ĐẠI HỌC QUY NHƠN
KHOA CÔNG NGHỆ THÔNG TIN

----------o0o----------

BÁO CÁO BÀI TẬP LỚN

HỌC PHẦN XỬ LÝ NGÔN NGỮ TỰ NHIÊN

PHÂN TÍCH QUAN ĐIỂM

Sinh viên thực hiện:	**Nguyễn Như Ý**
Lớp:	**Khoa học máy tính K42**
Mã sinh viên:	**4251050188**
Giảng viên hướng dẫn	**TS. Lê Quang Hùng**

Quy nhơn tháng 11 năm 2021

Mục lục

I. GIỚI THIỆU

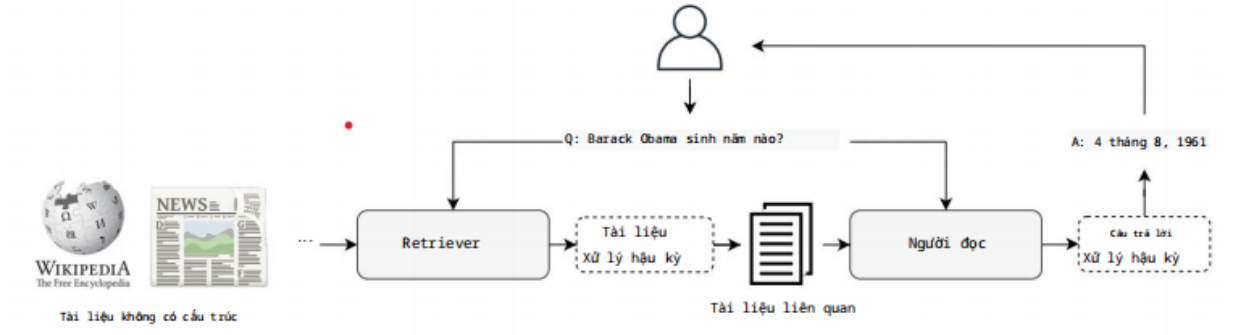
Với nhu cầu trao đổi thông tin của con người ngày càng cao, thông tin tràn ngập trên mọi phương tiện truyền thông, đặc biệt là sự phát triển rộng rãi của mạng toàn cầu Internet, hằng ngày con người phải xử lý một lượng thông tin khổng lồ. Những thắc mắc của người dùng dưới dạng truy vấn sẽ được tìm kiếm và trả về một cách ngắn gọn, súc tích, chính xác nhất những gì mà họ mong muốn. Đó chính là mục tiêu của hệ thống hỏi-đáp tự động.

Trả lời câu hỏi miền mở (OpenQA) là một nhiệm vụ quan trọng trong Xử lý ngôn ngữ tự nhiên (NLP), nhằm mục đích trả lời một câu hỏi dưới dạng ngôn ngữ tự nhiên dựa trên các tài liệu phi cấu trúc quy mô lớn. Gần đây, số lượng tài liệu nghiên cứu về OpenQA đã tăng vọt, đặc biệt là về các kỹ thuật tích hợp với tính năng Đọc hiểu máy thần kinh (MRC). Mặc dù các công trình nghiên cứu này đã nâng cao hiệu suất lên tầm cao mới trên bộ dữ liệu điểm chuẩn, chúng hiếm khi được đề cập trong các cuộc khảo sát hiện có về hệ thống QA. Trả lời câu hỏi (QA) nhằm mục đích cung cấp câu trả lời chính xác để trả lời câu hỏi của người dùng bằng ngôn ngữ tự nhiên. Đó là một nhiệm vụ lâu đời có từ những năm 1960 [1]. So với công cụ tìm kiếm, hệ thống QA nhằm mục đích trình bày câu trả lời cuối cùng cho một câu hỏi trực tiếp thay vì trả về danh sách các đoạn trích hoặc siêu liên kết có liên quan, do đó mang lại hiệu quả và thân thiện với người dùng hơn.

II. TỔNG QUAN VỀ BÀI TOÁN PHÂN TÍCH QUAN ĐIỂM

1. Phát biểu bài toán

Hệ thống trả lời câu hỏi (QA) trên miền mở chấp nhận các câu hỏi ngôn ngữ tự nhiên làm đầu vào và trả lại câu trả lời chính xác từ nội dung được chôn trong kho ngữ liệu văn bản lớn như Wikipedia. Việc xây dựng các hệ thống như vậy cho các ứng dụng thực tế trong lịch sử là khá khó khăn và liên quan. Bài báo cáo này cố gắng làm cho các hệ thống như vậy đơn giản hơn để triển khai và khai thác.



2. Một số mô hình Question Answering

a. Mô hình Seq2seq

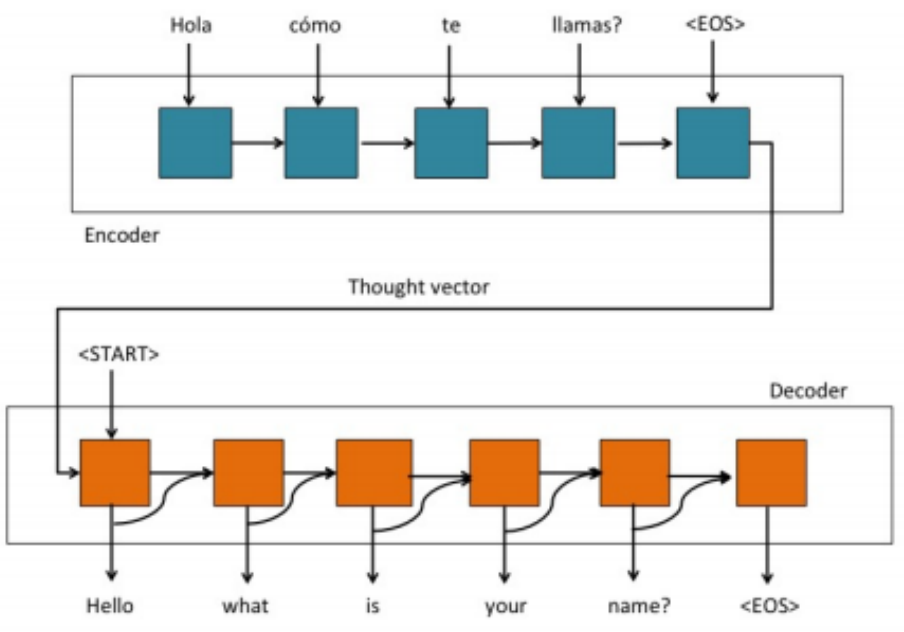


Figure 3.1: Diagram of the sequence to sequence architecture

Mô hình Seq2seq sử dụng kiến trúc Encoder-Decoder có độ dài đầu vào và đầu ra khác nhau. Kiến trúc Encoder-Decoder được coi là hai khối

- Mã hóa (Encoder) và Giải mã (Decoder), hai khối này được kết nối với nhau thông qua Vector trung gian (Context Vector):

-Bộ mã hóa - Encoder: Bộ mã hóa xử lý từng token trong chuỗi đầu vào, và nó cố gắng nhồi nhét toàn bộ thông tin đầu vào vào một vector có độ dài cố định, tức là "vector trung gian". Sau đó bộ mã hóa sẽ chuển vector này sang bộ giải mã.

-Vector trung gian - Context Vector: Vector này có chức năng gói gọn toàn bộ ý nghĩa của chuỗi đầu vào và giúp bộ giải mã đưa ra được quyết định chính xác. Đây là trạng thái ẩn nằm cuối chuỗi và được tính bởi bộ mã hóa, vector này sau đó cũng hoạt động như trạng thái ẩn đầu tiên của bộ giải mã.

-Bộ giải mã - Decoder: Bộ giả mã sử dụng Vector trung gian và cố gắng dự đoán chuỗi đích.

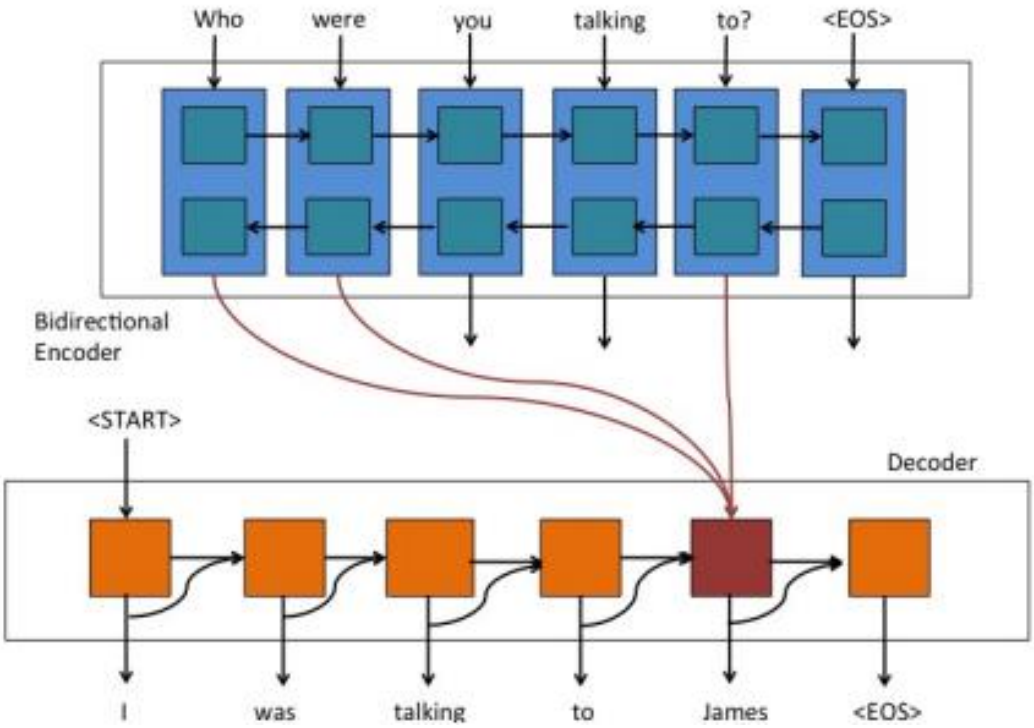
Hạn chế của mô hình Encoder

Vì bộ mã hóa chuyển đổi toàn bộ chuỗi đầu vào thành một vector có độ dài cố định và sau đó bộ giải mã dự đoán chuỗi đầu ra nên:

-Kiến trúc này chỉ hoạt động tốt với các chuỗi ngắn.

-Khó để bộ mã hóa ghi nhớ các chuỗi dài thành một vector có độ dài cố định.

b. Mô hình Attention



Không giống như Seq2Seq đã trình bày trước đây, đối với cơ chế chú ý, Thay vì chỉ sử dụng vectơ trạng thái ẩn cuối cùng làm vectơ ngữ cảnh, đối với mỗi từ để giải mã, bộ giải mã tính toán một vectơ ngữ cảnh với tổng trọng số của tất cả các vectơ trạng thái ẩn của bộ mã hóa. một bộ mã hóa hai chiều được sử dụng để mã hóa từng câu đầu vào thành hai vectơ trạng thái ô ẩn.

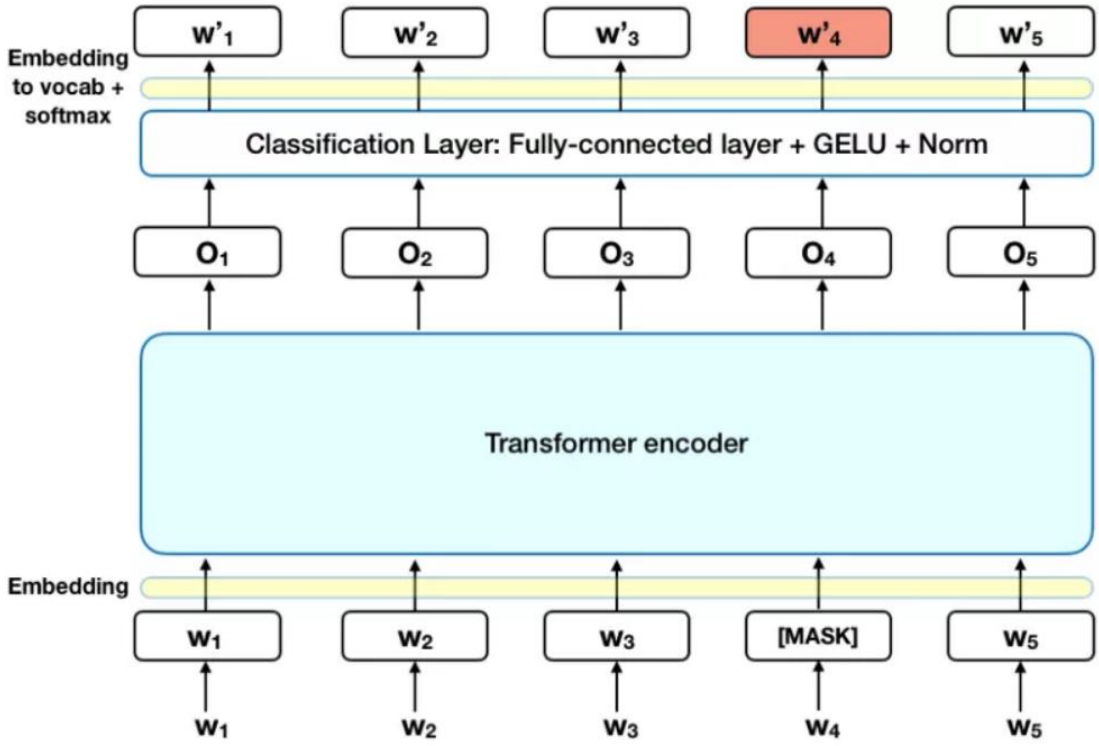
III. MÔ HÌNH BERT

BERT là viết tắt của cụm từ Bidirectional Encoder Representation from Transformer có nghĩa là mô hình biểu diễn từ theo 2 chiều ứng dụng kỹ thuật Transformer. BERT được thiết kế để huấn luyện trước các biểu diễn từ (pre-train word embedding). Điểm đặc biệt ở BERT đó là nó có thể điều hòa cân bằng bối cảnh theo cả 2 chiều trái và phải.

Biểu đồ dưới đây là một mô tả cấp cao của bộ mã hóa Transformer. Đầu vào là một chuỗi các mã thông báo, đầu tiên được nhúng vào các vectơ và sau đó được xử lý trong mạng neural. Đầu ra là một chuỗi các vectơ có kích thước H, trong đó mỗi vectơ tương ứng với một mã thông báo đầu vào có cùng chỉ mục.

Khi đào tạo các mô hình ngôn ngữ, có một thách thức trong việc xác định mục tiêu dự đoán. Nhiều mô hình dự đoán từ tiếp theo theo trình tự (ví dụ: Đứa trẻ về nhà từ ___), một cách tiếp cận định hướng vốn đã hạn chế việc học theo ngữ cảnh. Để vượt qua thử thách này, BERT sử dụng hai chiến lược đào tạo:

-Masked LM (MLM)



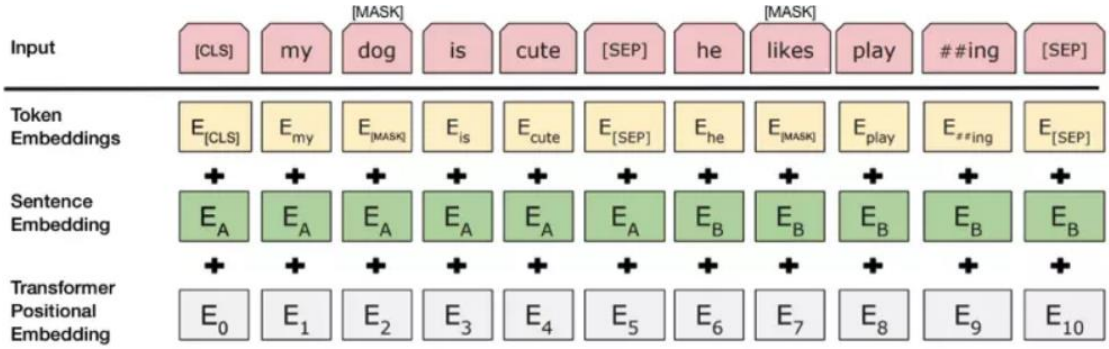
Trước khi cho các chuỗi từ vào BERT, 15% số từ trong mỗi chuỗi được thay thế bằng mã thông báo [MASK]. Mô hình sau đó cố gắng dự đoán giá trị ban đầu của các từ bị che, dựa trên ngữ cảnh được cung cấp bởi các từ khá không bị che ở trong chuỗi. Về mặt kỹ thuật, dự đoán của các từ đầu ra yêu cầu:

+ Thêm một lớp phân loại ở đầu ra của bộ mã hóa.

+ Nhân các vectơ đầu ra với ma trận nhúng, chuyển đổi chúng thành các chiều từ vựng.

+ Tính xác suất của mỗi từ trong từ vựng với hàm softmax.

-Dự đoán câu tiếp theo (Next Sentence Prediction (NSP))



Trong quy trình đào tạo BERT, mô hình nhận các cặp câu làm đầu vào và học cách dự đoán nếu câu thứ hai trong cặp là câu tiếp theo trong tài liệu gốc. Trong quá trình đào tạo, 50% đầu vào là một cặp trong đó câu thứ hai là câu tiếp theo trong tài liệu gốc, trong khi 50% còn lại, một câu ngẫu nhiên từ kho văn bản được chọn làm câu thứ hai. Giả định là câu ngẫu nhiên sẽ bị ngắt kết nối từ câu đầu tiên.

Khi thực hiện huấn luyện mô hình BERT, Masked LM và Dự đoán câu tiếp theo (NSP) được huấn luyện cùng nhau, với mục tiêu tối giản hóa sự kết hợp hàm mất mát của hai chiến lược.

IV. CÁC BƯỚC THỰC NGHIỆM VÀ KẾT QUẢ

a. Các bước thực nghiệm

Sử dụng 20 Newsgroup dataset làm kho dữ liệu văn bản

Trước tiên tải tập dữ liệu vào một mảng bằng cách sử dụng **scikit-learning** (là thư viện máy học phần mềm) và nhập **ktrain modules**

```
In [2]: # load 20newsgroups datset into an array
        from sklearn.datasets import fetch_20newsgroups
        remove = ('headers', 'footers', 'quotes')
        newsgroups_train = fetch_20newsgroups(subset='train', remove=remove)
        newsgroups_test = fetch_20newsgroups(subset='test', remove=remove)
        docs = newsgroups_train.data +  newsgroups_test.data

In [3]: import ktrain
        from ktrain.text.qa import SimpleQA
```

Bước 1: Lập chỉ mục tài liệu

Đầu tiên, lập chỉ mục các tài liệu vào công cụ tìm kiếm việc này sẽ được sử dụng để nhanh chóng truy xuất các tài liệu có khả năng chứa câu trả lời cho một câu hỏi. Để làm như vậy, cần phải chọn một vị trí chỉ mục, vị trí này phải là một thư mục chưa tồn tại.

Vì bài đăng trên nhóm tin thường có kích thước nhỏ và vừa bộ nhớ, nên đặt `commit_every` giá trị lớn để tăng tốc quá trình chỉ mục. Điều này có nghĩa kết quả sẽ không được viết cho đến khi kết thúc. Nếu gặp sự cố, có thể giảm giá trị này.

```
In [4]: INDEXDIR = '/tmp/myindex'

In [5]: SimpleQA.initialize_index(INDEXDIR)
        SimpleQA.index_from_list(docs, INDEXDIR, commit_every=len(docs),
                                 multisegment=True, procs=4, # these args speed up indexing
                                 breakup_docs=True         # this slows indexing but speeds up answer retrieval
                                 )
```

Đối với những tập tài liệu quá lớn để tải vào danh sách Python, có thể sử dụng `SimpleQA.index_from_folder` danh sách này sẽ thu thập thông tin một thư mục và lập chỉ mục tất cả các tài liệu văn bản thuần túy (ví dụ: `.txt`) theo mặc định.

Nếu tài liệu của bạn ở các định dạng như `.pdf`, `.docx` `.pptx`, bạn có thể cung cấp `use_text_extraction=True` đối số `index_from_folder`, đối số này sẽ sử dụng gói textract để trích xuất văn bản từ các loại tệp khác nhau và lập chỉ mục văn bản này vào công cụ tìm kiếm để truy xuất câu trả lời. Bạn cũng có thể chuyển đổi chúng thành `.txt` theo cách thủ công bằng `ktrain.text.textutils.extract_copy` hoặc các công cụ như Apache Tika hoặc textract .

Bước 2: Tạo một phiên bản QA

Tiếp theo, tạo phiên bản QA (Lưu ý rằng, theo mặc định, `SimpleQA` sử dụng TensorFlow. Để sử dụng PyTorch, hãy cung cấp`framework='pt'`dưới dạng tham số). Bước này sẽ tự động tải xuống mô hình BERT SQuAD nếu chưa tồn tại trong hệ thống.

```
In [6]: qa = SimpleQA(INDEXDIR)
```

Trong khoảng 3 mã dòng, đã xây dựng một hệ thống QA end-to-end hiện có thể sử dụng để tạo câu trả lời cho các câu hỏi. Tiếp theo, đặt câu hỏi cho hệ thống.

Bước 3: Đặt câu hỏi

Gọi `ask` phương thức để đưa ra câu hỏi cho kho dữ liệu văn bản đã được lập chỉ mục và truy xuất câu trả lời. Sử dụng `qa.display` phương pháp này để hiển thị 5 kết quả hàng đầu trong Jupyter notebook. Các câu trả lời được suy ra bằng cách sử dụng mô hình BERT được tinh chỉnh trên tập dữ liệu SQuAD. Mô hình sẽ lược qua các đoạn văn và tìm câu trả lời cho ứng viên.

Theo mặc định, `ask` hiện đang sử dụng `batch_size` giá trị là 8, nhưng nếu cần , có thể thử nghiệm bằng cách đặt `batch_size` thông số.

Lưu ý [20 Newsgroup dataset] bao gồm sự từ đầu đến giữa những năm 1990 do đó các tham chiếu đến sự kiện gần đây sẽ không tồn tại.

b. Kết quả

1. Câu hỏi về không gian

Như bạn có thể thấy, câu trả lời ứng cử viên hàng đầu chỉ ra rằng tàu thăm dò không gian Cassini đã được phóng vào tháng 10 năm 1997, điều này có vẻ đúng. Câu trả lời đúng sẽ không phải lúc nào cũng là câu trả lời hàng đầu, nhưng nó nằm trong trường hợp này.

```
In [7]: answers = qa.ask('When did the Cassini probe launch?')
        qa.display_answers(answers[:5])
```

	Candidate Answer	Context	Confidence	Document Reference
0	in october of 1997	cassini is scheduled for launch aboard a titan iv / centaur in october of 1997 .	0.819032	59
1	on january 26,1962	ranger 3, launched on january 26,1962 , was intended to land an instrument capsule on the surface of the moon, but problems during the launch caused the probe to miss the moon and head into solar orbit.	0.151229	8525
2	- 10 / 06 / 97	key scheduled dates for the cassini mission (vvejga trajectory)--10 / 06 / 97-titan iv / centaur launch 04 / 21 / 98-venus 1 gravity assist 06 / 20 / 99-venus 2 gravity assist 08 / 16 / 99-earth gravity assist 12 / 30 / 00-jupiter gravity assist 06 / 25 / 04-saturn arrival 01 / 09 / 05-titan probe release 01 / 30 / 05-titan probe entry 06 / 25 / 08-end of primary mission (schedule last updated 7 / 22 / 92) - 10 / 06 / 97	0.029694	59
3	* 98	cassini * * * * * * * * * * * * * * * * * * 98 ,115 * * * *	0.000026	5356
4	the latter part of the 1990s	scheduled for launch in the latter part of the 1990s , the craf and cassini missions are a collaborative project of nasa, the european space agency and the federal space agencies of germany and italy, as well as the united states air force and the department of energy.	0.000017	18684

Lưu ý, vì sử dụng `index_from_list` để lập chỉ mục tài liệu nên cột cuối cùng (tức **Document Reference** - tài liệu tham khảo) hiển thị chỉ mục danh sách danh được liên kết với bài đăng trong nhóm tin chứa câu trả lời, có thể sử dụng để xem xét toàn bộ tài liệu có chứa câu trả lời. Nếu sử dụng `index_from_folder` để lập chỉ mục tài liệu, cột cuối cùng sẽ hiển thị đường dẫn và tên tệp dữ liệu. Các giá trị **Document Reference** có thể tùy chỉnh bằng cách cung cấp một `references` tham số cho `index_from_list`

Để xem văn bản của tài liệu có chứa câu trả lời trên cùng, bỏ ghi chú và thực hiện:

```
In [8]: #print(docs[59])
```

2. Câu hỏi tôn giáo

```
In [9]: answers = qa.ask('Who was Muhammad?')
        qa.display_answers(answers[:5])
```

	Candidate Answer	Context	Confidence	Document Reference
0	the holy prophet of islam	just a small reminder to all my muslim brothers, did _ ever _ the holy prophet of islam (muhammad pbuh), say to anyone who called himself a muslim :	0.762464	1278
1	the messenger of allah	muhammad is the messenger of allah , and those who are with him are firm against the unbelievers and merciful among each other.	0.201862	4876
2	is the last prophet of islam	muhammad peace and blessings of allah be upon him (saw) is the last prophet of islam .	0.035121	4640
3	either a liar, or he was crazy (a modern day mad mahdi) or he was actually who he said he was	the book says that muhammad was either a liar, or he was crazy (a modern day mad mahdi) or he was actually who he said he was .	0.000293	4934
4	[mahound ' s	muhammad ' s [mahound ' s] integrity is not really impugned in this part of the story, and there ' s no reason to think this was rushdie ' s intent : gibreel, as the archangel, produces the verses (divine and satanic), though he does not know their provenance.	0.000138	15852

Ở đây chúng ta thấy được những quan điểm khác nhau về việc Muhammad, người sáng lập đạo Hồi, đã được tranh luận và thảo luận trong bộ tài liệu này.

3. Câu hỏi kỹ thuật

```
In [10]: answers = qa.ask('What causes computer images to be too dark?')
         qa.display_answers(answers[:5])
```

	Candidate Answer	Context	Confidence	Document Reference
0	if your viewer does not do gamma correction	if your viewer does not do gamma correction , then linear images will look too dark, and gamma corrected images will ok.	0.937990	13873
1	is gamma correction	this, is gamma correction (or the lack of it).	0.045165	13873
2	so if you just dump your nice linear image out to a crt	so if you just dump your nice linear image out to a crt , the image will look much too dark.	0.010337	13873
3	that small color details	the algorithm achieves much of its compression by exploiting known limitations of the human eye, notably the fact that small color details are not perceived as well as small details of light and dark.	0.002114	6987
4	that small color details	the algorithm achieves much of its compression by exploiting known limitations of the human eye, notably the fact that small color details are not perceived as well as small details of light and dark.	0.002114	12344

Như đã thấy, *gamma correction* là câu trả lời hàng đầu.

V. KẾT QUẢ

Sử dụng thư viện Gradio để tạo giao diện. Gradio là cách nhanh nhất để demo mô hình học máy với giao diện web thân thiện để mọi người có thể sử dụng nó, ở bất cứ đâu!

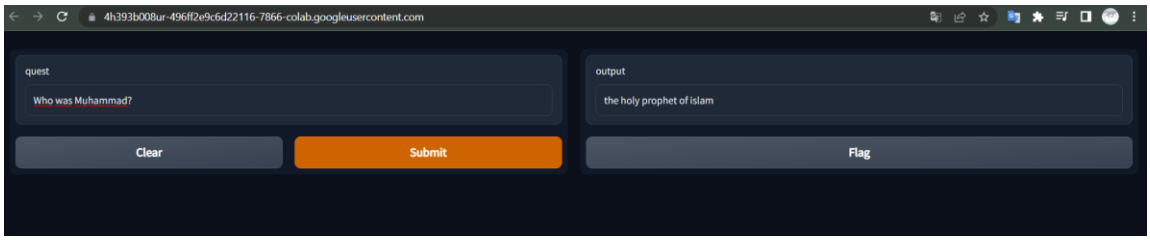
Đã có giao diện:

```
import gradio as gr

def greet(quest):
    answers = qa.ask(quest)
    for key,value in answers[0].items():
      if key == 'answer':
        return value

demo = gr.Interface(fn=greet, inputs="text", outputs="text")

demo.launch()
```



VI. KẾT LUẬN

Sử dụng BERT cho một bài toán cụ thể là tương đối đơn giản:

BERT có thể được sử dụng cho nhiều bài toán ngôn ngữ, trong khi chỉ thêm một lớp nhỏ vào mô hình lõi:

Các bài toán phân loại như phân tích tình cảm được thực hiện tương tự như phân loại Câu tiếp theo, bằng cách thêm một lớp phân loại trên đầu ra của Transfomer cho mã thông báo [CLS].

Trong bài toán Trả lời Câu hỏi (Question Answering) (ví dụ: SQuAD v1.1), phần mềm nhận được một câu hỏi liên quan đến chuỗi văn bản và được yêu cầu đánh dấu câu trả lời trong chuỗi. Sử dụng BERT, một mô hình Hỏi và Đáp có thể được đào tạo bằng cách học thêm hai vectơ đánh dấu điểm bắt đầu và kết thúc của câu trả lời.

Tóm lại, BERT chắc chắn là một bước đột phá trong việc sử dụng Machine Learning để xử lý ngôn ngữ tự nhiên. Thực tế là nó có thể tiếp cận và cho phép tinh chỉnh nhanh sẽ có khả năng cho phép một loạt các ứng dụng thực tế trong tương lai.

TÀI LIỆU THAM KHẢO

1. https://gradio.app/quickstart/

2. https://towardsdatascience.com/build-an-open-domain-question-answering-system-with-bert-in-3-lines-of-code-da0131bc516b

3. https://nbviewer.org/github/amaiya/ktrain/blob/master/examples/text/question_answering_with_bert.ipynb

4. https://github.com/amaiya/ktrain

5. https://nbviewer.org/github/amaiya/ktrain/blob/master/examples/text/question_answering_with_bert.ipynb?fbclid=IwAR0EyAo18cu0MNzgFruXt6BeAJSnrpUWEueYPkjHitH4L-eQQV37b1Tqt4U

6. https://www.techtarget.com/searchenterpriseai/definition/BERT-language-model

7. https://itzone.com.vn/vi/article/bert-mo-hinh-ngon-ngu-hien-dai-cho-xu-ly-ngon-ngu-tu-nhien-nlp/