HỆ THỐNG HỎI ĐÁP TIẾNG VIỆT HOÀN CHỈNH DỰA TRÊN TÀI LIỆU NGƯỜI DÙNG CUNG CẤP

VIETNAMESE END-TO-END QUESTION ANSWERING SYSTEM BASED ON USER-PROVIDED DOCUMENT

SVTH: Nguyễn Huy Tưởng

Lớp 19TCLC_DT4, Khoa Công nghệ Thông tin, Trường Đại học Bách Khoa - Đại học Đà Nẵng
Email: huytuong010101@gmail.com

GVHD: TS. Nguyễn Văn Hiệu, KS. Hồ Minh Hoàng

Khoa Công nghệ Thông tin, Trường Đại học Bách Khoa - Đại học Đà Nẵng
Email: nvhieuqt@dut.udn.vn, minhhoangho99@gmail.com

Tóm tắt - Hiện nay, hệ thống hỏi đáp đang rất phổ biến, được ứng dụng rộng rãi trong nhiều lĩnh vực như bán hàng, tư vấn viên, giáo dục,... Nghiên cứu này đề xuất một mô hình trích xuất câu trả lời từ văn bản Tiếng Việt dựa trên mô hình ngôn ngữ BARTPho - đây là mô hình ngôn ngữ mới nhất đã được huấn luyện trên tập dữ liệu lớn dành cho tiếng Việt. Nghiên cứu này cũng đồng thời đề xuất một pipeline hoàn chỉnh cho hệ thống hỏi đáp dựa trên tài liệu người dùng cung cấp. Đồng thời xây dựng một hệ thống hỏi đáp Tiếng Việt, hệ thống này cho phép người dùng bổ sung các tài liệu và hỏi đáp trên các tài liệu đó. Hiệu quả thực hiện trên dữ liệu Tiếng Việt đạt kết quả ấn tượng với điểm F1 đạt 77.00%. Mô hình đề xuất này mở ra các ứng dụng trong lĩnh vực hỏi đáp cho ngôn ngữ Tiếng Việt với mức độ thích ứng với dữ liệu mới nhanh, không cần thiết phải huấn luyện lại mô hình cho mỗi tập dữ liệu mới.

Từ khóa - hệ thống hỏi đáp, tìm kiếm tài liệu, trích xuất câu trả lời dựa trên ngữ cảnh, transformer, xử lý ngôn ngữ tự nhiên

Abstract - Currently, the question and answer system is very popular, widely applied in many fields such as sales, consulting, education,... This study proposes a model to extract answers from the text in Vietnamese language based on the language model BARTPho - this is the latest language model trained on a large data set for Vietnamese. This study also proposes a complete pipeline for a Q&A system based on user-provided documents. At the same time, building a Vietnamese question-and-answer system, which allows users to add documents and ask questions on those documents. Performance on Vietnamese data achieved impressive results with F1 score of 77.00%. This model for export opens up applications in the field of Q&A for the Vietnamese language with its fast adaptability to new data, without the need to retrain the model for each new dataset.

Key words - Q&A system, document retrieval,, context-based answer extraction, transformer, natural language processing

1. Đặt vấn đề

Hệ thống hỏi đáp là một ứng dụng cực kỳ phổ biến của lĩnh vực xử lý ngôn ngữ tự nhiên, được ứng dụng trong nhiều lĩnh vực khác nhau. Mục tiêu của hệ thống hỏi đáp là thay thế con người trả lời những câu hỏi đã được huấn luyện từ trước. Ưu điểm khi áp dụng hệ thống hỏi đáp là tiết kiệm được thời gian, công sức. Đồng thời hệ thống hỏi đáp cũng có thời gian phản hồi nhanh hơn con người. Tuy nhiên vấn đề quan trọng khi ứng dụng hệ thống hỏi đáp vào thực tế là huấn luyện và độ chính xác của câu trả lời, đối với người sử dụng non-tech, việc huấn luyện cần được đơn giản hóa mức tối thiểu và đồng thời, hệ thống cần đạt được độ chính xác nhất định. Một trong những cách huấn luyện hệ thống hỏi đáp phổ biến hiện nay là cung cấp cho các mô hình AI một bộ các chủ đề, mỗi chủ đề bao gồm nhiều câu hỏi và câu trả lời. Từ đó mô hình sẽ được huấn luyện để phân loại một câu hỏi do người dùng cung cấp thuộc chủ đề nào, từ đó lựa chọn câu trả lời tương ứng. Bài nghiên cứu [1] là một trong những bài nghiên cứu theo hướng phát triển trên. Đồng thời cũng đã có những hệ thống theo phương pháp đó đã được xây dựng giúp người dùng dễ dàng tự huấn luyện và xây dựng cho mình một hệ thống hỏi đáp như Rasa[2]. Tuy đã được ứng dụng rộng rãi, tuy nhiên nhược điểm của phương pháp trên là người xây dựng hệ thống hỏi đáp cần liệt kê tất cả chủ đề mà người dùng có thể hỏi, đồng thời cần liệt kê tất cả các câu hỏi mà người dùng có thể hỏi. Sau đó cần tốn một khoảng thời gian để mô hình được huấn luyện. Gần đây, một phương pháp mới được đề xuất đó là trích xuất câu trả lời từ ngữ cảnh (context)[3]. Với phương pháp này, câu trả lời trích xuất từ một đoạn văn bản cho trước. Từ đó, chúng ta có thể xây dựng một hệ thống đáp dựa trên tài liệu do người dùng cung cấp. Nhờ vậy, thay vì cần nắm bắt tất cả câu hỏi hệ thống cần trả lời, chỉ cần cung cấp cho hệ thống một bộ tài liệu đủ lớn và bao quát thông tin cần trả lời, từ đó mô hình sẽ tự động trích xuất câu trả lời từ tài liệu được cung cấp.

Trong bài nghiên cứu này, tôi đề xuất một mô hình trích xuất câu trả lời từ văn bản dựa trên mô hình ngôn ngữ BARTPho - một pretrained model lớn và mới nhất dành cho Tiếng Việt hiện nay, đồng thời từ đó đề xuất một pipeline cho hệ thống hỏi đáp Tiếng Việt dựa trên tài liệu người dùng cung cấp. Với pipeline đề xuất, người dùng có thể liên tục bổ sung tài liệu cho hệ thống hỏi đáp và hệ thống cập nhật những tài liệu đó phục vụ cho việc trả lời câu hỏi. Với khả năng thích ứng với dữ liệu mới gần như ngay lập tức, đồng thời với những tối ưu để mô hình có thể xử lý cả câu hỏi có hoặc không có câu trả lời, bài nghiên cứu có thể xem như một pipeline cơ bản có các nghiên cứu trong lĩnh vực hỏi đáp với ngôn ngữ Tiếng Việt.

2. Nghiên cứu liên quan

Theo bài nghiên cứu[15] một hệ thống hỏi đáp hoàn chỉnh dựa trên ngữ cảnh thường bao gồm 2 phần chính: Tìm kiếm văn bản và trích xuất câu trả lời từ văn bản. Trong đó phần tìm kiếm văn bản giúp mô hình chọn ra những văn bản tri thức phù hợp, có khả năng chứa câu trả lời cao nhất. Sau đó, phần trích xuất câu trả lời sẽ tìm câu trả lời trong các văn bản đã tìm được ở bước trước đây, lựa chọn câu trả lời tốt nhất để phản hồi.

Một số phương thức tìm kiếm tài liệu:

- Probabilistic Retrieval Model: Phương pháp này cố gắng ước tính xác suất phù hợp của mỗi văn bản với câu truy vấn, thuật toán phổ biến hiện nay của phương thức này có thể kể đến ldf

TF-IDF, BM25,...

- Language Models: Phương thức này sử dụng các mô hình ngôn ngữ học sâu để tính mức độ phù hợp giữa câu truy vấn và các văn bản.
- Semantic Search: Với sự phát triển của các mô hình học sâu hiện nay, đây có lẽ là phương thức cho độ chính xác cao nhất. Phương thức này tạo ra biểu diễn ngữ nghĩa của câu truy vấn và văn bản, từ đó tính độ tương đồng về ngữ nghĩa của tất cả các văn bản.

Một số phương thức trích xuất câu trả lời:

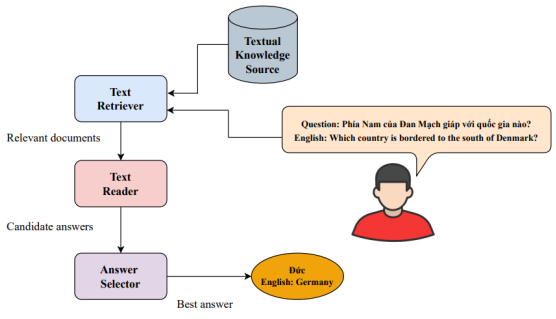
- Machine Learning: Một số nghiên cứu trước đây đã sử dụng những thuật toán cơ bản như SVM, Random Forests,.. để xác định câu trả lời dựa vào cấu trúc ngữ pháp hay đặc trưng của văn bản.
- Deep Learning: Gần đây nhiệm vụ trích xuất câu trả lời đang dần tập trung vào sử dụng các mạng học sâu để xác định vị trí của câu trả lời trong văn bản. Một số kiến trúc phổ biến được sử dụng hiện nay có thể kể đến như LSTM, Transformer,...

Đã có một số bài nghiên liên quan đến xây dựng hệ thống hỏi đáp dựa trên ngữ cảnh trong Tiếng Việt, trong đó bài nghiên cứu [4] đã đề xuất một pipeline hoàn chỉnh cho hệ thống hỏi đáp đa ngôn ngữ. Bài nghiên cứu đề xuất pipeline gồm 2 module chính: Tìm kiếm tài liệu và trích xuất câu trả lời. Trong đó ở module tìm kiếm tài liệu, hệ thống sử dụng phương pháp TF-IDF [5] để tìm kiếm các đoạn văn ngắn liên quan và sử dụng mô hình XLM-RoBERTa[6] kết hợp với giải pháp Token classification để tìm kiếm vị trí bắt đầu, kết thúc của câu hỏi trong đoạn văn. Mặc dù đã đạt được độ chính xác tương đối ấn tượng tuy nhiên, phương pháp trên vẫn còn một số khuyết điểm:

- Tài liệu cung cấp cần phải chia nhỏ theo từng đoạn văn.
- Chưa xử lý ngoại lệ câu hỏi không có câu trả lời

Trong phần tiếp theo, tôi sẽ đề xuất một số phương pháp để cải thiện và khắc phục những nhược điểm trên.

3. Phương pháp đề xuất



Hình 1: *Pipeline được đề xuất bởi* [4]

Tại bài nghiên cứu [4] tác giải đề đề xuất một pipeline cơ bản cho hệ thống hỏi đáp. Dựa trên Pipeline này, tôi đề xuất một pipeline mới gồm 2 module chính: Tìm kiếm tri thức và trích xuất câu trả lời. Trong đó module tìm kiếm tri thức được chia thành 2 phần tìm kiếm tài liệu và tìm kiếm đoạn văn, có chức năng tìm kiếm ra các đoạn văn có khả năng chứa câu trả lời cho câu hỏi, sau đó module trích xuất câu trả lời sẽ tiến hành trích xuất câu trả lời và chọn câu trả lời tốt nhất phản hồi cho người dùng. Đồng thời, huấn luyện mô hình để có thể xử lý được các câu hỏi không có câu trả lời.

3.1. Module tìm kiếm tri thức

3.1.1. Tìm kiếm tài liệu

Hiện nay hệ thống Search Engine đang rất phát triển, cùng với đó là những thuật toán tìm kiếm được đề xuất, trong đó thuật toán BM25 là một trong những thuật toán tìm kiếm dữ liệu văn bản phổ biến nhất hiện nay.

Thuật toán này được phát triển dựa trên thuật toán Okapi BM25 và đã được áp dụng rộng rãi trong các công cụ tìm kiếm như Lucene hay Elasticsearch. Cách hoạt động của thuật toán BM25 là đánh giá mỗi từ trong truy vấn và tính điểm cho mỗi tài liệu để xác định mức độ phù hợp của nó với truy vấn. Các yếu tố quan trọng để tính điểm bao gồm tần suất xuất hiện của từ trong tài liệu, tần suất xuất hiện của từ trong tài liệu so với tần suất xuất hiện của từ trong toàn bộ tập văn bản, số lượng từ trong tài liệu và độ quan trọng của từng từ trong truy vấn.

Công thức(1) dưới đây được sử dụng để tính độ liên quan giữa query Q và văn bản D

$$score(D,\ Q)\ =\ \sum_{i=1}^{n} IDF(q_i).\frac{f(q_i,D).(k+1)}{f(q_i,D)+k.(1-b+b.\frac{|D|}{avgD})} \quad (1)$$

Trong đó:

- n là số lượng term có trong query q (term được hiểu là một từ có nghĩa)
- $f(q_i,\ D)$ là tần suất xuất hiện của term q_i trong văn bản D
- k và b là hai tham số trong đó mặc định k = 1.2 và b = 0.75
- |D| là độ dài văn bản
- avgD là độ dài trung bình trọng tập các văn bản

IDF(q_i) là hàm Inverse Document Frequency với công thức:

$$IDF(q_i)\ =\ log(\frac{N-n(q_i)+0.5}{n(q_i)+0.5}) \quad (2)$$

Trong đó:

- N là tổng số lượng văn bản
- n(q_i) là số lượng văn bản có chưa term q_i

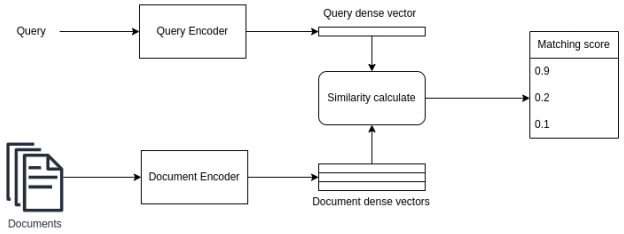
Qua công thức trên có thể thấy mức độ tương quan giữa câu query và văn bản bị ảnh hưởng bởi các yếu tố:

- Các từ trong câu query xuất hiện càng nhiều trong văn bản thì mức độ tương quan càng cao (thể hiện qua hàm $f(q_i,\ D)$). Điều này hoàn toàn phù hợp với thực tế khi văn bản đó có càng nhiều từ chung với query thì khả năng cao cả 2 điều đang thể hiện một ý nghĩa. Hoặc ít nhất đang cùng mô tả về một vấn đề.
- Mức độ ảnh hưởng của một từ là càng thấp nếu như nó xuất hiện trong càng nhiều văn bản (được thể hiện qua hàm $IDF(q_i)$). $IDF(q_i)$ đã có ý nghĩa rất lớn khi hỗ trợ cho yếu tố ở trên, ngoài việc một từ xuất hiện càng nhiều tuy nhiên khi từ đó quá phổ biến (xuất hiện ở rất nhiều văn bản, các từ thường xuyên dùng như là, thế , nếu,...) thì cũng không được đánh giá cao.
- Văn bản đó sẽ được đánh giá cao hơn nếu như có độ dài ngắn hơn. Điều này có nghĩa là một văn bản có tương quan cao với query tuy nhiên văn

bàn đó càng ngắn, nghĩa là văn bản đó càng cô đọng, súc tích thì sẽ được đánh giá cao hơn những văn bản dài.

3.1.2. Tìm kiếm đoạn văn

Sau khi tìm kiếm được các tài liệu liên quan đến đoạn văn, đây là một văn bản khá dài vì vậy chưa thể trích xuất câu trả lời từ văn bản đó. Do đó chúng ta cần chia nhỏ tài liệu thành các đoạn văn nhỏ (overlap lẫn nhau), sau đó cần xếp hạng để lựa chọn các đoạn văn liên quan nhất đến câu hỏi. Tìm kiếm, xếp hạng văn bản trong NLP thường sử dụng phương pháp tìm kiếm dựa trên từ khóa (keyword-based search) hoặc truy xuất dựa trên vector (vector-based retrieval). Tuy nhiên, trong một số trường hợp, các phương pháp này có thể không đảm bảo sự chính xác và đầy đủ trong việc truy xuất thông tin. Dense Retrieval giải quyết vấn đề này bằng cách sử dụng mô hình học sâu (deep learning model) để biểu diễn các câu hoặc văn bản thành các dense vectors. Các vector này bao gồm thông tin ngữ nghĩa, ngữ cảnh của câu hoặc văn bản đó. Phương pháp Dense Retrieval thường sử dụng mạng neural học sâu để biểu diễn câu hoặc văn bản. Quá trình Dense Retrieval bao gồm hai giai đoạn chính: index và truy xuất. Trong giai đoạn index, các câu hoặc văn bản trong tập dữ liệu được mã hoá thành các dense vector bằng cách sử dụng mô hình học sâu. Vector này sẽ được lưu trữ trong một cơ sở dữ liệu có hiệu suất cao, chẳng hạn như Elasticsearch. Trong giai đoạn truy xuất, khi người dùng đưa ra một câu hoặc câu hỏi, mô hình Dense Retrieval sẽ biểu diễn câu đó thành một vector tương tự bằng cách sử dụng mô hình học sâu đó. Sau đó, quá trình truy xuất được thực hiện bằng cách tính toán độ tương đồng (similarity) giữa vector này và các vector trong cơ sở dữ liệu. Các câu hoặc văn bản có độ tương đồng càng cao thì càng tự tin về mức độ liên quan giữa câu truy vấn và văn bản.



Hình 2: *Dense retrieval method*

Mô hình mã hóa văn bản được sử dụng là mô hình keepitreal/vietnamese-sbert[7] được cung cấp bởi thư viện Transformer. Mô hình dựa trên mô hình gốc là RobertaModel[13] và đã được huấn luyện trên bộ dữ liệu Tiếng Việt với hàm mất mát Cosine Similarity. Từ đó với mỗi đầu vào là một câu/đoạn văn, mô hình cho ra 1 vector 768 chiều với các đầu vào càng giống nhau về mặt ý nghĩa thì độ đo Cosine Similarity sẽ càng lớn

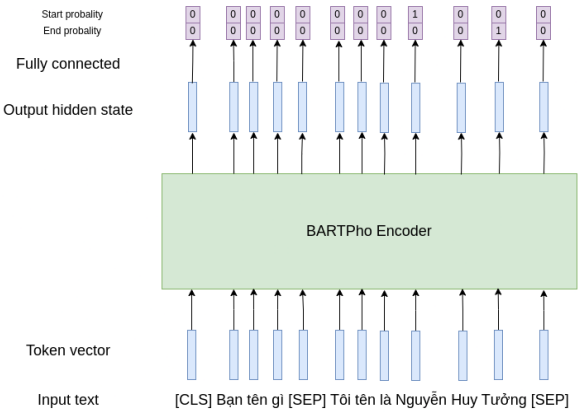
Bảng 1*: Ví dụ sử dụng phương pháp Dense Retrieval*

Câu hỏi	**Đoạn văn**	**Độ liên quan**
Trường đại học	Trường Đại học Bách khoa là một trong ba Trường Đại học Bách	0.596
Bách Khoa Đà Nẵng có địa chỉ ở đâu?	khoa của Việt Nam giữ vai trò là trung tâm đào tạo cán bộ kỹ thuật công nghệ và các nhà quản lý có trình độ cao, đồng thời là trung tâm nghiên cứu khoa học và chuyển giao công nghệ có vai trò chủ đạo trong việc triển khai, ứng dụng công nghệ tiên tiến phục vụ cho sự nghiệp công nghiệp hóa, hiện đại hóa đất nước, đặc biệt là khu vực miền Trung - Tây Nguyên. Trường Đại học Bách Khoa Đà Nẵng tọa lạc ở số 54, đường Nguyễn Lương Bằng, Thành Phố Đà Nẵng.	
Trường đại học Bách Khoa Đà Nẵng có địa chỉ ở đâu?	Các nhà khoa học của Trường Đại học Bách khoa đã triển khai thực hiện thành công một số nhiệm vụ nghiên cứu cơ bản cấp nhà nước và nhiều đề tài cấp Bộ trọng điểm, cấp Bộ, cấp Tỉnh, thành phố Đà Nẵng. Hoạt động nghiên cứu khoa học không chỉ góp phần trực tiếp vào việc thực hiện tốt mục tiêu đào tạo sinh viên ở các bậc học khác nhau từ đại học tới tiến sĩ, mà đồng thời, các kết quả nghiên cứu khoa học do lực lượng khoa học – công nghệ của trường thực hiện cũng đã phục vụ rất hiệu	0.476

3.2. Module trích xuất câu trả lời

Đối với mô hình trích xuất câu trả lời, tôi sử dụng Pretrained mô hình BARTPho[9], chỉnh sửa lại kiến trúc để phù hợp với bài toán trích xuất câu trả lời.

BARTPho[9] là một biến thể của mô hình BART[14] (Bidirectional and AutoRegressive Transformer) được tinh chỉnh và đào tạo trên dữ liệu Tiếng Việt. Mô hình BARTPho được xây dựng dựa trên kiến trúc Transformer, một mô hình học sâu mạnh mẽ trong lĩnh vực xử lý ngôn ngữ tự nhiên. BARTPho được huấn luyện trên một tập dữ liệu lớn bao gồm rất nhiều văn bản Tiếng Việt ở nhiều lĩnh vực khác nhau. Quá trình huấn luyện này giúp mô hình hiểu và tổng quát hóa kiến thức về ngôn ngữ Tiếng Việt, từ đó có khả năng xử lý các nhiệm vụ như dịch máy, tạo mô tả,...



***Hình 3**: Kiến trúc mô hình trích xuất câu trả lời*

Để tái sử dụng mô hình BARTPho cho bài toán trích xuất câu trả lời, tôi sử dụng lại phần Encoder của mô hình BARTPho, phần này giúp ánh xạ mỗi từ Tiếng Việt trong một câu sang một feature vector 1024 phần tử, trong đó vector này nhờ vào cơ chế Positional Encoding và cơ chế Attention, nó đã thể hiện cho vị trí và ngữ nghĩa của nó trong câu. Từ đó thêm một lớp Fully Connected từ feature vector của mỗi token thành vector 2 phần tử để dự đoán xác xuất từ này có phải là từ bắt đầu hoặc kết thúc cho câu trả lời hay không.

Trước khi đưa vào mô hình, câu hỏi và đoạn văn sẽ được tách từ sử dụng công cụ VNCoreNLP[8], việc này là tuân thủ theo luồng xử lý của mô hình BARTPho để việc sử dụng Pretrained được hiệu quả hơn.

Văn bản đầu vào của mô hình có dạng **[CLS] question [SEP] context [SEP],** trong đó question và context lần lượt là câu hỏi và văn bản chứa câu trả lời. [CLS] và [SEP] là 2 token đặc biệt thể hiện cho việc bắt đầu và kết thúc một nội dung.

Ví dụ một văn bản đầu vào:

[CLS] Việt_Nam có bao nhiêu dân_tộc ? [SEP] Trên đất_nước Việt_Nam hiện tại có tổng_cộng 54 dân_tộc anh em . [SEP]

Văn bản này sau đó được mã hóa dưới dạng một vector chứa tất cả token index và được đưa trực tiếp vào mô hình. Đầu ra của mô hình trên là một Tensor 2 chiều có kích thước (2 x length_seq) với length_seq là số lượng token của văn bản đầu vào.

4. Thực nghiệm

4.1. Dataset

Mô hình được huấn luyện trên nhiều bộ dữ liệu bao gồm UIT-ViQuAD[10], phần Tiếng Việt của bộ MKQA[11] và bộ dữ liệu từ bert-vietnamese-question-answering [12]. Đồng thời bộ dataset củng được bổ sung hơn 800 bộ câu hỏi - văn bản - câu trả lời được thu thập thủ công từ Google về chủ đề hành chính với những câu trả lời dài. Sau khi thu thập dữ liệu đoạn văn từ Google, sử dụng công cụ ChatGPT để sinh ra mỗi đoạn văn 3 câu hỏi sau đó gán nhãn thủ công cho vị trí bắt đầu và kết thúc câu trả lời.

***Bảng 2**: Thống kê dữ liệu*

	Min number of token	***Max number of token***	***Mean number of token***
Question	*2*	*55*	*12*
Context	*5*	*766*	*146*

***Bảng 3**: Thống kê số lượng dữ liệu*

Train	**Validate**	**Test**
19341	2150	2737

4.2. Hàm mất mát

Như đã phân tích ở trên, bài toán chúng ta đang đưa về dự đoán vị trí của token bắt đầu câu trả lời và vị trí token kết thúc câu trả lời hay nói cách khác cần dự đoán xác suất một token là vị trí bắt đầu và xác suất một token thuộc vị trí kết thúc. Hàm mất mát khá phổ biến trong các bài toán phân loại chính là Cross Entropy.

Vì có 2 nhiệm vụ là dự đoán vị trí bắt đầu, vị trí kết thúc cho câu trả lời, do đó có 2 hàm loss bao gồm hàm loss để tối ưu cho vector xác suất dự đoán vị trí bắt đầu và vị trí kết thúc. Công thức [3], [4] dưới đây là hàm loss dự đoán xác suất token bắt đầu và kết thúc câu trả lời.

$$L_{start}(\hat{y}, y) = -\sum_{k}^{K} y_{start}^{k} log(\hat{y}_{start}^{k}) \quad (3)$$

$$L_{end}(\hat{y}, y) = -\sum_{k}^{K} y_{end}^{k} log(\hat{y}_{end}^{k}) \quad (4)$$

Trong đó:

- K là số lượng token của văn bản đầu
- y_{start}^{k} là xác suất token thứ k là token bắt đầu câu trả lời
- $\hat{y}_{start}^{k}$ = 1 nếu token thứ k là vị trí bắt đầu câu trả lời, ngược lại $\hat{y}_{start}^{k}$ = 0
- y_{end}^{k} là xác suất token thứ k là token kết thúc câu trả lời
- $\hat{y}_{end}^{k}$ = 1 nếu token thứ k là vị trí kết thúc câu trả lời, ngược lại $\hat{y}_{end}^{k}$ = 0

Hàm loss chính là sự kết hợp giữa hai hàm loss trên:

$$L = \alpha . L_{start} + (1 - \alpha) L_{end} \quad (5)$$

Với α là tham số, qua các thực nghiệm cho thấy α=0.4 cho kết quả tốt nhất.

4.3. Kết quả huấn luyện

Tham số, cấu hình huấn luyện:

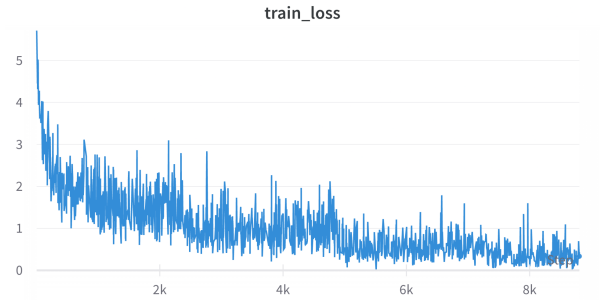
- Learning rate: 0.00002 (giảm tuyến tính xuống 0.000002)
- Optimizer: AdamW
- Batchsize: 8
- Environment: Ubuntu, GPU RTX 3090 24GB

Metric đánh giá:

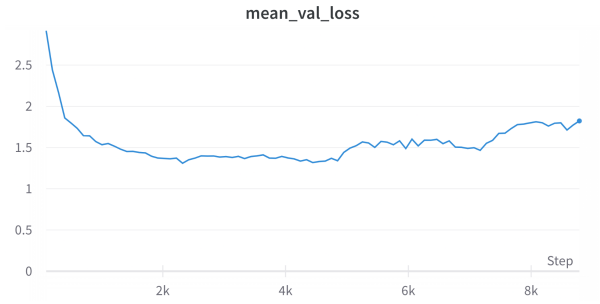
- Exact match: Có giá trị 1 hoặc 0 ứng với câu trả lời trùng khớp hoặc không trùng khớp với nhãn
- Precision: Tỷ lệ giữa số token chung giữa dự đoán và nhãn trên tổng số từ trong dự đoán.
- Recall: Tỷ lệ giữa số token chung giữa dự đoán và nhãn trên tổng số từ trong nhãn.
- F1 score: $F1 = 2.\frac{precision\,.\,recall}{precision + recall}$

***Bảng 4**: Kết quả huấn luyện tốt nhất trên tập validate*

	F1 score	***Precision***	***Recall***	***EM***
Train	*0.98*	*0.99*	*0.98*	*0.94*
Validate	*0.77*	*0.78*	*0.75*	*0.58*
Test	*0.76*	*0.78*	*0.74*	*0.59*



***Hình 4:** Hàm mất mát trên tập train theo optimize step*



***Hình 5**: Hàm mất mát trên tập validate theo optimize step*

5. Bàn luận

5.1. Huấn luyện với tập dữ liệu bổ sung

Qua kết quả huấn luyện ở trên, có thể thấy mô hình đã có thể trích xuất câu trả lời tương đối tốt. Mặc dù qua biểu đồ cho thấy mô hình đã xuất hiện tình trạng overfitting, tuy nhiên qua các thử nghiệm thêm Dropout và áp dụng loss regularization vẫn không thể cải thiện tình trạng overfitting nên hiện tại đang sử dụng early stopping để chọn được điểm hội tụ tốt nhất.

Tuy nhiên xem xét một ví dụ sau:

- Câu hỏi: Ai là người khám phá ra Châu Phi?
- Tài liệu: Nỗ lực đầu tiên được biết đến nhằm thuộc địa hóa lãnh thổ nay là Canada của người châu Âu bắt đầu khi người Norse định cư trong một thời gian ngắn tại L'Anse aux Meadows thuộc Newfoundland vào khoảng năm 1000 CN. Không có thêm hành động thám hiểm của người châu Âu cho đến năm 1497, khi đó thủy thủ người Ý John Cabot khám phá ra vùng duyên hải Đại Tây Dương của Canada cho Vương quốc Anh.
- Trả lời: thủy thủ người Ý John Cabot
- Confident: 0.77

Có thể thấy mặc dù câu trả lời không nằm trong tài liệu, mô hình vẫn trích xuất câu trả lời từ văn bản với mức độ confident rất cao. Vấn đề có thể do mô hình đang được huấn luyện để cố gắng trích xuất câu trả lời trong mọi trường hợp và không thể phân biệt câu hỏi nào không có câu trả lời trong đoạn văn. Việc này có thể dẫn đến các câu trả lời không chính xác khi áp dụng vào thực tế. Để khắc phục tình trạng trên, tôi tiến hành bổ sung dữ liệu với những câu hỏi không có câu trả lời trong đoạn văn. Đối với những trường hợp này, mô hình được huấn luyện để dự đoán token bắt đầu câu hỏi **[CLS]** sẽ là token bắt đầu và kết thúc câu trả lời.

Để xây dựng tập dữ liệu không có câu trả lời, tôi thực hiện 2 phương thức sau:

- Với mỗi câu hỏi trong tập dữ liệu ban đầu, lựa chọn ngẫu nhiên một đoạn văn khác khác chủ đề (sử dụng trường title trong tập dataset)
- Với mỗi câu hỏi trong tập dữ liệu ban đầu, xóa những câu chứa câu trả lời ở trong đoạn văn đi

Tập dữ liệu mới có tỉ lệ như sau:

- 50% câu hỏi có câu trả lời trong đoạn văn
- 25% câu hỏi với đoạn văn thuộc chủ đề khác
- 25% câu hỏi với đoạn văn đã bị xóa câu trả lời

Kết quả huấn luyện được thể hiện qua bảng sau

***Bảng 5:** Kết quả huấn luyện sau khi bổ sung dữ liệu*

	F1 score	***Precision***	***Recall***	***EM***
Train	*0.99*	*0.99*	*0.99*	*0.98*
Validate	*0.8*	*0.81*	*0.78*	*0.7*
Test	*0.77*	*0.79*	*0.75*	*0.67*

Qua quá trình bổ sung dữ liệu và huấn luyện lại mô hình, mô hình đã đạt F1 score 0.77 trên tập kiểm thử. Đồng thời cải thiện việc trích xuất câu trả lời không khớp với câu hỏi, cụ thể với ví dụ trên, mô hình đã dự đoán là không có câu trả lời với confident 0.99. Từ đó có thể thấy khả năng cải thiện của mô hình khi có thể tổng quát hóa trên tập dữ liệu lớn hơn, mở ra hướng cải thiện độ chính xác cho mô hình và pipeline.

5.2. Cải thiện với Named Entity Recognition:

Mặc dù đã huấn luyện với những câu hỏi không có câu trả lời, một số trường hợp mô hình vẫn cố gắng trích xuất câu trả lời mặc dù không có chứa câu trả lời trong đoạn văn. Một trong những trường hợp đặc biệt là những câu hỏi chứa danh từ riêng, mô hình chưa gặp những danh từ đó nhiều dẫn đến việc không hiểu đó là một token quan trọng trong câu. Ví dụ như:

- Câu hỏi: Ai là hiệu trưởng Trường Đại học Bách khoa Hà Nội?
- Tài liệu: "Trước khi được công nhận Hiệu trưởng Trường ĐH Bách khoa, PGS.TS. Nguyễn Hữu Hiếu đảm nhiệm các chức vụ: giảng viên khoa

Điện; Phó trưởng khoa; Trưởng khoa Điện; Phó Hiệu trưởng nhiệm kỳ 2017-2022..
- Trả lời: Nguyễn Hữu Hiếu
- Confident: 0.94

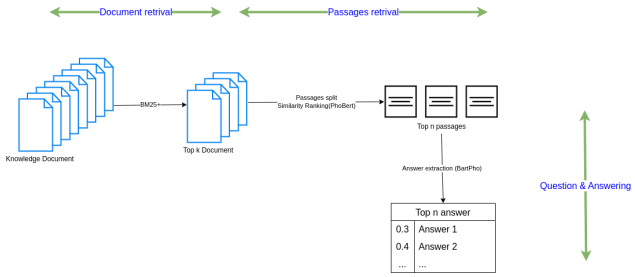
Có thể thấy Hà Nội là một danh từ riêng quan trọng trong câu hỏi tuy nhiên mô hình trích xuất câu trả lời đã bỏ qua nó. Để khắc phục tình trạng này, có thể sử dụng một công cụ khác để phát hiện các danh từ riêng, từ đó có thể ra quyết định rằng đoạn văn không chưa câu trả lời khi nó không có chưa danh từ riêng. Công cụ xử lý ngôn ngữ tự nhiên có hỗ trợ Named Entity Recognition phổ biến hiện nay có thể kể đến là VNCoreNLP, công cụ này hỗ trợ nhận diện danh từ riêng chỉ người, chỉ địa điểm và tổ chức. Dưới đây là một vài kết quả:

***Bảng 6**: Kết quả Named Entity Recognition*

Câu hỏi	Danh từ quan trọng
Ai là hiệu trưởng Trường Đại học Bách khoa Hà Nội?	Đại học, Bách khoa, Hà Nội
Diện tích của tỉnh Bắc Giang là bao nhiêu?	Bắc Giang
Đại tướng Võ Nguyên Giáp sinh năm bao nhiêu?	Võ Nguyên Giáp

Sau khi trích xuất được những danh từ quan trọng trong câu hỏi, có thể đưa ra quyết định đoạn văn không chứa câu trả lời nếu nó không chưa những danh từ này.

6. Xây dựng hệ thống hỏi đáp



***Hình 6:** Mô hình hỏi đáp*

Tài liệu tri thức cho hệ thống hỏi đáp được lưu trữ dưới dạng văn bản, những dữ liệu này do người dùng cung cấp và mô hình sẽ trích xuất câu trả lời từ kho tài liệu này.

Pipeline hệ thống hỏi đáp bao gồm những bước chính sau:

- Bước 1: Sử dụng câu hỏi như một câu truy vấn, sử dụng thuật toán BM25 để tìm kiếm n tài liệu liên quan.
- Bước 2: Với những tài liệu đã tìm kiếm được, thực hiện tách thành các đoạn văn nhỏ hơn overlap lẫn nhau. Sau đó sử dụng phương pháp Dense Retrieval để xếp hạng và lựa chọn k đoạn văn có độ tương quan cao nhất.
- Bước 3: Sử dụng mô hình đã xây dựng để trích xuất câu trả lời của k đoạn văn trên, lựa chọn đoạn văn cho câu trả lời với confident cao nhất để phản hồi tới người dùng.

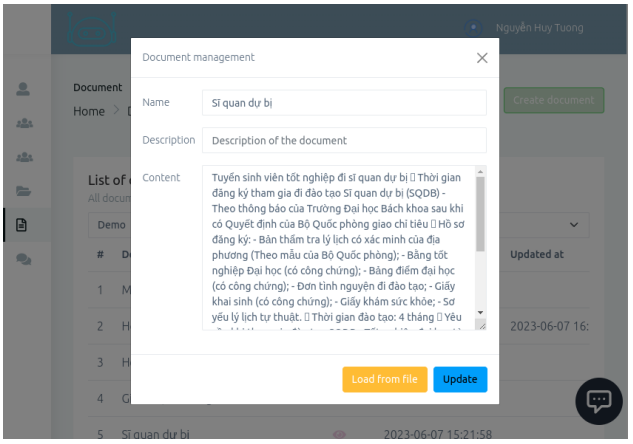
Lưu ý rằng confident được tính bằng:

$$Score = d * \frac{P_{start} + P_{end}}{2} \quad (7)$$

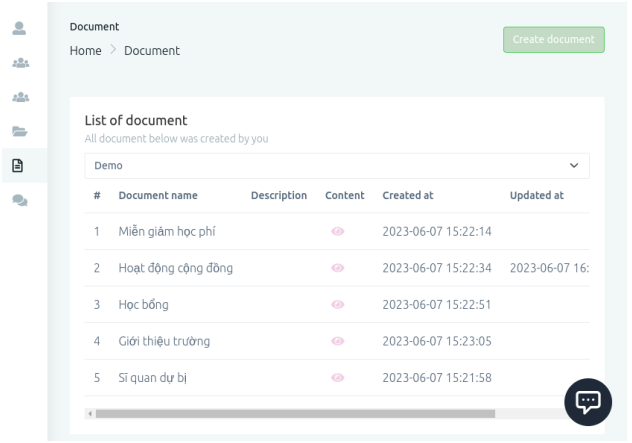
Trong đó: d là giá trị Cosine Similarity của vector câu hỏi và vector đoạn văn được normalize về khoảng [0, 1]. P_{start}, P_{end} lần lượt là xác suất của token bắt đầu, token kết thúc câu trả lời.

Để tối ưu hóa quá trình triển khai, những tài liệu tri thức được lưu trữ ở Elasticsearch đồng thời phương pháp BM25 và việc tính cosine similarity giữa vector câu hỏi và vector đại diện cho các đoạn văn cũng được thực hiện ở Elasticsearch.

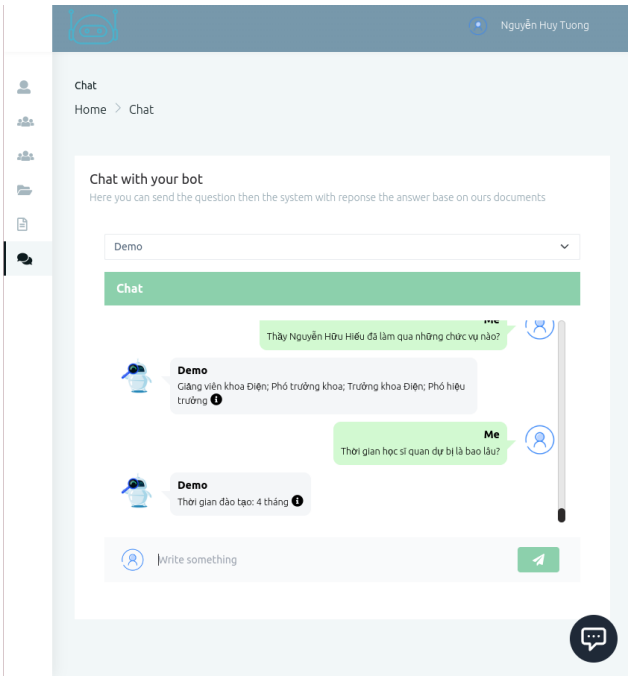
Dưới đây là kết quả của hệ thống sau khi triển khai:



***Hình 7**: Giao diện thêm một tài liệu mới*



***Hình 8**: Giao diện danh sách các tài liệu*



***Hình 9**: Giao diện hỏi đáp về tài liệu đã thêm*

Hệ thống cho phép người dùng thêm các tài liệu liên quan (Hình 7) và xem danh sách các tài liệu đã thêm (Hình 8). Sau đó người dùng có thể hỏi đáp về những thông tin đã trong các tài liệu đã được thêm (Hình 9).

7. Kết luận

Nghiên cứu này đã đề xuất một pipeline hoàn chỉnh cho một hệ thống hỏi đáp Tiếng Việt dựa trên tài liệu người dùng cung cấp. Hệ thống hỏi đáp có thể hoạt động liên tục, không bị ngắt quãng do không cần huấn luyện lại khi người dùng cung cấp một tài liệu mới. Đồng thời nghiên cứu đã xây dựng thành công mô hình trích xuất câu trả lời từ văn bản ngữ cảnh với độ chính xác đạt 0.77 F1 score. Mô hình được huấn luyện trên cả dữ liệu có câu trả lời và không có câu trả lời, từ đó giúp mô hình linh động hơn khi có thể xử lý cả những câu trả lời mà hệ thống hỏi đáp không có khả năng phản hồi (không có tài liệu tri thức liên quan).

Để cải thiện độ chính xác của mô hình củng như hệ thống hỏi đáp, có thể có một số hướng phát triển như: Huấn luyện mô hình trích xuất câu trả lời trên tập dữ liệu lớn hơn. Đồng thời huấn luyện mô hình mã hóa câu hỏi và đoạn văn phục vụ cho việc tìm kiếm đoạn văn thay vì đang sử dụng mô hình vietnamese-sbert[7] đã đã huấn luyện trước.

Tài liệu tham khảo

[1] Trang Huyen Nguyen and M. R. Shcherbakov, “A Neural Network based Vietnamese Chatbot,” Nov. 2018, doi: https://doi.org/10.1109/sysmart.2018.8746962.

[2] “Conversational AI Platform | Superior Customer Experiences Start Here,” Rasa, Dec. 2020. https://rasa.com/.

[3] Wissam Siblini, M. Challal, and C. Pasqual, “Delaying Interaction Layers in Transformer-based Encoders for Efficient Open Domain Question Answering,” Arxiv-vanity.com, 2019. https://www.arxiv-vanity.com/papers/2010.08422/.

[4] Kiet Van Nguyen, Phong Nguyen-Thuan Do, Nhat Duy Nguyen, T. Huynh, A. V. Nguyen, and Ngan Luu-Thuy Nguyen, “XLMRQA: Open-Domain Question Answering on Vietnamese Wikipedia-Based Textual Knowledge Source,” pp. 377–389, Jan. 2022, doi: https://doi.org/10.1007/978-3-031-21743-2_30.

[5] Wikipedia Contributors, “tf–idf,” Wikipedia, May 27, 2023. https://en.wikipedia.org/wiki/Tf%E2%80%93idf.

[6] A. Conneau et al., “Unsupervised Cross-lingual Representation Learning at Scale,” Jul. 2020, doi: https://doi.org/10.18653/v1/2020.acl-main.747.

[7] “keepitreal/vietnamese-sbert · Hugging Face,” Huggingface.co, 2023. https://huggingface.co/keepitreal/vietnamese-sbert.

[8] T. H. Vu, Dat Quoc Nguyen, Dai Hai Nguyen, M. Dras, and M. H. Johnson, “VnCoreNLP: A Vietnamese Natural Language Processing Toolkit,” Jan. 2018, doi: https://doi.org/10.18653/v1/n18-5012.

[9] N. L. Tran, D. M. Le, and D. Q. Nguyen, “BARTpho: Pre-trained Sequence-to-Sequence Models for Vietnamese,” arXiv.org, 2021. https://arxiv.org/abs/2109.09701.

[10] V. Nguyen, D.-V. Nguyen, A. G.-T. Nguyen, and N. L.-T. Nguyen, “A Vietnamese Dataset for Evaluating Machine Reading Comprehension,” arXiv.org, 2020. https://arxiv.org/abs/2009.14725.

[11] S. Longpre, Y. Lu, and J. Daiber, “MKQA: A Linguistically Diverse Benchmark for Multilingual Open Domain Question Answering,” arXiv.org, 2020. https://arxiv.org/abs/2007.15207.

[12] mailong25, “mailong25/bert-vietnamese-question-answering: Vietnamese question answering system with BERT,” GitHub, Jan. 12, 2023. https://github.com/mailong25/bert-vietnamese-question-answering..

[13] Y. Liu et al., “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” arXiv.org, 2019. https://arxiv.org/abs/1907.11692.

[14] M. Lewis et al., “BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension,” arXiv.org, 2019. https://arxiv.org/abs/1910.13461.

[15] F. Zhu, W. Lei, C. Wang, J. Zheng, S. Poria, and T.-S. Chua, “Retrieving and Reading: A Comprehensive Survey on Open-domain Question Answering,” arXiv.org, 2021. https://arxiv.org/abs/2101.00774.

[16] A. Conneau et al., “Unsupervised Cross-lingual Representation Learning at Scale,” arXiv.org, 2019. https://arxiv.org/abs/1911.02116 .