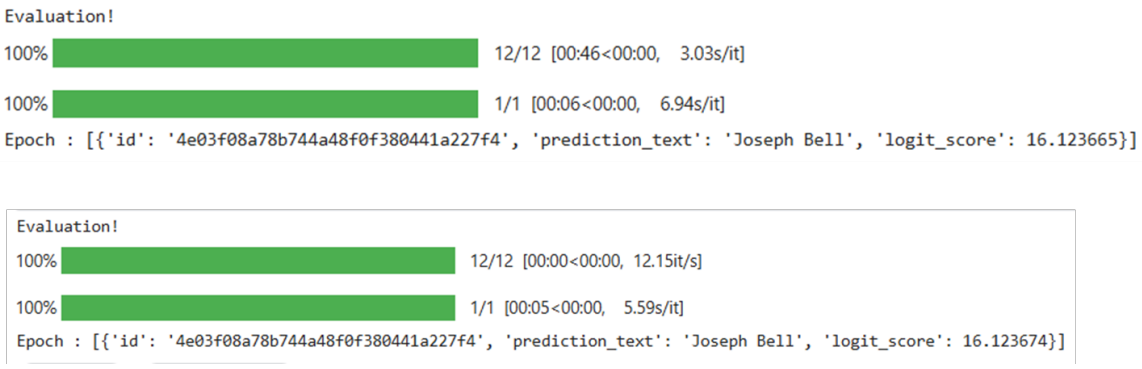
3.5.2.2 Đánh giá mô hình

Thời gian chạy của mô hình

Thí nghiệm sau được tôi thực hiện trên môi trường làm việc của Kaggle trong hai điều kiện khác nhau, bao gồm: thực thi mô hình với CPU và thực thi mô hình với GPU P100 của Kagle. Kết quả thu được trong bảng 3.6

Bảng 3.6: Chi tiết kết quả thí nghiệm thời gian thực thi của mô hình giữa CPU và GPU P100 của Kaggle

Điều kiện đánh giá	**Nội dung đoạn văn**	**Độ dài đoạn văn**	**Câu hỏi**	**Thời gian phản hồi**
Sử dụng CPU	Mô tả về nhân vật Sherlock Holmes tại trang Wikipedia	9334 từ	Ai là nhân vật truyền cảm hứng cho Arthur Conan Doyle tạo ra Sherlock Holmes	52 giây
Sử dụng GPU P100 (Kaggle)	Mô tả về nhân vật Sherlock Holmes tại trang Wikipedia	9334 từ	Ai là nhân vật truyền cảm hứng cho Arthur Conan Doyle tạo ra Sherlock Holmes	6 giây



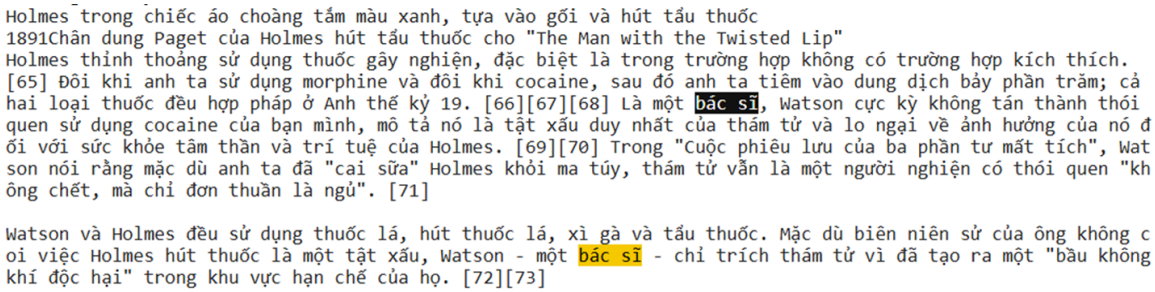
Hình 3.8: Hình ảnh thể hiện thời gian thực thi của hệ thống khi sử dụng CPU so với khi sử dụng GPU P100

Kết quả trả về của mô hình

Mô hình PhoBERT hoạt động dựa trên việc trích xuất nội dung chính có

trong câu hỏi được cung cấp. Các trường hợp lỗi của mô hình bao gồm:

- Mô hình phản hồi sai đối với các câu hỏi chứa quá ít thông tin liên quan đến nội dung có trong đoạn văn bản.
- Mô hình phản hồi sai đối với các câu hỏi không chứa thông tin trong đoạn văn bản được cung cấp.
- Mô hình phản hồi sai đối với các câu hỏi dài. Câu hỏi dài dẫn đến việc trích xuất dư thừa hoặc không đầy đủ các ý chính, các ngữ cảnh trong câu.



Hình 3.9: Hình ảnh mô tả đoạn văn chứa nội dung câu trả lời đến câu hỏi mô hình phản hồi chính xác

Trong hình 3.9, đoạn văn bản đề cập đến việc Watson là một bác sĩ, tuy nhiên không đề cập đến từ "nghề nghiệp"trong đoạn văn này. Vì vậy khi đặt hai câu hỏi liên quan nghề nghiệp của Watson, hệ thống sẽ phản hồi theo hai cách khác nhau:

Bảng 3.7: Thông tin phản hồi khi đặt hai câu hỏi khác nhau liên quan đến nghề nghiệp của Watson

Câu hỏi	**Câu trả lời phản hồi**
Watson là ai	bác sĩ
Nghề nghiệp của Watson là gì	nuôi ong

Trong trường hợp câu hỏi "Watson là ai"mô hình có thể trả lời được vì trong đoạn văn trên có nội dung liên quan trong câu "Là một bác sĩ, Watson...". Tuy nhiên, trong câu hỏi thứ hai "Nghề nghiệp của Watson là gì", mô hình phản hồi sai vì ngữ cảnh của câu hỏi bị ảnh hưởng bởi từ "nghề nghiệp", hình 3.10 mô tả đoạn văn chứa nội dung liên quan đến từ "nghề nghiệp"

Trong His Last Bow, độc giả được kể rằng Holmes đã nghỉ hưu tại một trang trại nhỏ ở Sussex Downs và lấy nghề nuôi ong làm nghề chính của mình. [51] Động thái này không xác định niên đại chính xác, nhưng có thể được cho là không muộn hơn năm 1904 (vì nó được nhắc đến hồi tưởng trong "Cuộc phiêu lưu của vết bẩn thứ hai", được xuất bản lần đầu tiên vào năm đó). [52] Câu chuyện kể về Holmes và Watson sắp nghỉ hưu để hỗ trợ nỗ lực chiến tranh của Anh. Chỉ có một cuộc phiêu lưu khác, "Cuộc phiêu lưu của Bờm sư tử", diễn ra trong thời gian thám tử nghỉ hưu. [53]

Hình 3.10: Hình ảnh mô tả đoạn văn chứa nội dung câu trả lời đối với câu hỏi mà mô hình phản hồi sai

3.5.2.3 Thảo luận

Từ kết quả thực nghiệm của mô hình PhoBERT, kết quả cho thấy rằng mô hình PhoBERT cho kết quả tốt đối với tiếng Việt, kết quả thu được EM = 53.8473 và F1-Score = 77.9264 đối với tập dữ liệu UIT-ViQuAD.

Tuy nhiên mô hình PhoBERT vẫn gặp phải vấn đề trong việc trả lời các câu hỏi dài hoặc các câu hỏi chứa không đủ dữ kiện có trong đoạn văn. Nguyên nhân lý giải cho các trường hợp này bởi vì mô hình PhoBERT thực hiện mã hóa thông tin ngữ cảnh theo hai chiều của câu. Đối với các câu hỏi dài, hoặc các câu hỏi không đủ dữ kiện trong đoạn văn dẫn đến việc biểu diễn thông tin ngữ cảnh của mô hình bị dư thừa hoặc thiếu thông tin.

Mặc khác, mô hình PhoBERT chưa thể nắm bắt được hết các từ quan trọng ở trong câu. Giải pháp cho trường hợp này được đưa ra là tìm cách để đánh trọng số cho các từ trước khi đưa vào mô hình PhoBERT bằng các phương pháp như POS Tagging, phân loại câu hỏi...

3.6 Tiểu kết chương 3

Nội dung chương 3 này mô tả quá trình huấn luyện và tinh chỉnh mô hình PhoBERT cho nhiệm vụ trả lời câu hỏi. Dữ liệu được sử dụng để huấn luyện mô hình là tập dữ liệu UIT-ViQuAD. Phương pháp đánh giá mô hình trong nhiệm vụ trả lời câu hỏi được sử dụng phổ biến là độ chính xác tuyệt đối (Exact Match) và F1-Score. Mô hình PhoBERT được sử dụng có cấu hình là PhoBERT_base, kết quả thu được với tập dữ liệu UIT-ViQuAD khá cao và có thể áp dụng được trong hệ thống thực tế. Tuy nhiên chương 3 cũng đã trình bày ra các lỗi mà mô hình gặp phải và sự phụ thuộc nhiều của thời gian phản hồi mô hình vào cấu hình GPU của hệ thống

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Kết luận

Trong khóa luận này, tôi đã xây dựng mô hình trả lời câu hỏi dựa trên trích xuất nội dung của đoạn văn bản được cung cấp. Mô hình được huấn luyện dựa trên tập dữ liệu UIT-ViQuAD được giới thiệu bởi nhóm tác giả đến từ Trường Đại học Công nghệ Thông tin, Đại học Quốc gia Thành phố Hồ Chí Minh với kích thước tập dữ liệu bao gồm 25270 cặp câu hỏi và đoạn văn bản cho quá trình huấn luyện và 3 187 cặp câu hỏi, đoạn văn bản cho quá trình đánh giá. Kết quả huấn luyện của mô hình PhoBERT-base thu được với tập đánh giá là EM = 53.8473 và F1-Score = 77.9264.

Từ quá trình tìm hiểu các mô hình Transformer và mô hình BERT, tôi nhận thấy rằng các mô hình này mang lại những tiến bộ đáng kể trong việc mã hóa không gian vector của các từ vựng. Cách mã hoá này cho phép mô hình linh hoạt hơn trong quá trình huấn luyện thông qua từ và ngữ cảnh của các từ trong câu. Vì mô hình BERT mã hóa các từ thành vector theo biểu diễn ngữ cảnh của chúng do đó cho kết quả tốt hơn so với các mô hình mạng nơ-ron hồi quy truyền thống.

Thông qua quá trình nghiên cứu, tôi đã nắm vững các nguyên lý hoạt động của mô hình BERT và ứng dụng của mô hình này. Khác với mô hình Transformer, mô hình BERT chỉ tồn lại lớp Mã hóa Encoder- tác dụng của lớp Mã hóa này nhằm biểu diễn một từ đầu vào thành một vector được biểu diễn theo ngữ cảnh của các từ khác trong câu.

Kết quả thu được từ quá trình thực nghiệm cho thấy rằng hệ thống trả lời câu hỏi trích xuất câu trả lời dược xây dựng từ mô hình PhoBERT cho kết quả

tốt đối với tiếng Việt. Tuy nhiên mô hình vẫn gặp phải vấn đề về mặt thời gian phản hồi trong quá trình xử lý với dữ liệu đầu vào có kích thước lớn, phụ thuộc nhiều vào GPU của hệ thống. Mô hình phản hồi sai đối với các câu hỏi cung cấp không đầy đủ ngữ cảnh hoặc các câu hỏi có kích thước lớn, dẫn đến việc trích xuất thiếu các ý chính, các ngữ cảnh của câu.

Hướng phát triển

Hướng phát triển trong tương lai của đề tài có thể tập trung vào các khía cạnh sau đây để nâng cao hiệu quả và khả năng ứng dụng của mô hình PhoBERT đối với việc phát triển hệ thống trả lời câu hỏi cho tiếng việt:

Thực hiện gán nhãn các từ loại (POS Tagging) giúp cho việc phân tích câu hỏi, tìm ra những ý chính có trong đoạn văn bản trở nên dễ dàng, đánh trọng số lên các từ để tăng độ chính xác của mô hình.

Sử dụng thuật toán BM25 [16] để có thể tìm kiếm nội dung liên quan đến câu hỏi cung cấp nhanh hơn. Ngoài ra, việc sử dụng thuật toán BM25 còn mở ra khả năng huấn luyện mô hình cho việc trả lời câu hỏi trong một nguồn tri thức giới hạn nhất định [17].

Tài liệu tham khảo

[1] Anh Gia Tuan Nguyen Ngan Luu Thuy Nguyen Kiet Van Nguyen, Duc Vu Nguyen. A vietnamese dataset for evaluating machine reading comprehension, 2020.

[2] Anh Tuan Nguyen Dat Quoc Nguyen. Phobert: Pre-trained language models for vietnamese. 2020.

[3] Billy Wan Dapeng Hong. Attention-based recurrent neural networks for question answering. 2017.

[4] Peter Ondruska Mohit Iyyer James Bradbury Ishaan Gulrajani Victor Zhong Romain Paulus Richard Socher Ankit Kumar, Ozan Irsoy. Ask me anything: Dynamic memory networks for natural language processing, 2015.

[5] Niki Parmar Jakob Uszkoreit Llion Jones Aidan N. Gomez Łukasz Kaiser Illia Polosukhin Ashish Vaswani, NoamShazeer. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017)*, 2017.

[6] Kenton Lee Kristina Toutanova Jacob Devlin, Ming-Wei Chang. Bert: Pre-training of deep bidirectional transformers for language understanding. 2018.

[7] *VLSP 2021 - Vietnamese Machine Reading Comprehension Result (https://aihub.ml/competitions/public_submissions/35)*.

[8] Alice K. Wolf Carol Chomsky Bert F. Green, Jr. and Kenneth Laughery Lincoln Laboratory. Baseball: an automatic question-answerer. 1961.

[9] Richard Socher Jeffrey Pennington and Christopher D. Manning. Glove: Global vectors for word representation. in empirical methods in natural language processing (emnlp), pages 1532- 1543, 2014.

[10] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[11] Mohit Iyyer Matt Gardner Christopher Clark Kenton Lee Matthew Peters, Mark Neumann and Luke Zettlemoyer. Deep contextualized word representations. *In NAACL*, 2018.

[12] Tim Salimans Ilya Sutskever Alec Radford, Karthik Narasimhan. Improving language understanding by generative pre-training, 2018.

[13] Duc Vu Nguyen Anh Gia-Tuan Nguyen Ngan Luu-Thuy Kiet Van Nguyen, Tin Huynh Van. New vietnamese corpus for machine reading comprehension. *Health News*, 2021.

[14] Naman Goyal Jingfei Du Mandar Joshi Danqi Chen Omer Levy Mike Lewis Luke Zettlemoyer Veselin Stoyanov Yinhan Liu, Myle Ott. Roberta: A robustly optimized bert pretraining approach. 2019.

[15] Alexandra Birch Rico Sennrich, Barry Haddow. Neural machine translation of rare words with subword units. 2015.

[16] Hugo Zaragoza Stephen E. Robertson. The probabilistic relevance frame-

work: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 2009.

[17] Đặng Văn Nghiêm Trần Thị Minh Khoa Đặng Thị Phúc, Nguyễn Thanh Long. Xây dựng hệ thống tự động giải đáp thắc mắc về các quy định học tập tại trường Đại học công nghiệp bằng kỹ thuật học sâu. 2023.