ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BÁO CÁO MÔN HỌC
Project III

Đề tài

Bài toán Question Answering trong cuộc thi Zalo AI Challenge 2019

Sinh viên: Hồ Sỹ Thế - 20200614
Lớp: Khoa học Máy tính 04 - K65
GVHD: PGS.TS. Lê Thanh Hương

Hà Nội, Ngày 24 tháng 1 năm 2024

MỞ ĐẦU

Ngày nay, Trí tuệ nhân tạo là cụm từ không con xa lạ với bất cứ ai trong chúng ta. Công nghệ trí tuệ nhân tạo đã bước ra ngoài đời thực và giúp ích cho con người trong rất nhiều công việc, từ thay thế những công việc đơn giản cho tới cả tạo ra các tác phẩm nghệ thuật phức tạp. Có thể nói đến việc AI có thể giúp ích cho con người trong việc giải quyết vấn đề kiến thức của con người, tiêu biểu là việc khả năng trả lời câu hỏi của AI. Nhận thấy tiềm năng ứng dụng này, em quyết định thực hiện đề tài 'Giải quyết bài toán Question Answering'.

Báo cáo này cung cấp cái nhìn tổng quan toàn diện về phương pháp của em đã thực hiện đề thực hiện đề tài. Đề tài đặt ra không chỉ là cơ hội để thử nghiệm kiến thức của bản thân mà còn là cơ hội để em áp dụng lý thuyết vào thực tế, từ đó xây dựng kỹ năng thực hành và tư duy giải quyết vấn đề trong lĩnh vực học máy nói chung và NLP nói riêng.

Em xin gửi lời cảm ơn chân thành đến cô PGS.TS. Lê Thanh Hương đã góp ý và đưa ra các gợi ý các phương pháp giúp em thực hiện được đề tài này. Em xin chân thành cảm ơn!

MỤC LỤC

1. Giới thiệu bài toán

1.1 Đặt vấn đề

Đi cùng với sự phát triển của trí tuệ loài người là việc con người đặt ra nhiều câu hỏi. Trả lời câu hỏi là một khía cạnh cốt lõi của trí tuệ con người. Từ việc tìm hiểu kiến thức cho đến việc tham gia các cuộc thảo luận, khả năng trả lời câu hỏi chính là một trong những yếu tố quan trọng giúp chúng ta tiếp cận và hiểu rõ hơn về thế giới xung quanh. Trong thời đại số hóa hiện đại, việc tự động hóa quá trình này thông qua máy móc là một bước tiến lớn, mang lại nhiều tiện ích và ứng dụng trong thực tế. Bài toán Question Answering (QA) trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP) đặt ra mục tiêu chính là tạo ra hệ thống có khả năng trả lời câu hỏi một cách chính xác, dựa trên nguồn dữ liệu hoặc tri thức đã có.

Bài toán QA không chỉ là một lĩnh vực nghiên cứu thuần túy, mà còn là nền tảng cho hàng loạt ứng dụng thực tế. Trợ lý ảo như Siri, Google Assistant, Alexa đều dựa trên nền tảng của QA, giúp người dùng truy vấn thông tin, thiết lập lịch trình, hoặc thậm chí điều khiển các thiết bị thông minh. Bên cạnh đó, các hệ thống hỗ trợ khách hàng trực tuyến, dịch vụ tìm kiếm thông tin y học, và nhiều ứng dụng khác đều được hỗ trợ bởi khả năng trả lời câu hỏi tự động. Điểm đặc biệt là với sự phát triển của công nghệ và thuật toán, chất lượng và độ chính xác của các hệ thống QA ngày càng được nâng cao, mở ra nhiều khả năng mới trong tương lai.

Nhận thấy bài toán này có tính hấp dẫn và thực tế cao, nên em đề xuất xây dựng, tìm ra một mô hình học sâu có thể giải quyết hiểu quả bài toán QA.

1.2 Mô hình bài toán

Bài toán QA được phát biểu như sau: Cho câu hỏi và đoạn tài liệu, nhiệm vụ bài toán là cho biết câu trả lời cho câu hỏi đó có nằm trong đoạn tài liệu đã cho hay không.

- Đầu vào: Câu hỏi (hay còn gọi là truy vấn - query) và đoạn tài liệu (document).
- Đầu ra: Nhãn **true** nếu câu trả lời có trong đoạn văn, **false** nếu ngược lại.

Để đánh giá một phương pháp tốt hay tồi hơn các phương pháp khác, tiêu chí đánh giá được sử dụng là điểm F1 macro.

1.3 Định hướng phương pháp

Sau khi phân tích dữ liệu và tham khảo từ nhiều nguồn khác, các phương pháp được cân nhắc phải phù hợp với dữ liệu và phù hợp với khả năng thực hiện (cài đặt, tài nguyên

huấn luyện). Cuối cùng, các phương pháp được đề xuất:

- Baseline: BERT - Phương pháp đã được thực hiện trong học phần 'Thực tập Kỹ thuật'.
- BERT/RoBERTa + Question Classification
- BERT/RoBERTa/DistilBERT + Labeling
- Seq2Seq: ViT5

Mục tiêu của dự án: Cải thiện độ chính xác so với giải pháp **Baseline** của bài toán Question Answering trên bộ dữ liệu của cuộc thi Zalo AI Challenge 2019.

1.4 Thông tin dự án

Mã nguồn dự án được lưu trữ tại GitHub cá nhân: `https://github.com/hsthe29/ProjectIII-QuestionAnswering`

2. Dữ liệu

Dữ liệu là đối tượng rất quan trọng trong việc huấn luyện các mô hình, kết quả của mô hình phụ thuộc rất lớn vào dữ liệu đầu vào. Càng nhiều dữ liệu với độ đa dạng và phù hợp cao với ngữ cảnh thì mô hình xử lý càng tốt và cho dự đoán càng chính xác. Trong bài toán này, các bộ dữ liệu được sử dụng được liệt kê dưới đây:

2.1 Zalo AI Challenge 2019 Vietnamese Wiki Question Answering (ZaloQA)

Dataset được lấy từ cuộc thi Zalo AI Challenge 2019 track Vietnamese Wiki Question Answering. Dataset gồm 2 file:

- train.json: 18108 mẫu dữ liệu
- test.json: 501 mẫu dữ liệu

Mỗi mẫu dữ liệu bao gồm các trường

- *id*: ID của mẫu dữ liệu
- *question*: Câu hỏi
- *title*: Tiêu đề của mẫu dữ liệu
- *text*: Đoạn tài liệu
- *label*: Nhãn cho biết xem trong *text* có câu trả lời cho *question* không.

Để thuận tiện hơn cho việc xử lý dữ liệu cho việc huấn luyện, cần chuyển định dạng file từ json sang tsv (Tab Separated Values). Sau khi chuyển, xem trước tập dữ liệu:

question	title	document	label
Quang Hải giành được chức vô địch U21 quốc gia năm bao nhiêu tuổi	Nguyễn Quang Hải (sinh 1997)	Năm 2013 , Nguyễn Quang Hải giành chức vô địch U21 quốc gia 2013 cùng với đội trẻ Hà Nội T&T và tạo nên cú sốc khi trở thành cầu thủ 16 tuổi đầu tiên giành được danh hiệu vô địch U21 quốc gia	true
Tên chính thức của Na Uy là gì	Greenland	Năm 1261 , Greenland trở thành thuộc địa của Vương quốc Na Uy . Tới lượt mình Na Uy lại tham gia vào Liên minh Kalmar năm 1397 và sau này là Liên minh Đan Mạch-Na Uy	false
Bảng Anh là gì	Bảng xếp hạng đại học thế giới Quacquarelli Symonds	Bảng xếp hạng đại học thế giới Quacquarelli Symonds , tiếng Anh : QS World University Rankings , là bảng xếp hạng thường niên về thứ hạng các trường đại học trên thế giới của tổ chức giáo dục Quacquarelli Symonds (QS) - Anh Quốc	false

2.2 The Text REtrieval Conference Question Classification (TREC)

Bộ dữ liệu Text REtrieval Conference (TREC) Question Classification chứa 5500 câu hỏi được gắn nhãn trong tập huấn luyện và 500 câu hỏi khác cho tập kiểm tra.

Bộ dữ liệu có 6 nhãn lớp thô và 50 nhãn lớp mịn. Độ dài trung bình của mỗi câu là 10 (từ), số lượng từ vựng là khoảng 8700.

Dữ liệu được thu thập từ bốn nguồn: 4.500 câu hỏi tiếng Anh do USC xuất bản (Hovy et al., 2001), khoảng 500 câu hỏi được xây dựng thủ công cho một số lớp hiếm, 894 câu hỏi TREC 8 và TREC 9 và 500 câu hỏi từ TREC 10 phục vụ cho như tập kiểm tra. Những câu hỏi này được dán nhãn thủ công.

Bộ dữ liệu gốc ở tiếng Anh, để áp dụng cho tiếng Việt ta cần chuyển nó sang tiếng

Việt. Việc này được thực hiện bằng Google Translate API.

Mỗi mẫu dữ liệu bao gồm các trường:

- *text*
- *coarse_label*
- *fine_label*

Link: `https://huggingface.co/datasets/trec`

2.3 SQuAD 2.0

Stanford Question Answering Dataset (SQuAD) là một bộ dữ liệu đọc hiểu, bao gồm các câu hỏi được đặt ra trên một tập hợp các bài viết trên Wikipedia, trong đó câu trả lời cho mỗi câu hỏi là một đoạn văn bản từ tài liệu tương ứng hoặc câu hỏi không thể trả lời được.

SQuAD 2.0 kết hợp 100.000 câu hỏi trong SQuAD 1.1 với hơn 50.000 câu hỏi không thể trả lời. Để hoạt động tốt trên SQuAD 2.0, hệ thống không chỉ phải trả lời các câu hỏi khi có thể mà còn phải xác định khi nào đoạn văn không có câu trả lời nào và tránh trả lời.

Bộ dữ liệu SQuAD 2.0 dùng tiếng Anh, để có thể áp dụng vào tiếng Việt thì ta cần phải dịch bộ dữ liệu từ tiếng Anh sang tiếng Việt. Việc dịch ngôn ngữ được thực hiện bằng cách sử dụng Google Translate API.

Mỗi mẫu dữ liệu bao gồm các trường:

- *text*
- *coarse_label*
- *fine_label*

Link: `https://rajpurkar.github.io/SQuAD-explorer/`

2.4 ViQuAD 2.0

Bộ dữ liệu mới dành cho ngôn ngữ tiếng Việt để đánh giá các mô hình MRC. Bộ dữ liệu này bao gồm hơn 23.000 cặp câu hỏi-câu trả lời do con người tạo ra dựa trên 5.109 đoạn của 174 bài viết tiếng Việt từ Wikipedia.

Cấu trúc bộ dữ liệu giống với bộ SQuAD 2.0.

Link: `https://sites.google.com/uit.edu.vn/kietnv/datasets`

2.5 Facebook MLQA

MLQA (MultiLingual Question Answering) là bộ dữ liệu chuẩn để đánh giá hiệu suất trả lời câu hỏi đa ngôn ngữ. MLQA bao gồm hơn 5K phiên bản QA trích xuất (12K bằng tiếng Anh) ở định dạng SQuAD bằng bảy ngôn ngữ - tiếng Anh, tiếng Ả Rập, tiếng Đức, tiếng Tây Ban Nha, tiếng Hindi, tiếng Việt và tiếng Trung giản thể. MLQA có tính song song cao, với các phiên bản QA song song trung bình giữa 4 ngôn ngữ khác nhau.

Cấu trúc bộ dữ liệu giống với bộ SQuAD 2.0.

Bảng sau đây cho biết lượng dữ liệu trong mỗi ngôn ngữ:

Split	**en**	**de**	**es**	**ar**	**zh**	**vi**	**hi**
dev	1148	512	500	517	504	511	507
test	11590	4517	5254	5335	5137	5495	4918

Link: `https://github.com/facebookresearch/MLQA`

3. Các phương pháp thực hiện

3.1 Baseline: BERT

3.1.1 Phương pháp

Với mỗi cặp query Q và document D có nhãn là **Label** sẽ được nối lại thành một chuỗi đầu vào là:

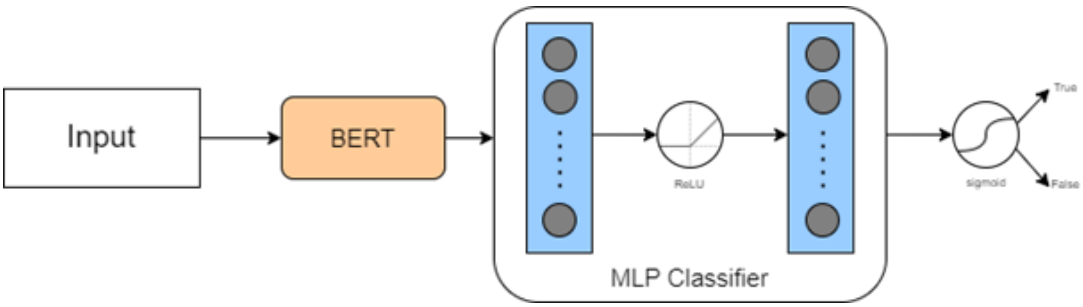
$$\textbf{Input} = \text{Encode}\{\textbf{[CLS]}\ Q\ \textbf{[SEP]}\ D\ \textbf{[SEP]}\}$$

Trong đó, toán từ Encode là toán tử mã hóa câu đầu vào thành các token ids. Vì mô hình BERT nhận đầu vào với độ dài tối đa là 512 và không phải **Input** nào cũng thỏa mãn điều kiện đó. Do đó, kích thước tối đa của query Q được đặt thành $L_Q = 64$, và những token thừa (bên phải) của query Q và document D sẽ được loại bỏ.

Sau đó, **Input** sẽ được đưa vào một mô hình pretrained-BERT cho ra các vector đặc trưng d-chiều của các token trong ngữ cảnh câu, kí hiệu là BERT(**Input**). Tuy nhiên không cần dùng toàn bộ câu, chỉ cần vector đặc trưng của token **[CLS]** cũng đã mang đủ thông tin của đầu vào để dùng cho việc phân loại (do tác vụ tiền huấn luyện NSP của BERT). Thêm nữa, khi sử dụng BERT để trích xuất vector embedding, một số người chọn lấy vector từ tầng cuối cùng của BERT vì nó chứa thông tin ngữ cảnh phong phú nhất. Tuy nhiên, một số nghiên cứu đã chỉ ra rằng việc sử dụng vector embedding từ các tầng gần cuối thường mang lại kết quả tốt hơn. Lý do cho điều này có thể là do tầng cuối cùng của BERT tập trung vào việc học các đặc trưng phức tạp và cụ thể cho các tác vụ huấn luyện là MLM và NSP, trong khi các tầng gần cuối hơn tập trung vào việc học các đặc trưng chung và bao quát hơn. Do đó, vector embedding từ các tầng gần cuối có thể mang lại một cái nhìn tổng quát hơn về ngữ nghĩa của từ, trong khi vector embedding từ tầng cuối cùng có thể quá tập trung vào các chi tiết cụ thể. Vì vậy, vector đặc trưng sau khi đi qua BERT là vector đặc trưng của token **[CLS]** của 2 tầng cuối cùng, ký hiệu là CLS2(BERT(**Input**)).

Lớp phân loại là mạng Multilayer Perceptron (MLP) bao gồm 2 lớp Fully Connected. Đầu ra sau khi đi qua lớp mạng này là:

$$\textbf{Output} = \text{MLP}(\text{CLS2}(\text{BERT}(\textbf{Input})))$$



Hình 3.1 Baseline-Finetuning BERT

3.1.2 Huấn luyện

Hàm mất mát Cross Entropy Loss:

$$\mathscr{L}_{\textbf{Input}} = -\sum_{c=1}^{C} \textbf{Label}_c \log(softmax(\textbf{Output})_c)$$

Tham số huấn luyện:

- Epoch: 10
- Max learning rate: 2e-5
- Batch size: 16
- AdamW Optimizer
- Warmup Linear Scheduler (warmup steps: 10%)

Accelerator: GPU P100 (Kaggle).

Thời gian huấn luyện: 7 GPU hours.

3.2 BERT/RoBERTa + Question Classification

3.2.1 Phương pháp

Huấn luyện thêm tác vụ phụ *Phân loại câu hỏi (QC)* bên cạnh tác vụ chính của bài toán QA. Với ý tưởng khi dùng thêm tác vụ QC là để:

- Hướng dẫn tìm kiếm: Biết trước loại câu hỏi giúp mô hình xác định tốt hơn ngữ cảnh giữa câu hỏi và đoạn tài liệu từ đó giúp đưa ra kết quả chính xác hơn.
- Phân tích câu hỏi và câu trả lời: Sau khi biết loại câu hỏi, mô hình có thể tùy chỉnh cách phân tích và đánh giá mối quan hệ giữa câu hỏi và câu trả lời. Ví dụ: một câu hỏi được phân loại là "Khi nào" sẽ yêu cầu câu trả lời chứa thông tin về thời gian, trong khi danh mục "Ai" sẽ yêu cầu tên hoặc danh tính của một cá nhân, ...

Để kết hợp 2 tác vụ, bên cạnh bộ dữ liệu ZaloQA thì phương pháp này sẽ dùng thêm bộ dữ liệu TREC. Quá trình huấn luyện mô hình sẽ gồm 2 giai đoạn:

- Giai đoạn 1: Huấn luyện trên tác vụ QC. Mục tiêu dự đoán nhãn cho câu hỏi:

  $$\textbf{Input} = \{\textbf{[CLS]}\ Q\ \textbf{[SEP]}\}$$

  Nhãn dự đoán:

  $$L = \text{QCModel}\,(\textbf{Input})$$

  Đầu vào của tác vụ QC là câu hỏi, và đầu ra là nhãn của câu hỏi được mã hóa số.

- Giai đoạn 2: Huấn luyện tác vụ QA.

  Với mỗi query Q và document D sẽ dùng mô hình đã huấn luyện QC để dự đoán nhãn của query Q là L. Sau đó kết hợp đầu vào dưới dạng:

  $$\textbf{Input} = \{\textbf{[CLS]}\ L\ \textbf{[SEP]}\ Q\ \textbf{[SEP]}\ D\ \textbf{[SEP]}\}$$

  Sau đó đưa vào mô hình QA để huấn luyện.

Mô hình cho cả 2 pha đều có kiến trúc giống nhau và giống như phương pháp **Baseline**, chỉ có mạng MLP phân loại là khác nhau về số neuron đầu ra tương ứng với số nhãn của tác vụ. Ở tác vụ QA, đầu ra với mỗi ví dụ gồm 2 số tương ứng 2 nhãn **false** và **true**, còn ở tác vụ QC, đầu ra với mỗi vị dụ bằng với số lượng nhãn trong tập dữ liệu TREC.

3.2.2 Huấn luyện

Hàm mất mát Cross Entropy Loss:

$$\mathscr{L}_{\textbf{Input}} = -\sum_{c=1}^{C} \textbf{Label}_c \log(softmax(\textbf{Output})_c)$$

Tham số huấn luyện QC:

- Epoch: 6
- Max learning rate: 1e-5
- Batch size: 16
- AdamW Optimizer
- Warmup Linear Scheduler (warmup steps: 10%)

Tham số huấn luyện QA:

- Epoch: 10
- Max learning rate: 2e-5
- Batch size: 16
- AdamW Optimizer
- Warmup Linear Scheduler (warmup steps: 10%)

Accelerator: GPU P100 (Kaggle).

Thời gian huấn luyện: 10 GPU hours.

3.3 BERT/RoBERTa/DistilBERT + Labeling

3.3.1 Phương pháp

Phương pháp này sử dụng kết hợp bộ dữ liệu ZaloQA và ViQuAD 2.0. Và vì bộ ViQuAD 2.0 có nhiều mẫu dữ liệu có kích thước dài hơn mô hình cho phép nên phải phân đoạn document D thành từng phân đoạn nhỏ P có kích thước 512 và các phân đoạn liên tiếp sẽ chồng chéo nhau một đoạn có kích thước 256 (bảo đảm answer có thể nằm gọn trong phân đoạn).

Đối với bộ dữ liệu ViQuAD 2.0, mỗi câu hỏi có thể chứa câu trả lời (answer) chính là câu trả lời trong đoạn tài liệu, hoặc không có. Do đó, để có thể tận dụng nhãn từ dữ liệu, hai kiểu gán nhãn dưới đây được thực hiện:

- **Hard Labeling**: **1** nếu answer nằm trọn trong phân đoạn P của document D, **0** nếu ngược lại.
- **Soft Labeling**:
  - Với những mẫu mà không chứa câu trả lời: **0** cho mọi phân đoạn.
  - Với những phân đoạn mà chứa toàn bộ câu trả lời: **1**
  - Với những phân đoạn không chứa phần nào của câu trả lời: **0**
  - Với những phân đoạn chứa một phần câu trả lời: dùng công thức

$$0.5\left(1+\frac{|\text{present}|}{|\text{answer}|}\right)$$

    Với $|u|$ là số token có trong chuỗi u.

Mô hình sử dụng có kiến trúc giống với mô hình ở phương pháp **Baseline**, trong đó phần text encoder ngoài BERT ra có sử dụng thêm 2 loại mô hình khác là XLM-RoBERTa và DistilBERT.

3.3.2 Huấn luyện

Hàm mất mát Cross Entropy Loss:

$$\mathscr{L}_{\textbf{Input}} = -\sum_{c=1}^{C} \textbf{Label}_c \log(softmax(\textbf{Output})_c)$$

Vì ba bộ Text Encoder BERT/RoBERTa/DistilBERT sử dụng có số lượng tham số khác khau nên tham số huấn luyện có đôi chút khác biệt.

Tham số huấn luyện BERT:

- Epoch: 10
- Max learning rate: 2e-5
- Batch size: 16
- AdamW Optimizer
- Warmup Linear Scheduler (warmup steps: 10%)

Tham số huấn luyện RoBERTa:

- Epoch: 10
- Max learning rate: 2e-5
- Batch size: 16
- AdamW Optimizer
- Warmup Linear Scheduler (warmup steps: 10%)

Tham số huấn luyện DistilBERT:

- Epoch: 6
- Max learning rate: 1e-5
- Batch size: 16
- AdamW Optimizer
- Warmup Linear Scheduler (warmup steps: 10%)

Accelerator: GPU P100 (Kaggle).

Thời gian huấn luyện tổng: Khoảng 22 GPU hours.

3.4 Seq2Seq: ViT5

3.4.1 Phương pháp

Các phương pháp trên chỉ thực hiện theo cùng một kiểu đó chính là dùng Text Encoder lấy vector embedding của câu rồi đưa vào một mạng phân loại. Việc đó khiến cho ta không biết khả năng hiểu ngữ cảnh giữa query Q và document D của mô hình có tốt hay không, hay ta không hiểu rõ mô hình sẽ chú ý theo hướng như thế nào giữa câu hỏi và đoạn tài liệu.

Khác với các mô hình phân loại ở trên, phương pháp này nâng cao khả năng truy xuất thông tin (Information Retrieval) nhờ dùng mô hình Seq2Seq dựa trên ý tưởng là có thể tăng cường khả năng hiểu ngữ cảnh giữa query và document của mô hình trong quá trình pretrain từ đó giúp mô hình nâng cao khả năng hiểu và tăng được độ chính xác. Cụ thể, quá trình pretrain trước khi áp dùng cho bộ dữ liệu ZaloQA sẽ sử dụng một lượng lớn dữ liệu tăng cường từ các bộ dữ liệu ViQuAD 2.0, SQuAD 2.0 và Facebook MLQA kết hợp lại.

Mô hình Seq2Seq sử dụng là ViT5. ViT5 là mô hình Seq2Seq dựa trên kiến trúc T5 được tiền huấn luyện cho tiếng Việt. ViT5 được đào tạo trên kho văn bản tiếng Việt đa dạng và chất lượng cao. ViT5 về hai tác vụ sinh văn bản là Abstractive Text Summarization và Named Entity Recognition.

Mô hình trải qua 2 giai đoạn huấn luyện:

- Pretrain: Sử dụng các bộ dữ liệu ViQuAD 2.0, SQuAD 2.0 và Facebook MLQA. Với mỗi cặp query Q - document D và answer A (nếu có câu trả lời), đầu vào được kết hợp như sau:

$$\textbf{Input} = \text{Encode}\{Q \texttt{ </s> } D \texttt{ </s>}\}$$

$$\textbf{Label} = \begin{cases} \text{Encode('Có. } \{A\}\text{')} & , \exists A \\ \text{Encode('Không.')} & , \text{otherwise} \end{cases}$$

  Chuỗi sinh ra trong quá trình huấn luyện, chứa xác suất từng của từng token:

$$\textbf{Output} = \text{ViT5}(\textbf{Input})$$

$$T = |\textbf{Input]}| = |\textbf{Output}|$$

- Fine-tuning:Sử dụng bộ dữ liệu ZaloQA. Với mỗi cặp query Q - document D và label l, đầu vào được kết hợp như sau:

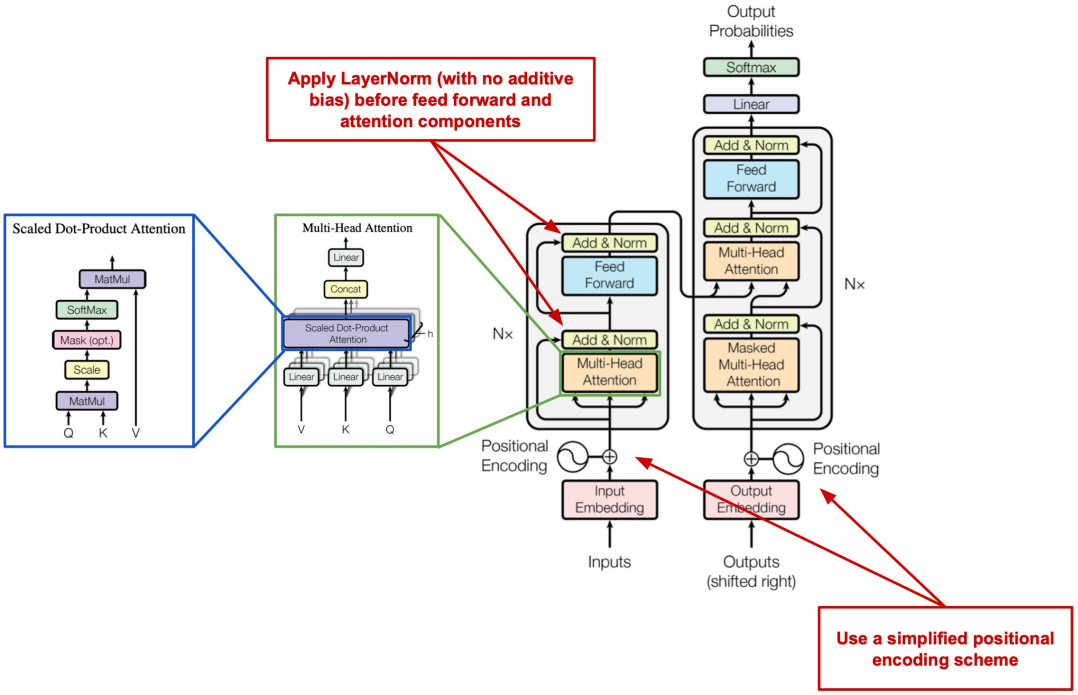
$$\textbf{Input} = \text{Encode}\{Q \texttt{ </s> } D \texttt{ </s>}\}$$

$$\textbf{Label} = \begin{cases} \text{Encode('Có.')} & ,l = \text{true} \\ \text{Encode('Không.')} & ,l = \text{false} \end{cases}$$

Chuỗi sinh ra trong quá trình huấn luyện, chứa xác suất của từng token:

$$\textbf{Output} = \text{ViT5}(\textbf{Input})$$

$$T = |\textbf{Input}]| = |\textbf{Output}|$$



Hình 3.2 Kiến trúc mô hình T5

3.4.2 Huấn luyện

Hàm mất mát Cross Entropy Loss trên chuỗi T:

$$\mathscr{L}_{\textbf{Input}} = -\frac{1}{|T|}\sum_{t=1}^{|T|}\log(\textbf{Output}^{t}_{\textbf{Label}^{t}})$$

Tham số huấn luyện Pretrain:

- Epoch: 10
- Max learning rate: 2e-5
- Batch size: 16
- AdamW Optimizer

- Warmup Linear Scheduler (warmup steps: 10%)

Tham số huấn luyện Fine-tuning:

- Epoch: 6
- Max learning rate: 1e-5
- Batch size: 16
- AdamW Optimizer
- Warmup Linear Scheduler (warmup steps: 10%)

Accelerator: GPU P100 (Kaggle).

Thời gian huấn luyện tổng: Khoảng 40 GPU hours.

4. Kết quả thực nghiệm

4.1 So sánh kết quả

Từ các phương pháp trên, bảng đánh giá được thiết lập dựa trên kết quả của các mô hình sau đây:

- **Baseline**
- **BERT + QC**
- **XLM-RoBERTa + QC**
- **BERT + Soft Labeling**
- **XLM-RoBERTa + Soft Labeling**
- **DistilBERT + Soft Labeling**
- **BERT + Hard Labeling**
- **XLM-RoBERTa + Hard Labeling**
- **DistiBERT + Hard Labeling**
- **ViT5**

Models	F1 macro (%)	F1 weighted (%)
Baseline	81.21	83.61
BERT + QC(92%)	81.16	83.02
XLM-RoBERTa + QC(92%)	81.73	84.72
BERT + Soft Labeling	81.65	84.02
XLM-RoBERTa + Soft Labeling	82.43	85.52
DistilBERT + Soft Labeling	80.39	81.82
BERT + Hard Labeling	66.12	71.18
XLM-RoBERTa + Hard Labeling	68.64	72.27
DistiBERT + Hard Labeling	65.36	70.85
ViT5	<u>84.61</u>	<u>87.26</u>

Bảng 4.1 Kết quả của các mô hình khác nhau khi đánh giá trên tập test của ZaloQA. Kết quả tốt nhất được gạch chân

4.2 Nhận xét

Dựa vào bảng kết quả, ta có thể thấy phương pháp sử dụng mô hình ViT5 đạt điểm đánh giá cao nhất và là phương pháp tốt nhất của dự án này. Không ngoài dự đoán khi mà các bước huấn luyện được thiết kế giúp tăng cường khả năng suy luận của mô hình ViT5 và đã cho kết quả thực nghiệm rất tốt, còn đối với các phương pháp khác khi so sánh kết quả với **Baseline**:

- BERT/RoBERTa + QC: Kết quả gần như xấp xỉ **Baseline**. Nguyên nhân có thể do việc gán nhãn là thừa đối với mẫu đã cho, khi đưa dữ liệu vào mô hình thì các lớp Self Attention của mô hình đã có thể tính được biểu diễn của ngữ cảnh dựa vào câu hỏi mà câu trả lời, đưa thêm nhãn của câu hỏi là việc không cần thiết. Do đó, các phương pháp sau đó không sử dụng thêm tác vụ QC.
- BERT/RoBERTa/DistilBERT + Soft Labeling: Có nâng độ chính xác so với phương pháp **Baseline** tuy nhiên không đáng kể. Do phương pháp gán nhãn đã tận dụng được những context đúng trong passage của đoạn tài liệu được cho. Tuy nhiên, phương pháp gãn nhãn trong công thức được nêu có thể làm mô hình trở nên trung tính (thay vì cố gắng xấp xỉ hóa các xác suất gấn 0 và 1 thì lại xấp xỉ về khoảng giữa) và có tỉ lệ các mẫu *Hard Negative* lớn.
- BERT/RoBERTa/DistilBERT + Hard Labeling: Kết quả tệ hơn rất nhiều do phương pháp gán nhãn cứng bỏ qua nhiều context đúng đối với câu hỏi trong passage của đoạn tài liệu, làm cho mô hình khó nắm bắt được chính xác ngữ cảnh của câu hỏi và đoạn tài liệu.

Như đã đề cập ở phần **1.2. Định hướng phương pháp**, mục tiêu thực hiện đề tài này để cải tiến kết quả của phương pháp **Baseline**. Như vậy, mục tiêu của đề tài đã đạt được. Tuy nhiên vẫn có thể còn nhiều giải pháp tốt hơn chưa được áp dụng, Đồng thời ta cũng cần nghiên cứu kỹ hơn tính chất của các mô hình ngôn ngữ lớn và cách chúng hoạt động để đưa ra các giải thích có ý nghĩa cho kết quả. Một cách giúp mở rộng bài toán và giúp mô hình học được tốt hơn ý nghĩa của câu hỏi và ngữ cảnh là thực hiện đa tác vụ, kết hợp giải nhiều bài toán và sử dụng tri thức học được để hỗ trợ giải các tác vụ khác. Nếu có thể, những cách làm này sẽ được triển khai và thử nghiệm tiếp tục trong thời gian tới.

5. Các phương pháp nghiên cứu thêm

Ngoài các phương pháp đã thực hiện được trình bày ở phía trên, phần này trình bày thêm một số phương pháp khác đã được tìm hiểu nhưng chưa áp dụng được do đang gặp một số vấn đề trong quá trình cài đặt mã nguồn.

5.1 ColBERT

Paper ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT

5.1.1 Ý tưởng

ColBERT (**Co**ntextualized **L**ate interaction over **BERT**) được đề xuất trong paper "ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT" (Omar Khattab and Matei Zaharia, 2023) là một mô hình tìm kiếm nhanh và chính xác, cho phép tìm kiếm dựa trên BERT trên các bộ sưu tập văn bản lớn trong thời gian rất ngắn (khoảng vài mili giây). ColBERT giới thiệu một kiến trúc tương tác muộn mà độc lập mã hóa truy vấn và tài liệu sử dụng BERT và sau đó sử dụng một bước tương tác mạnh mẽ nhưng đơn giản mô hình hóa sự tương đồng giữa chúng.

Cơ chế hoạt động của ColBERT bao gồm các bước sau:

- Mã hóa Truy vấn (Query) và Tài liệu (Document): Truy vấn và văn bản Tài liệu được mã hóa riêng biệt thành các context vector bằng cách sử dụng hai mô hình BERT khác nhau, vector ngữ cảnh của truy vấn q và tài liệu d lần lượt là E_q và E_d .

- Tương tác muộn (Late interaction): Các context vector được cho phép chú ý đến nhau và tính điểm liên quan cho mỗi cặp truy vấn-tài liệu. Độ tương đồng được tính qua phép toán tích vô hướng.

- Tính toán Score: Sử dụng các phép tính độ tương đồng (similarity) mạnh mẽ nhưng hiệu quả để tìm các tài liệu phù hợp với truy vấn. Score được tính qua công thức:

$$S_{q,d} = \sum_{i \in [|E_q|]} \max_{j \in [|E_d|]} E_{q_i} \cdot E_{d_j}^T$$

Các tài liệu sau đó được xếp hạng dựa trên điểm số này, và top-k tài liệu có điểm số cao nhất sẽ được chọn làm kết quả trả về. Điều này cho phép ColBERT tìm kiếm nhanh chóng và chính xác trên các kho dữ liệu văn bản lớn

Thuật ngữ "tương tác muộn" (late interaction) trong ColBERT mô tả cách mà mô

hình này xử lý truy vấn và tài liệu. Trong nhiều mô hình tìm kiếm dựa trên BERT khác, truy vấn và tài liệu được ghép lại với nhau và sau đó được đưa qua mô hình BERT để tính toán điểm liên quan (Cross Encoder). Điều này có thể tốn kém về mặt tính toán vì mỗi cặp truy vấn-tài liệu đều phải được xử lý qua mạng neural lớn.

Trái lại, ColBERT sử dụng cơ chế "tương tác muộn" nơi mà truy vấn và tài liệu được mã hóa độc lập bằng BERT, và sau đó một bước tương tác đơn giản nhưng mạnh mẽ được sử dụng để mô hình hóa sự tương đồng giữa chúng. Bằng cách trì hoãn (delaying) nhưng vẫn giữ lại sự tương tác tinh tế này, ColBERT có thể tận dụng sự biểu diễn phức tạp của BERT trong khi vẫn có khả năng tính toán trước các biểu diễn tài liệu, giúp tăng tốc độ xử lý truy vấn.

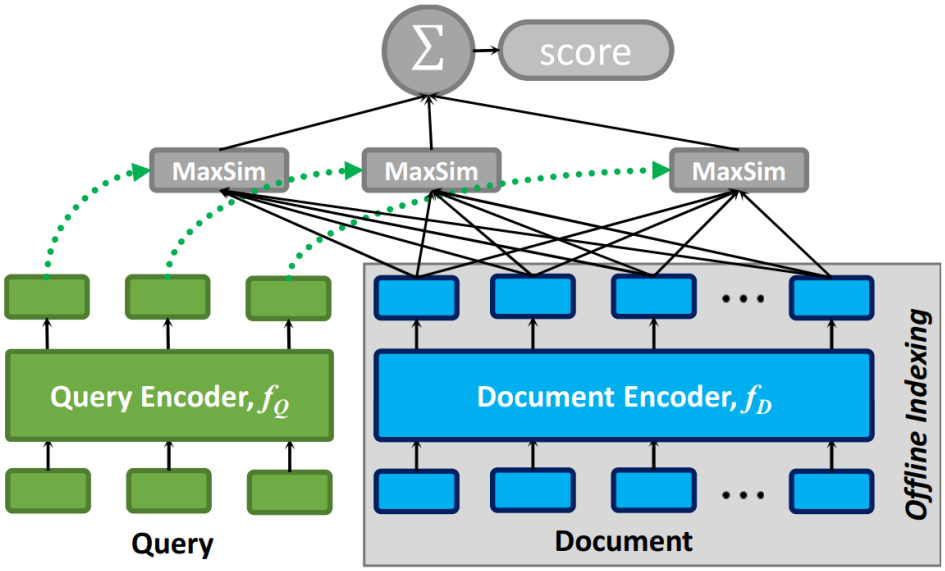
Điều này cho phép ColBERT vượt qua chất lượng của các mô hình khác, trong khi vẫn mở rộng hiệu quả đến các kho dữ liệu lớn.

Ưu điểm:

- Hiệu suất: ColBERT cung cấp hiệu suất tìm kiếm nhanh và chính xác trên các kho dữ liệu văn bản lớn.
- Độc lập mã hóa: ColBERT mã hóa truy vấn và tài liệu một cách độc lập, giúp giảm thiểu chi phí tính toán.
- Tương tác muộn: ColBERT sử dụng cơ chế tương tác muộn, cho phép mô hình hóa sự tương đồng giữa truy vấn và tài liệu sau khi đã mã hóa.

Nhược điểm:

- Phụ thuộc vào BERT: ColBERT phụ thuộc vào BERT để mã hóa truy vấn và tài liệu, do đó nó có thể bị ảnh hưởng bởi các hạn chế của BERT, như khả năng hiểu ngữ cảnh phức tạp.
- Độ phức tạp: Mặc dù ColBERT giảm thiểu chi phí tính toán bằng cách sử dụng cơ chế tương tác muộn, nhưng việc tính toán các nhúng ngữ cảnh từ BERT vẫn có thể tốn kém.



Hình 5.1 Kiến trúc của ColBERT với một query q và một document d.

5.1.2 Vấn đề khi cài đặt

Khi cài đặt mã nguồn mô hình, một số vấn đề sau xảy ra:

- ColBERT nguyên bản dùng cho tiếng Anh, để dùng cho tiếng Việt thì cần mô hình BERT được tiền huấn luyện cho tiếng Việt. Tuy nhiên, vì chưa nắm rõ các module trong project ColBERT nên quá trình tích hợp gặp phải một số khó khăn và chưa thực hiện được.

5.2 Fine-grained Attention Alignment

Paper FAA: Fine-grained Attention Alignment for Cascade Document Ranking

5.2.1 Ý tưởng

Mô hình **F**ine-grained **A**ttention **A**lignment (FAA) được đề xuất trong paper "FAA: Fine-grained Attention Alignment for Cascade Document Ranking" (Li et al., 2023) là một mô hình xếp hạng tài liệu theo độ liên quan với truy vấn. Mô hình này được thiết kế để xử lý các tài liệu dài, vượt quá giới hạn chiều dài của các mô hình Transformer truyền thống bằng cách chia document thành nhiều đoạn văn (passage) có kích thước cố định.

Mô hình FAA bao gồm hai giai đoạn:

Giai đoạn 1: Passage Selector

Trong giai đoạn này, mô hình sử dụng một mô hình Encoder đã được tiền huấn luyện để học các đại diện cho cả tài liệu và truy vấn. Sau đó, mô hình sử dụng hàm chấm điểm để tính toán độ liên quan của đoạn văn với truy vấn.

• Passage Selector sử dụng một mô hình Encoder để học và tạo các vector biểu diễn ẩn cho cả tài liệu (chính xác hơn là các đoạn của tài liệu) và truy vấn. Mô hình Encoder này được cấu hình với một số lượng lớp, số tham số nhỏ hơn so với các mô hình Encoder khác để giảm độ trễ truy vấn trên các tài liệu dài. Vector biểu diễn ẩn của truy vấn q và đoạn văn q_i ký hiệu lần lượt là $\text{Enc}_{psg}(q) \in \mathbb{R}^d$ và $\text{Enc}_{psg}(q_i) \in \mathbb{R}^d$

• Sau khi tính toán được các vector biểu diễn ẩn, mô hình sử dụng hàm chấm điểm để tính toán độ liên quan của các đoạn văn q_i với truy vấn q. Hàm chấm điểm này là hàm *scaled dot-product* có công thức:

$$\mathscr{R}_{psg}(q, p_i) = \frac{\text{Enc}_{psg}(q)^T \text{Enc}_{psg}(p_i)}{\sqrt{d}}$$

• Tiếp theo, Passage Selector sẽ chọn k đoạn văn có điểm phù hợp cao nhất để tạo thành $\bar{\mathbb{P}}$, được công thức hóa là:

$$\bar{\mathbb{P}} = \underset{\bar{\mathbb{P}} \subset \mathbb{P}, ||\bar{\mathbb{P}}||=k}{\arg\max} \sum_{p_i \in \bar{\mathbb{P}}} \mathscr{R}_{psg}(q, p_i)$$

Giai đoạn 2: Document Ranker

Trong giai đoạn này, mô hình sử dụng một Encoder khác để thực hiện mã hóa chéo (cross encoder). Kiến trúc thực hiện toàn bộ sự chú ý trên truy vấn cũng như các đoạn được trích xuất và đã được chứng minh là có hiệu quả trong việc xếp hạng (Hofstätter et al., 2021). Mô hình sử dụng kết quả phân tích này để tính toán điểm phù hợp của tài liệu đối với truy vấn.

• Sau khi *Passage Selector* lấy ra top-k passages $\bar{\mathbb{P}}$, tất cả các đoạn được chọn trong $\bar{\mathbb{P}}$ trước tiên được ghép lại với nhau thành $\hat{\mathbb{P}} = \{\bar{p}_1; \bar{p}_2; \ldots; \bar{p}_k\}$. Sau đó chúng tôi ghép nối truy vấn (q) và các đoạn được ghép $\hat{\mathbb{P}}$ làm đầu vào của *Document Ranker* với mã thông báo [CLS] và [SEP], được ký hiệu là u:

$$u = \{[\text{CLS}]; q; [\text{SEP}]; \hat{\mathbb{P}}; [\text{SEP}]\}$$

• Khác với mã hóa truyền thống chỉ sử dụng vector biểu diễn ẩn của mã thông báo phân loại [CLS] làm biểu diễn cho toàn bộ chuỗi u, bài báo đề xuất hợp nhất các điểm liên quan của cấp độ đoạn văn đã chọn từ *Passage Selector* để tạo ra biểu diễn ẩn $\text{Enc}_{doc}(u)$ của chuỗi đầu vào u. Cụ thể, chúng tôi kí hiệu vector nhúng của [CLS] là $E_{[CLS]}$ và biểu thị vector nhúng của mã thông báo trong $\bar{\mathbb{P}}$ là $\{E_{1,1}, E_{1,2}, \ldots, E_{i,j}, \ldots\}$, trong đó $E_{i,j}$ biểu thị vector nhúng của mã thông báo thứ j trong đoạn văn được chọn thứ i $\bar{p}_i$.

• Với mỗi đoạn văn được chọn bởi *Passage Selector*, tính toán vector nhúng trung bình của mỗi đoạn đã chọn:

$$\text{MeanPool}(\bar{p}_i) = \sum_{z=1}^{l} E_i^z / l$$

trong đó l là kích thước của $\bar{p}_i$.

• Sau đó, bài báo tính tích của điểm liên quan ở cấp độ đoạn văn từ bộ chọn, gọi là Pasage-selector Guided Vector (E_{PGV}), được chính thức hóa như sau:

$$E_{PGV} = \sum_{t=1}^{k} \text{MeanPool}(\bar{p}_i) \cdot \mathscr{R}_{psg}(q, p_i)$$

• Cuối cùng hợp nhất E_{PGV} với $E_{[CLS]}$ để có được đồng biểu diễn khớp cấp tài liệu:

$$\text{Enc}_{doc}(u) = E_{[CLS]} + \lambda \cdot E_{PGV}$$

trong đó λ là tham số để kiểm soát trọng số của E_{PGV}.

• Sau khi có được vector biểu diễn của u, vector được đưa vào mạng Multi-layer Perceptron (MLP) để tính điểm liên quan của truy vấn q và tài liệu d:

$$\mathscr{R}(q,d) = \text{MLP}(\text{Enc}_{doc}(u))$$

Mô hình FAA đã được thử nghiệm trên các tập dữ liệu document ranking tiêu chuẩn. Kết quả thực nghiệm cho thấy mô hình FAA có thể cải thiện đáng kể độ chính xác so với các mô hình xếp hạng tài liệu truyền thống.

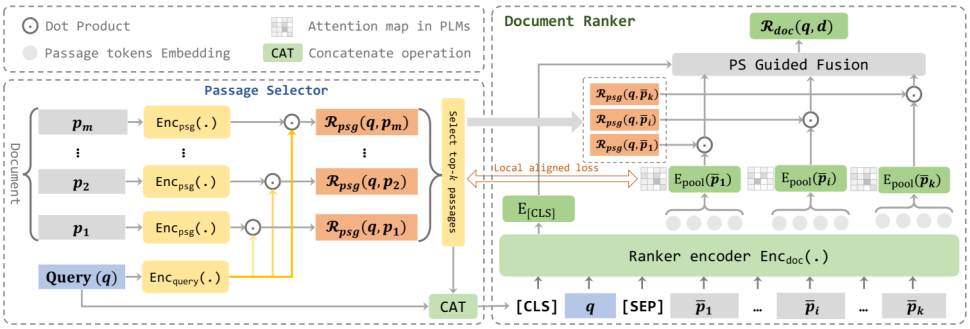
Ưu điểm:

• Có thể xử lý các tài liệu dài

• Có thể cải thiện độ chính xác của việc xếp hạng tài liệu

Nhược điểm:

• Cần thêm một giai đoạn phân tích độ liên quan Có thể phức tạp hơn các mô hình xếp hạng tài liệu truyền thống



Hình 5.2 Kiến trúc của FAA.

5.2.2 Vấn đề khi cài đặt

Khi cài đặt mã nguồn mô hình, một số vấn đề sau xảy ra:

• Một số biến không đề cập giá trị. Giá trị các biến λ và τ trong các công thức như

$$\text{Enc}_{doc}(u) = E_{[\text{CLS}]} + \lambda \cdot E_{\text{PGV}}$$

$$\mathscr{A}^{doc}(q, \bar{p}_k) = \frac{\exp(\alpha_{q \to \bar{p}_k}/\tau)}{\sum_{\bar{p} \in \bar{\mathbb{P}}} \exp(\alpha_{q \to \bar{p}}/\tau)}$$

Không được đề cập trong paper.

• Ở đầu vào của *Document Ranker*, ta thấy rõ được rằng độ dài của vector u là khá dài, cần một mô hình đầu vào lớn và cần tài nguyên tính toán lớn hơn nhiều. Do tài nguyên còn khá hạn chế (chỉ chạy trên Colab và Kaggle) nên điều kiện này thì hiện tại chưa đáp ứng được.

6. Đánh giá

KẾT LUẬN

Qua báo cáo này, em đã trình bày toán bộ công việc trong quá trình thực hiện đề tài *Bài toán Question Answering trong cuộc thi Zalo AI Challenge 2019*. Đây là một bài toán hay và quá trình làm việc với bài toán này đã mang lại cho em những hiểu biết và ý nghĩa nhất định.

Trong quá trình làm việc, em đã cố gắng thu thập thông tin và học hỏi, thử nghiệm các giải pháp cho vấn đề, tuy nhiên do chưa có nhiều kiến thức chuyên sâu về bài toán đặt ra cũng như sự đa dạng trong việc áp dụng các giải pháp nên kết quả không thể tránh khỏi những thiếu sót, hạn chế. Em rất mong nhận được đóng góp để có thể hoàn thiện tốt hơn nữa.

Một lần nữa em xin gửi lời cảm ơn chân thành tới cô PGS.TS. Lê Thanh Hương đã tận tình giúp đỡ trong quá trình nghiên cứu, tìm hiểu và thực hiện đề tài đã lựa chọn.

TÀI LIỆU THAM KHẢO

[1] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale, 2020.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[3] Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert, 2020.

[4] Zhen Li, Chongyang Tao, Jiazhan Feng, Tao Shen, Dongyan Zhao, Xiubo Geng, and Daxin Jiang. FAA: Fine-grained attention alignment for cascade document ranking. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1688–1700, Toronto, Canada, July 2023. Association for Computational Linguistics.

[5] Long Phan, Hieu Tran, Hieu Nguyen, and Trieu H. Trinh. Vit5: Pretrained text-to-text transformer for vietnamese language generation, 2022.

[6] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023.

[7] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020.