ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

****

XỬ LÍ NGÔN NGỮ TỰ NHIÊN
BÁO CÁO BÀI TẬP LỚN

NHÓM 5
CHỦ ĐỀ: QUESTION ANSWERING

Giảng viên: TS. Trần Hồng Việt

Sinh viên: Hoàng Vũ Duy Anh - 18020001
Lưu Hoàng Nam - 18020921
Nguyễn Công Thuận - 18021250

Mục lục

Danh sách hình vẽ

1 Giới thiệu

1.1 Giới thiệu về Question Answering

Lĩnh vực *Truy hồi thông tin (Information Retrieval)* là một lĩnh vực tập trung vào tìm kiếm tài liệu có bản chất *phi cấu trúc (unstructured)* như văn bản, hình ảnh, video... sao cho *phù hợp (relevant)* với một *nhu cầu thông tin (information need)* nào đó, từ một *tập hợp dữ liệu lớn (large collections)*.

Trong lĩnh vực đó, nhiệm vụ *hỏi đáp (Question Answering – QA)* là nhiệm vụ tự động trả lời những câu hỏi được đưa ra dưới dạng ngôn ngữ tự nhiên. Các câu hỏi trong một miền ứng dụng cụ thể có thể được trả lời thông qua các kỹ thuật xử lý ngôn ngữ tự nhiên. Theo đó, một hệ thống Question Answering sẽ tìm ra câu trả lời cho một câu hỏi bằng cách truy vấn hoặc tìm kiếm ở trong một hệ tri thức nào đó, có thể là một nguồn thông tin, hay một kho kiến thức.... Thông thường, hệ tri thức đó sẽ được cung cấp cho hệ thống từ một vài thông tin trước đó, một vài tài liệu có sẵn, hay từ một trang web chứa một lượng thông tin rất đầy đủ như Wikipedia.

Tuỳ thuộc vào lượng thông tin có sẵn để hỗ trợ trong việc đưa ra câu trả lời, hệ thống Question Answering có thể được chia thành hai loại, bao gồm:

- **Closed-domain Question Answering:** trả lời những câu hỏi liên quan đến một miền ứng dụng nhất định, chẳng hạn như trong lĩnh vực Y tế, hoặc lĩnh vực Chính trị...
- **Open-domain Question Answering:** trả lời những câu hỏi liên quan đến mọi thứ có thể, và cần phải dựa vào một kho kiến thức đủ rộng lớn. Một ví dụ của loại hệ thống này là DrQA, được phát triển bởi Facebook Research. Theo đó, nguồn dữ liệu được cung cấp cho hệ thống DrQA là một lượng lớn các bài viết trên Wikipedia, và do những bài viết đó liên quan đến rất nhiều chủ đề khác nhau, nên hệ thống này có thể được coi là một hệ thống Open-domain Question Answering.

1.2 Các tập dữ liệu nổi tiếng

Đối với nhiệm vụ Question Answering, để có thể hỗ trợ trong việc huấn luyện và đánh giá một hệ thống, cộng đồng Xử lý ngôn ngữ tự nhiên đã tạo ra một số tập dữ liệu đã được chuẩn hoá. Trong các tập dữ liệu, dữ liệu của mỗi câu hỏi thường bao gồm ba thành phần cơ bản như sau:

- `"question"`: là một câu hỏi để huấn luyện hoặc đánh giá.
- `"text"`: là một đoạn văn có thể chứa câu trả lời cho câu hỏi, hoặc không chứa bất kỳ câu trả lời nào.
- `"answer"`: là câu trả lời cho câu hỏi, có thể có hoặc không tuỳ vào tập dữ liệu huấn luyện hay đánh giá.

Trong số các tập dữ liệu đó, có một số tập dữ liệu được sử dụng phổ biến như sau, chủ yếu phù hợp với hệ thống Open-domain Question Answering:

1. **Stanford Question Answering Dataset (SQuAD):** là một bộ dữ liệu được tạo ra bởi các nhà nghiên cứu tại Đại học Stanford. Bộ dữ liệu này bao gồm hơn 100,000 câu hỏi liên quan đến một số chủ đề có trên Wikipedia, với câu trả lời cho mỗi câu hỏi là một hoặc một vài cụm từ có

trong đoạn văn tương ứng với chủ đề trên Wikipedia. Tuy nhiên, trong một vài dữ liệu, câu hỏi có thể không có câu trả lời.

2. **Natural Questions (NQ):** là một bộ dữ liệu được cung cấp bởi Google. Bộ dữ liệu này bao gồm hơn 300,000 câu hỏi được người dùng hỏi trong thực tế qua công cụ Google Search, với câu trả lời được trích xuất trong một đoạn của một bài viết tại Wikipedia.

3. **Question Answering in Context (QuAC):** là một bộ dữ liệu liên quan đến hành động tìm kiếm thông tin trong một cuộc hội thoại giữa một học sinh và một giáo viên. Trong bộ dữ liệu này, học sinh sẽ đưa ra một loạt các câu hỏi về một chủ đề trên Wikipedia, và câu trả lời sẽ được giáo viên đưa ra thông qua những đoạn ngắn của bài viết đó trên Wikipedia. Bộ dữ liệu này bao gồm khoảng 100,000 cặp câu hỏi - câu trả lời.

4. **ELI5 (Explain Like I'm Five):** là một bộ dữ liệu được tạo ra bởi Facebook Research, bao gồm hơn 270,000 cặp câu hỏi - câu trả lời từ subreddit ELI5, cùng với một số dữ liệu trên CommonCrawl.

5. **NewsQA:** là một tập dữ liệu bao gồm hơn 100,000 cặp câu hỏi - câu trả lời liên quan đến các chủ đề trên trang báo điện tử CNN.

1.3 Kiến trúc của một hệ thống Open-domain Question Answering

Trong một bài báo được đưa ra bởi Lewis, et al., 2020, kiến trúc của hệ thống Open-domain Question Answering có thể được phân thành 3 loại, theo mức độ phức tạp tăng dần như sau:

- **Retriever - Reader:** một hệ thống có thể nhớ và đưa ra câu trả lời cho một câu hỏi đã được huấn luyện từ trước.
- **Retriever - Generator:** một hệ thống có thể trả lời những câu hỏi chưa được biết trước, bằng cách chọn ra câu trả lời phù hợp nhất trong quá trình huấn luyện.
- **Generator:** một hệ thống có thể trả lời những câu hỏi chưa được biết trước, trong đó, câu trả lời của câu hỏi đó không có trong quá trình huấn luyện.

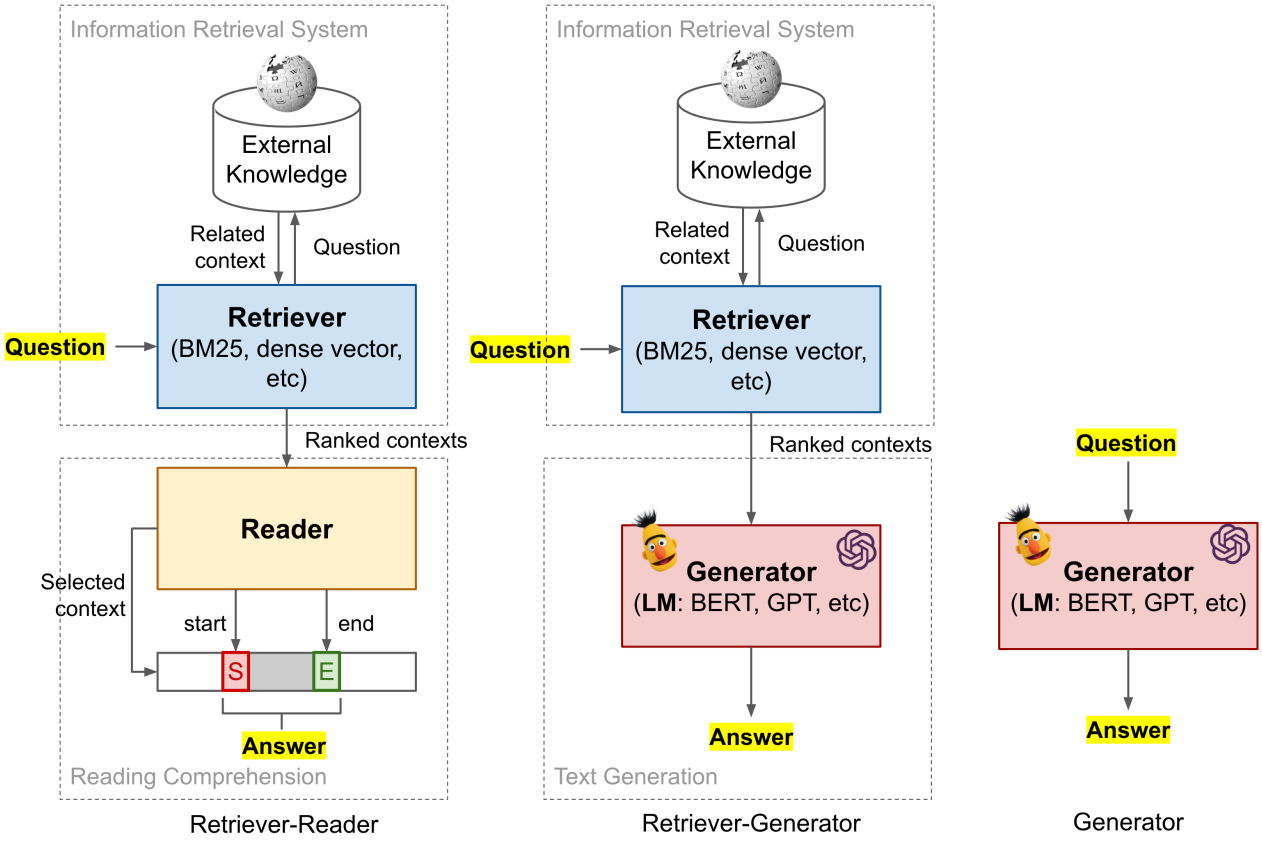
3 loại kiến trúc được mô tả như ở Hình 1 dưới đây.

1.3.1 Kiến trúc Retriever - Reader

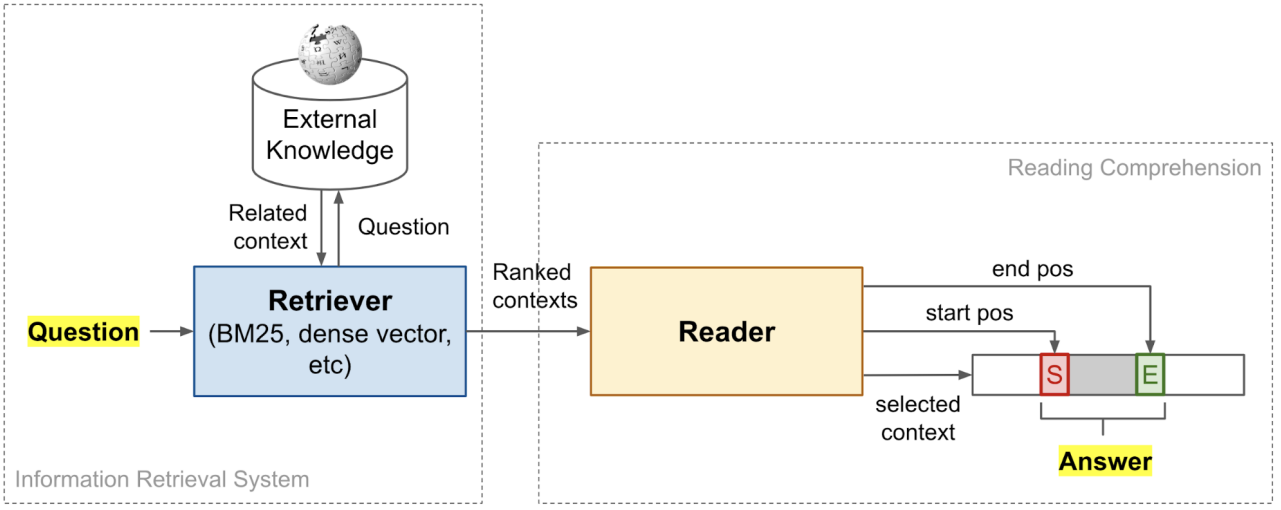
Đối với kiến trúc này, khi cần trả lời một câu hỏi, nếu một hệ thống không có bất kỳ thông tin nào, hoặc không đủ lớn để có thể ghi nhớ thông tin có trong tập dữ liệu huấn luyện, hệ thống đó sẽ khó có thể đưa ra một câu trả lời chính xác cho câu hỏi đó. Chính vì vậy, với kiến trúc này, hệ thống cần phải được cung cấp một kho kiến thức rộng lớn, chẳng hạn như Wikipedia, để từ đó có thể xác định được những đoạn có thể liên quan đến câu trả lời.

Theo đó, quá trình tìm kiếm câu trả lời cho một câu hỏi nào đó có thể được chia ra thành 2 bước như sau:

1. Xác định được đoạn có thông tin liên quan đến câu hỏi trong kho kiến thức được cung cấp,
2. Trích xuất được câu trả lời trong đoạn thông tin nhận được.



Hình 1: Tổng quan về ba kiến trúc của hệ thống Open-domain Question Answering



Hình 2: Kiến trúc Retriever - Reader

Kiến trúc này được đưa ra lần đầu tiên tại bài báo "Document retriever Question-Answering" bởi Chen et al., 2017. Theo đó, 2 thành phần Retriever và Reader có thể được huấn luyện và cài đặt một cách độc lập, hoặc được huấn luyện cùng nhau.

1.3.1.1 Thành phần Retriever

Để tạo ra thành phần Retriever, có 2 cách tiếp cận phổ biến là sử dụng một hệ thống truy hồi thông tin dựa vào các đặc trưng sau khi áp dụng TF-IDF (cách truyền thống), hoặc dựa vào các embedding

vectors của đoạn văn được tạo bởi mạng nơ-ron (cách hiện đại).

Cách truyền thống

Mô hình **DrQA** sử dụng một cách tìm kiếm hiệu quả dựa vào mô hình vector space. Theo đó, mỗi truy vấn và mỗi đoạn văn được biểu diễn theo một vector bag-of-word, trong đó mỗi từ được biểu diễn bởi một trọng số tính theo phương pháp TF-IDF (term frequency × inverse document frequency). Chi tiết về mô hình này sẽ được mô tả tại **Mục 2**.

Cách hiện đại sử dụng mạng nơ-ron

Thay vì biểu diễn đoạn văn theo một vector space thưa, với các phương pháp tiên tiến như mạng nơ-ron (chẳng hạn như MLP, LSTM, bidirectional LSTM...), một đoạn văn có thể được biểu diễn dưới dạng một dense vector. Theo đó, một số mô hình đi theo cách tiếp cận sau:

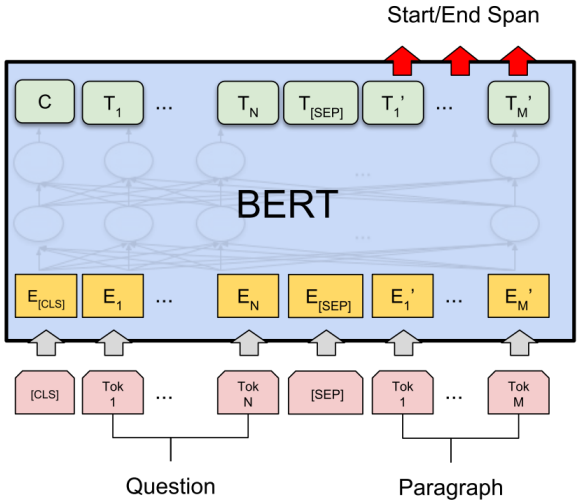
$$h_x = E_x(x) \qquad h_z = E_z(z) \qquad score(x, z) = h_x^T h_z$$

1. Trích xuất dense vector biểu diễn câu hỏi x và đoạn văn z bằng cách sử dụng một mạng nơ-ron,
2. Nhân 2 vector với nhau để thu được điểm đánh giá tương quan, từ đó xếp hạng những đoạn văn có mức độ liên quan đến câu hỏi cao nhất.

1.3.1.2 Thành phần Reader

Thành phần Reader được sử dụng với mục đích trích xuất được câu trả lời từ một đoạn văn nhất định. Theo đó, có một số phương pháp sử dụng mô hình mạng nơ-ron được áp dụng phổ biến đối với bài toán này, bao gồm:

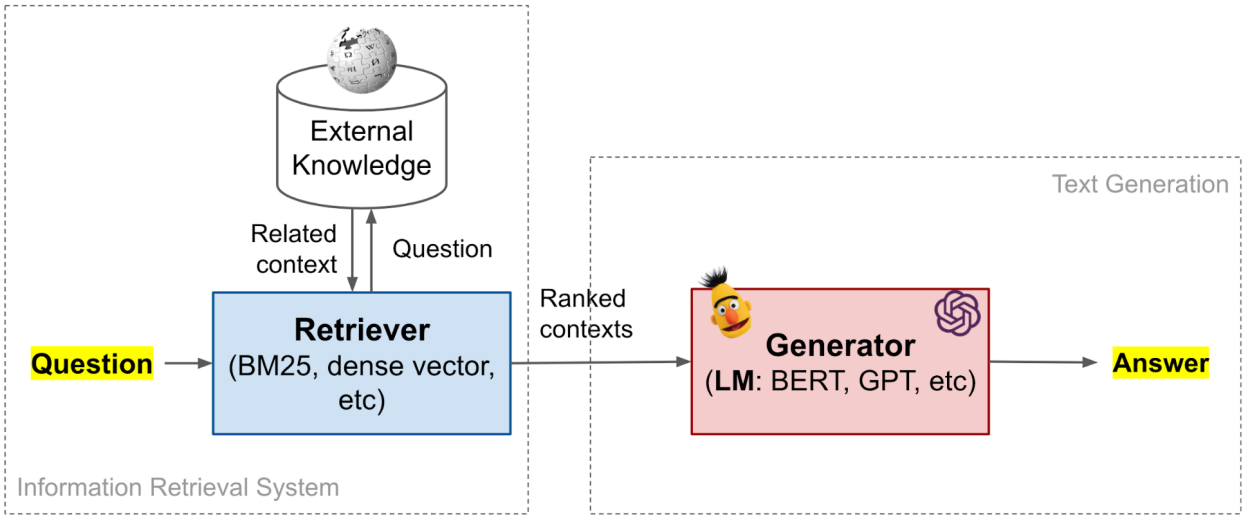
- **Bi-directional LSTM:** là mô hình được sử dụng trong DrQA.
- **Các mô hình BERT**



Hình 3: Mô hình BERT được sử dụng trong hệ thống Question Answering

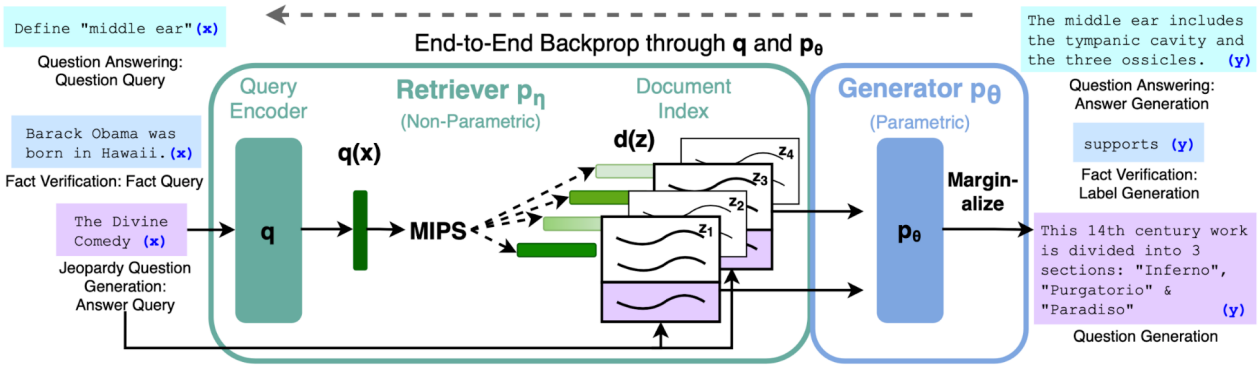
1.3.2 Kiến trúc Retriever - Generator

Khác với kiến trúc Retriever - Reader, đối với kiến trúc Retriever - Generator, mục tiêu của bước thứ 2 là sinh ra một câu trả lời cho câu hỏi hơn là trích xuất câu trả lời dựa vào một đoạn văn.



Hình 4: Kiến trúc Retriever - Generator

Theo đó, sau khi những đoạn văn đã được tìm kiếm và xếp hạng, thành phần Generator sẽ sử dụng mô hình BERT, GPT hoặc các mô hình tương tự để có thể sinh ra câu trả lời cho câu hỏi một cách hiệu quả nhất. Một mô hình hệ thống đặc trưng cho kiến trúc này là *RAG* (Retrieval-Augmented Generation), được giới thiệu bởi Lewis et al., 2020.

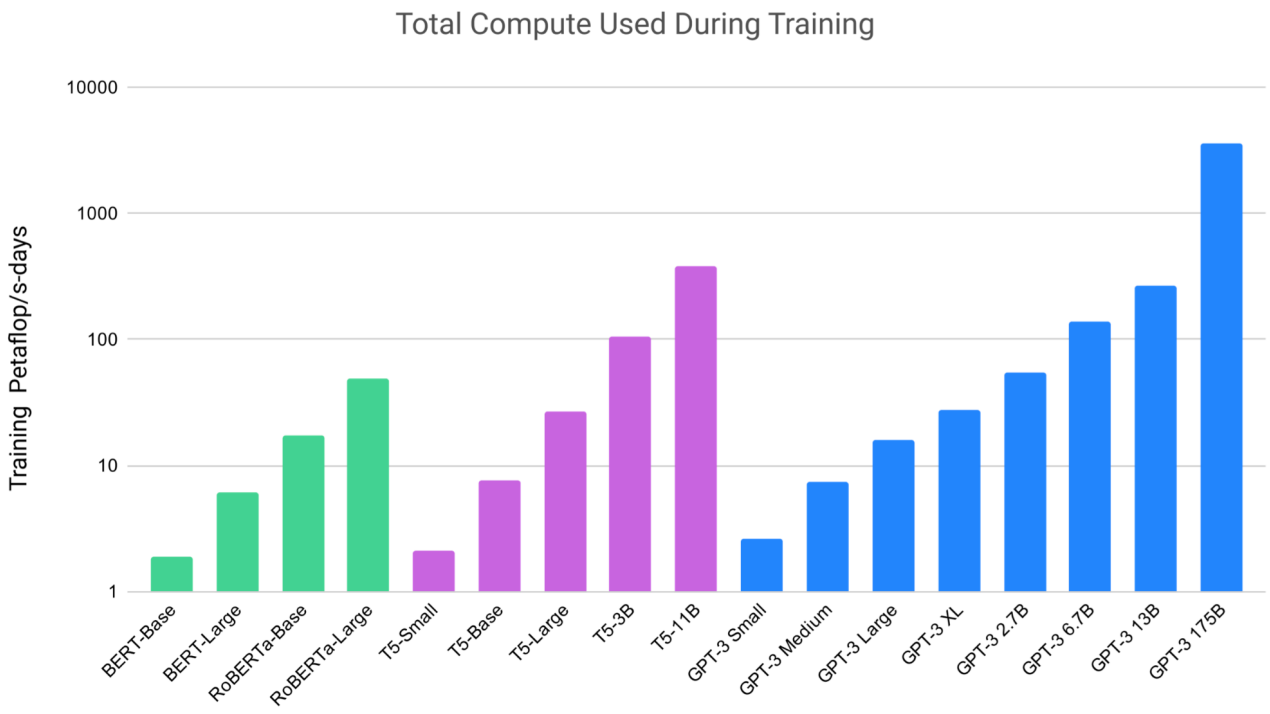


Hình 5: Mô hình RAG

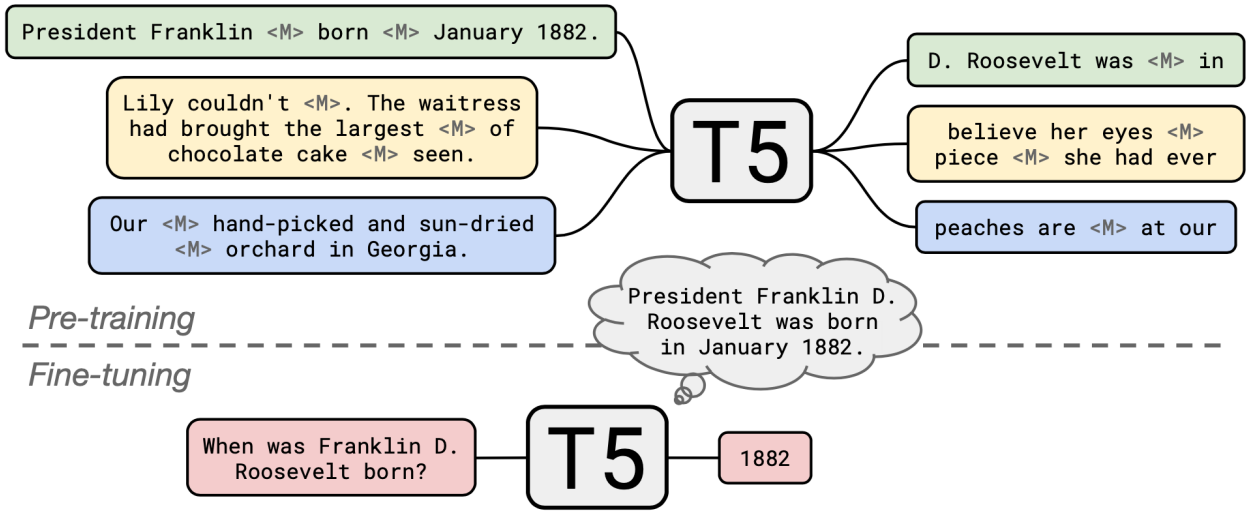
1.3.3 Kiến trúc Generator

Đối với một số mô hình ngôn ngữ có kích cỡ lớn, chúng đã được huấn luyện dựa trên một tập lớn các dữ liệu văn bản, từ đó chúng có thể ghi nhớ những kiến thức thông qua các trọng số trong những mô hình đó. Chính vì vậy, những mô hình đó có thể được sử dụng để trả lời những câu hỏi mà không cần cung cấp một ngữ cảnh cụ thể.

Theo đó, trong một bài báo được giới thiệu bởi Roberts et al. (2020), họ đã đánh giá khả năng thực tế của mô hình ngôn ngữ T5 sau khi đã được fine-tuned để trả lời câu hỏi trong điều kiện không biết trước ngữ cảnh.



Hình 6: Kích cỡ của một số mô hình ngôn ngữ phổ biến



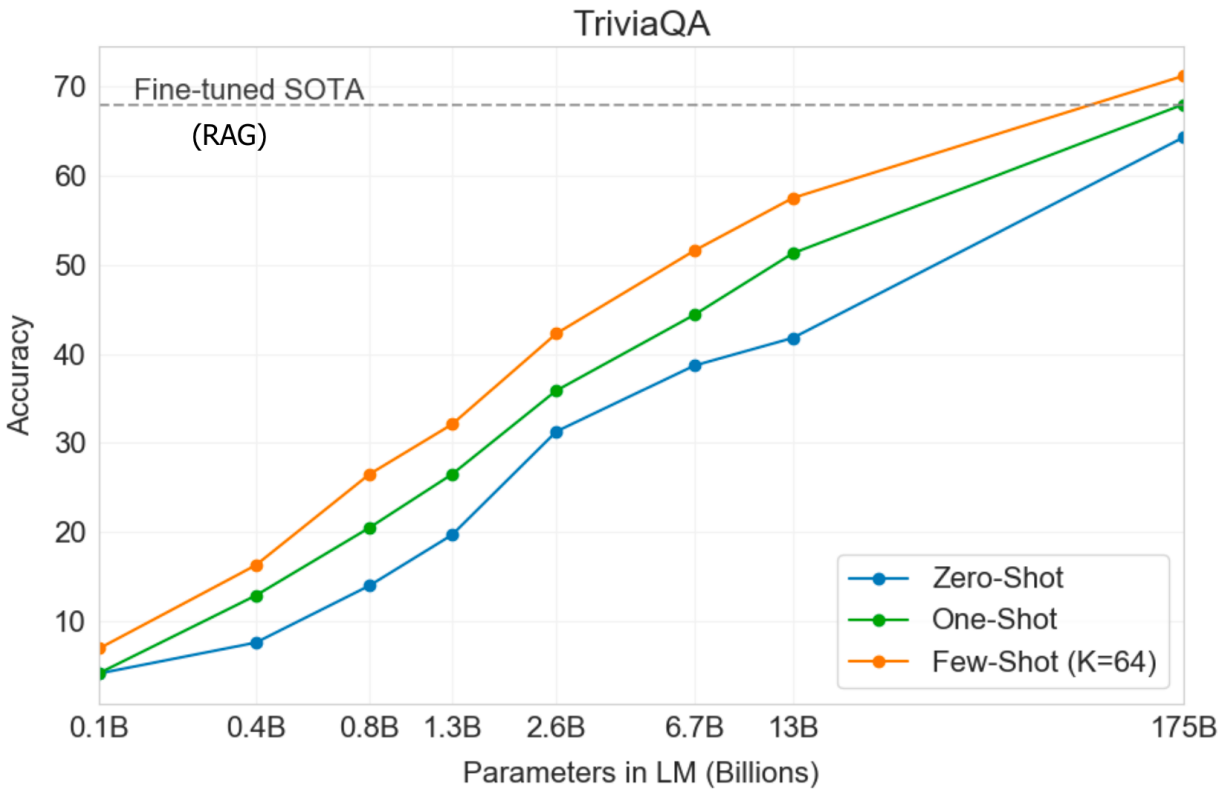
Hình 7: Mô hình T5 được fine-tuned để trả lời câu hỏi

Ngoài ra, trong một số trường hợp, một mô hình không nhất thiết cần phải được fine-tuned để có thể sử dụng trong việc trả lời câu hỏi, chẳng hạn như mô hình GPT-3. Theo đó, trong một bài báo được giới thiệu bởi Brown et al., 2020, họ đã đánh giá khả năng trả lời câu hỏi của mô hình qua 3 trường hợp cụ thể như sau:

1. **few-shot learning:** mô hình GPT-3 có thể nhận được một số chỉ dẫn về ngữ cảnh,
2. **one-shot learning:** mô hình GPT-3 chỉ được nhận duy nhất **một** chỉ dẫn về ngữ cảnh,
3. **zero-shot learning:** mô hình GPT-3 không được nhận bất kỳ một chỉ dẫn nào về ngữ cảnh.

Theo đó, với tập dữ liệu TriviaQA, khả năng của mô hình GPT-3 có thể đạt được tương đương,

hoặc thậm chí còn vượt trội hơn so với một mô hình khác đã được fine-tuned. Ngoài ra, cũng có thể thấy khả năng hoạt động của GPT-3 tỉ lệ thuận với kích cỡ của mô hình, như Hình 8 dưới đây.

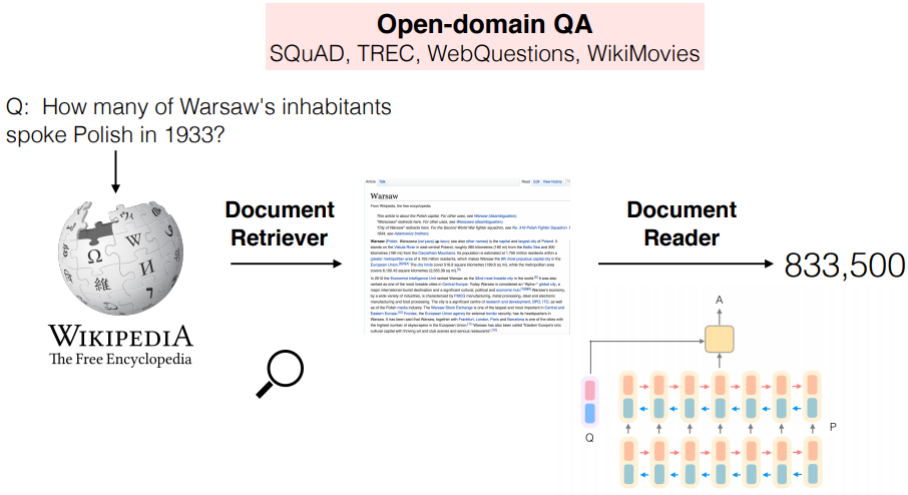


Hình 8: Khả năng của GPT-3 đối với tập dữ liệu TriviaQA

2 DrQA

2.1 Tổng quan

Mục tiêu của DrQA là trả lời các câu hỏi thuộc dạng open-domain. DrQA sử dụng nguồn dữ liệu từ Wikipedia làm cơ sở tri thức (knowledge base).



Hình 9: Tổng quan về hệ thống DrQA

Phương pháp bao gồm 2 phần chính là bộ lấy tài liệu (Document Retriver) và bộ đọc tài liệu (Document Reader). Trong đó

1. **Document Retriever** thu hẹp phạm vi tìm kiếm, từ tập các tài liệu (document) ban đầu, lấy ra một tập nhỏ các tài liệu có liên quan nhất. Cụ thể với DrQA, tập tài liệu là tập các bài viết trên Wikipedia, Document Retriever sẽ lấy ra k = 5 tài liệu liên quan nhất đến câu hỏi.

2. **Document Reader** nhận đầu vào là tập các bài viết liên quan được lấy ra từ bước Document Retriever, sau đó tìm ra đoạn văn có khả năng cao nhất là đáp án.

2.2 Document Retriever

DrQA sử dụng cách đánh giá bằng TF-IDF, một phương pháp hiệu quả được sử dụng trong các công cụ tìm kiếm (search engine) mà không cần qua quá trình học. TF-IDF là viết tắt của term frequency - inverted document frequency. Giá trị này xuất phát từ một nhận xét rằng: một từ sẽ càng có ý nghĩa với việc tìm kiếm nếu nó xuất hiện nhiều lần trong một tài liệu, và nó chỉ xuất hiện trong một số ít các tài liệu.
Để tính toán giá trị này, trước tiên, truy vấn và các tài liệu được chuyển về dưới dạng vector bag-of-word. Mỗi từ t ứng với tài liệu d và tập tài liệu D sẽ được tính giá trị tf-idf như sau:

$$tfidf(t, d, D) = tf(t, d) \times idf(t, D)$$
$$tf(t, d) = log(1 + freq(t, d))$$
$$idf(t, D) = \log \frac{|D|}{|d \in D : t \in d|}$$

Với $freq(t, d)$ là số lần từ t xuất hiện trong tài liệu d. t ở đây có thể là unigram hoặc bigram term. DrQA sử dụng một tokenizer để thực hiện phân đoạn câu cũng như xác định từ loại cho các từ trong

câu. Đối với một tài liệu d, giá trị của tài liệu đối với truy vấn q được tính như sau:

$$score(q, d) = \sum_{t \in q} tfidf(t, d).$$

DrQA mặc định lấy ra $k = 5$ tài liệu có điểm số cao nhất với câu hỏi.

2.3 Document Reader

Document reader sẽ nhận đầu vào là một câu hỏi q được biểu diễn dưới dạng $\{q_1, q_2, ..., q_l\}$ với l là số tokens ở trong câu hỏi và một tập các tài liệu. Trong đó, mỗi tài liệu được chia thành các đoạn văn, mỗi đoạn văn p được biểu diễn bởi $\{p_1, p_2, ..., p_m\}$ với m là số tokens của đoạn văn. Mô hình document reader sẽ có 3 phần chính: paragraph encoding, question encoding và prediction.

2.3.1 Paragraph Encoding

Ta biểu diễn các token p_i dưới dạng một vector đặc trưng $\tilde{p_i}$. Vector đặc trưng $\tilde{p_i}$ bao gồm những thành phần sau:

- Word embedding: $f_{emb}(p_i) = E(p_i)$. Glove word embedding được sử dụng để thực hiện việc embedding. Đa số các từ pre-trained word embedding đều được cố định và chỉ thực hiện fine-tune 1000 từ thường xuyên xuất hiện nhất.
- Exact match: $f_{exact\_match}(p_i) = \Pi(p_i \in q)$. Sử dụng 3 binary features để biểu diễn p_i có khớp với một từ ở trong câu hỏi q, có thể ở dạng ban đầu (giống như trong câu hỏi), không phân biệt chữ hoa, thường (case-insensitive), hoặc ở dạng nguyên bản (lemma form, ví dụ như các từ break, breaks, broke và broken có chung dạng nguyên bản là break).
- Đặc trưng của token: $f_{token}(p_i) = (POS(p_i), NER(p_i), TF(p_i))$. Các đặc trưng về từ loại (POS - part of speech), loại thực thể của các đối tượng có tên (NER - named entity recognition), và tần suất xuất hiện của từ (TF - term frequency).
- Aligned question embedding: $f_{align}(p_i) = \sum_j a_{ij} E(q_j)$, trong đó trọng số attention a_{ij} thể hiện sự gióng hàng (alignment) giữa p_i và mỗi từ q_j trong câu hỏi. Cụ thể, a_{ij} được tính bằng tích chấm giữa các ánh xạ phi tuyến (nonlinear mappings) của các word embedding:

  $$a_{ij} = \frac{exp(\alpha(E(p_i)) \cdot \alpha(E(q_j)))}{\sum_{j'} exp(\alpha(E(p_i)) \cdot \alpha(E(q_{j'})))}$$

  trong đó $\alpha()$ là một lớp fully-connected với hàm activation phi tuyến ReLU. So với đặc trưng exact match, đặc trưng này bổ sung thêm các gióng hàng tương đối (soft alignment) giữa các từ tương tự nhưng không giống nhau (ví dụ: car và vehicle).

Tập vector đặc trưng $\tilde{p}_1, \tilde{p}_2, \ldots, \tilde{p}_m$ sau đó sẽ được đưa qua LSTM để thu được:

$$\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_m\} = RNN(\{\tilde{p}_1, \tilde{p}_2, \ldots, \tilde{p}_m\})$$

với $\mathbf{p}_i$ được kì vọng sẽ mã hóa các thông tin về ngữ cảnh.

2.3.2 Question Encoding

Câu hỏi q được encode thành tổng có trọng số của các từ trong câu hỏi. Cụ thể, vector $\mathbf{q}$ được biểu diễn dưới dạng $\mathbf{q} = \sum b_j \times q_j$, trong đó b_j thể hiện độ quan trong của mỗi từ trong câu hỏi. Công thức tính b_j

$$b_j = softmax(w^T E(x_j))$$

Trong đó w là một vector có thể học được.

2.3.3 Prediction

Sau khi có các vector đặc trưng của câu hỏi và các đoạn văn, với mỗi vị trí i, ta cần tính toán xác suất là vị trí bắt đầu $P_{start}(i)$ và vị trí kết thúc $P_{end}(i)$.

$$P_{start}(i) \propto exp(\mathbf{p}_i \mathbf{W}_s \mathbf{q})$$
$$P_{end}(i) \propto exp(\mathbf{p}_i \mathbf{W}_e \mathbf{q})$$

Trong đó, $\mathbf{W}_s$ và $\mathbf{W}_e$ là các tham số có thể học được. Ta sẽ cần tìm 2 chỉ số i_s và i_e sao cho $P_{start}(i_s) \times P_{end}(i_e)$ đạt giá trị lớn nhất với điều kiện $i_s \leq i_e \leq i_s + 15$.