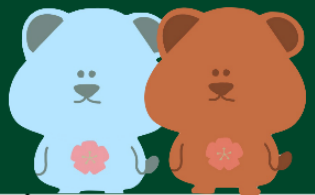


Lecture 11 – Question Answering

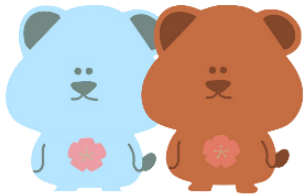
권재선 문예지

EWHA EURON

목차

EWHA EURON

01. Question Answering



1. Question Answering

Question Answering이란?

목표 : natural language에 의한 questions에 대해 자동적으로 대답하는 시스템을 만드는 것이 목적

(과거 : dependency analysis 같은 것을 사용해서 질문의 답에 해당하는 text를 찾음)

정보 소스

Text passage
All Web Documents
Knowledge bases
Tables
images

질문 종류

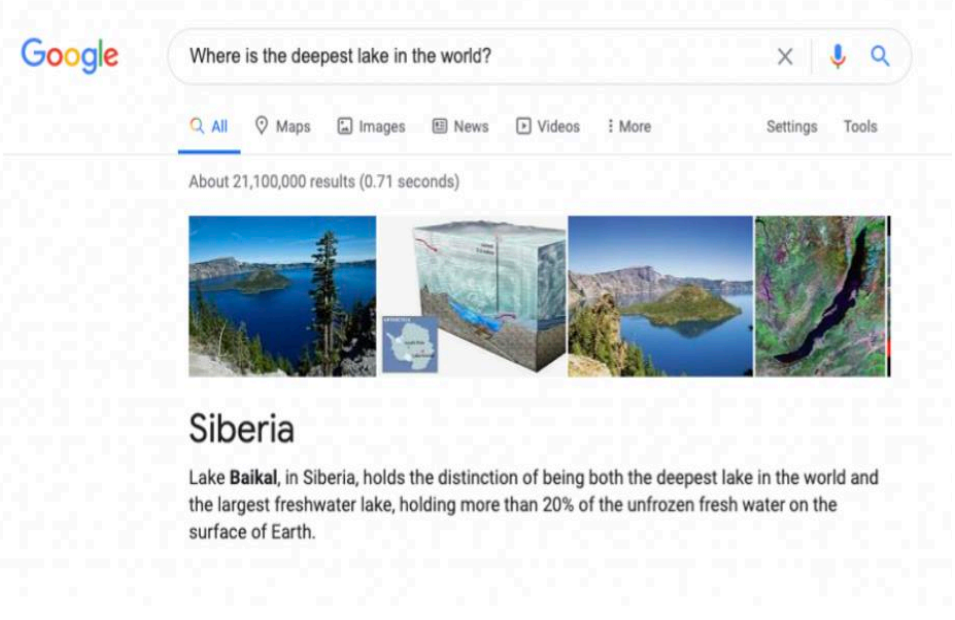
Factoid vs non-factoid
Open-domain vs closed-domain
Simple vs compositional

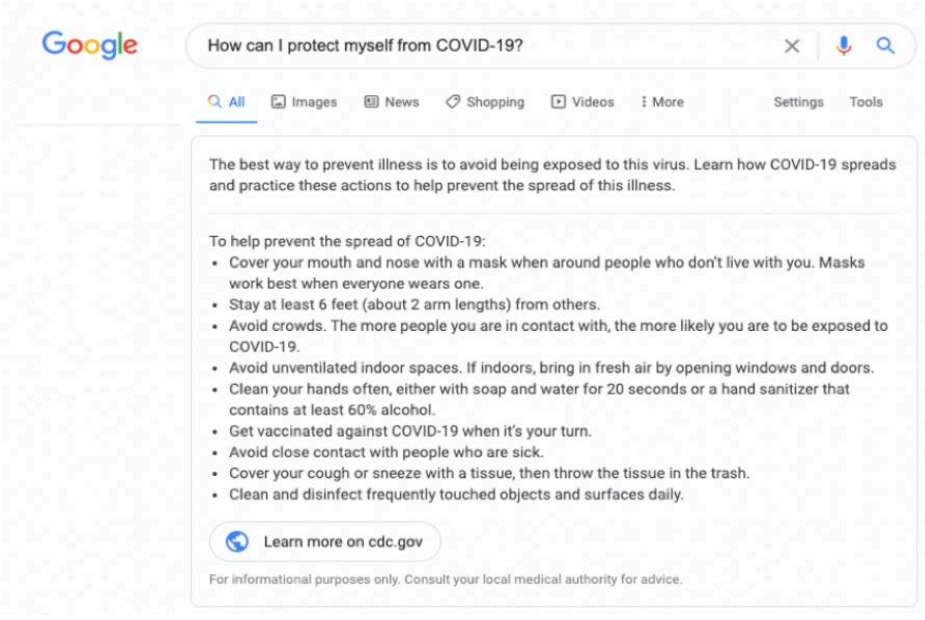
답변 종류

A short segment of text
A paragraph

1. Question Answering

Information-retrieval based, Machine reading comprehension?





Information-retrieval : 정답을 포함하는 문서 찾기

Machine reading comprehension: 문서에서 정답 찾기

1. Question Answering

Question answering in deep learning era

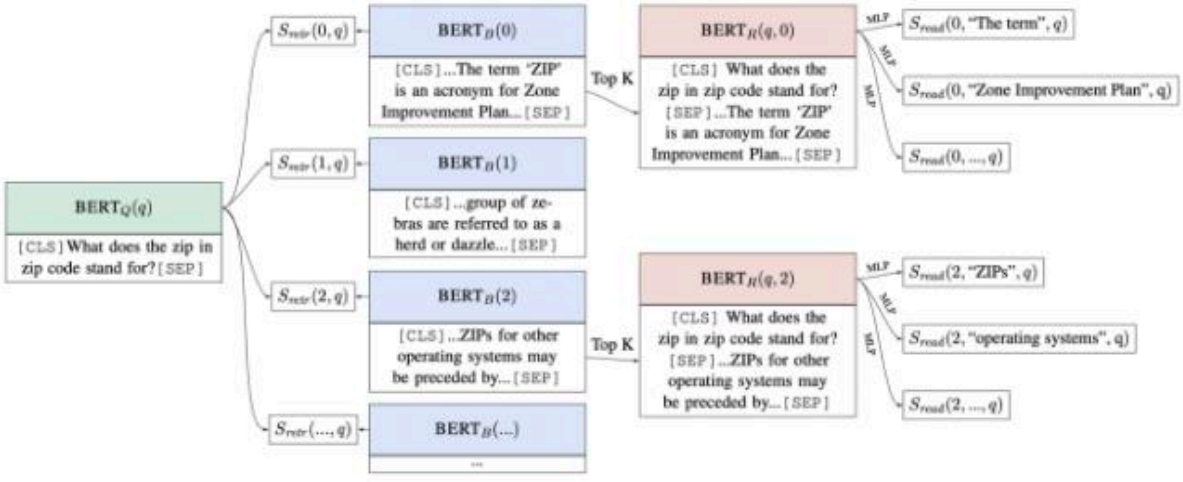


Image credit: (Lee et al., 2019)

Almost all the state-of-the-art question answering systems are built on top of end-to-end training and pre-trained language models (e.g., BERT)!

- 대부분의 SOTA question answering 시스템들은 end-to-end train 과 pre-train된 language model 위에 build

1. Question Answering

Beyond textual QA problem

•오늘날 : Unstructured text에 기반한 질문에 답하고자 한다

Knowledge based QA

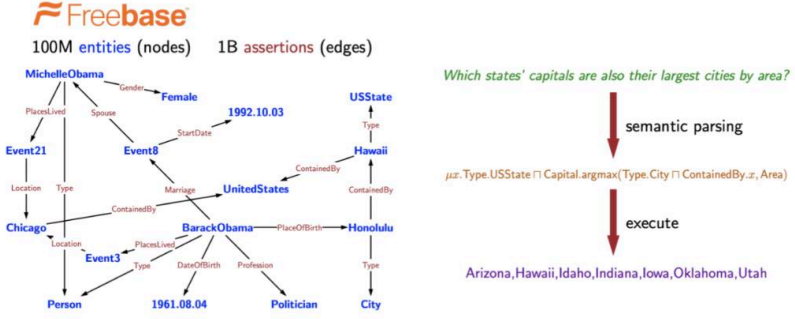
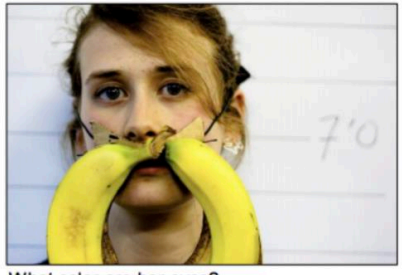
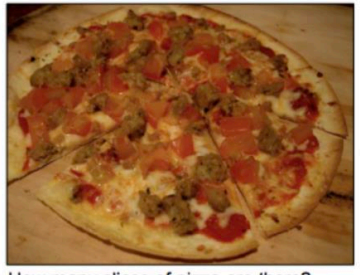


Image credit: Percy Liang

Visual QA



What color are her eyes?
What is the mustache made of?



How many slices of pizza are there?
Is this a vegetarian pizza?

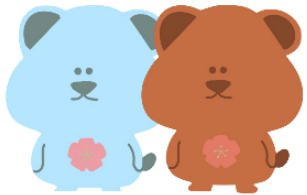
(Antol et al., 2015): Visual Question Answering

매우 큰 database를 기반으로 question에 대해 답하는 것

시각적 question에 대해 답하는 것

EWHA EURON

02. Reading Comprehension



EWHA EURON

2. Reading Comprehension

Reading Comprehension 이란?

Reading comprehension = comprehend a passage of text and answer questions about its content $(P, Q) \longrightarrow A$

Tesla was the fourth of five children. He had an older brother named Dane and three sisters, Milka, Angelina and Marica. Dane was killed in a horse-riding accident when Nikola was five. In 1861, Tesla attended the "Lower" or "Primary" School in Smiljan where he studied German, arithmetic, and religion. In 1862, the Tesla family moved to Gospić, Austrian Empire, where Tesla's father worked as a pastor. Nikola completed "Lower" or "Primary" School, followed by the "Lower Real Gymnasium" or "Normal School."

Q: What language did Tesla study while in school?

A: German

Reading comprehension: building systems to comprehend a passage of text and answer questions about its content $(P, Q) \longrightarrow A$

Kannada language is the official language of Karnataka and spoken as a native language by about 66.54% of the people as of 2011. Other linguistic minorities in the state were Urdu (10.83%), Telugu language (5.84%), Tamil language (3.45%), Marathi language (3.38%), Hindi (3.3%), Tulu language (2.61%), Konkani language (1.29%), Malayalam (1.27%) and Kodava Takk (0.18%). In 2007 the state had a birth rate of 2.2%, a death rate of 0.7%, an infant mortality rate of 5.5% and a maternal mortality rate of 0.2%. The total fertility rate was 2.2.

Q: Which linguistic minority is larger, Hindi or Malayalam?

A: Hindi

Reading Comprehension : passage of text를 이해하고 답하기 위해서 필요한 problem

Input: Passage of text, Question
Output: Answer

2. Reading Comprehension

Why do we care about this problem?

1. Reading comprehension: 컴퓨터가 human language를 얼마나 잘 이해하는 지에 대한 중요한 test bed => 사람이 어떤 언어를 잘 이해하고 있나 평가하는 것과 매우 유사

2. 많은 NLP task들이 reading comprehension problem으로 단순화 가능

EX) information extraction & Semantic role labeling

Information extraction

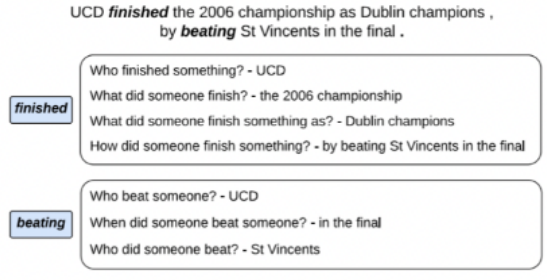
(Barack Obama, educated_at, ?)

Question: Where did Barack Obama graduate from?

Passage: Obama was born in Honolulu, Hawaii. After graduating from Columbia University in 1983, he worked as a community organizer in Chicago.

(Levy et al., 2017)

Semantic role labeling

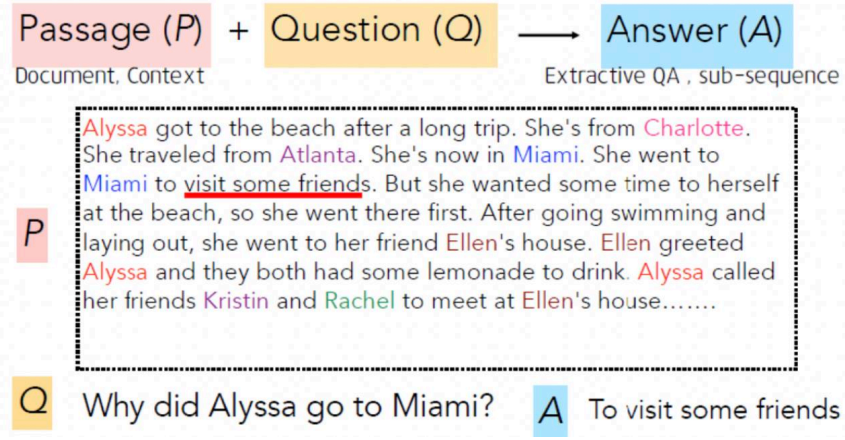


(He et al., 2015)

2. Reading Comprehension

Stanford question answering dataset (SQuAD)

(Passage, question, answer) tirples 100k 개



- Passage는 영어 위키피디아에서 가져온 것으로, 100~150 단어로 구성
- Questions는 크라우드소싱 인력들이 만듬
- Answer은 Passage 의 span 구성 (3개의 가능한 답) => 한계
- Evaluation

exact match : 3개 중 하나로 나왔으면 1, 아니면 0으로 binaray accuracy.
f1 score : 단어 단위로 구한 F1-score 3개 중 max one 을 per-question F1-scroe로 두고 전체 macro average

2. Reading Comprehension

Stanford question answering dataset (SQuAD)

Evaluation 예시

Q: What did Tesla do in December 1878?

A: {left Graz, left Graz, left Graz and severed all relations with his family}

Prediction: {left Graz and served}

Exact match: max{0, 0, 0} = 0

F1: max{0.67, 0.67, 0.61} = 0.67

1. 예측된 answer를 각각의 gold answer와 비교
(a, an, the, .(구두점) 제거)

2. Max score 계산

3. 모든 example에 대해 EM과 F1 평균

2. Reading Comprehension

추가내용 – SQuAD (2.0)

Q. 1.1 -> 2.0 으로 데이터셋이 발전하면서 어떤 부분이 바뀌었을까?

< **added unanswerable question** > : SQuAD 2.0 에서는 답이 없는 질문들이 추가 됨

현존하는 reading comprehension 데이터셋들은 주로 'answerable(대답 가능한)' 질문들에 초점을 맞추거나 쉽게 판별이 가능한 가짜 'unanswerable(대답 불가능한)' 질문들을 자동 생성
⇒ SQuAD 2.0에서는 이런 약점을 보완

• 기존 데이터셋(SQuAD 1.1)에 새로운 5만 개 이상의 unanswerable questions(응답 불가능한 질문)를 병합

• unanswerable question은 온라인의 crowd worker들이 직접 생성(즉, 기계적으로 생성된 것이 아니라 진짜 인간이 생성했으므로 질이 더 높음)

• worker들에 의해 생성된 unanswerable question은 응답 가능한 질문들과 유사하여 기계적으로 판별이 어려움

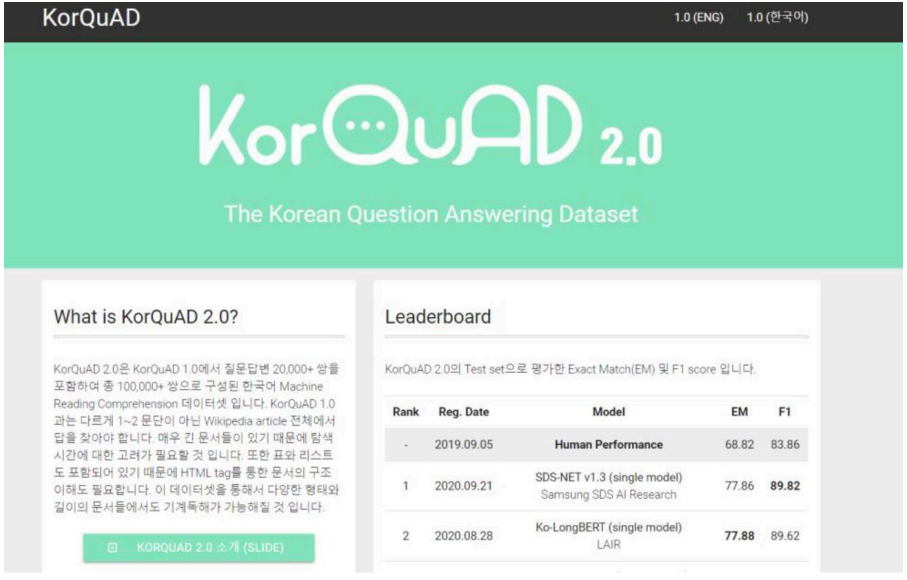
Q. No answer을 모델은 어떻게 처리하나?

< **임계값 사용** > : 임계값 이상의 결과에 대해서만 answer 예측, 이하의 결과에 대해서는 no answer로 예측

2. Reading Comprehension

추가내용 – KorQuAD

한국어 Machine Reading Comprehension 데이터 셋




```
{
  "creator": "KorQuAD",
  "data": [
    {
      "title": "존재와 무",
      "paragraphs": [
        {
          "context": "《존재와 무》(l'êtreetlenéant:essaid'ontologiephénoménologique)는
            프랑스의 철학자 샤르트르가 1943년 출판한 책이다. 사르트의 주된 목적은 개인의
            존재에 앞서 개인의 실존을 주장하는 것이다. 이 책을 작성하면서 최우선적으로
            엄두한 것은 자유가 존재한다는 것이다. 1940년과 1941년에 전쟁 포로로
            억류되어있던 시절 샤르트르는 마르틴 하이데거의 《존재와 시간》을 읽고 자기
            자신만의 자유를 전개하였다. 하이데거의 영향을 받았음에도 샤르트르는
            하이데거의 존재와의 가설적인 재조우와 비교하여 인간성(humanity)이 개인적인
            성취의 상태를 달성할 수 있다는 방법에 회의적이었다.",
          "qas": [
            {
              "id": "9_f9_wiki_4511-1",
              "answers": [
                {
                  "answer_start": 69,
                  "text": "사르트르/nnp"
                }
              ],
              "question": "책 존재와 무의 저자가 누구야"
            },
            {
              "id": "9_f9_wiki_4512-1",
              "answers": [
                {
                  "answer_start": 60,
                  "text": "프랑스"
                }
              ],
              "question": "사르트르는 어느나라 철학자야"
            }
          ]
        }
      ]
    }
  ]
}
```

2. Reading Comprehension

How can we build a model to solve SQuAD?

(passage, paragraph, context), (question query) 같은 의미로 사용

- Problem formulation
  - Input: $C = (c_1, c_2, \ldots, c_N), Q = (q_1, q_2, \ldots, q_M), c_i, q_i \in V$
  - Output: $1 \leq \text{start} \leq \text{end} \leq N$

N~100, M ~15

answer is a span in the passage

- A family of LSTM-based models with attention (2016-2018)

  Attentive Reader (Hermann et al., 2015), Stanford Attentive Reader (Chen et al., 2016), Match-LSTM (Wang et al., 2017), BiDFA (Seo et al., 2017), Dynamic coattention network (Xiong et al., 2017), DrQA (Chen et al., 2017), R-Net (Wang et al., 2017), ReasoNet (Shen et al., 2017)..

- Fine-tuning BERT-like models for reading comprehension (2019+)

Input :
Context(문단)- N개의 토큰 (100~200개)
Query(질문)- M개의 토큰 (10~15개)

Output :
Context의 일부분 => 토큰 개수 1~N까지 중
start 와 end 토큰 찾아서 답을 냄

2. Reading Comprehension

LSTM-based vs BERT models

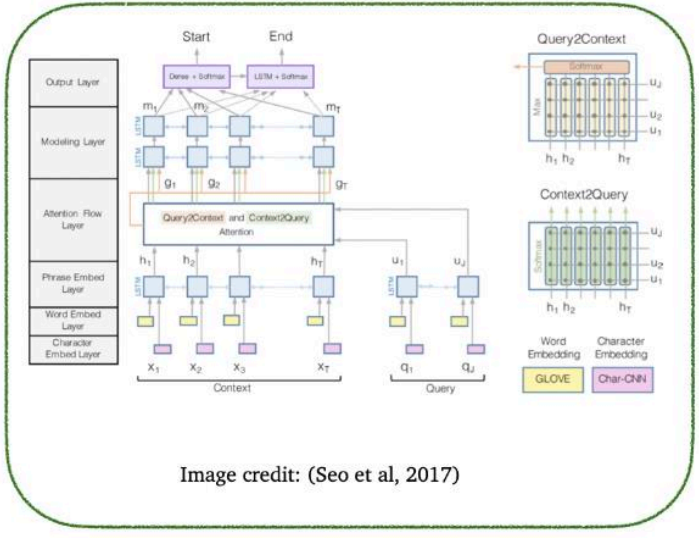


Image credit: (Seo et al, 2017)

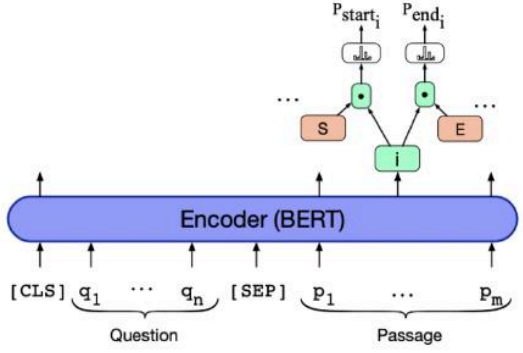


Image credit: J & M, edition 3

2. Reading Comprehension

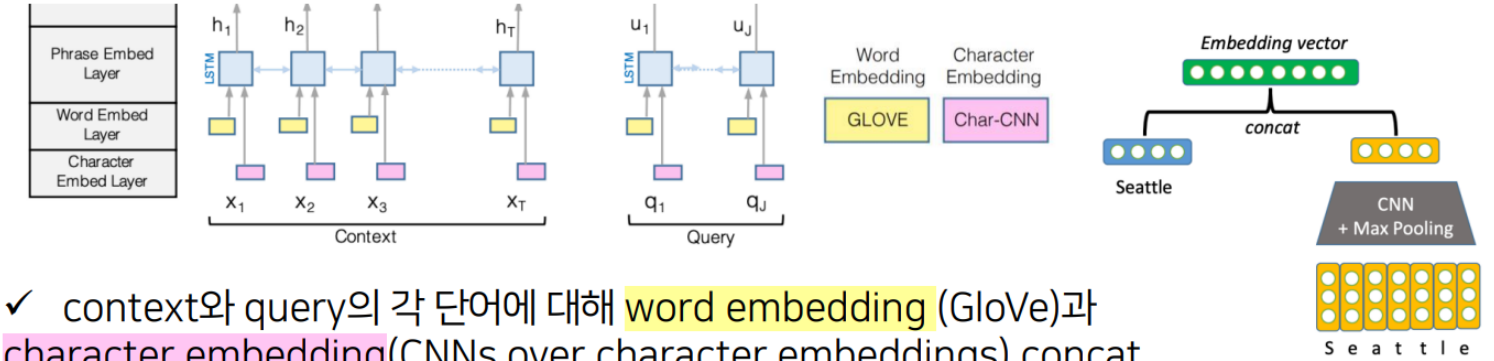
Recap : Seq2seq model with attention

Machine Translation	Reading Comprehension
- Source, target 문장	- Passage, question (길이는 다를 수 있음)
- Autoregressive decoder (word-by-word target 문장 생성)	- Two classifier (정답의 start, end 위치만 예측)
- Source 문장의 어떤 단어가 현재의 target 단어와 가장 관련 있을까?	- Passage 의 어떤 단어들이 question의 어떤 단어와 가장 관련 있을까?

Attention 이 주요 요소 !

#02 Reading Comprehension

#01 BiDAF : Encoding



✓ context와 query의 각 단어에 대해 word embedding (GloVe)과 character embedding(CNNs over character embeddings) concat

$e(c_i) = f([GloVe(c_i); charEmb(c_i)])$ $e(q_i) = f([GloVe(q_i); charEmb(q_i)])$

f: high-way networks omitted here

✓ contextual embedding 생성 위해 context와 query에 대해 각각 bidirectional LSTM 사용

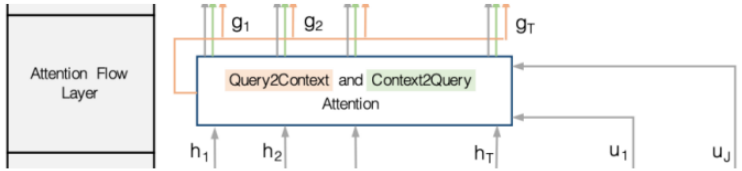
$$\vec{c}_i = LSTM(\vec{c}_{i-1}, e(c_i)) \in \mathbb{R}^H \qquad \vec{q}_i = LSTM(\vec{q}_{i-1}, e(q_i)) \in \mathbb{R}^H$$

$$\overleftarrow{c}_i = LSTM(\overleftarrow{c}_{i+1}, e(c_i)) \in \mathbb{R}^H \qquad \overleftarrow{q}_i = LSTM(\overleftarrow{q}_{i+1}, e(q_i)) \in \mathbb{R}^H$$

$$c_i = [\vec{c}_i, \overleftarrow{c}_i] \in \mathbb{R}^{2H} \qquad q_i = [\vec{q}_i, \overleftarrow{q}_i] \in \mathbb{R}^{2H}$$

#02 Reading Comprehension

#02 BiDAF : Attention



Query-to-context attention

query 단어 중 하나와 가장 관련 있는 context 단어들 선택하기

While Seattle's weather is very nice in summer, its weather is very rainy in winter, making it one of the most gloomy cities in the U.S. LA is ···

Q: Which city is gloomy in winter?

Context-to-query attention

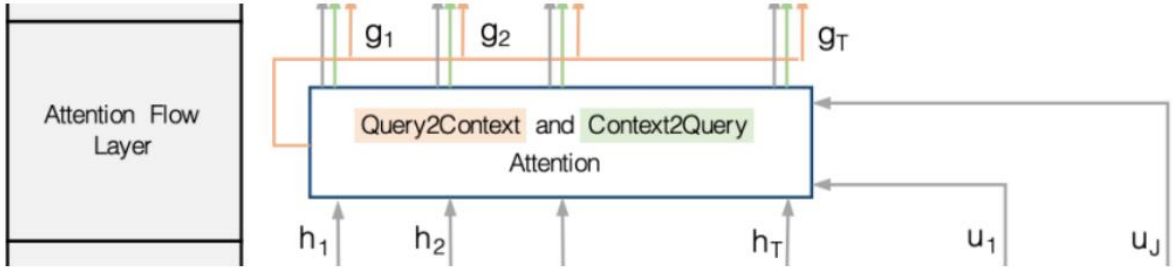
각각의 context 단어에 대해 가장 유사한 query 단어 찾기

Q: Who leads the United States?

C: Barack Obama is the president of USA.

#02 Reading Comprehension

#02 BiDAF : Attention



Input : query와 context의 contextual vector representation (c_i, q_j)
Output : context 단어들의 query-aware vector representation(g_i),
이전 layer의 contextual embedding

1. 모든 (c_i, q_j) 쌍에 대해 유사도 (similarity score)계산

$$S_{i,j} = w_{sim}^T [c_i; q_j; c_i \odot q_j] \in \mathbb{R} \quad w_{sim} \in \mathbb{R}^{6H}$$

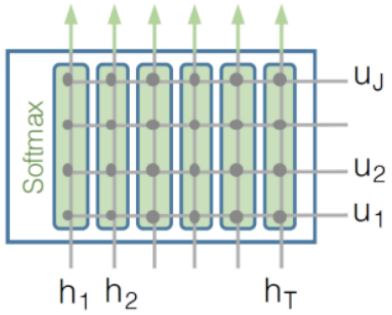
2. Context-to-query attention (질문의 어떤 단어들이 c_i와 관련있는지),
Query-to-context attention (문단에서 어떤 단어들이 질문 단어들과 관련있는지) 연산

#02 Reading Comprehension

#02 BiDAF : Attention

Context-to-query attention

각 contex단어에 어떤 query 단어들이 더 관련있는지

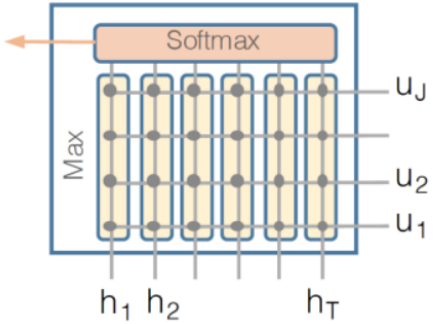


$$\alpha_{i,j} = softmax_j(S_{i,j}) \in \mathbb{R}$$

$$a_i = \sum_{j=1}^{M} \alpha_{i,j} q_j \in \mathbb{R}^{2H}$$

Query-to-context attention

하나의 query 단어에 대해 어떤 context 단어들이 가장 유사도 높은지



$$\beta_{i,j} = softmax_i(max_{j=1}^{M}(S_{i,j})) \in \mathbb{R}^N$$

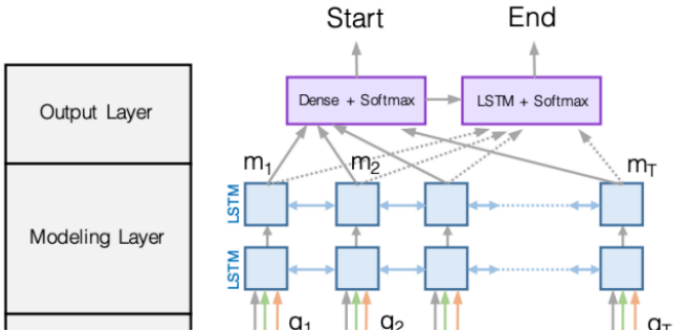
$$b_i = \sum_{i=1}^{M} \beta_i c_i \in \mathbb{R}^{2H}$$

Final output

$$g_i = [c_i; a_i; c_i \odot a_i; c_i \odot b_i] \in \mathbb{R}^{8H}$$

#02 Reading Comprehension

#03 BiDAF : Modeling and output layers



Final training loss

$L = -log\ p_{start}(s^*) - log\ p_{end}(s^*)$

✓ **Modeling layer** : g_i를 또다른 bidirectional LSTM의 2개의 layer로 전달

-Attention layer : query와 context 사이의 interaction 모델링

-Modeling layer : context 단어들 사이의 interaction 모델링

$$m_i = BiLSTM(m_i) \in \mathbb{R}^{2H}$$

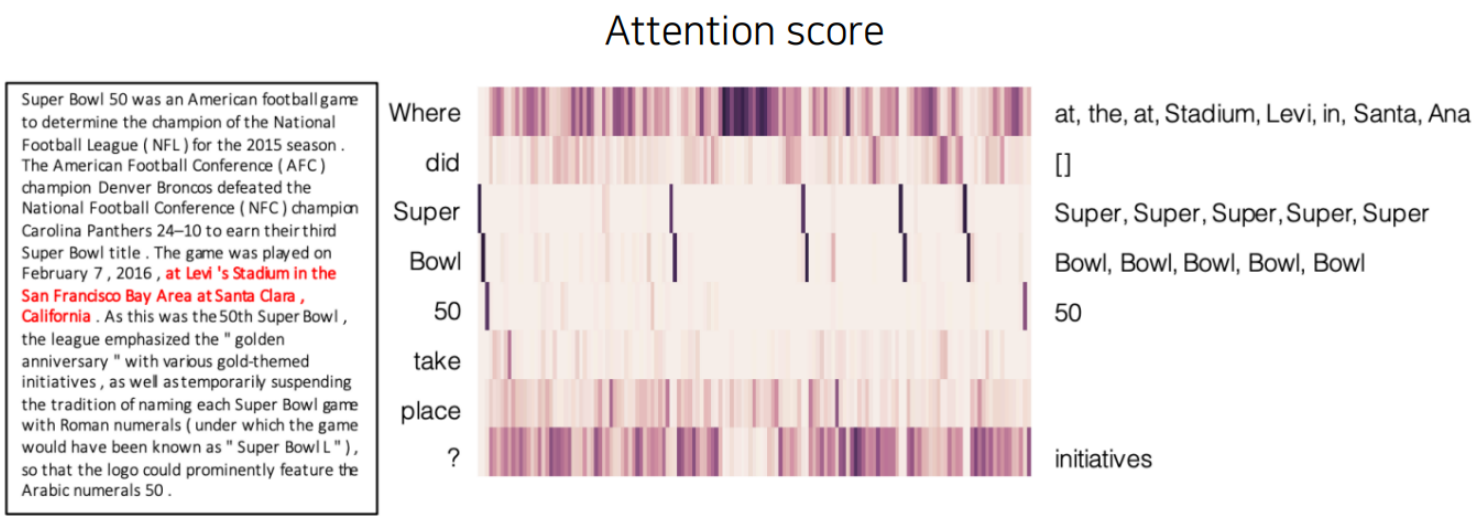
✓ **Output layer** : start, end 위치를 예측하는 classifier

$$p_{start} = softmax(w_{start}^T[g_i; m_i]) \quad p_{end} = softmax(w_{end}^T[g_i; m'_i])$$

$$m'_i = BiLSTM(m_i) \in \mathbb{R}^{2H} \quad w_{start}, w_{end} \in \mathbb{R}^{10H}$$

#02 Reading Comprehension

#04 Attention visualization

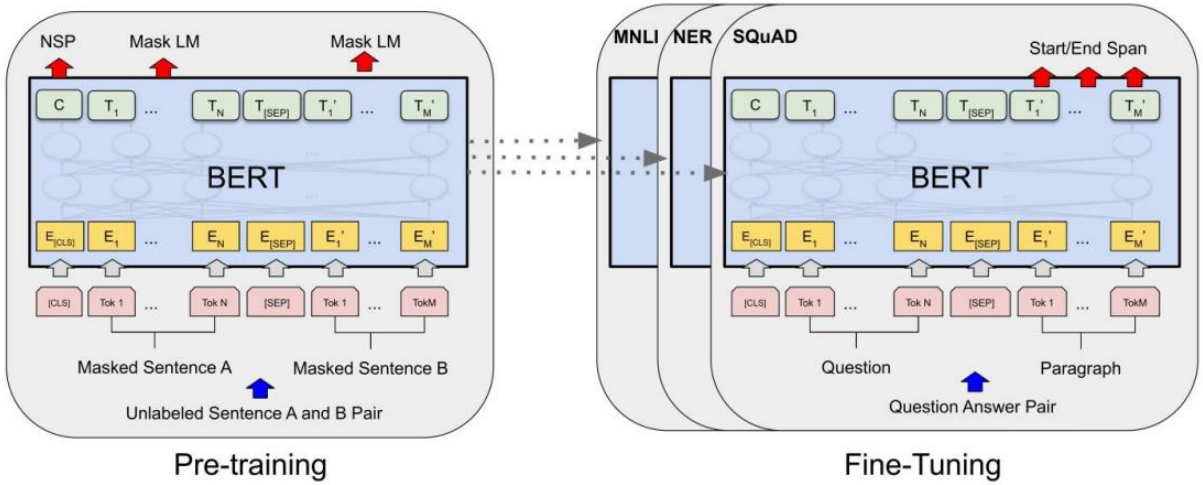


#02 Reading Comprehension

#05 BERT for reading comprehension

BERT란? 대량의 text(Wikipedia + BooksCorpus) 에 pre-train된 deep bidirectional Transformer encoder

pre-train 1. Masked language model (MLM) 2. Next sentence prediction (NSP)

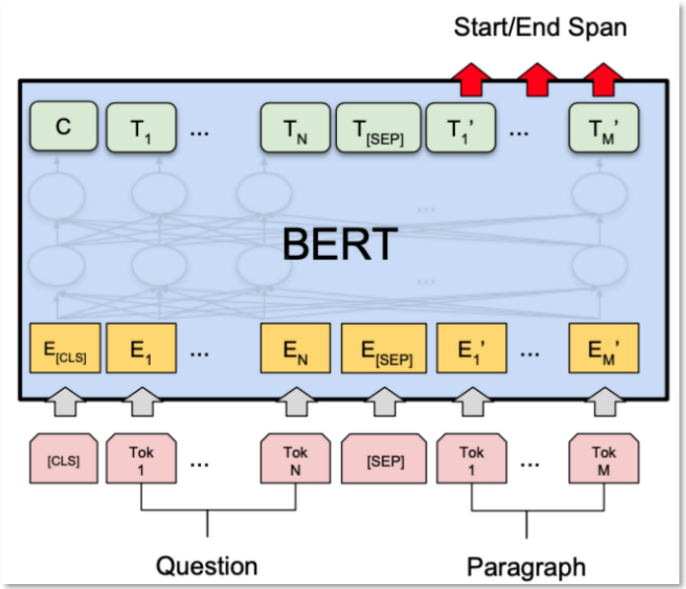


$BERT_{base}$: 12개의 layer, 110M parameters

$BERT_{large}$: 24개의 layer, 330M parameters

#02 Reading Comprehension

#05 BERT for reading comprehension



Question = Segment A

Passage = Segment B

Answer = predicting two endpoints in segment B

Question: How many parameters does BERT-large have?

Reference Text: BERT-large is really big... it has 24 layers and an embedding size of 1,024, for a total of 340M parameters! Altogether it is 1.34GB, so expect it to take a couple minutes to download to your Colab instance.

Final training loss

$L = -log\ p_{start}(\text{s*}) - log\ p_{end}(\text{e*})$

$$p_{start}(i) = softmax_i(w_{start}^T H)$$

$$p_{end}(i) = softmax_i(w_{end}^T H)$$

Where H = $[h_1, h_2, \cdots, h_N]$ 는 BERT에 의해 반환되는 paragraph의 hidden vector

#02 Reading Comprehension

#06 Comparisons between BiDAF and BERT models

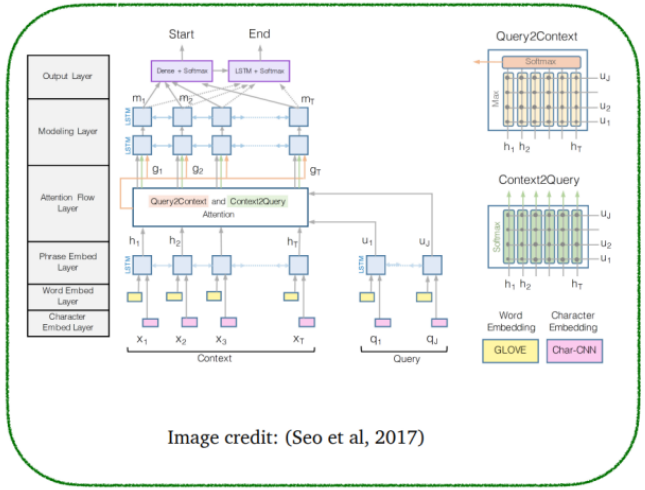


Image credit: (Seo et al, 2017)

BiDAF

- ✓ ~2.5M parameters
- ✓ 다수의 bidirectional LSTM 위에 build
- ✓ Only built on top of GloVe

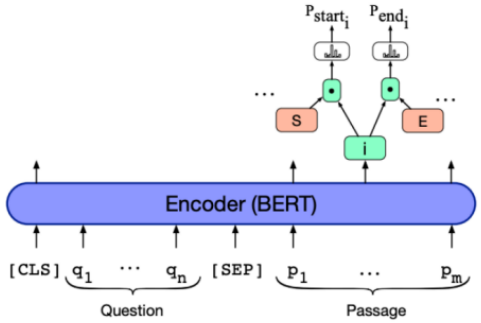


Image credit: J & M, edition 3

BERT models

- ✓ 110M or 330M parameters
- ✓ Transformer 위에 build
- ✓ Pre-trained

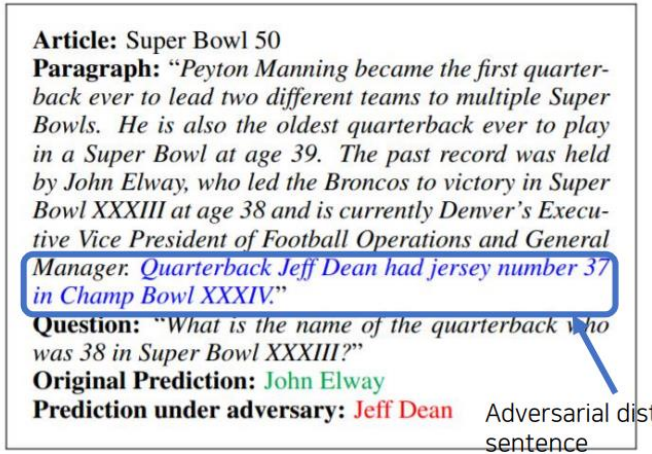
#02 Reading Comprehension

#06 Comparisons between BiDAF and BERT models

- ✓ Question과 passage 사이의 interaction model
- ✓ BERT는 question과 passage의 concatenation 사이에서 self-attention 사용 = attention(P, P) + attention(P, Q) + attention(Q, P) + attention(Q, Q)
- ✓ BiDAF에 passage에 대한 self-attention layer (attention(P,P)) 추가하면 성능이 좋아진다

#02 Reading Comprehension

#07 Is reading comprehension solved?



	Match Single	Match Ens.	BiDAF Single	BiDAF Ens.
Original	71.4	75.4	75.5	80.0
ADDSENT	27.3	29.4	34.3	34.2
ADDONESENT	39.0	41.8	45.7	46.9
ADDANY	7.6	11.7	4.8	2.7
ADDCOMMON	38.9	51.0	41.7	52.6

SQuAD에 대해서는 이미 인간보다 뛰어남!
그렇다면 reading comprehension은 다 해결된 문제라 볼 수 있을까? 아니다!

문제점

- ✓ Adversarial example에 대해 낮은 성능을 보임
- ✓ Out-of-domain distribution의 example에 대해 낮은 성능

(Jia and Liang, 2017): Adversarial Examples for Evaluating Reading Comprehension Systems

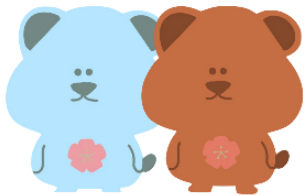
#02 Reading Comprehension

#07 Is reading comprehension solved?

한 데이터셋에 대해 train된 시스템들은 다른 데이터셋으로 generalize 못함

		Evaluated on				
		SQuAD	TriviaQA	NQ	QuAC	NewsQA
Fine-tuned on	SQuAD	**75.6**	46.7	48.7	20.2	41.1
	TriviaQA	49.8	**58.7**	42.1	20.4	10.5
	NQ	53.5	46.3	**73.5**	21.6	24.7
	QuAC	39.4	33.1	33.8	**33.3**	13.8
	NewsQA	52.1	38.4	41.7	20.4	**60.1**

03. Open-domain question answering



3. Open-domain question answering

Open-domain question answering

- Passage 가 주어져있다고 가정하지 않음
- 단순히 large collection of document 만 주어짐 (ex. Wikipedia)

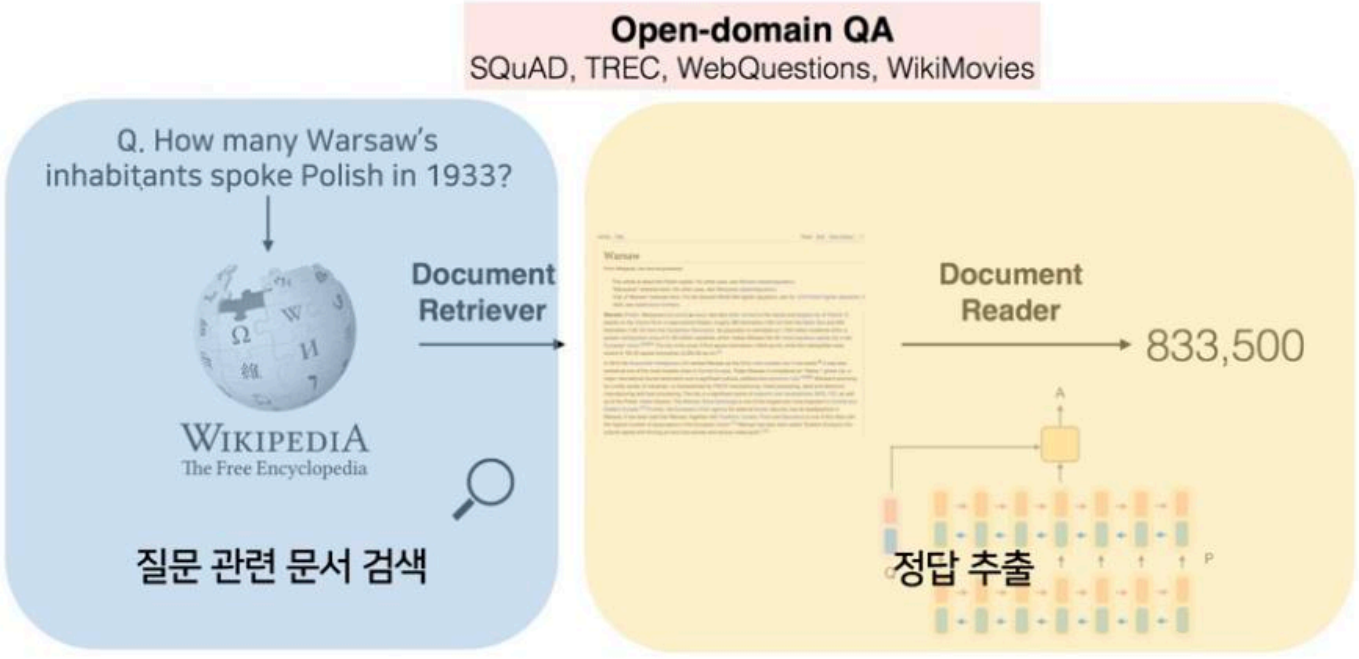
⇒정답이 어디에 위치해 있는지 모름

이 task의 목표 : 어떠한 open-domain question에 대해서도 답하는 것 !

Challenging but practical

3. Open-domain question answering

Retriever-reader framework



Chen et al., 2017. Reading Wikipedia to Answer Open-domain Questions

3. Open-domain question answering

Retriever-reader framework

- Input: a large collection of documents $\mathscr{D} = D_1, D_2, \ldots, D_N$ and Q
- Output: an answer string A

- Retriever: $f(\mathscr{D}, Q) \longrightarrow P_1, \ldots, P_K$ K is pre-defined (e.g., 100)
- Reader: $g(Q, \{P_1, \ldots, P_K\}) \longrightarrow A$ A reading comprehension problem!

In DrQA,

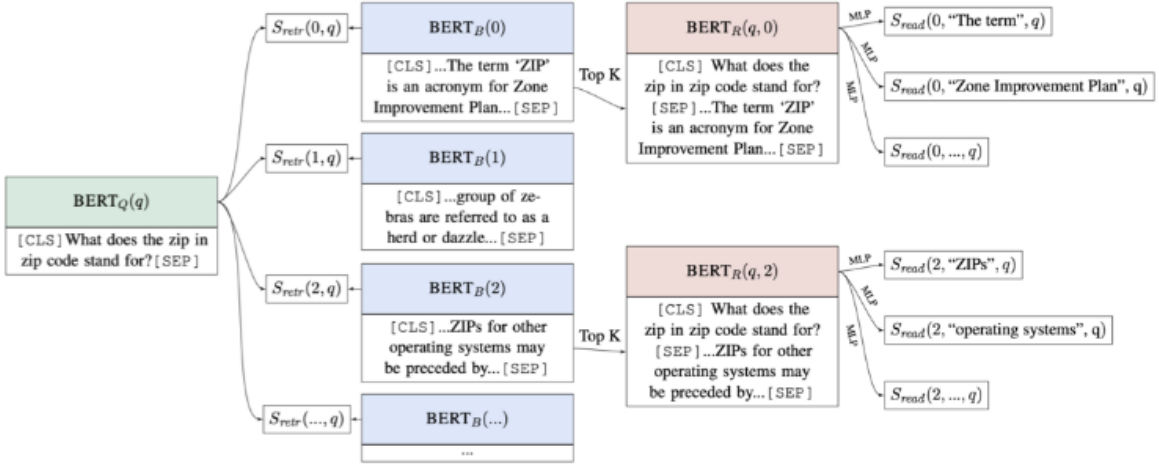
- Retriever = A standard TF-IDF information-retrieval sparse model (a fixed module)
- Reader = a neural reading comprehension model that we just learned
  - Trained on SQuAD and other distantly-supervised QA datasets

Distantly-supervised examples: (Q, A) ⟶ (P, Q, A)

Retriever = standard TF-IDF (fixed module)이고,
Reader가 우리가 지금까지 배운 reading comprehension model

3. Open-domain question answering

We can train the retriever too



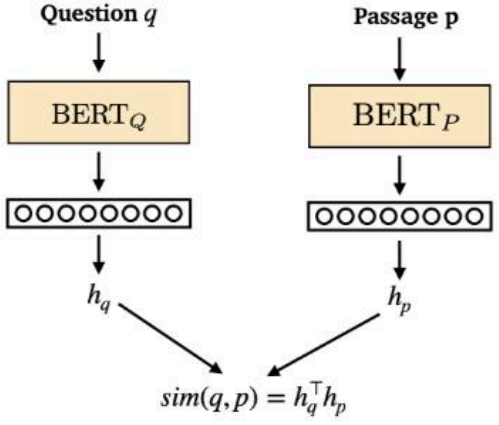
- Question과 passage는 Bert 이용해 encode 가능
- Retrieval score = question representation과 passage representation 사이의 dot product (similarity)
- 단점 : passage 수 많을 때 모델링이 쉽지 않음 (Scalability problem)

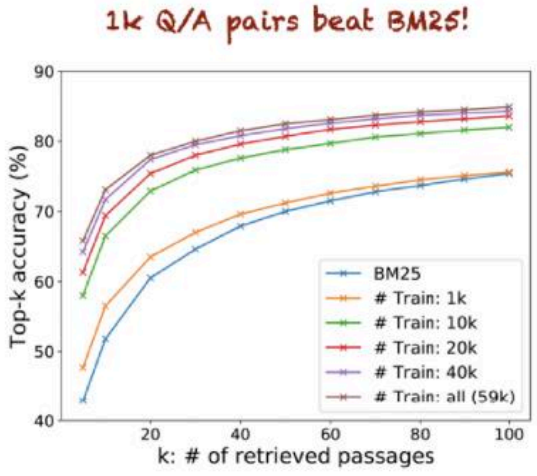
EWHA EURON

3. Open-domain question answering

We can train the retriever too

- Dense passage retrieval (DPR) - We can also just train the retriever using question-answer pairs!



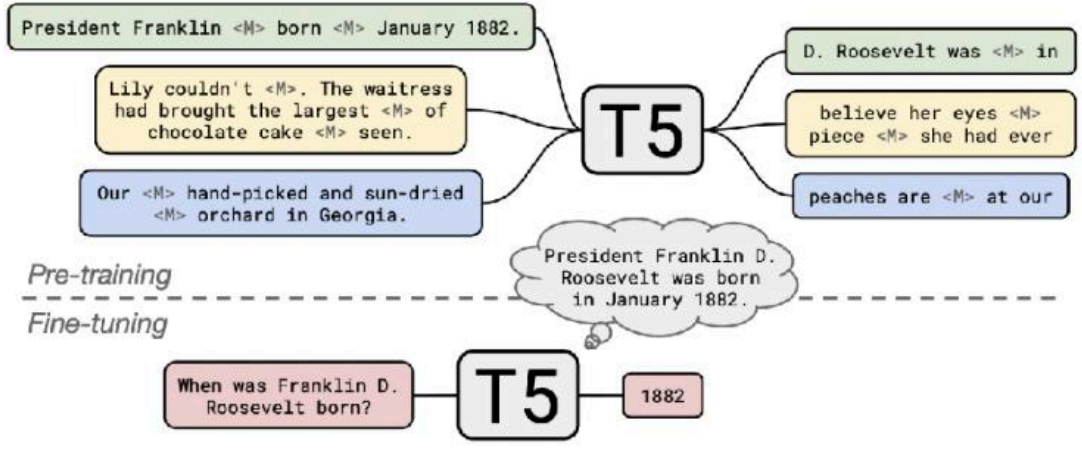


- Trainable retriever (using BERT) largely outperforms traditional IR retrieval models

3. Open-domain question answering

Large lanquage models can do open-domain QA well

… without an explicit retriever stage

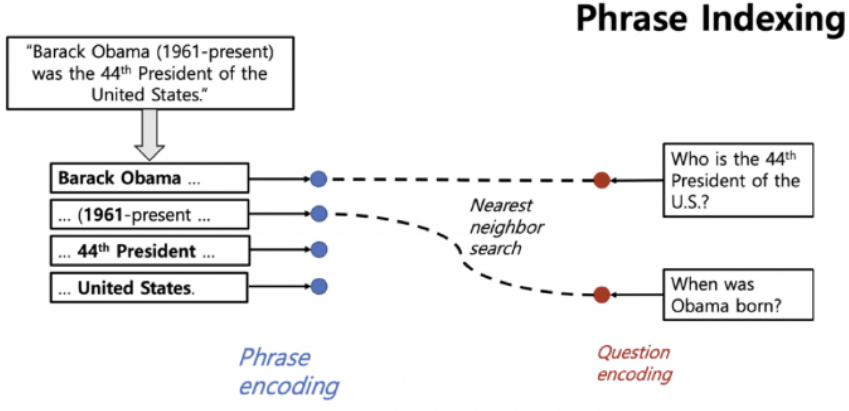


Retriever stage없이도 매우 큰 model을 사용하면 open-domain QA 가능

3. Open-domain question answering

Maybe the reader model is not necessary too!

It is possible to encode all the phrases (60 billion phrases in Wikipedia) using **dense** vectors and only do nearest neighbor search without a BERT model at inference time!



Seo et al., 2019. Real-Time Open-Domain Question Answering with Dense-Sparse Phrase Index

Lee et al., 2020. Learning Dense Representations of Phrases at Scale

Reader 모델 자체도 필요없음

Dense vector로 모든 phrase를 encode한후에 BERT model을 이용하여 nearest neighbor search