Question Answering

Group 5
Natural Language Processing
INT3406 1

Thành viên trong nhóm

Nguyễn Công Thuận
180201250

Hoàng Vũ Duy Anh
18020001

Lưu Hoàng Nam
18020921

Outline

1. Tổng quan về Hệ thống Question Answering
2. DrQA
3. Cải tiến hệ thống
4. Ứng dụng cho Tiếng Việt
5. Demo

Tổng quan về
Hệ thống Question Answering

Question Answering System

- Trả lời câu hỏi được đưa ra
- Truy vấn hoặc tìm kiếm ở trong một hệ tri thức cho trước

Closed-domain

- Trả lời câu hỏi liên quan đến một miền ứng dụng cụ thể

Open-domain

- Trả lời câu hỏi liên quan đến nhiều miền ứng dụng
- Cần một kho kiến thức rộng lớn như Wikipedia

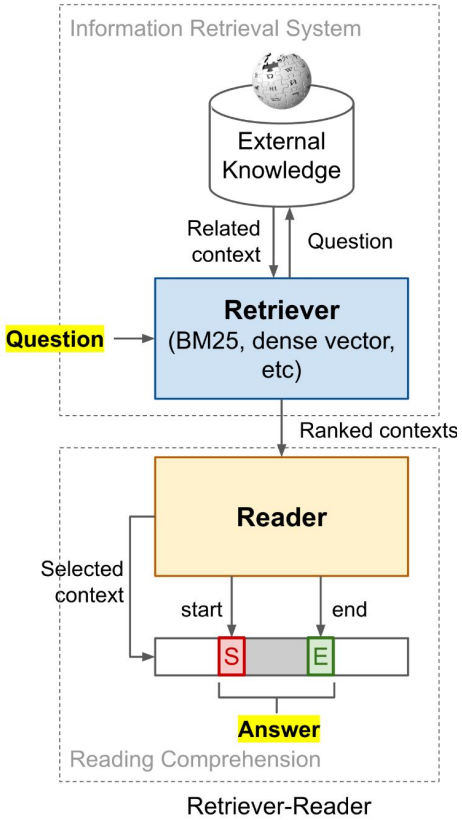
Tập dữ liệu

- Dữ liệu chủ yếu gồm 3 thành phần cơ bản:
  - `“question”`: câu hỏi
  - `“text”`: đoạn văn có thể chứa câu trả lời hoặc không
  - `“answer”`: câu trả lời

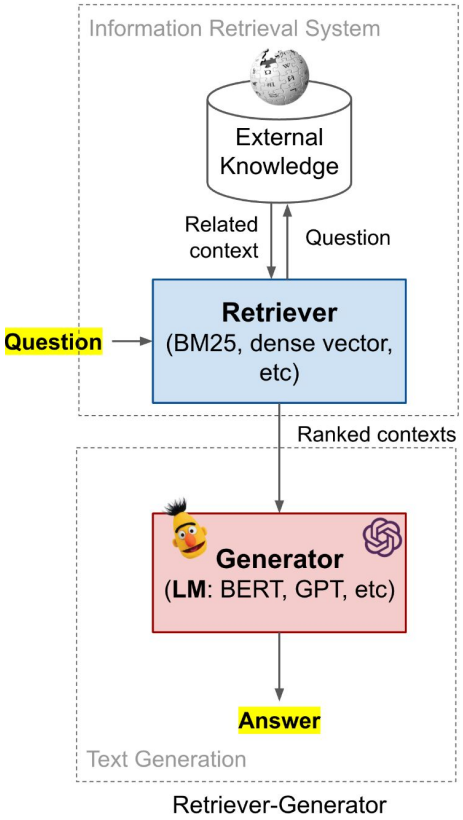
https://rajpurkar.github.io/SQuAD-explorer/

Kiến trúc Hệ thống

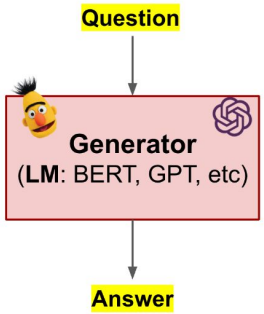
- 3 loại kiến trúc
  - Retriever - Reader
  - Retriever - Generator
  - Generator



Retriever-Reader



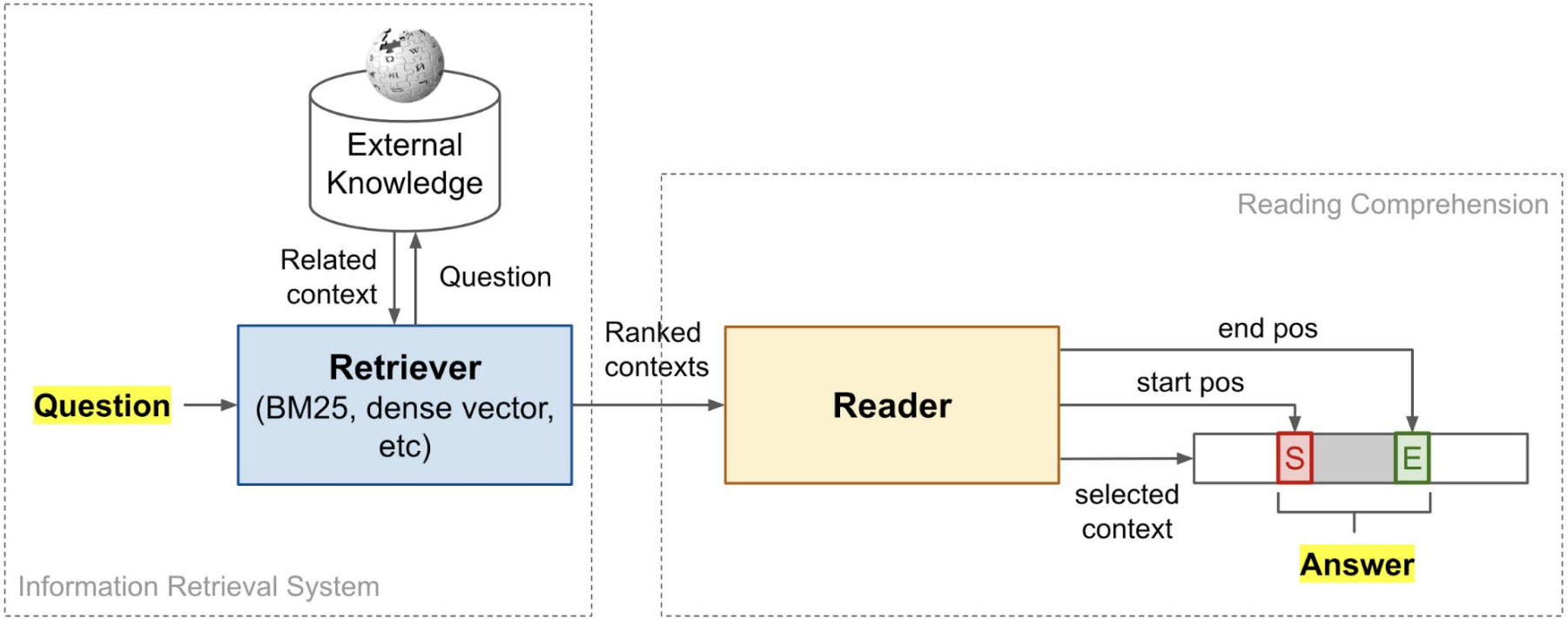
Retriever-Generator



Generator

Kiến trúc Retriever - Reader

- 2 bước thực hiện
  - Xác định được các đoạn văn có thể có thông tin liên quan đến câu hỏi trong kho kiến thức được cung cấp
  - Trích xuất được câu trả lời từ các đoạn thông tin nhận được


External Knowledge
Related context
Question
Retriever
(BM25, dense vector, etc)
Question
Information Retrieval System
Ranked contexts
Reader
Reading Comprehension
end pos
start pos
S
E
selected context
Answer

Kiến trúc Retriever - Reader

- **Thành phần Retriever**
  - **Input**: Câu hỏi
  - **Output**: Những đoạn văn (context) liên quan đến câu hỏi

Kiến trúc Retriever - Reader

- **Thành phần Retriever**
  - 2 cách tiếp cận
    - Cách tiếp cận truyền thống: TF - IDF
    - Cách tiếp cận hiện đại: mạng nơ-ron

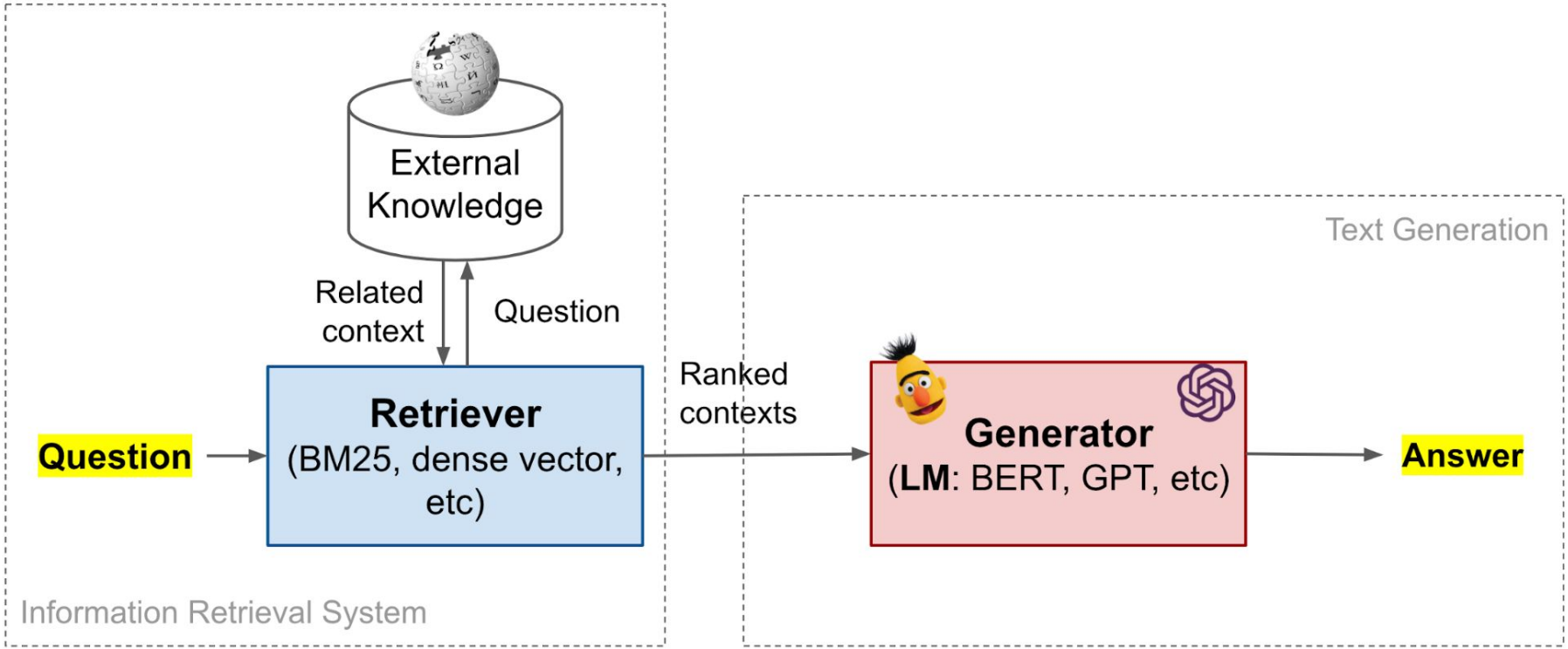
      VD: MLP, LSTM...
  - Xếp hạng những đoạn văn (context) có mức độ liên quan đến câu hỏi lớn nhất

Kiến trúc Retriever - Reader

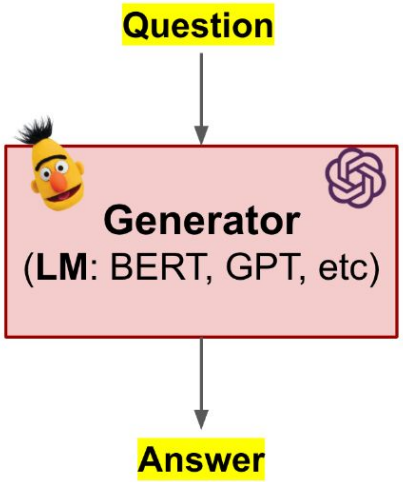
- Thành phần Reader
  - Input: Những đoạn văn (context) liên quan đến câu hỏi
  - Output: Câu trả lời
  - Sử dụng mạng nơ-ron
    - Bi-directional LSTM
    - BERT

Kiến trúc Retriever - Generator

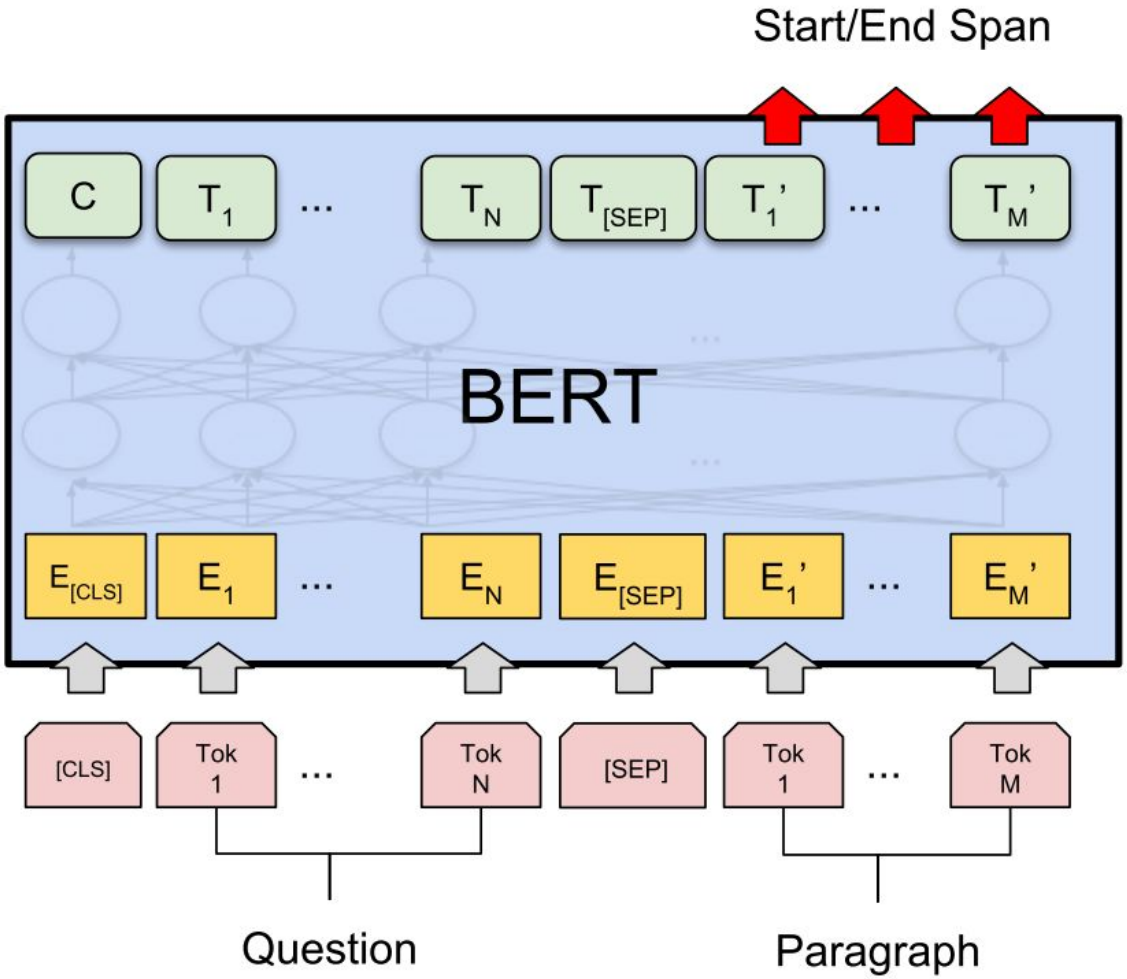
- 2 bước thực hiện:
  - Xác định được đoạn có thông tin liên quan đến câu hỏi trong kho kiến thức được cung cấp
  - Sinh ra câu trả lời từ các đoạn văn liên quan, sử dụng các mạng nơ-ron như BERT, GPT...


External Knowledge
Related context
Question
Retriever
(BM25, dense vector, etc)
Question
Information Retrieval System
Ranked contexts
Text Generation
Generator
(LM: BERT, GPT, etc)
Answer

Kiến trúc Generator



Generator


Start/End Span
C
T1
...
TN
T[SEP]
T1'
...
TM'
BERT
E[CLS]
E1
...
EN
E[SEP]
E1'
...
EM'
[CLS]
Tok 1
...
Tok N
[SEP]
Tok 1
...
Tok M
Question
Paragraph

DrQA

Tổng quan về Hệ thống DrQA

- Được công bố trong một bài báo vào năm 2017, sau khi tập dữ liệu SQuAD ra đời
- Sử dụng kiến trúc Retriever - Reader

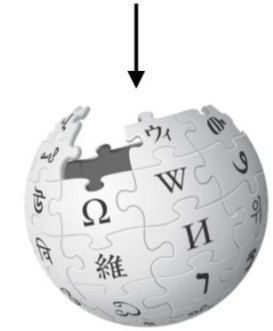
Tổng quan về Hệ thống DrQA

- Mục tiêu: Trả lời câu hỏi thuộc dạng open-domain
- Sử dụng Wikipedia làm cơ sở tri thức

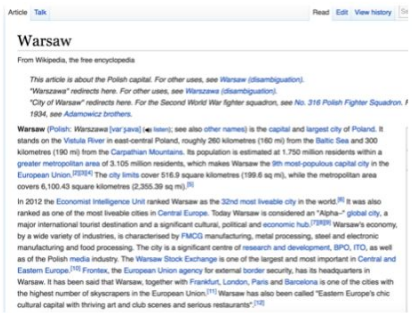
Open-domain QA

SQuAD, TREC, WebQuestions, WikiMovies

Q: How many of Warsaw's inhabitants spoke Polish in 1933?

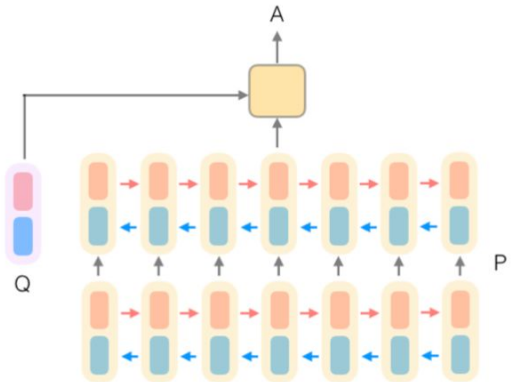


Document Retriever



Document Reader

833,500



Kiến trúc của DrQA

- **Document Retriever**
  - Thu hẹp phạm vi tìm kiếm từ nguồn dữ liệu Wikipedia
  - Lấy ra k = 5 tài liệu liên quan nhất đến câu hỏi
  - Đánh giá mức độ liên quan theo TF - IDF

TF - IDF

$$tf(t,d) = \log(1 + freq(t,d))$$

$$idf(t,D) = \log \frac{|D|}{|d \in D : t \in d|}$$

$$tfidf(t,d,D) = tf(t,d) \times idf(t,D)$$

TF - IDF

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

Kiến trúc của DrQA

- **Document Reader**
  - Tìm ra câu trả lời trong 5 tài liệu được lấy ra
  - Câu hỏi $q = \{q_1, q_2, \dots, q_l\}$

    l: số token trong câu hỏi
  - Đoạn văn $p = \{p_1, p_2, \dots, p_m\}$

    m: số token trong đoạn văn

Kiến trúc của DrQA

- Document Reader
  - 3 bước chính
    - Paragraph Encoding
    - Question Encoding
    - Prediction

Paragraph Encoding

- Biểu diễn các token p_i dưới dạng vector đặc trưng $\tilde{p}_i$, gồm các thành phần

Paragraph Encoding

- Biểu diễn các token p_i dưới dạng vector đặc trưng $\tilde{p}_i$, gồm các thành phần
  - Word embedding
    - Sử dụng Glove word embedding

$$f_{emb}(p_i) = E(p_i)$$

Paragraph Encoding

- Biểu diễn các token p_i dưới dạng vector đặc trưng $\tilde{p}_i$, gồm các thành phần
  - Exact match
    - Sử dụng 3 binary feature để kiểm tra mức độ khớp

$$f_{exact\_match}(p_i) = \Pi(p_i \in q)$$

Paragraph Encoding

- Biểu diễn các token p_i dưới dạng vector đặc trưng $\tilde{p}_i$, gồm các thành phần
  - Đặc trưng của token
    - Part-Of-Speech, Name Entity Recognition, Term Frequency

$$f_{token}(p_i) = (POS(p_i), NER(p_i), TF(p_i))$$

Paragraph Encoding

- Biểu diễn các token p_i dưới dạng vector đặc trưng $\tilde{p}_i$, gồm các thành phần
  - Aligned question embedding

$$f_{align}(p_i) = \Sigma_j a_{i_j} E(q_j)$$

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

Paragraph Encoding

- Tập vector đặc trưng $\tilde{p}_1, \tilde{p}_2, \ldots, \tilde{p}_m$ sẽ được đưa qua RNN

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

- Mã hoá thông tin của đoạn văn

Question Encoding

- Vector **q** được biểu diễn

$$\mathbf{q} = \sum b_j \times q_j$$

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

Prediction

- Tính toán xác suất vị trí bắt đầu và kết thúc

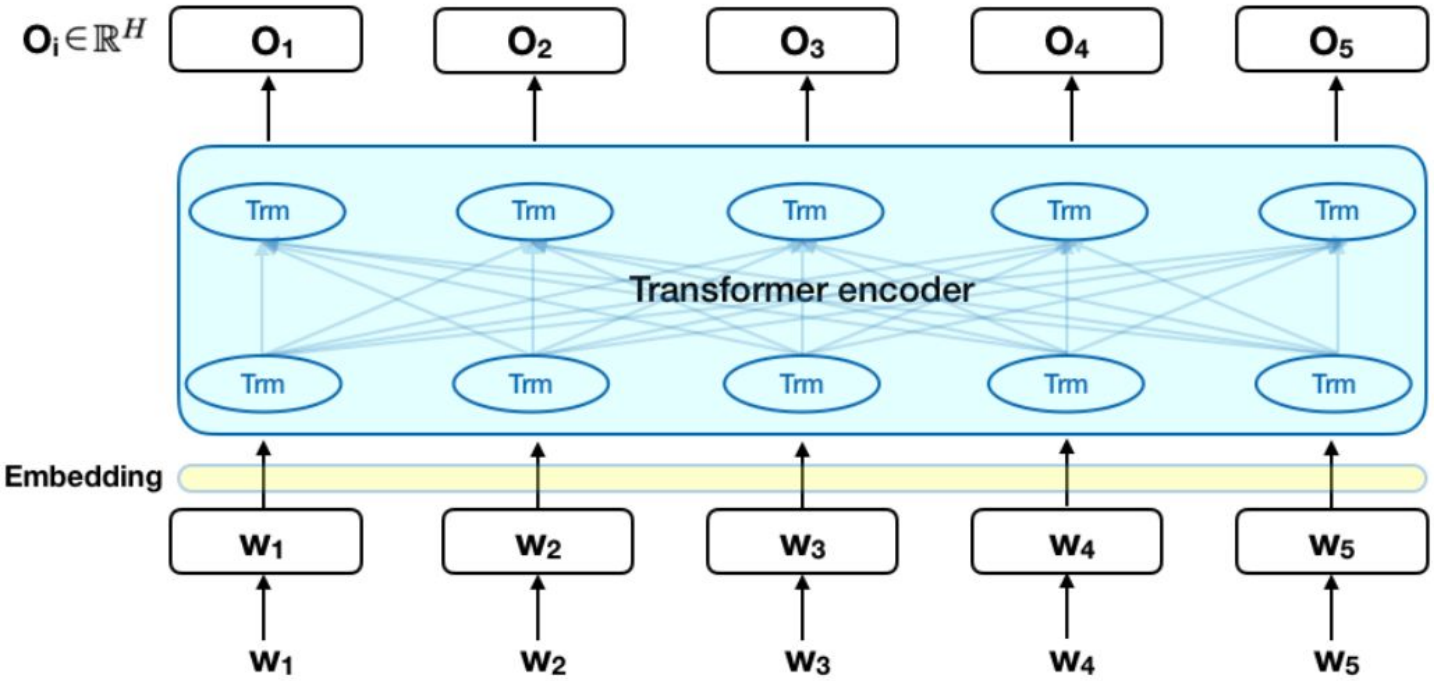
$$P_{start}(i) \propto exp(\mathbf{p}_i \mathbf{W}_s \mathbf{q})$$

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

Cải tiến hệ thống

Cải tiến

- Document Reader
  - Các mô hình BERT
    - Transformer architecture
    - Deeply bidirectional


$O_i \in \mathbb{R}^H$
O1
O2
O3
O4
O5
Trm
Transformer encoder
Embedding
W1
W2
W3
W4
W5

Cải tiến

- Document Retriever
  - TF-IDF
- Document Reader
  - Các mô hình BERT
    - DistilBERT
    - MobileBERT
    - ALBERT

Kết quả đánh giá

Model	EM	F1	Latency
DrQA (Chen et al., 2017)	29.5	-	~0.5s
DrQA retriever + DistilBERT-base (n_docs=5)	31.0	35.2	2.58s
n_docs=4	**31.9**	**36.9**	**2.07s**
n_docs=3	31.6	35.5	1.53s
n_docs=2	30.3	35.1	1.03s
DrQA retriever + MobileBERT	**32.0**	**37.5**	**2.35s**
DrQA retriever + AlBERT	-	-	-

Cải tiến

- **Document Retriever**
  - **BM25 Retriever trong Anserini**

$$\text{score}(D,Q) = \sum_{i=1}^{n} \text{IDF}(q_i, D) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{avgdl}\right)}$$

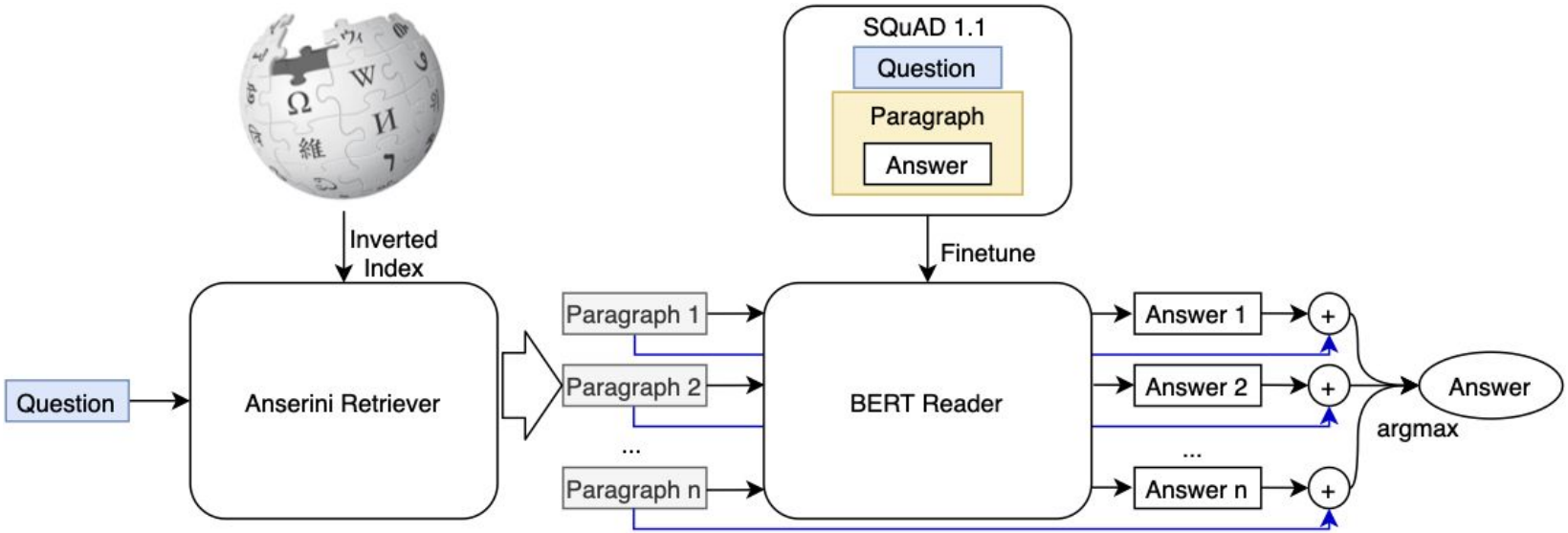
$$\text{IDF}(q_i, D) = \log \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5}$$

N	số lượng văn bản có trong ngữ liệu
$n(q_i)$	số lượng văn bản chứa từ q_i
$f(q_i, D)$	tần suất xuất hiện từ q_i trong văn bản D
$\|D\|$	số lượng từ có trong văn bản D
$avgdl$	độ dài trung bình của các văn bản
k_1	tham số tự chọn, thường là 2
b	tham số tự chọn, thường là 0.75

Cải tiến

- Document Retriever
  - BM25 Retriever
- Document Reader
  - Mô hình BERT

BERTserini


SQuAD 1.1
Question
Paragraph
Answer
Inverted
Index
Finetune
Question
Anserini Retriever
Paragraph 1
Paragraph 2
...
Paragraph n
BERT Reader
Answer 1
Answer 2
...
Answer n
+
+
+
Answer
argmax

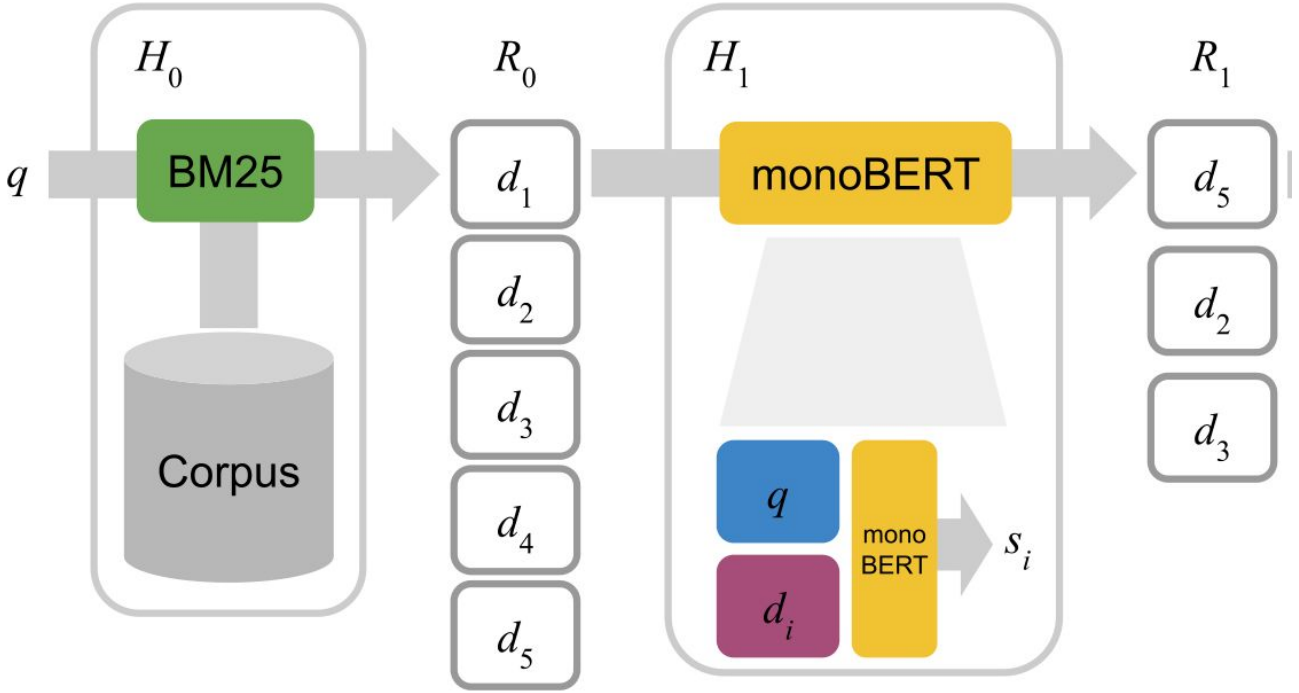
Kết quả đánh giá

Model	EM	F1	R
BERTserini (Article, $k = 5$)	19.1	25.9	63.1
BERTserini (Paragraph, $k = 29$)	36.6	44.0	75.0
BERTserini (Sentence, $k = 78$)	34.0	41.0	67.5
BERTserini (Paragraph, $k = 100$)	**38.6**	**46.1**	**85.8**

Kết quả đánh giá

Dataset	EM	F_1
Open SQuAD-dev (Paragraph, k = 30)	37.3	43.9

Re-rank docs with BERT, Dense retrieval...



Ứng dụng cho Tiếng Việt

Dữ liệu Tiếng Việt

- **SQuAD-translate:** ~100k
- **vi-wiki:** 710
- **XQuAD:** 1000+ , 10 ngôn ngữ
- **MLQA:** 6000+ , 7 ngôn ngữ
- **UIT-ViQuAD:** 23000+

Mô hình thử nghiệm

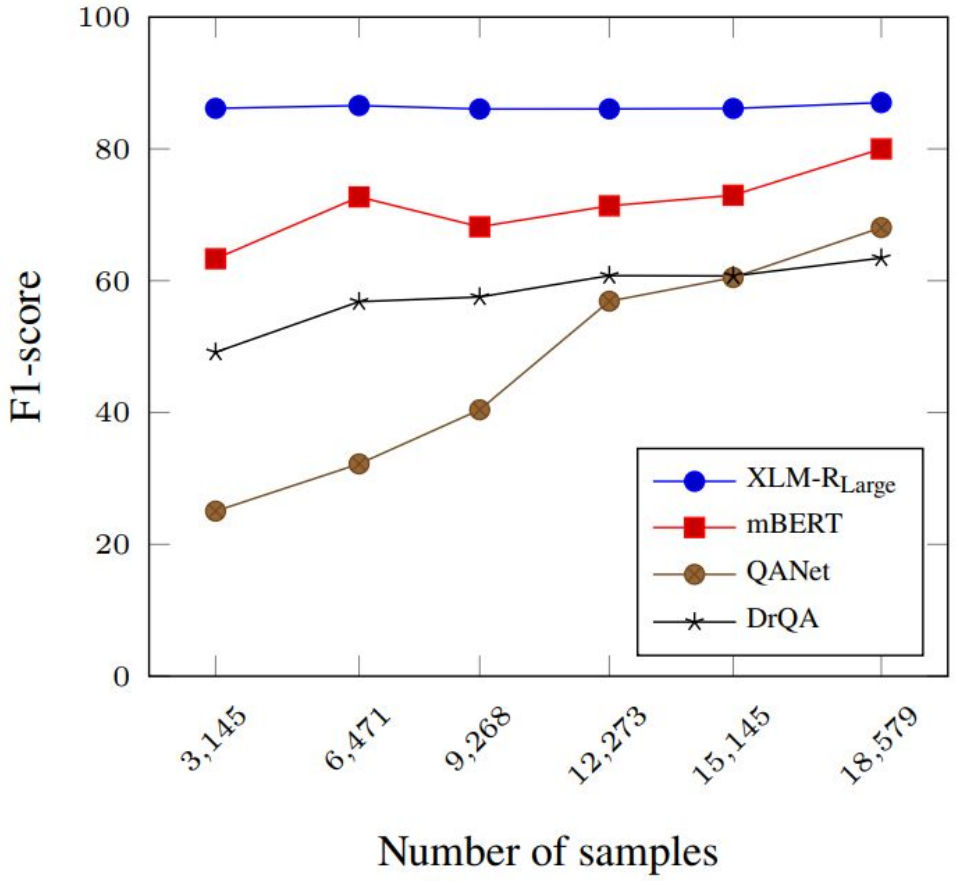
- ALBERT-vi
- PhoBERT
- XLM-RoBERTa - hiện đang là SOTA
  - Pretrained trên 2.5TB dữ liệu, 100 ngôn ngữ, vocab~250k
  - Cross-lingual transfer, zero-shot

Kết quả đánh giá

Dữ liệu huấn luyện	Mô hình	Tham số	Throughput	vi-wiki-test	MLQA-dev
SQuAD-translate (~100k pairs)	BERT-base [24]	110M	-	43.2 / 65.9	-
	ALBERT-vi-base	12M	12.2/s	32.4 / 48.8	26.2 / 42.1
	PhoBERT-base	135M	17.6/s	45.0 / 63.6	37.6 / 57.2
	XLM-R-base	270M	15.1/s	45.9 / 65.5	40.9 / 59.8
MLQA + XQuAD (~7000 pairs)	XLM-R-base	270M	15.1/s	52.3 / 67.0	44.4 / 64.5
	XLM-R-large	550M	4.9/s	**60.4 / 73.9**	**51.1 / 70.4**

XLM-RoBERTa

An experiment by UIT NLP research group in the UIT-ViQuAD paper



Demo

Thanks for listening!