BACHELOR THESIS

The proposed Dual Encoder model for Open-domain question answering system: Case study in Vietnamese COVID-19 topic

Le Vu Loi - 20173240

Talented class of Computer Science

Supervisor: Assoc. Prof. Pham Van Hai
Department of Information System

July 9, 2021

- Kính thưa thầy Chủ tịch Hội đồng và các thầy cô trong hội đồng, thưa toàn thể các bạn. Em là Lê Vũ Lợi, sinh viên lớp Tài năng Công nghệ thông tin K62. Hôm nay em xin trình bày đồ án tốt nghiệp của mình với tên đề tài là "The proposed Dual Encoder model for Open-domain question answering system: Case study in Vietnamese COVID-19 topic". Đề tài được thực hiện dưới sự hướng dẫn của PGS.TS. Phạm Văn Hải. Sau đây em xin được bắt đầu.

Outline

- Bố cục phần trình bày của em gồm có 5 phần: thứ nhất là phần giới thiệu tổng quan, thứ 2 là các nghiên cứu liên quan, thứ 3 là mô hình đề xuất, thứ 4 là áp dụng mô hình nghiên cứu cho dữ liệu tiếng Việt. Thứ 5 là kết quả thực nghiệm và cuối cùng là kết luận và hướng phát triển.

Open-domain question answering

Machine Reading Comprehension

Summarization

Word embedding

Dependency Parsing

Named Entity Recognition

Open-domain question answering

Part-of-speec Tagging

Machine Translation

Language Modeling

Sentiment Analysis

Dialogue Management

Sentence Embedding

Information retrieval

Keyword Extraction

- Em xin đi vào phần đầu tiên là giới thiệu tổng quan.
- Bài toán em nghiên cứu mang tên là Open-domain question answering (hệ thống trả lời câu hỏi miền mở)
- Đây là một bài toán ở mức độ tương đối khó trong lĩnh vực xử lý ngôn ngữ tự nhiên, nó dựa trên các bài toán nền tảng hơn như: word embedding,language modeling, question answering, information retrieval hay machine reading comprehension.

Open-domain question answering

- Combination of **retriever** (Information Retrieval) and **reader** (Machine Reading Comprehension)
  - "Skim through" a large data source to find a subset of relevant documents.
  - "Swallow" each document to find the exact answer(s).

- Cụ thể hơn thì có thể xem bài toán open-domain question answering là sự kết hợp của bài toán information retrieval và machine reading comprehension.
- Trong đó, thành phần retriever sẽ thực hiện tìm kiếm sơ bộ trên một kho văn bản kích thước lớn có sẵn để lọc ra một tập nhỏ các văn bản liên quan nhất tới một câu hỏi của người dùng
- Tiếp đó, thành phần reader có nhiệm vụ đọc kĩ tập văn bản này để tìm ra câu trả lời chính xác nhất

Problem formulation

- **Input**
  - A question in human natural language.

    E.g. Who is the founder of Google?
- **Output**
  - A list of answers for the input question

    E.g. [Larry Page, Sergey Brin]
- **Constraints**
  - The system answers only factoid question.

- Phát biểu bài toán Open-domain question answering
- Bài toán nhận đầu vào là một câu hỏi của người dùng dưới dạng ngôn ngữ tự nhiên, đầu ra của bài toán là một hoặc một danh sách các câu trả lời tương ứng với câu hỏi đã cho. Ràng buộc của bài toán là hệ thống chỉ trả lời các câu hỏi có tính chất factoid
- Một câu hỏi có tính chất factoid nếu nó liên quan đến một sự thật hiển nhiên. Ví dụ như câu hỏi "Thành phố nào là thủ đô của Việt Nam". Ví dụ về câu hỏi không có tính chất factoid đó là câu "Tại sao anh ta không thích làm điều này". Câu hỏi này phụ thuộc vào một ngữ cảnh cụ thể nên không có tính chất factoid. Các câu hỏi dạng yes/no, câu hỏi trắc nghiệm hay câu hỏi suy diễn cũng không thuộc dạng câu hỏi factoid.

Related works

[1] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. arXiv preprint arXiv:1704.00051, 2017.

[2] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. arXiv preprint arXiv:2004.04906, 2020.

- Phần thứ hai em xin trình bày về các nghiên cứu liên quan

Reading Wikipedia to answer open-domain questions

- **Retriever**: bigram hashing and TF-IDF matching
- **Reader**: Multi-layer recurrent neural network
- Potential improvements: using neural network to better capture documents' semantics

- Nghiên cứu thứ nhất mang tên "Reading wikipedia to answer open-domain question answering"
- Nghiên cứu sử dụng thuật toán bigram hashing và TF-IDF matching cho thành phần retriever và mạng hồi quy nhiều tầng cho thành phần reader.
- Nghiên cứu này đã đánh dấu một cột mốc trong các nghiên cứu về bài toán ODQA và thu hút sự chú ý của một loạt các nghiên cứu sau đó đối với cùng chủ đề.
- Một hướng cải thiện tiềm năng cho nghiên cứu này đó là sử dụng deep learning cho thành phần retriever để mô hình có thể bắt được ngữ nghĩa của các văn bản một cách tốt hơn.

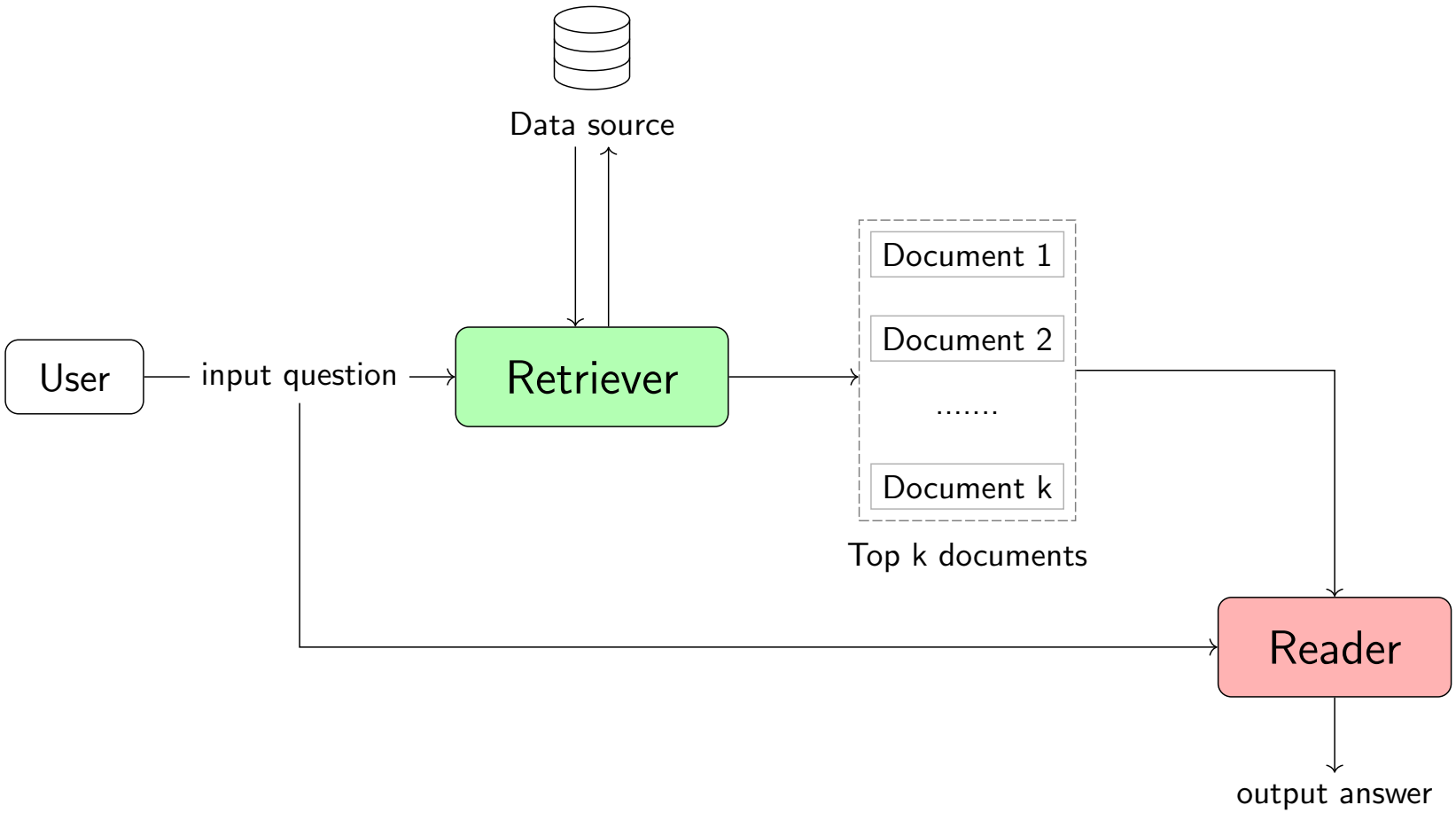
Dense passage retrieval

- **Retriever**: Dual-encoder
- **Reader**: Cross-encoder
- Successfully use neural network to solve information retrieval.
- Potential improvements: More challenging learning task for the system to gain deeper language understanding.

- Nghiên cứu sử dụng kiến trúc dual encoder cho thành phần retriever và kiến trúc cross-encoder cho thành phần reader. Đây là các kiến trúc mạng nơ ron được sử dụng gần đây trong lĩnh vực xử lý ngôn ngữ tự nhiên.
- Nghiên cứu này lần đầu tiên sử dụng deep learning cho bài toán Informational retrieval và kết quả thu được vượt xa so với hướng tiếp cận truyền thống không sử dụng deep learning.
- Một hướng cải thiện tiềm năng cho nghiên cứu này đó là sử dụng mục tiêu học khó hơn để thách thức mô hình và giúp mô hình đạt được mức độ hiểu ngôn ngữ sâu hơn.

System pipeline

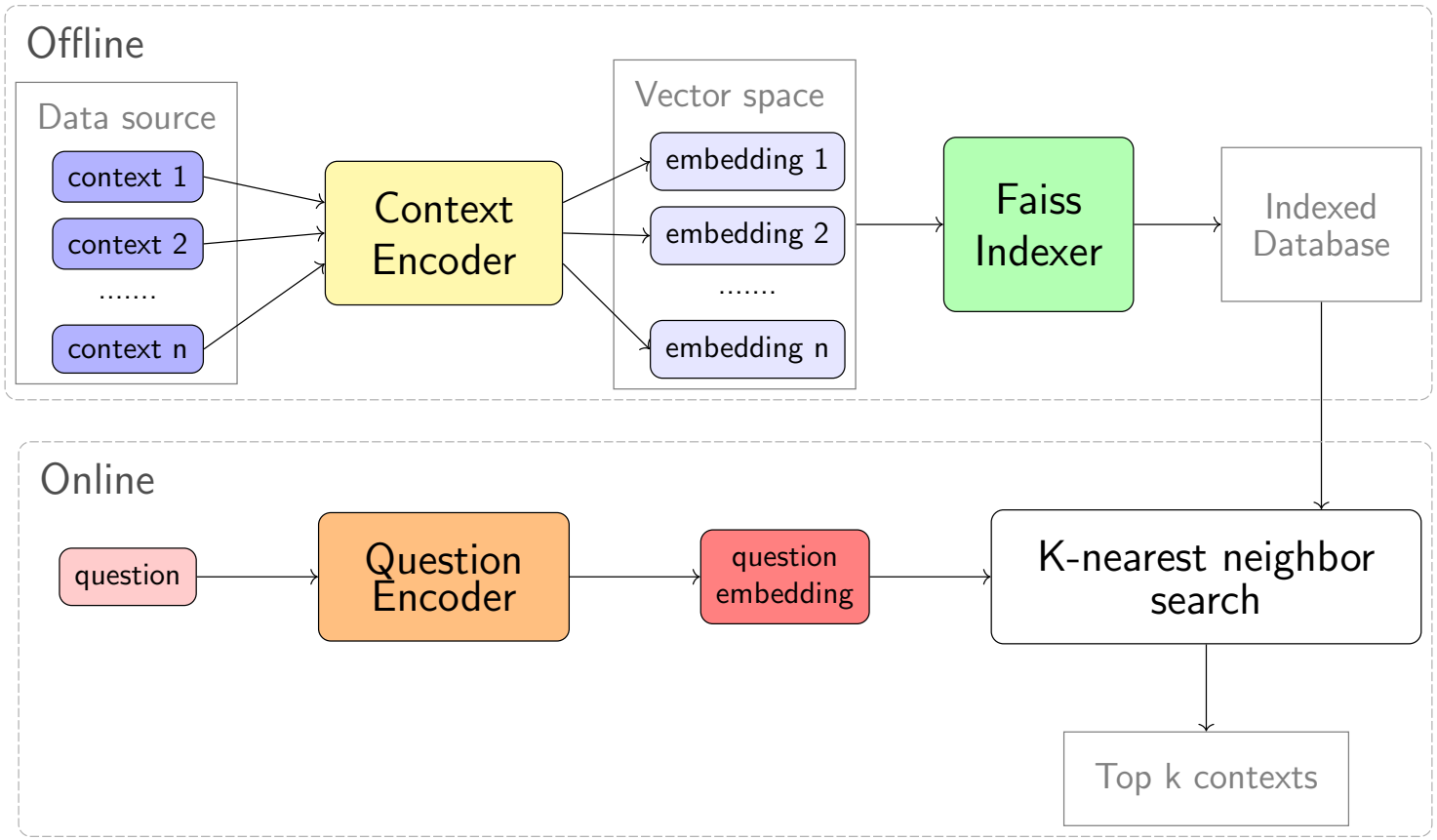
- Open-domain question answering = Retriever + Reader



- Phần thứ 3 của bài trình bày em xin đi vào mô hình đề xuất
- Trước tiên, em sẽ trình bày về kiến trúc tổng quan của hệ thống open-domain question answering.
- Hệ thống gồm 2 thành phần chính là retriever và reader.
- Thành phần retriever có nhiệm vụ tìm kiếm trong một kho văn bản với kích thước lớn để tìm ra một tập con các văn bản có liên quan nhất tới câu hỏi đầu vào của người dùng
- Thành phần reader có nhiệm vụ đọc các văn bản trả về bởi retriever và tìm ra câu trả lời chính xác

Dense retriever: Dual encoder architecture

- Dense retriever is based on Dual encoder architecture.



Workflow of a dense retriever

- Em sẽ đi vào thành phần đầu tiên là retriever. Thành phần này dựa trên kiến trúc dual encoder, bao gồm 2 mạng mã hóa độc lập là Context Encoder và Question Encoder.
- Như thể hiện trên hình vẽ thì luồng làm việc của retriever bao gồm 2 pha. Ở pha offline, mạng context encoder sẽ mã hóa tất cả các văn bản trong kho văn bản thành các vector trong không gian vector. Các vector này sau đó được đánh chỉ mục để phục vụ việc tìm kiếm nhanh ở pha online.
- Ở pha online, khi người dùng đưa ra một câu hỏi thì hệ thống sẽ sử dụng thành phần question encoder để mã hóa câu hỏi thành vector question embedding. Vector này sau được tìm kiếm trên kho văn bản mã hóa đã được đánh chỉ mục để trả về top-k văn bản có liên quan nhất đến câu hỏi người dùng.

Training dense retriever

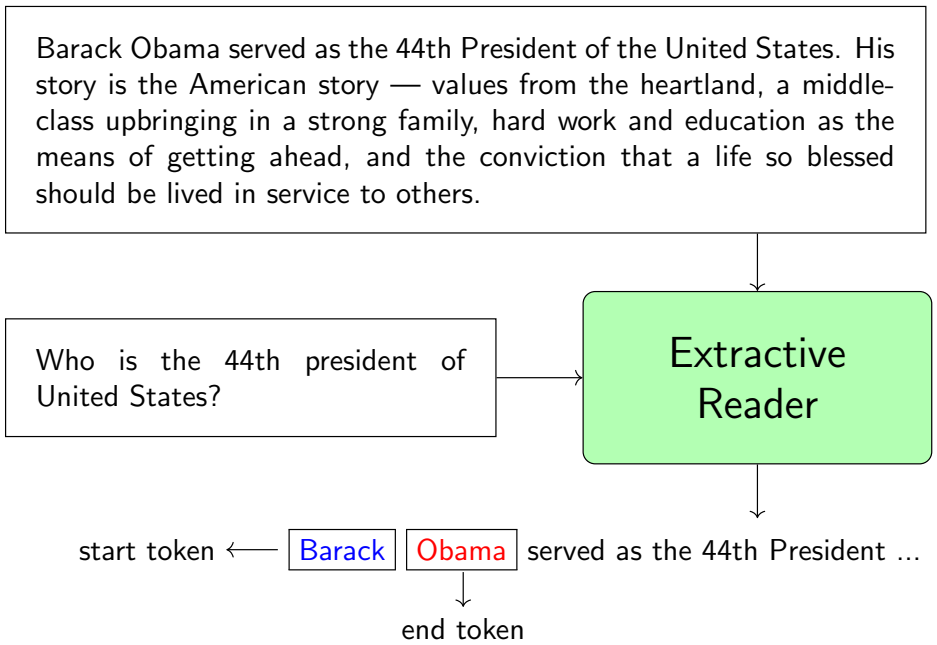
- Jointly train question encoder and context encoder.
- Training data: a training sample consists of:
  - q: input question.
  - p^+: positive context, which is the document that contains the answers.
  - $\left\{p_j^-\right\}_{j=1}^m$: m negative contexts, which are documents that do not contain the answers.
- Loss function (per one training sample): negative log-likelihood

$$\mathcal{L} = -\log\left\{\frac{\exp\left[\text{sim}\left(q, p^+\right)\right]}{\exp\left[\text{sim}\left(q, p^+\right)\right] + \sum_{j=1}^{m}\exp\left[\text{sim}\left(q, p_j^-\right)\right]}\right\} \quad (1)$$

- Để huấn luyện thành phần retriever, ta thực hiện tối ưu đồng thời 2 mạng question encoder và context encoder.
- Một mẫu dữ liệu sử dụng huấn luyện retriever bao gồm một câu hỏi, một văn bản positive (tức là văn bản chứa câu trả lời tương ứng với câu hỏi đã cho) và m văn bản negative, là các văn bản không chứa câu trả lời cho câu hỏi đã cho.
- Hàm loss sử dụng là hàm negative log likelihood. Hàm này cố gắng cực đại hóa độ tương đồng giữa câu hỏi đầu vào với văn bản positive, đồng thời cực tiểu hóa độ tương đồng giữa câu hỏi với các văn bản negative.

Extractive reader: Cross encoder architecture

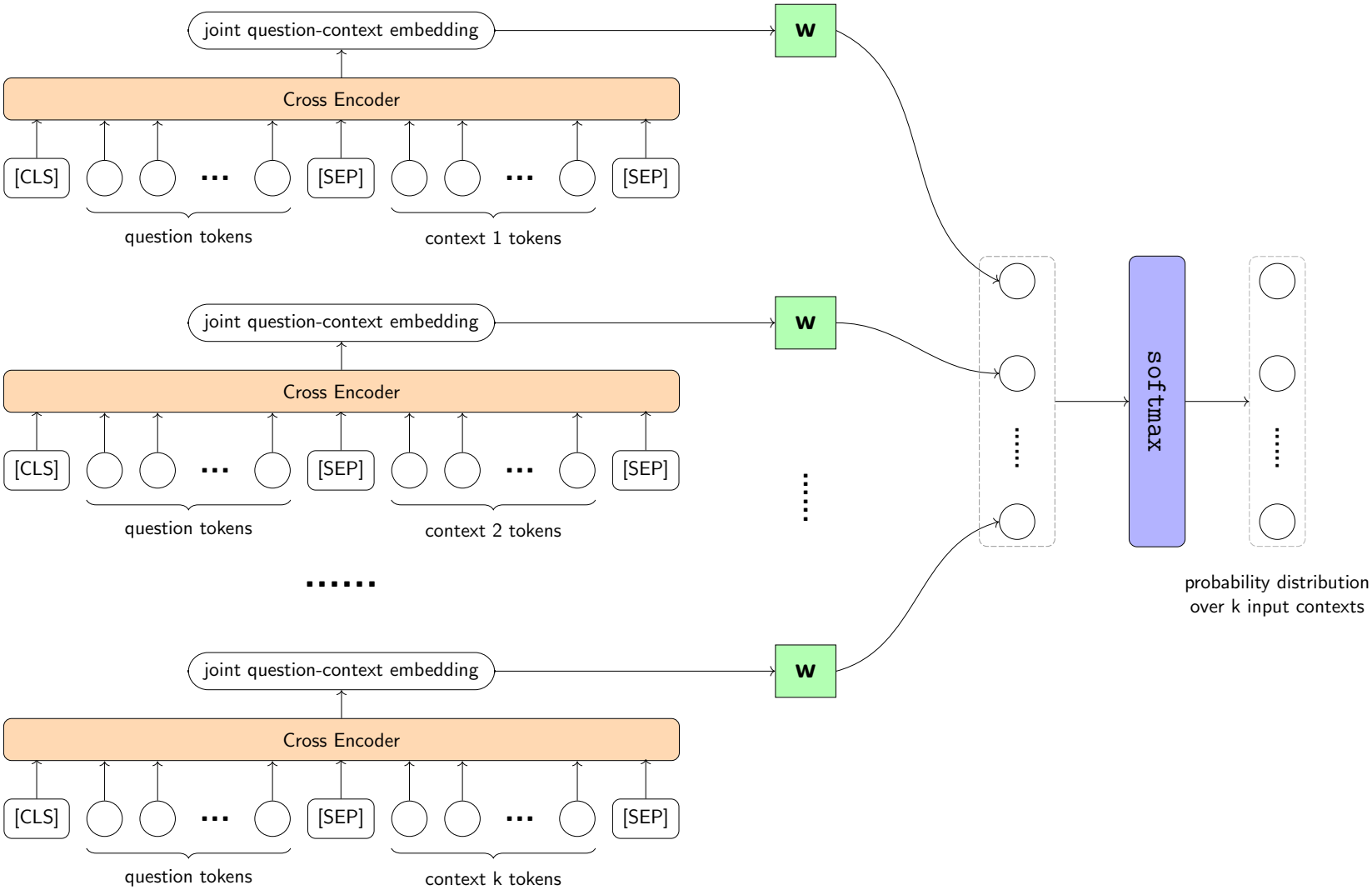
- Extractive reader's task is to predict the start and end position of answer in the documents returned by dense retriever.



- Extractive reader consists of 2 components, in which each component follows a cross encoder architecture:
  - Re-ranker: re-rank documents returned by dense retriever.
  - Single-document reader: read one document to extract answers.

- Thành phần thứ 2 của hệ thống là reader. Cụ thể đồ án của em sử dụng extractive reader. Extractive ở đây có nghĩa là reader sẽ trích xuất ra câu trả lời từ văn bản nó đang đọc, đồng nghĩa với việc câu trả lời phải là một cụm từ nằm trong văn bản, một ví dụ được đưa ra như hình ảnh trên slide. Một loại reader khác cũng được cộng đồng nghiên cứu quan tâm mang tên là generative reader. Đối với loại này thì câu trả lời không bị ràng buộc phải nằm trong văn bản mà mô hình có thể sinh ra bất kỳ câu trả lời nào mà nó nghĩ là hợp lý nhất. Tuy nhiên loại reader này đòi hỏi số lượng tham số lớn cũng như đặt ra nhiều thách thức hơn trong việc huấn luyện.
- Đối với thành phần reader thì em lại chia nhỏ thành 2 thành phần con đó là re-ranker và single-document reader. Thành phần re-ranker có nhiệm vụ sắp xếp lại thứ hạng của các văn bản trả về bởi retriever và sau đó thành phần single-document reader sẽ đọc các văn bản được xếp hạng bởi re-ranker theo thứ tự từ trên xuống dưới.

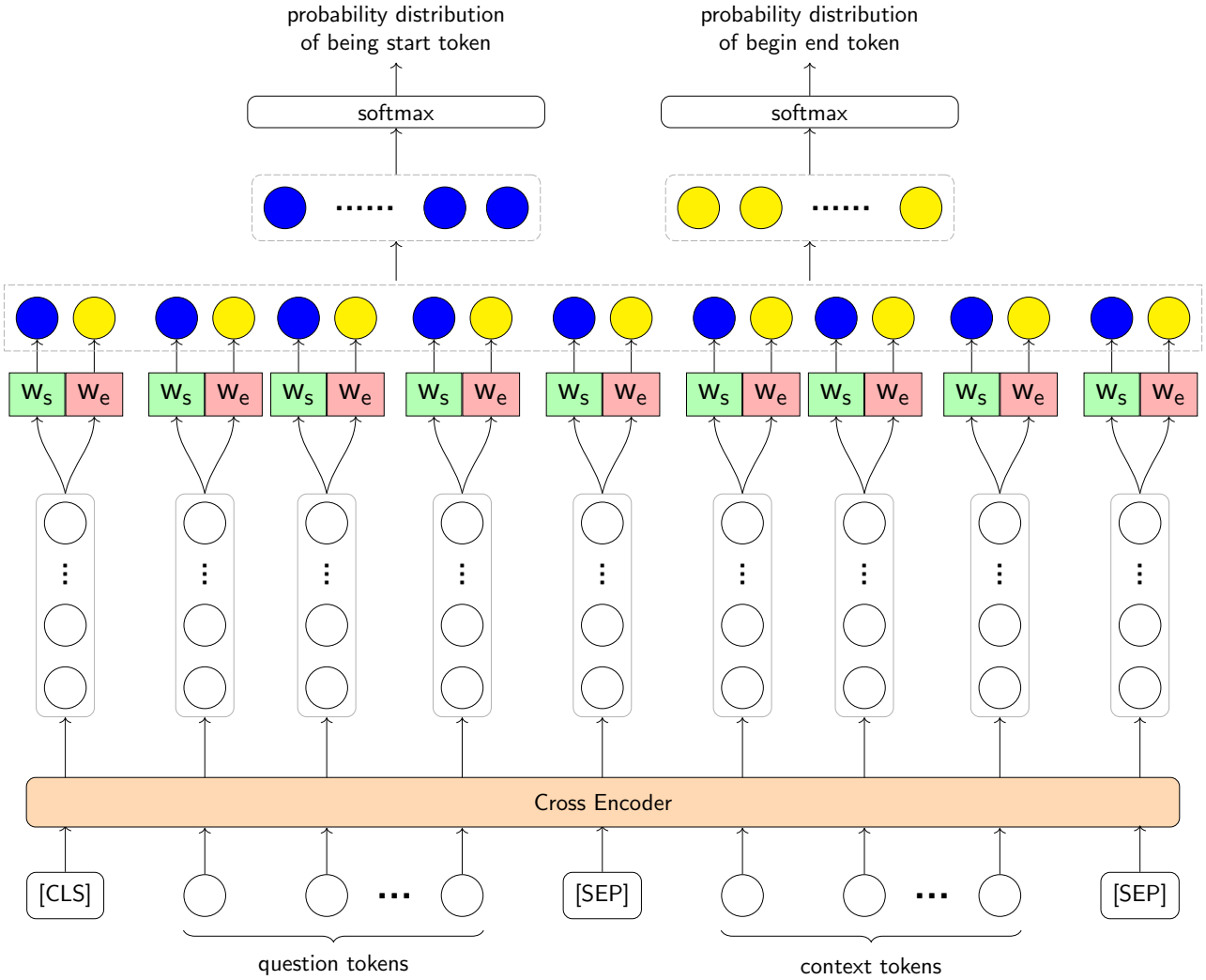
Re-ranker



Model architecture

- Slide này trình bày kiến trúc của thành phần re-ranker. Re-ranker tuân theo kiến trúc cross-encoder, nhận đầu vào là k chuỗi ghép nối giữa câu hỏi đầu vào và k văn bản đang cần được xếp hạng. Mỗi chuỗi này sau khi đi qua mạng cross-encoder sẽ trả về một vector mã hóa đồng thời cho câu hỏi và văn bản. Vector này sau đó được nhân với một ma trận w và trả về một score tương ứng với khả năng mà văn bản tương ứng là văn bản chứa câu trả lời. Cuối cùng k score này được chuẩn hóa bằng hàm softmax để trả về một phân phối xác suất ứng với xác suất một văn bản có chứa câu trả lời.

Single-document reader



Model architecture

- Tiếp đến là kiến trúc của thành phần single-document reader. Tương tự như thành phần re-ranker, single-document reader cũng tuân theo kiến trúc cross encoder. Thay vì xử lý đồng thời k chuỗi đầu vào thì single-document chỉ xử lý một chuỗi đầu vào, là ghép nối giữa câu hỏi của người dùng với văn bản mà single-document reader đang xử lý. Mỗi một token trong chuỗi đầu vào được mã hóa thành một vector và vector này sau đó được nhân lần lượt với các ma trận ws, we để sinh ra 2 score tương ứng với khả năng token đang xét là vị trí bắt đầu hay vị trí kết thúc của câu trả lời (2 score được biểu diễn bằng các hình tròn màu xanh dương và màu vàng như trên hình vẽ). Tương tự re-ranker, 2 score này cũng được chuẩn hóa qua hàm softmax để sinh ra 2 phân phối xác suất tương ứng với xác suất token đang xét là vị trí bắt đầu hay kết thúc của câu trả lời.

Proposed method: stratified loss for training dual encoder

- Idea: additional loss for learning difference between hard negative and normal negative contexts.
- Stratified loss
  - Assumptions: a batch of b training samples $\mathcal{D}$, where the i-th training sample $\mathcal{D}_i$ consists of:
    - q_i: input question.
    - p_i^+: positive context.
    - $\left\{p_{i,j}^-\right\}_{j=1}^m$: m hard negative contexts.
  - Loss formula

$$\begin{aligned}\mathcal{L} = & -\log\left\{\frac{\exp\left[\text{sim}\left(q_i, p_i^+\right)\right]}{\exp\left[\text{sim}\left(q_i, p_i^+\right)\right] + \sum_{j=1}^{m}\exp\left[\text{sim}\left(q_i, p_{i,j}^-\right)\right]}\right\} \\ & -\sum_{j=1}^{m}\log\left\{\frac{\exp\left[\text{sim}\left(q_i, p_{i,j}^-\right)\right]}{\exp\left[\text{sim}\left(q_i, p_{i,j}^-\right)\right] + \sum_{k\in\{1,2,\ldots,b\}\setminus\{i\}}\exp\left[\text{sim}\left(q_i, p_k^+\right)\right]}\right\}\end{aligned} \tag{2}$$

- Tiếp theo em xin trình bày về đề xuất sử dụng hàm stratified loss để huấn luyện retriever.
- Ý tưởng của hàm loss là học được sự khác nhau giữa các văn bản hard negative và văn bản negative thông thường. Đặc điểm của văn bản hard negative là nó nội dung tương đồng với câu hỏi, tuy nhiên không chứa thông tin cần thiết để trả lời cho câu hỏi. Ví dụ nếu ta có một câu hỏi là "Ca sĩ A quê ở đâu?", và một văn bản đề cập đến việc "Ca sĩ A sinh năm bao nhiêu". Văn bản này sẽ được coi là hard negative đối với câu hỏi đã cho vì nó cùng đề cập tới đối tượng được nhắc đến trong câu hỏi là ca sĩ A nhưng lại không trả lời cho câu hỏi đó.
- Công thức cho hàm stratified được thể hiện như trên slide. Về cơ bản, đây vẫn là hàm negative log-likelihood, tuy nhiên nó xem xét 2 thành phần. Thành phần thứ nhất, ứng với dòng đầu tiên của công thức sẽ giúp mô hình phân biệt giữa các văn bản positive với văn bản hard negative, trong khi thành phần thứ hai ứng với tổng sigma ở dòng thứ 2 sẽ học sự khác nhau giữa văn bản hard negative với văn bản negative thông thường.
- Động cơ để em đến với đề xuất này đó là khi tìm hiểu và thực nghiệm mô hình, em nhận thấy mô hình phân biệt rất tốt giữa văn bản positive với các văn bản negative thông thường không có liên quan, tuy nhiên lại gặp khó khăn khi phải phân biệt giữa các văn bản có độ tương đồng lớn.

Case study on Vietnamese COVID-19 topic

- Building an open-domain question answering for Vietnamese COVID-19 topic requires:
  - Building a context source for COVID-19 topic, which contains all documents that the system searches during answering a question about COVID-19 topic.
  - Annotate data for training dense retriever and extractive reader (re-ranker and single-document reader).

- Phần thứ 4 của bài trình bày em xin được nói về một case study của hệ thống open-domain question answering sử dụng cho tiếng việt về chủ đề dịch bệnh covid-19.
- Thì việc xây dựng một hệ thống open-domain question answering trước hết bắt đầu bằng việc xây dựng dữ liệu, gồm có 2 bước:
  - Bước thứ nhất đó là xây dựng kho văn bản. Kho văn bản này cần đủ lớn để chứa được tất cả các thông tin liên quan đến chủ đề hỏi đáp mà người xây dựng đang hướng tới, cụ thể với đồ án của em là chủ đề dịch bệnh covid-19
  - Bước thứ 2 là gán nhãn dữ liệu sử dụng cho việc huấn luyện các thành phần của hệ thống

Data crawling for COVID-19 data

- Context source: 168,388 contexts/documents about medial topic, mainly crawled from `https://suckhoedoisong.vn/`
- Training data: 995 training samples, in which each sample consists of:
  - Input question
  - One positive context
  - One hard negative context
  - List of answers

- Đối với việc crawl dữ liệu, em thực hiện crawl khoảng 170000 văn bản từ các trang báo về y tế, chủ yếu là từ trang sức khỏe đời sống để sử dụng làm kho văn bản cho hệ thống.
- Dữ liệu huấn luyện thì em gán nhãn tổng cộng 995 mẫu dữ liệu, mỗi mẫu bao gồm:
  - câu hỏi đầu vào
  - 1 văn bản positive
  - 1 văn bản hard negative
  - Danh sách các câu trả lời tương ứng với câu hỏi đã cho

Data annotating

Gán nhãn dữ liệu hỏi đáp covid-19

WELCOME, **LEVULOI**. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Tag › Qa samples

Select qa sample to change

ADD QA SAMPLE +

Action: -------- Go 0 of 50 selected

LINK	ID	POSITIVE	QUESTION	HARD NEGATIVE	ANSWERS
Edit	68	**Dịch COVID-19: Ca tử vong thứ 59 là bệnh nhân nam 76 tuổi, viêm đa khớp ở Bắc Ninh** Suckhoedoisong.vn - Trưa 13/6, Tiểu ban điều trị - Ban Chỉ đạo Quốc gia phòng chống dịch COVID-19 thông báo ca tử vong số 59 là bệnh nhân nam, 76 tuổi ở Bắc Ninh có tiền sử viêm đa khớp mới được phát hiện, sống trong vùng có nhiều ca bệnh COVID-19.	Ca tử vong thứ 59 do Covid-19 bao nhiêu tuổi?	**Bệnh nhân COVID-19 ở Bắc Ninh tử vong, ca tử vong thứ 59** Ngày 23-5, bệnh nhân có kết quả xét nghiệm dương tính với SARS-CoV-2. Tình trạng suy hô hấp của bệnh nhân không cải thiện, tổn thương phổi tiến triển nặng dần. Bệnh nhân được đặt nội khí quản, thở máy, vận mạch, hội chẩn, chuyển Bệnh viện Bệnh nhiệt đới trung ương ngày 3-6, với chẩn đoán: sốc nhiễm khuẩn, suy đa tạng, viêm phổi ARDS nặng do SARS-CoV-2, viêm đa khớp, xuất huyết tiêu hóa do loét tá tràng.	76 tuổi
Edit	69	**Dịch COVID-19: Ca tử vong thứ 59 là bệnh nhân nam 76 tuổi, viêm đa khớp ở Bắc Ninh** Tiểu ban điều trị - Ban Chỉ đạo Quốc gia phòng chống dịch COVID-19 thông báo ca tử vong số 59: BN5355, nam, 76 tuổi, có địa chỉ tại Thuận Thành, Bắc Ninh. Tiền sử: Viêm đa khớp mới được phát hiện, sống trong vùng có nhiều ca bệnh COVID-19. Ngày 8/5	Ca tử vong thứ 59 do Covid-19 ở Việt Nam có địa chỉ ở đâu?	**Bệnh nhân COVID-19 ở Bắc Ninh tử vong, ca tử vong thứ 59** Theo tiểu ban điều trị, do bệnh nhân tuổi cao, thể trạng gầy yếu, không đáp ứng với điều trị, suy đa tạng ngày càng tăng nên tử vong sáng sớm ngày 12-6. Chẩn đoán tử vong: sốc nhiễm khuẩn, suy đa tạng, viêm phổi ARDS nặng do SARS-CoV-2 trên bệnh nhân	Bắc Ninh

- Trên hình là một giao diện web em sử dụng để gán nhãn và quản lý dữ liệu được gán nhãn, bao gồm các trường thông tin tương ứng như em đã trình bày trong slide trước.

Datasets

- Google Natural Question: preprocessed data taken from [2]
  - 58,880 training samples
  - 8,757 development samples
  - 3,610 test samples
  - Context source contains 21,015,324 contexts
  - To rapidly produce experiments, the context source is reduced to 700,000 contexts and 450 additional contexts are considered to cover all input questions in the test set.
- Vietnamese COVID-19 datatset
  - 995 training samples
  - Context source contains 168,388 contexts

- Em xin đi vào phần thứ 5 của bài trình bày đó là phần kết quả thực nghiệm
- Bộ dữ liệu mà em sử dụng cho phần thực nghiệm là bộ Google natural question, đã được tiền xử lý, lấy từ nghiên cứu số 2. Tập dữ liệu bao gồm 58880 mẫu dữ liệu huấn luyện, 8757 mẫu dữ liệu development và 3610 mẫu dữ liệu test. Kho văn bản bộ dữ liệu sử dụng bao gồm 21 triệu văn bản khác nhau.
- Đây là một kho văn bản có kích thước cực kỳ lớn và để có thể thực nghiệm số lượng lớn thì em đã thu gọn kích thước của kho văn bản xuống còn 700000, đảm bảo rằng kho văn bản mới luôn chứa văn bản positive cho tất cả các câu hỏi trong tập test. Một phần nhỏ câu hỏi dữ liệu trong tập test (khoảng 450 câu) không có văn bản positive tương ứng trong tập test sẽ được em gán nhãn thủ công và bổ sung.

Metrics

- Top-k hit scores
  - Measure retriever's accuracy
  - Top-k hit is reached if at least one of k contexts returned by the retriever contains answer(s) for input question.
- Exact match
  - Measure reader's accuracy
  - Measure end-to-end system's accuracy
  - An exact match hit is reached if answer(s) produced by the open-domain question answering system matches exactly the ground truth answer(s)

- Về độ đo sử dụng, em sử 2 độ đo. Độ đo thứ nhất là top-k hit, đo độ chính xác của retriever. retriever sẽ nhận được một điểm top-k hit nếu ít nhất một trong số k văn bản mà nó trả về có chứa câu trả lời cho câu hỏi đầu vào.
- Độ đo thứ 2 là exact match, đo độ chính xác của reader cũng độ chính xác của toàn bộ mô hình. Mô hình nhận một điểm exact match nếu câu trả lời nó dự đoán trùng khớp với câu hỏi ground truth.

System settings

- Using Google Cloud Platform
- Training and inference on Cloud TPUs
- Process data on VM Compute Engine

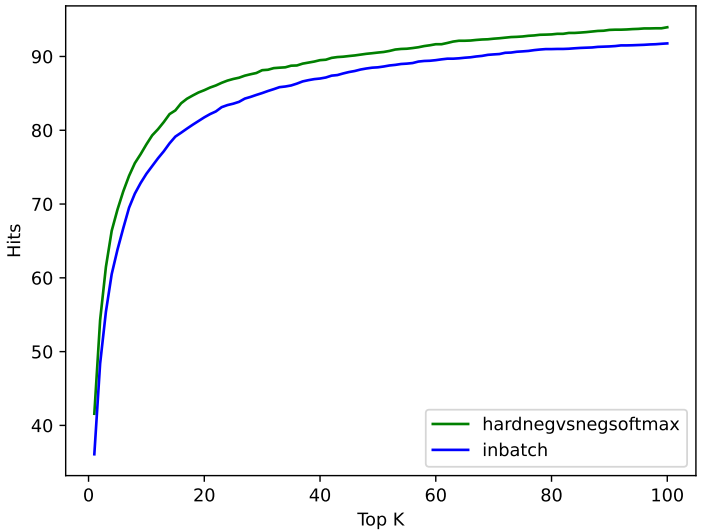
Hardware configurations

Cloud TPUs	VM Compute Engine
TPU v3-8 on-demand: • TPU version 3 • 8 TPU cores • 16GiB memory / TPU core	• OS: Ubuntu 20.04 • Disk: 30GB • RAM: 16GB • nCPUs: 4

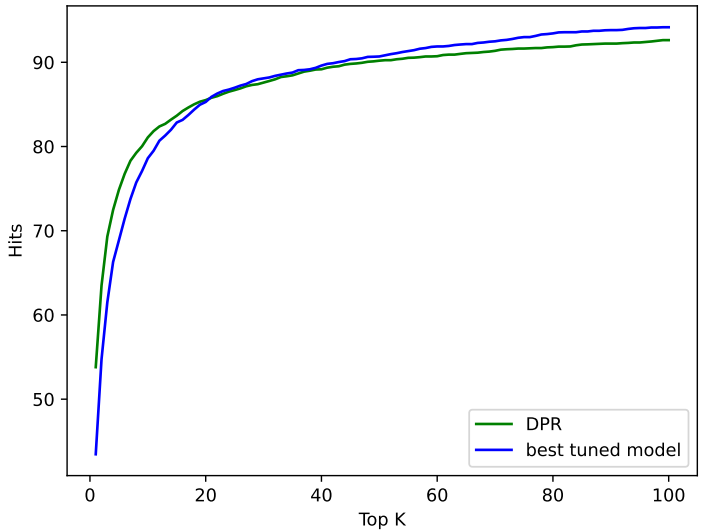
- Về thiết lập phần cứng, đồ án sử dụng huấn luyện mô hình trên Google Cloud TPU, với đặc điểm là tốc độ huấn luyện nhanh gấp nhiều lần so với khi sử dụng đồng thời nhiều GPU. Việc xử lý dữ liệu cũng được thực hiện trên các máy Compute Engine của Google.

Results on dense retriever

- Experimental results on dense retriever are conducted using Google Natural Question dataset.
- The proposed method was compared to the baseline in [2]



Comparasion results with baseline model implemented



Comparasion results with baseline model taken from released checkpoint

- Biểu đồ dưới đây thể hiện kết quả của mô hình mà em đã đề xuất trong phần 3. Hình bên trái là kết quả so sánh giữa mô hình baseline sử dụng trong nghiên cứu số 2 với đề xuất của em khi em tự implement lại mô hình baseline. Như có thể thấy mô hình đề xuất tương ứng với đường màu xanh lá cây có độ chính xác vượt trội hơn hẳn so với mô hình gốc. Ở hình bên phải là kết quả so sánh giữa mô hình đề xuất do em tự code với kết quả chạy khi sử dụng checkpoint mà tác giả bài báo cung cấp. Khi k nhỏ (khoảng nhỏ hơn 20) thì kết quả của checkpoint cao hơn mô hình đề xuất của em, còn với k lớn (lớn hơn 20) thì mô hình đề xuất của em đã dần dẫn tỏ ra tốt hơn.

Demo: Question Answering about COVID-19

Vietnamese Open-domain question answering for COVID-19 topic

Question

Khi nào Việt Nam có vắc xin chống covid-19 | Go

Việt Nam không có đỉnh dịch COVID - 19!

Còn nếu chúng ta chờ không được, đường biểu diễn bệnh cứ đi ngang hoài, chúng ta cần can thiệp bằng vắc xin. Thế giới đang cân nhắc xem xét, nếu tình hình tháng 5 tháng 6 tháng 7, dịch vẫn không xuống nổi, đến tháng 9, các nước buộc sẽ đưa vắc xin vào, không thể chờ thêm nữa. Vậy chúng ta có dự đoán khi nào hết dịch không, thưa bác sĩ? BS Trương Hữu Khanh: Thứ nhất, chúng ta làm tốt các biện pháp phòng ngừa dịch bệnh một cách quyết liệt mới có thể không còn bệnh nhân tại Việt Nam.

Vắc xin COVID-19 made in Vietnam đầu tiên dự kiến sẽ có vào cuối tháng 9/2021

Phó Thủ tướng nhắc lại nhận định của các chuyên gia, nhà khoa học cho rằng virus SARS-CoV-2 có thể có những biến đổi, tiếp tục tồn tại một số năm nữa. Cho đến giờ phút này nhiều khả năng các vắc xin phòng COVID-19 đều phải tiêm nhắc lại chứ không phải 1 đợt, hay 1 năm là xong. Dân số Việt Nam là 100 triệu người, vì vậy, chúng ta phải bằng các giải pháp để có vắc xin của Việt Nam, không chỉ phục vụ công tác phòng chống dịch COVID-19, mà còn chuẩn bị để ứng phó đối với những dịch bệnh có thể xảy ra trong tương lai.

Chủ tịch Quốc hội Vương Đình Huệ: Đẩy nhanh tiến độ thử nghiệm vắc xin COVID-19 Nano Covax để có thể sớm sản xuất trong nước

Đây là nhiệm vụ quan trọng để Việt Nam có thể chủ động phòng, chống COVID-19. Trong công tác phòng, chống dịch COVID-19 hiện nay, Chủ tịch Quốc hội nêu rõ, vắc xin là vũ khí quan trọng, mang tính quyết định sống còn đối với việc chấm dứt và chiến thắng đại dịch. "Nếu không sớm miễn dịch cộng đồng bằng việc tiêm chủng vắc xin COVID-19 sẽ rất khó để đẩy mạnh các hoạt động khác"- Chủ tịch Quốc hội nói.

- Trên đây là hình ảnh demo khi chạy hệ thống cho dữ liệu chủ đề covid-19. Có thể thấy kthi hỏi câu hỏi "Khi nào Việt Nam có vắc xin chống covid-19" thì hệ thống đã trả về các câu hỏi rất có liên quan tới câu hỏi đã cho.

Conclusion and future works

- Conclusion
  - Propose to train retriever model with stratified loss
  - Conduct a case study for open-domain question answering system in Vietnamese language for COVID-19 topic
  - Use Cloud TPUs to train large retriever model in short time
- Future works
  - Study machine reading comprehension problem to improve reader component
  - Study the relationship between open-domain question answering and automatic knowledge graph construction

- Phần cuối em xin đưa ra kết luận và hướng phát triển.
- Tổng kết về các kết quả đã được trong đồ án thì thứ nhất, đồ án đã đề xuất huấn luyện thành phần retriever của hệ thống open-domain question answering với hàm stratified loss, hiệu quả của đề xuất được chứng minh thông qua thực nghiệm.
- Thứ hai, đồ án thực hiện một case study của hệ thống open-domain question answering cho dữ liệu tiếng việt với chủ đề về dịch bệnh covid-19.
- Thứ 3, trong đồ án em cũng đã tìm hiểu và sử dụng TPU để huấn luyện các mô hình kích thước lớn, giảm thiểu đáng kể tài nguyên tính toán bao gồm CPU cũng như GPU so với nghiên cứu trước đó mà em tìm hiểu.

Thank you for your attention