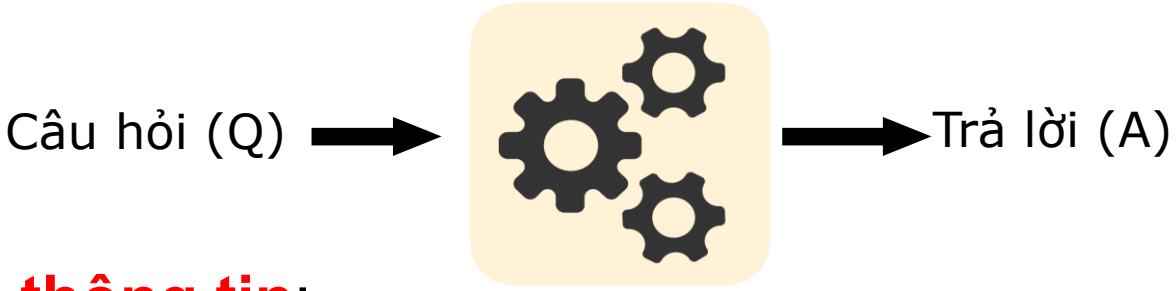
Bài 12
Các hệ thống hỏi đáp tự động
Question Answering Systems

Lê Thanh Hương
Trường Công nghệ Thông tin và Truyền thông, ĐHBKHN

Bài toán hỏi đáp tự động

- **Mục tiêu**: Xây dựng các hệ thống tự động trả lời các câu hỏi của con người bằng ngôn ngữ tự nhiên



- **Nguồn thông tin**:
  - Đoạn văn bản, các tài liệu trên web, cơ sở tri thức, cơ sở dữ liệu, tập các câu hỏi đáp có sẵn
- **Các dạng câu hỏi**: trả về giá trị/trả về không phải là giá trị, miền đóng/miền mở, đơn giản/phức tạp, …
- **Các dạng câu trả lời**: một vài từ, một đoạn, danh sách, có/không, …

Bài toán hỏi đáp tự động

Các dạng câu hỏi

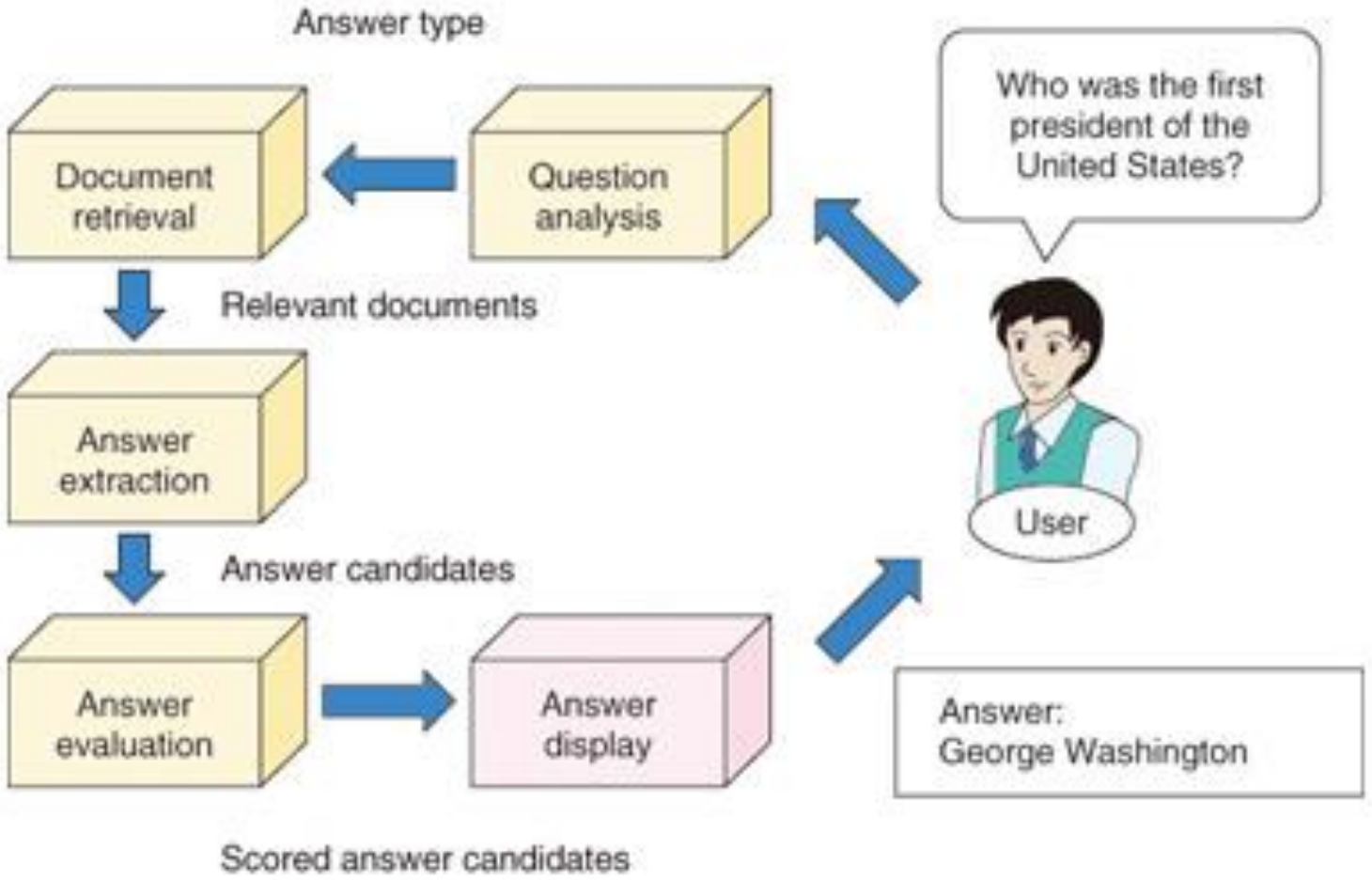
- Factoid queries: WH questions like when, who, where.
- Yes/ No queries: Is Berlin capital of Germany?
- Definition queries: what is leukemia?
- Cause/consequence queries: How, Why, What. what are the consequences of the Iraq war?
- Procedural queries: which are the steps for getting a Master degree?
- Comparative queries: what are the differences between the model A and B?
- Queries with examples: list of hard disks similar to hard disk X.
- Queries about opinion: What is the opinion of the majority of Americans about the Iraq war?

Các cách tiếp cận

- Khi có bộ dữ liệu QA cho trước
  - Đo độ tương đồng câu, lấy câu trả lời của câu hỏi tương đồng nhất.
    - VD: AskJeeves
  - Huấn luyện sử dụng học sâu để dự đoán câu trả lời
- Không có bộ dữ liệu QA, có CSDL hoặc CSTT
  - Phân tích câu hỏi (phân tích ngữ nghĩa sâu, so khớp mẫu,…), tìm câu trả lời (tra cứu CSDL, so khớp mẫu, suy diễn, …)
    - VD: TextMap, AskMSR, LCC, …
  - Tìm kiếm các tài liệu liên quan, tìm câu trả lời từ tài liệu liên quan

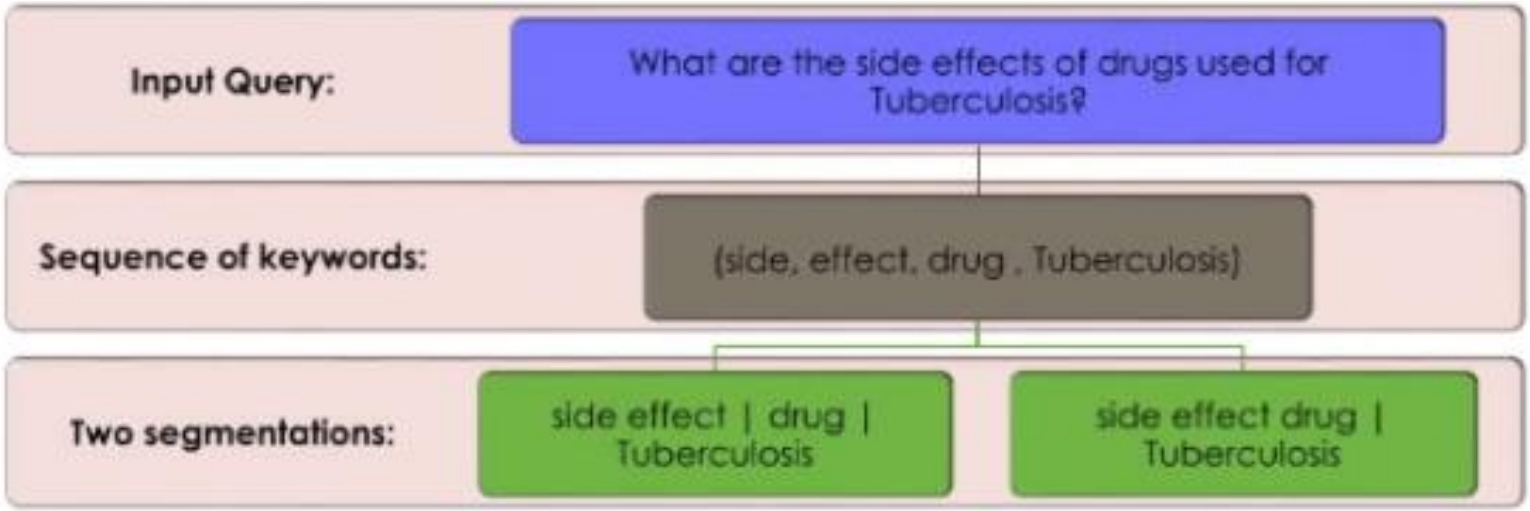
CÁC BÀI TOÁN CƠ BẢN

Hỏi đáp miền đóng

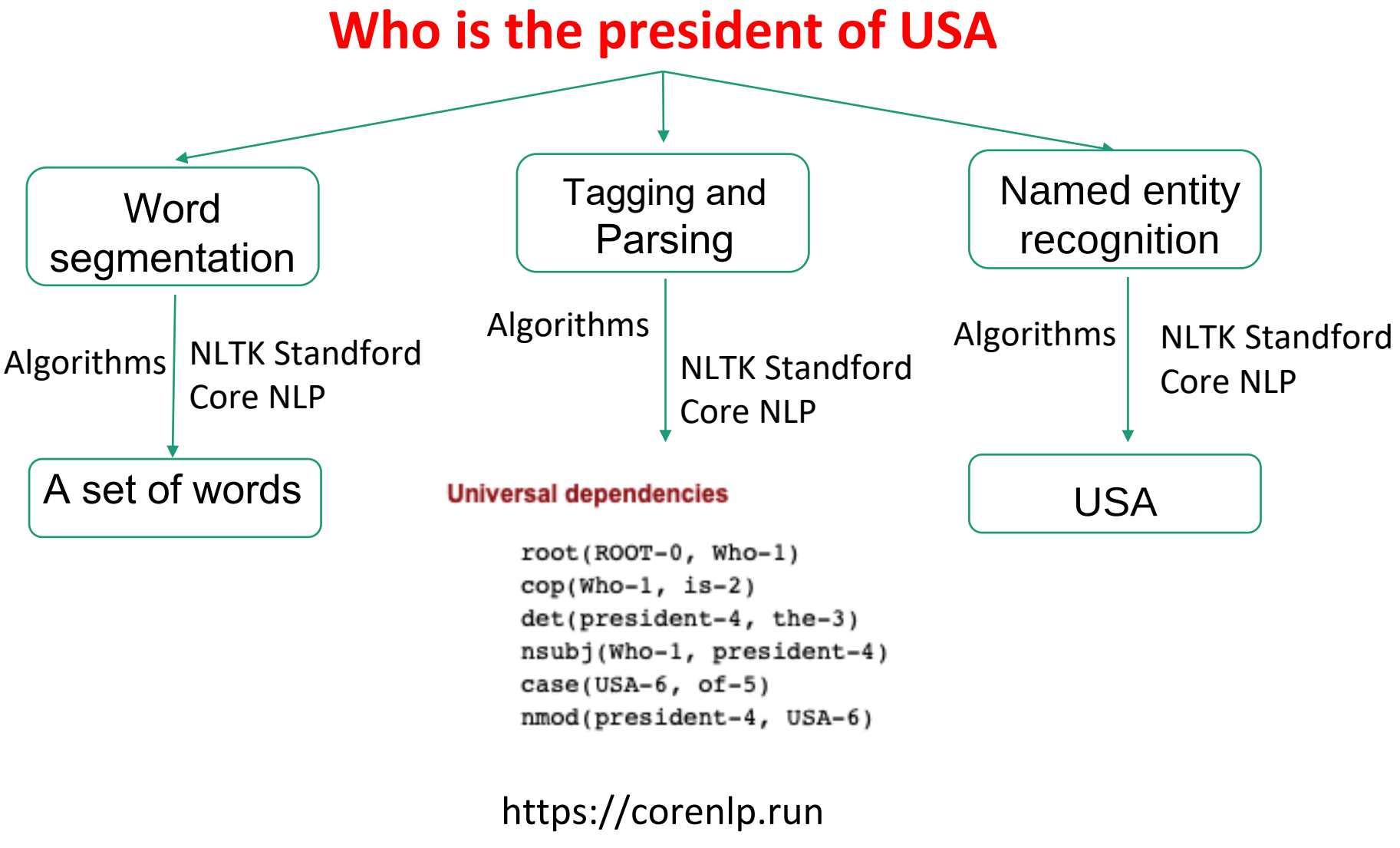


Phân tích câu hỏi

- Xác định các thành phần quan trọng nhất trong câu hỏi để tạo ra câu truy vấn theo từ khóa

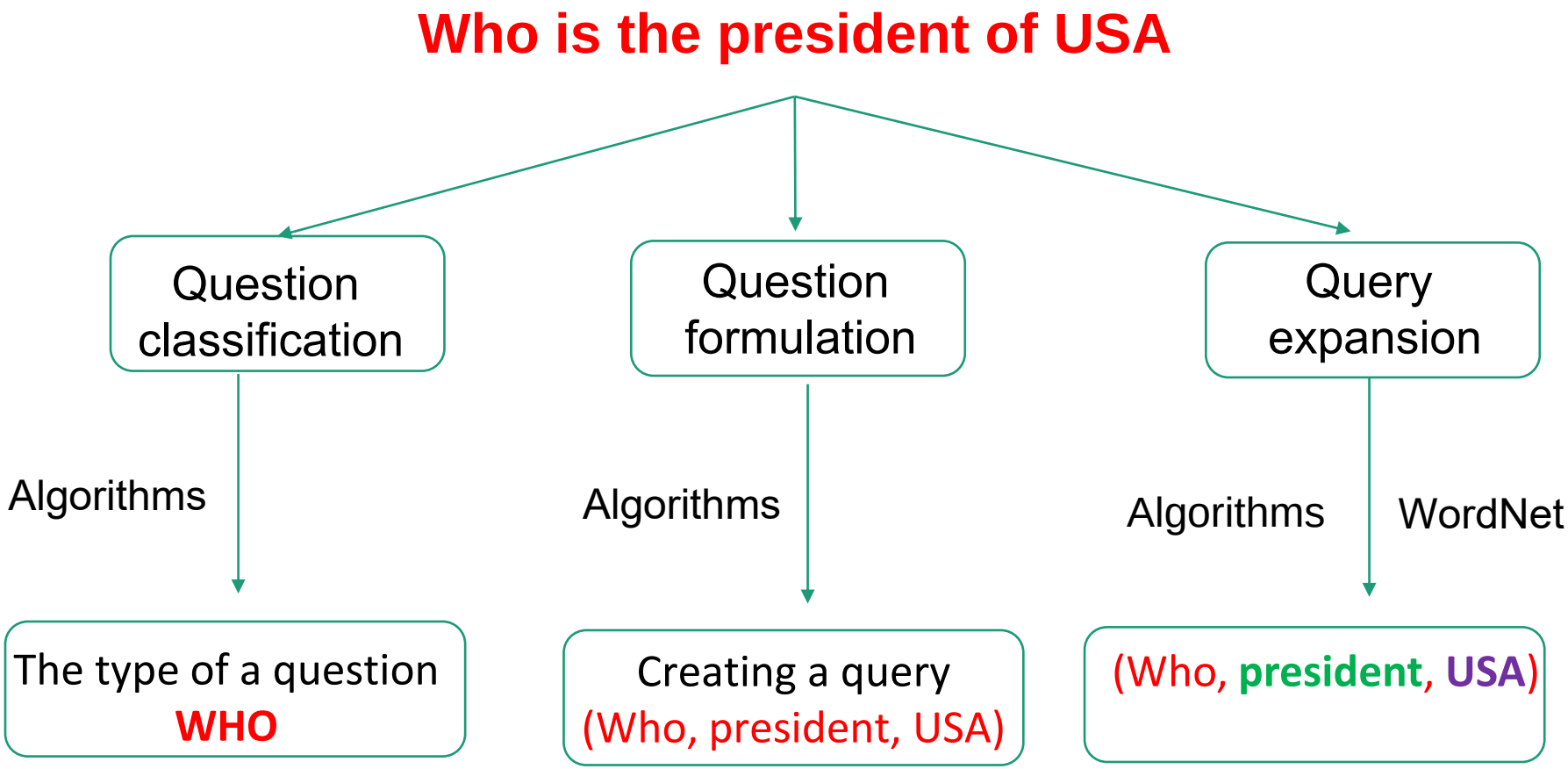


Tiền xử lý dữ liệu

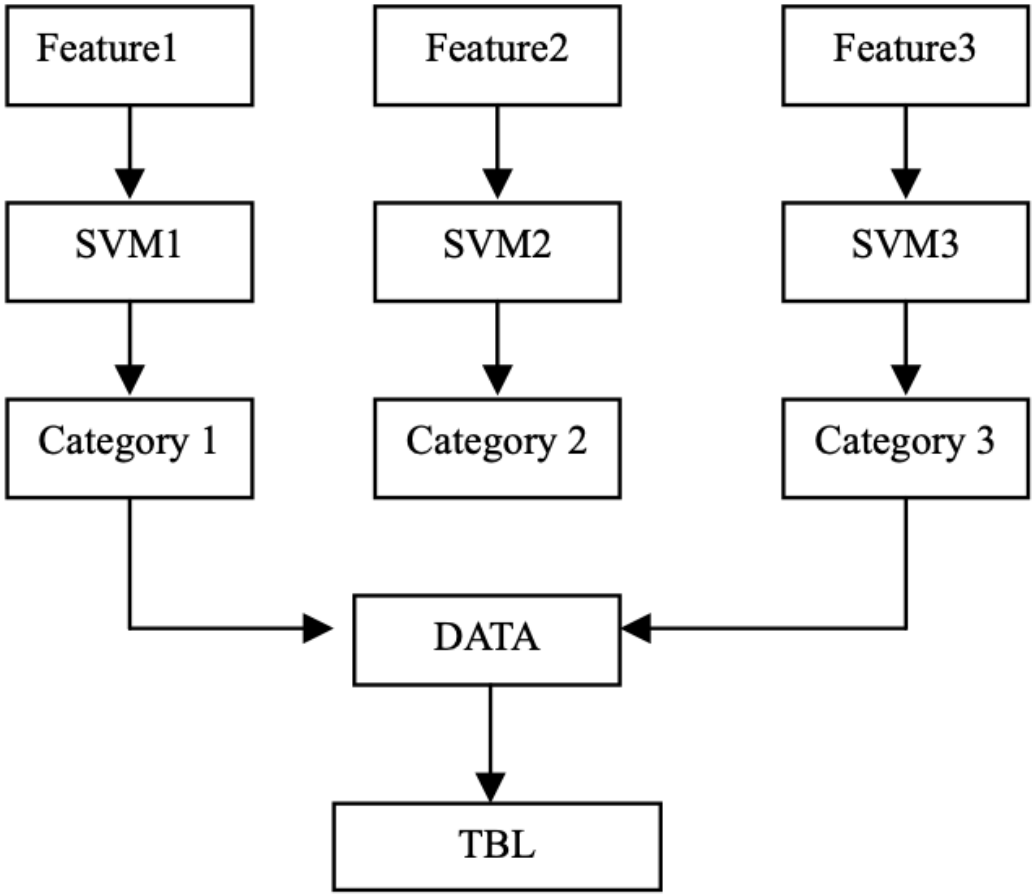


https://corenlp.run

Phân tích câu hỏi



Sử dụng SVM trong phân lớp câu hỏi



https://www.aclweb.org/anthology/I05-4009.pdf

Sử dụng SVM trong phân lớp câu hỏi

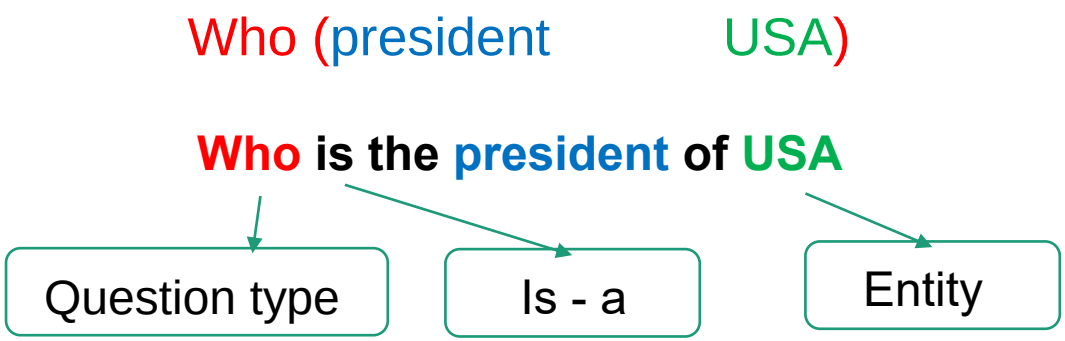
Tập đặc trưng của SVM

Kernel & feature \ Num. of Training		1000	2000	3000	4000	5500
Liner	Bag-of-word	79.6	81.2	83.4	85.8	84.8
	Wordnet	77.8	83.8	85.2	86.4	86.8
	Bi-gram	73.6	80.6	83.2	87.4	88.6
	Dependency	82.0	86.8	87.2	88.4	89.2
polyn omial	Bag-of-word	52.4	69.2	66.0	61.4	62.6
	Wordnet	48.4	69.8	70.0	68.8	73.2
	Bi-gram	27.6	49.2	46.4	49.6	50.8
	Dependency	73.0	78.8	81.8	82.4	85.2
RBF	Bag-of-word	68.8	73.2	80.2	81.4	83.6
	Wordnet	69.0	73.2	79.8	80.2	81.0
	Bi-gram	62.2	70.2	76.0	80.0	81.2
	Dependency	72.8	78.8	81.0	83.2	85.0
Sig moid	Bag-of-word	65.6	74.2	77.0	78.2	80.2
	Wordnet	74.2	82.6	83.4	83.8	84.4
	Bi-gram	68.6	74.4	79.8	83.2	84.8
	Dependency	75.2	78.0	82.4	83.4	85.2

https://www.aclweb.org/anthology/I05-4009.pdf

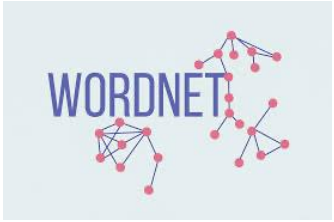
Tạo câu truy vấn

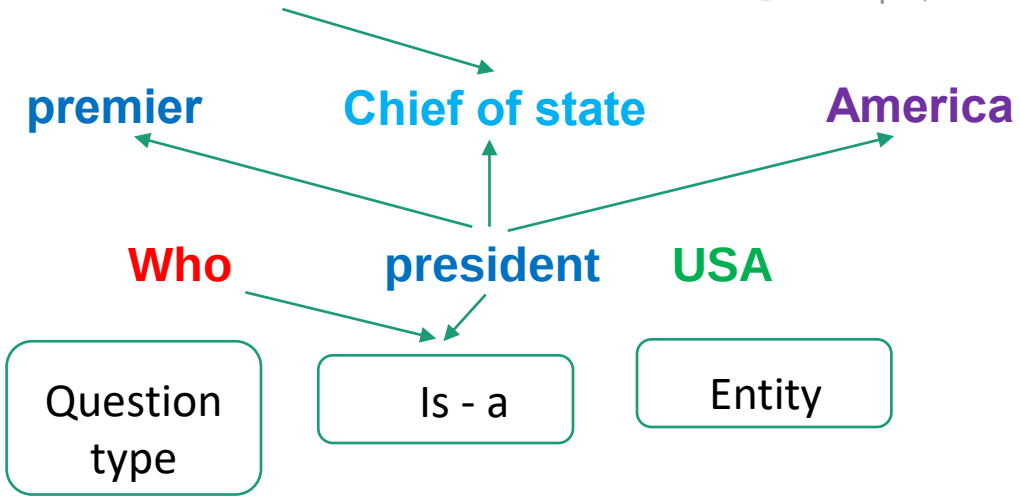
- Loại câu hỏi (intent classification)
  - WHO, WHERE, WHEN
  - WHAT, WHY, HOW
- Thực thể (slot filling, NER)
  - USA
- Quan hệ
  - Is-a; a-part-of; sub-class



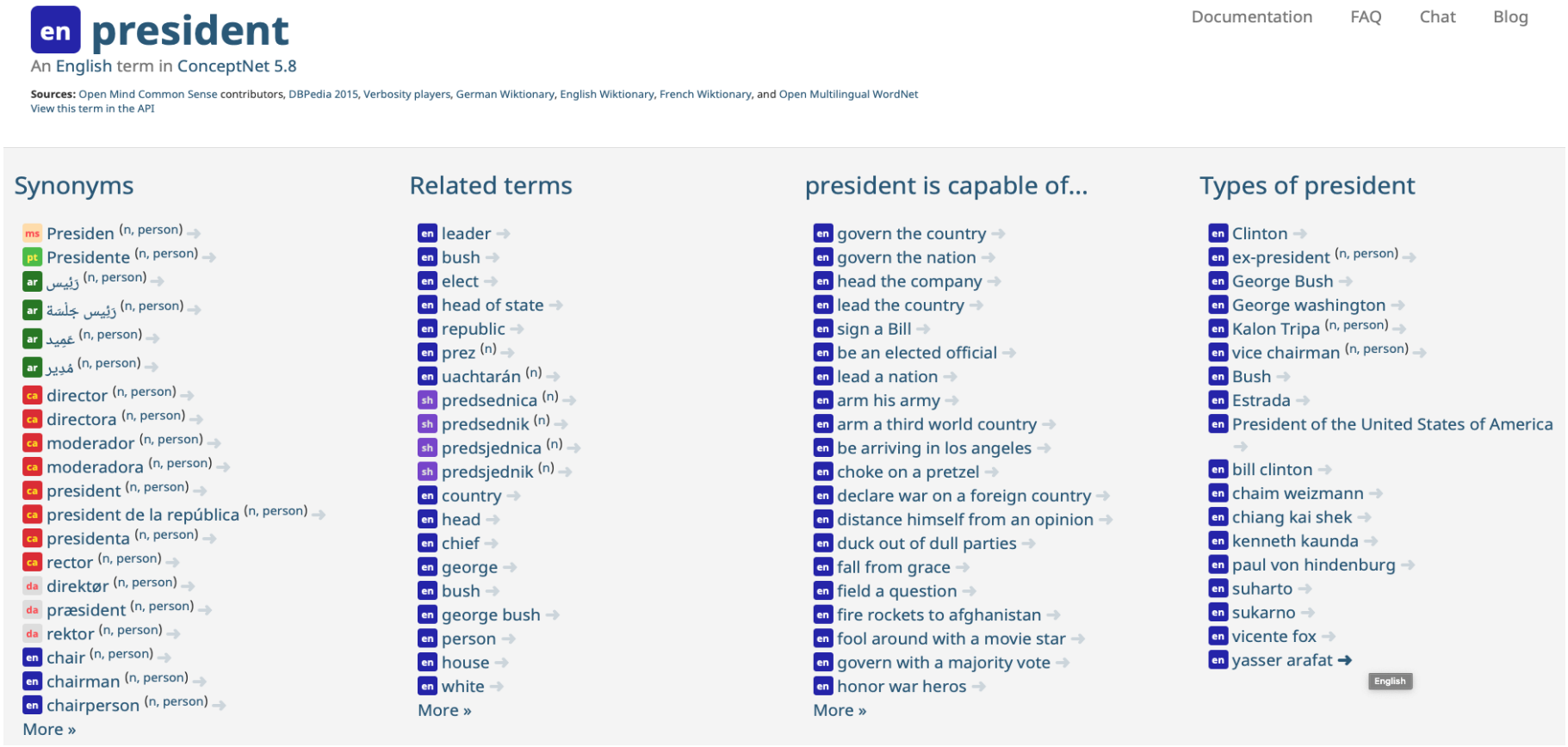
Mở rộng truy vấn

- Mở rộng câu truy vấn → tăng recall
- Kỹ thuật
  - Từ đồng nghĩa
  - WordNet hoặc knowledge graphs



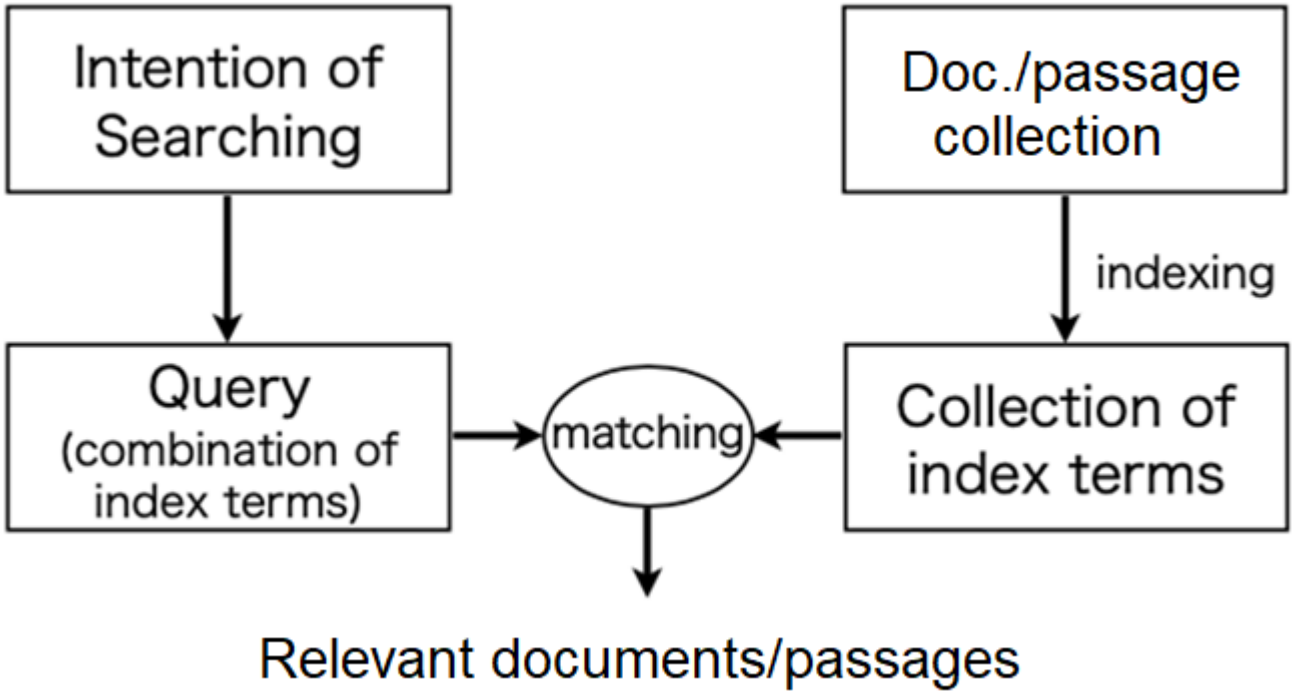


ConceptNet


en president
An English term in ConceptNet 5.8
Documentation FAQ Chat Blog
Sources: Open Mind Common Sense contributors, DBPedia 2015, Verbosity players, German Wiktionary, English Wiktionary, French Wiktionary, and Open Multilingual WordNet
View this term in the API
Synonyms
ms Presiden (n, person)
pt Presidente (n, person)
ar رَئِيس (n, person)
ar رَئِيس جَلْسَة (n, person)
ar عَمِيد (n, person)
ar مُدِير (n, person)
ca director (n, person)
ca directora (n, person)
ca moderador (n, person)
ca moderadora (n, person)
ca president (n, person)
ca president de la república (n, person)
ca presidenta (n, person)
ca rector (n, person)
da direktør (n, person)
da præsident (n, person)
da rektor (n, person)
en chair (n, person)
en chairman (n, person)
en chairperson (n, person)
More »
Related terms
en leader
en bush
en elect
en head of state
en republic
en prez (n)
en uachtarán (n)
sh predsednica (n)
sh predsednik (n)
sh predsjednica (n)
sh predsjednik (n)
en country
en head
en chief
en george
en bush
en george bush
en person
en house
en white
More »
president is capable of...
en govern the country
en govern the nation
en head the company
en lead the country
en sign a Bill
en be an elected official
en lead a nation
en arm his army
en arm a third world country
en be arriving in los angeles
en choke on a pretzel
en declare war on a foreign country
en distance himself from an opinion
en duck out of dull parties
en fall from grace
en field a question
en fire rockets to afghanistan
en fool around with a movie star
en govern with a majority vote
en honor war heros
More »
Types of president
en Clinton
en ex-president (n, person)
en George Bush
en George washington
en Kalon Tripa (n, person)
en vice chairman (n, person)
en Bush
en Estrada
en President of the United States of America
en bill clinton
en chaim weizmann
en chiang kai shek
en kenneth kaunda
en paul von hindenburg
en suharto
en sukarno
en vicente fox
en yasser arafat
English

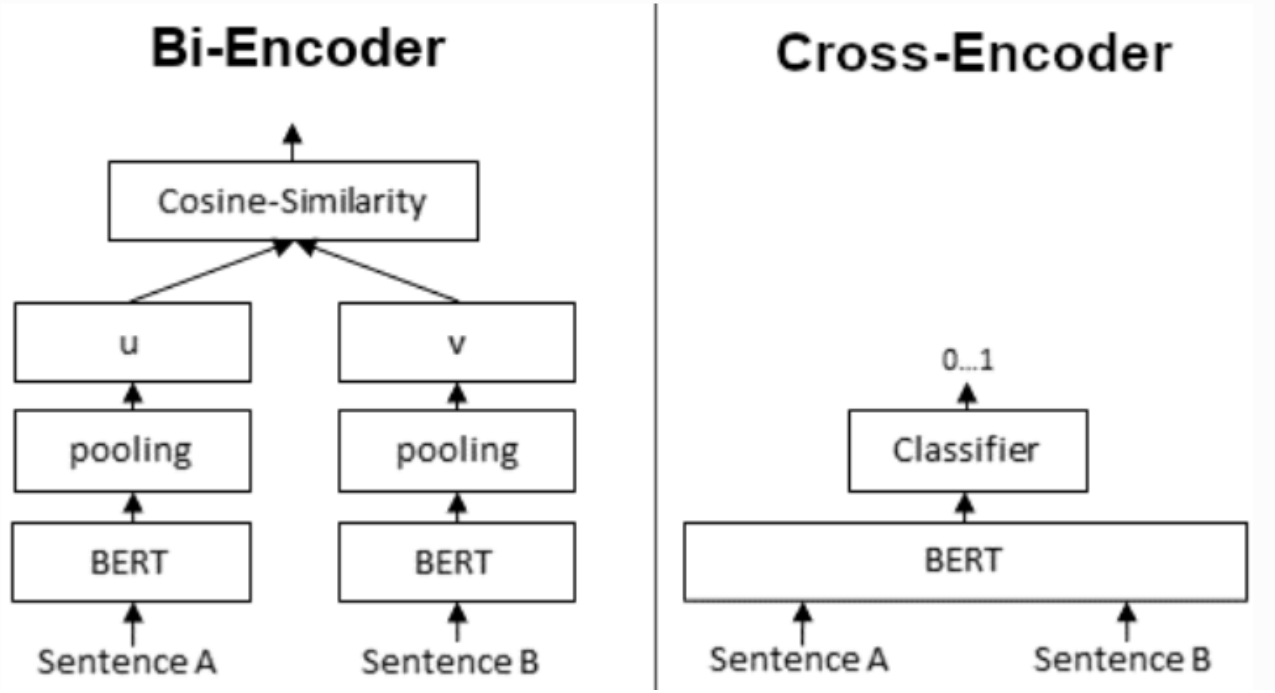
Truy vấn thông tin

- Sau khi có câu hỏi, hệ thống truy vấn thông tin

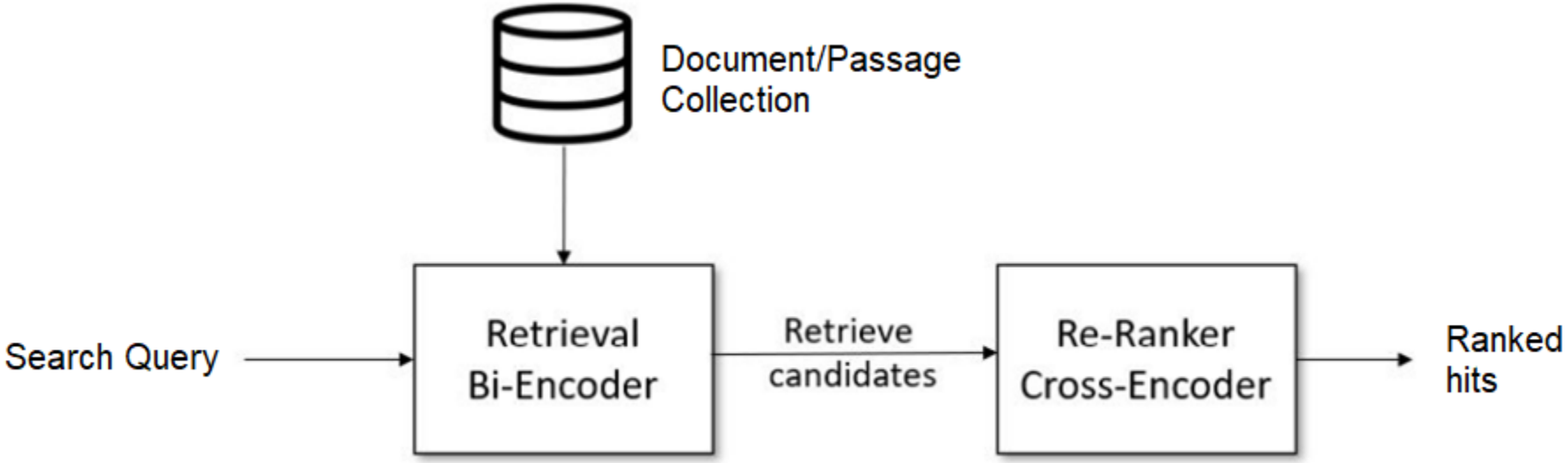


Tìm kiếm theo ngữ nghĩa

- Bi-Encoders: sinh sentence embedding cho từng câu đầu vào
- Cross-Encoder: Bert nhận đầu vào là cả 2 câu và tạo đầu ra là 1 giá trị trong khoảng (0,1), là độ tương đồng của 2 câu đầu vào.



Tìm kiếm theo ngữ nghĩa



- Cross-Encoder có độ chính xác cao hơn Bi-Encoders. Tuy nhiên, vì chạy chậm nên nó không thể làm việc với tập dữ liệu lớn.

➢Sử dụng bi-encoder trả về top N kết quả, sau đó dùng cross-encoder để xếp hạng lại top N kết quả đó (thường N = 100)

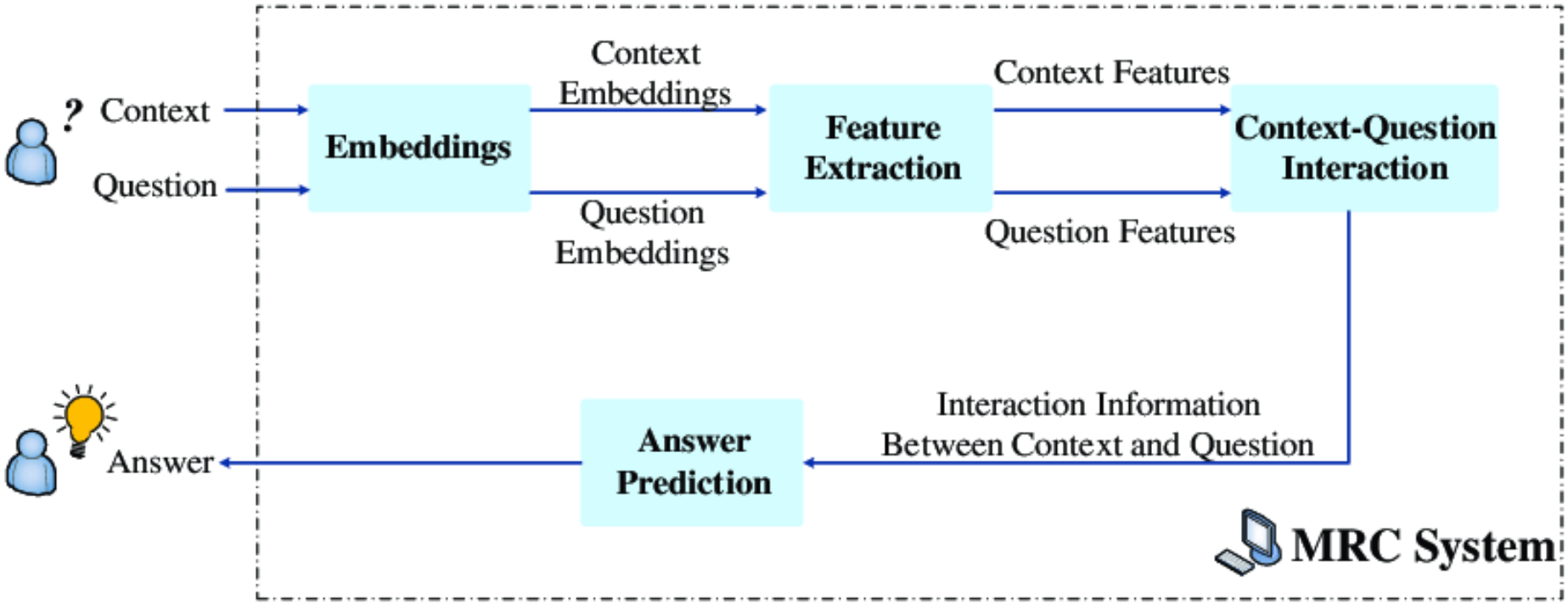
Cross encoder Fine tuning

- Tạo training dataset
  - Không dùng các cặp (Q,P) được dùng để test
  - Mẫu dương: các cặp (Q,P+)
  - Mẫu âm: các cặp (Q,P-) với P- là:
    - Các P không chứa câu trả lời của Q trong dataset
    - Các P thuộc top-k kết quả trả về của search (hard negative samples)

Trích xuất câu trả lời

- Đọc hiểu dựa trên học sâu

(Neural models for reading comprehension)



Machine Reading Comprehension (MRC)

Stanford question answering dataset (SQuAD)

- SQuAD được sử dụng rộng rãi trong việc xây dựng các hệ thống hỏi đáp.
- 100k mẫu đã gán nhãn (đoạn văn bản, câu hỏi, câu trả lời)
- Các đoạn được lấy từ English Wikipedia, 100~150 từ.
- Câu hỏi do cộng đồng tạo ra
- Mỗi câu trả lời là một xâu ngắn trong đoạn văn bản.

Hạn chế: không phải tất cả các câu hỏi đều có câu trả lời kiểu này!

- Mỗi câu hỏi có 3 câu trả lời mẫu, vì thường có nhiều cách trả lời

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under **gravity**. The main forms of precipitation include drizzle, rain, sleet, snow, **graupel** and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals **within a cloud**. Short, intense periods of rain in scattered locations are called "showers".

What causes precipitation to fall?
gravity

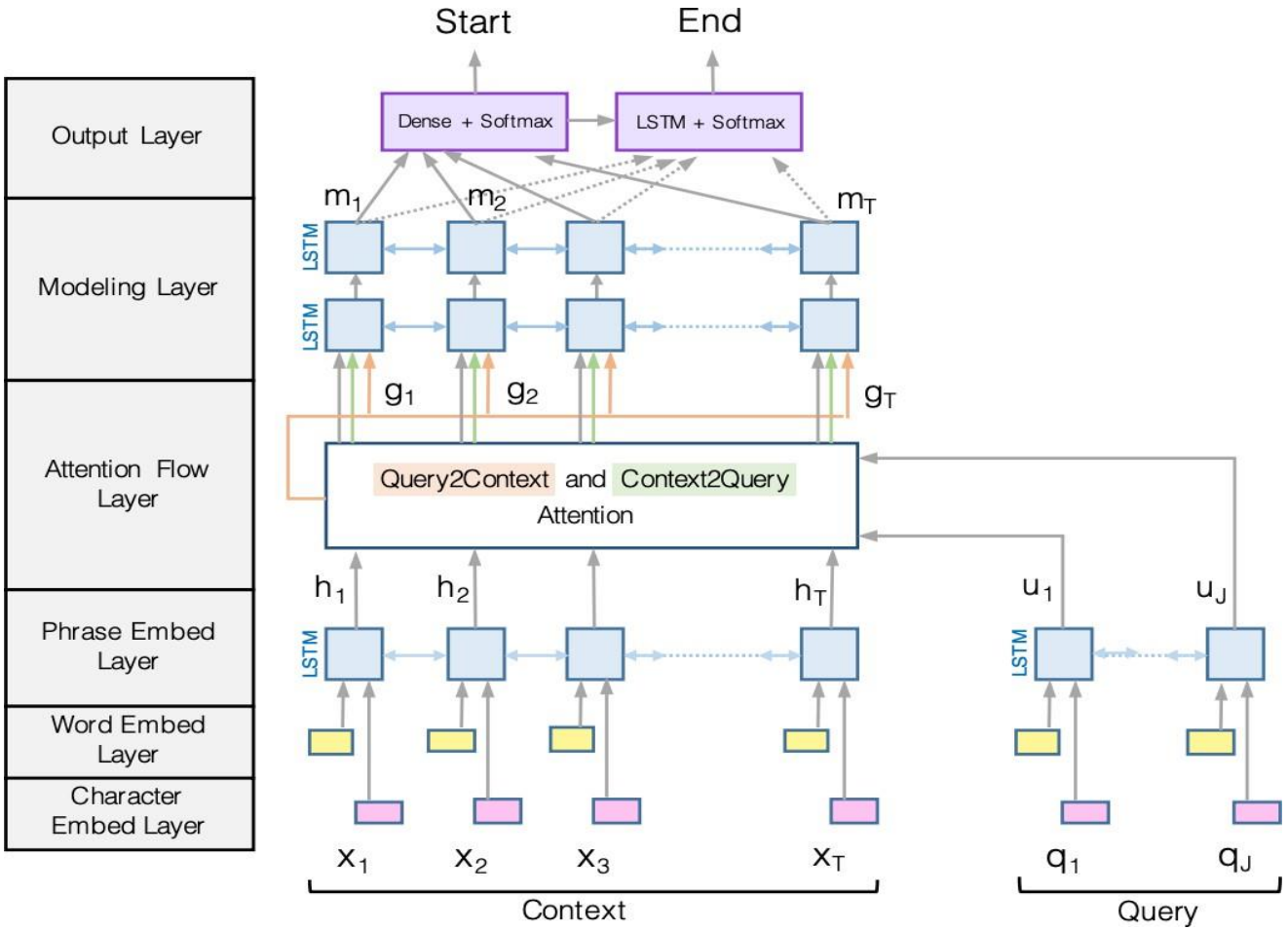
What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?
graupel

Where do water droplets collide with ice crystals to form precipitation?
within a cloud

(Rajpurkar et al., 2016): SQuAD: 100,000+ Questions for Machine Comprehension

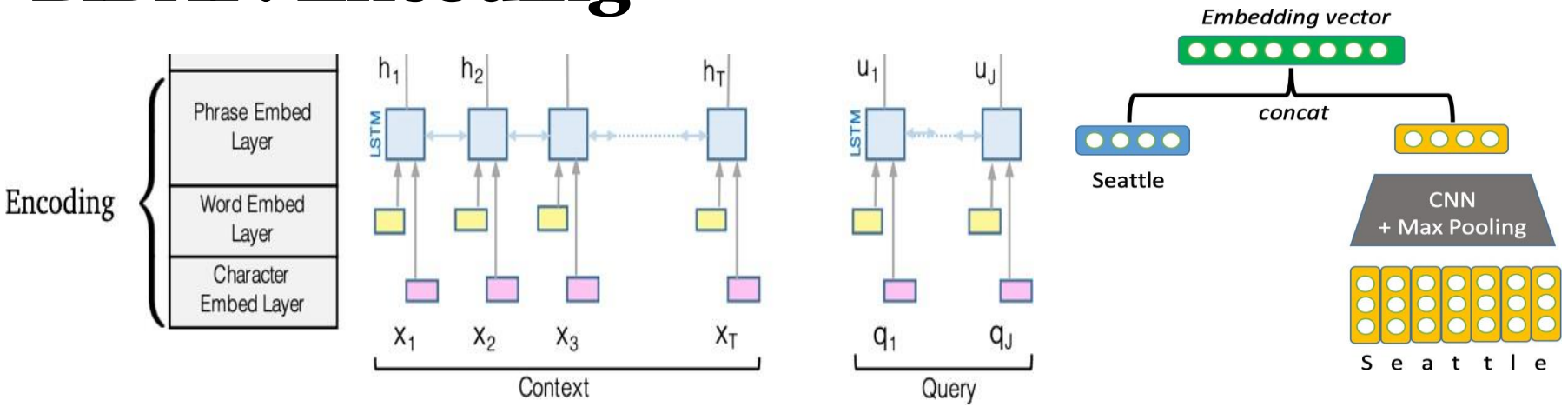
MÔ HÌNH BIDAF

BiDAF: the Bidirectional Attention Flow model



(Seo et al., 2017): Bidirectional Attention Flow for Machine Comprehension

BiDAF: Encoding



- Sử dụng word embedding (GloVe) nối với character embedding (CNNs mức ký tự) cho từng từ trong context và query.

$$e(c_i) = f([\text{GloVe}(c_i); \text{charEmb}(c_i)]) \quad e(q_i) = f([\text{GloVe}(q_i); \text{charEmb}(q_i)])$$

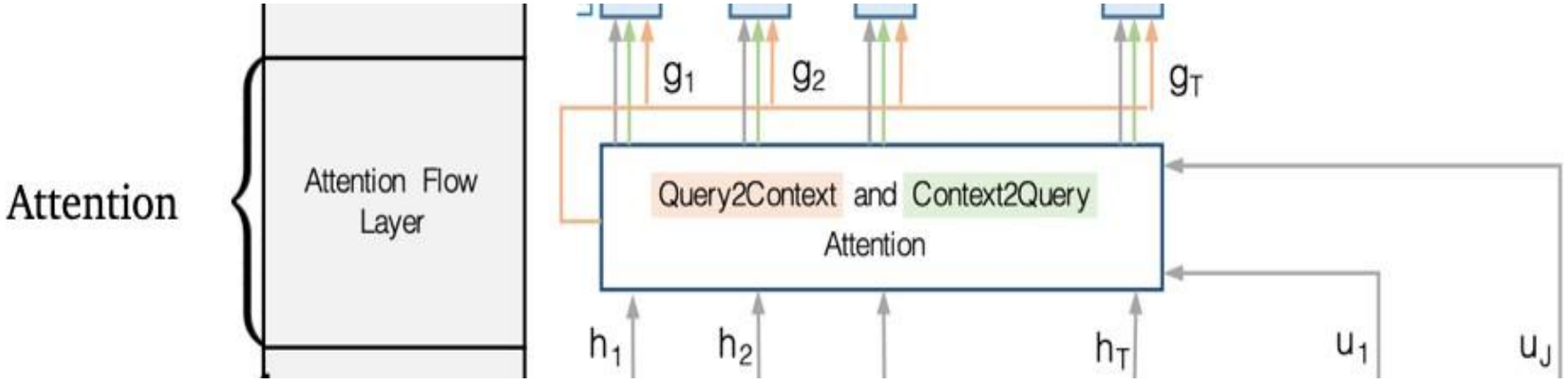
- Sử dụng 2 **bidirectional** LSTMs để sinh ra contextual embeddings cho context và query.

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

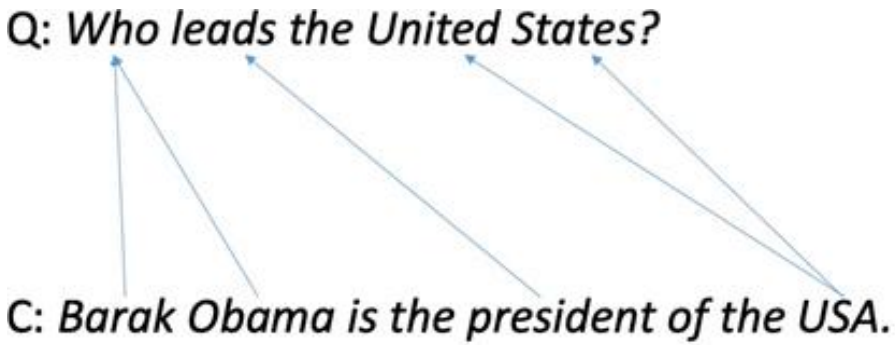
$$\overleftarrow{\mathbf{c}}_i = \text{LSTM}(\overleftarrow{\mathbf{c}}_{i+1}, e(c_i)) \in \mathbb{R}^H \qquad \overleftarrow{\mathbf{q}}_i = \text{LSTM}(\overleftarrow{\mathbf{q}}_{i+1}, e(q_i)) \in \mathbb{R}^H$$

$$\mathbf{c}_i = [\overrightarrow{\mathbf{c}}_i; \overleftarrow{\mathbf{c}}_i] \in \mathbb{R}^{2H} \qquad \mathbf{q}_i = [\overrightarrow{\mathbf{q}}_i; \overleftarrow{\mathbf{q}}_i] \in \mathbb{R}^{2H}$$

BiDAF: Attention

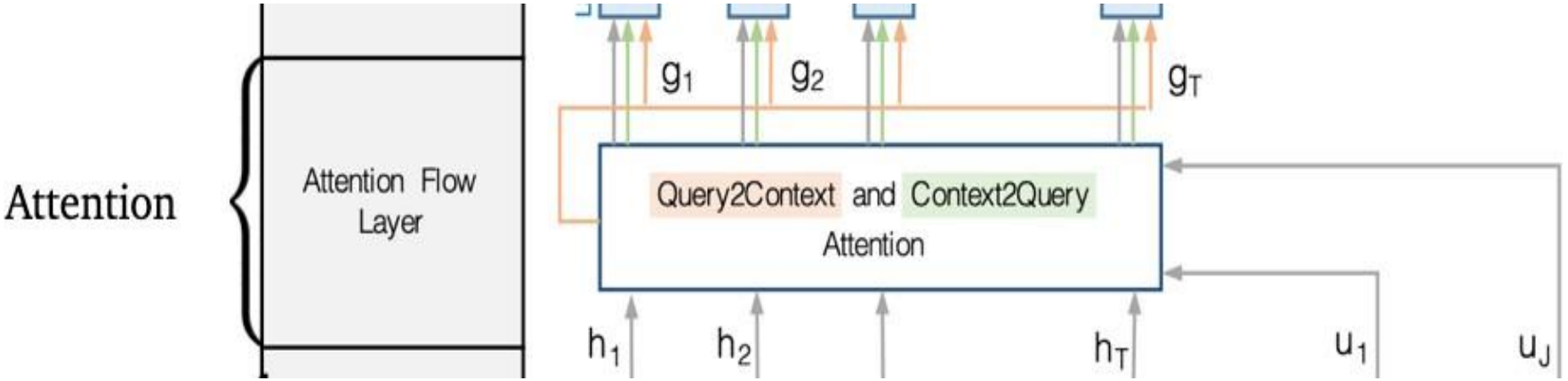


- Context-to-query attention: Với mỗi context word, tìm từ trong câu hỏi liên quan nhất với nó

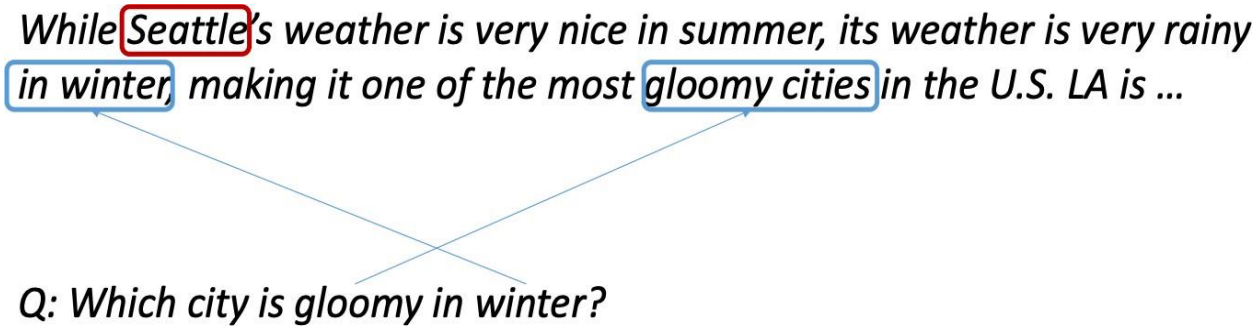


(Slides adapted from Minjoon Seo)

BiDAF: Attention

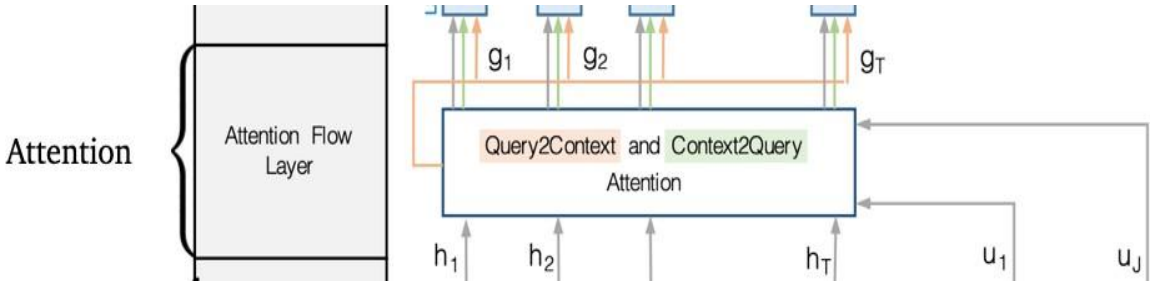


- Query-to-context attention: chọn từ trong context liên quan nhất đến 1 trong các từ trong câu hỏi.



(Slides adapted from Minjoon Seo)

BiDAF: Attention



The final output is

$$\mathbf{g}_i = [\mathbf{c}_i; \mathbf{a}_i; \mathbf{c}_i \odot \mathbf{a}_i; \mathbf{c}_i \odot \mathbf{b}] \in \mathbb{R}^{8H}$$

- Tính similarity score cho mỗi cặp (ci, qj):

$$S_{i,j} = \mathbf{w}_{\text{sim}}^{\intercal}[\mathbf{c}_i; \mathbf{q}_j; \mathbf{c}_i \odot \mathbf{q}_j] \in \mathbb{R} \qquad \mathbf{w}_{\text{sim}} \in \mathbb{R}^{6H}$$

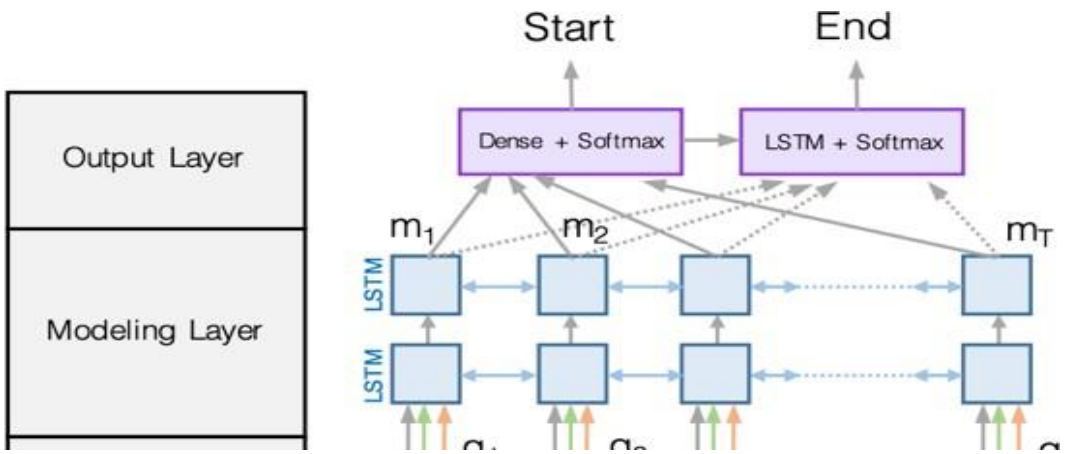
- Context-to-query attention (từ nào trong câu hỏi liên quan nhất đến context)

$$\alpha_{i,j} = \text{softmax}_j(S_{i,j}) \in \mathbb{R} \qquad \mathbf{a}_i = \sum_{j=1}^{M} \alpha_{i,j}\mathbf{q}_j \in \mathbb{R}^{2H}$$

- Query-to-context attention (từ nào trong context liên quan nhất đến từ trong câu hỏi)

$$\beta_i = \text{softmax}_i(\max_{j=1}^{M}(S_{i,j})) \in \mathbb{R}^N \qquad \mathbf{b} = \sum_{i=1}^{N} \beta_i \mathbf{c}_i \in \mathbb{R}^{2H}$$

BiDAF: Modeling and output layers



The final training loss is

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

Modeling layer: cho gi đi qua 2 tầng **bi-directional** LSTMs nữa.

- Attention layer mô hình hóa tương tác giữa query và context
- Modeling layer mô hình hóa tương tác giữa các từ nội dung

$$\mathbf{m}_i = \text{BiLSTM}(\mathbf{g}_i) \in \mathbb{R}^{2H}$$

- **Output layer:** 2 bộ phân loại dự đoán vị trí start và end :

$$p_{\text{start}} = \text{softmax}(\mathbf{w}_{\text{start}}^\intercal [\mathbf{g}_i; \mathbf{m}_i]) \qquad p_{\text{end}} = \text{softmax}(\mathbf{w}_{\text{end}}^\intercal [\mathbf{g}_i; \mathbf{m}_i'])$$

$$\mathbf{m}_i' = \text{BiLSTM}(\mathbf{m}_i) \in \mathbb{R}^{2H} \quad \mathbf{w}_{\text{start}}, \mathbf{w}_{\text{end}} \in \mathbb{R}^{10H}$$

BiDAF: Performance on SQuAD

F1 = **77.3%** trên SQuAD v1.1.

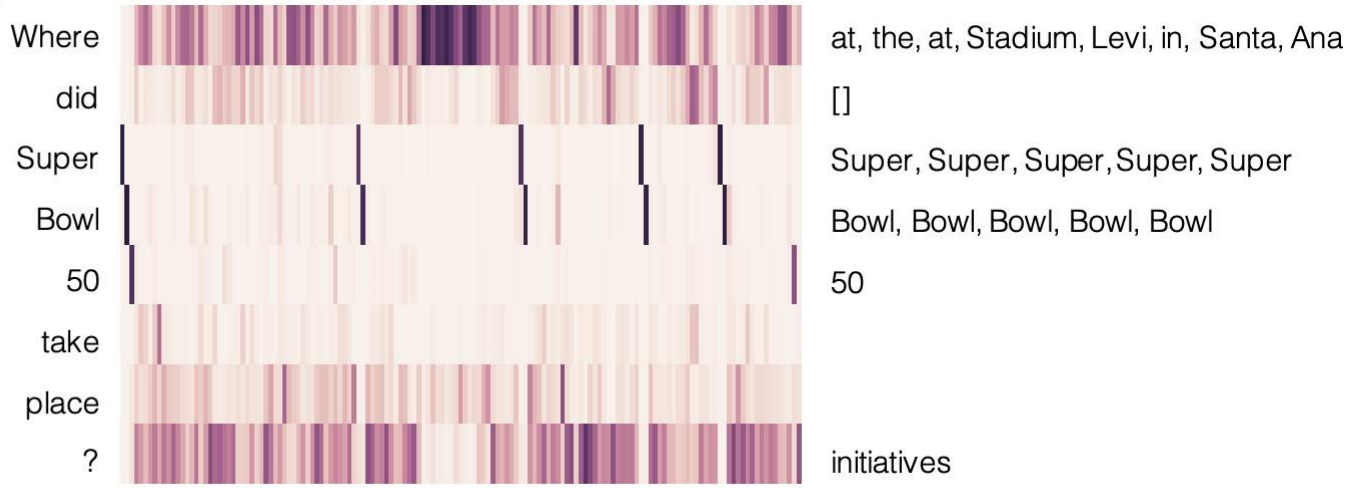
- Không có context-to-query attention — 67.7 F1
- Không có query-to-context attention — 73.7 F1
- Không có character embeddings — 75.4 F1

	Published[12]	LeaderBoard[13]
Single Model	EM / F1	EM / F1
LR Baseline (Rajpurkar et al., 2016)	40.4 / 51.0	40.4 / 51.0
Dynamic Chunk Reader (Yu et al., 2016)	62.5 / 71.0	62.5 / 71.0
Match-LSTM with Ans-Ptr (Wang & Jiang, 2016)	64.7 / 73.7	64.7 / 73.7
Multi-Perspective Matching (Wang et al., 2016)	65.5 / 75.1	70.4 / 78.8
Dynamic Coattention Networks (Xiong et al., 2016)	66.2 / 75.9	66.2 / 75.9
FastQA (Weissenborn et al., 2017)	68.4 / 77.1	68.4 / 77.1
BiDAF (Seo et al., 2016)	68.0 / 77.3	68.0 / 77.3
SEDT (Liu et al., 2017a)	68.1 / 77.5	68.5 / 78.0
RaSoR (Lee et al., 2016)	70.8 / 78.7	69.6 / 77.7
FastQAExt (Weissenborn et al., 2017)	70.8 / 78.9	70.8 / 78.9
ReasoNet (Shen et al., 2017b)	69.1 / 78.9	70.6 / 79.4
Document Reader (Chen et al., 2017)	70.0 / 79.0	70.7 / 79.4
Ruminating Reader (Gong & Bowman, 2017)	70.6 / 79.5	70.6 / 79.5
jNet (Zhang et al., 2017)	70.6 / 79.8	70.6 / 79.8
Conductor-net	N/A	72.6 / 81.4
Interactive AoA Reader (Cui et al., 2017)	N/A	73.6 / 81.9
Reg-RaSoR	N/A	75.8 / 83.3
DCN+	N/A	74.9 / 82.8
AIR-FusionNet	N/A	76.0 / 83.9
R-Net (Wang et al., 2017)	72.3 / 80.7	76.5 /84.3
BiDAF + Self Attention + ELMo	N/A	**77.9/ 85.3**
Reinforced Mnemonic Reader (Hu et al., 2017)	73.2 / 81.8	73.2 / 81.8

(Seo et al., 2017): Bidirectional Attention Flow for Machine Comprehension

Attention visualization

Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season . The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24–10 to earn their third Super Bowl title . The game was played on February 7 , 2016 , at Levi 's Stadium in the San Francisco Bay Area at Santa Clara , California . As this was the 50th Super Bowl , the league emphasized the " golden anniversary " with various gold-themed initiatives , as well as temporarily suspending the tradition of naming each Super Bowl game with Roman numerals (under which the game would have been known as " Super Bowl L ") , so that the logo could prominently feature the Arabic numerals 50 .



MÔ HÌNH BERT

Transformer

- Kiến trúc cho phép biến đổi một chuỗi thành một chuỗi khác với bộ mã hoá và giải mã
- Multi-head attention
- Feed forward layers
- Positional embeddings

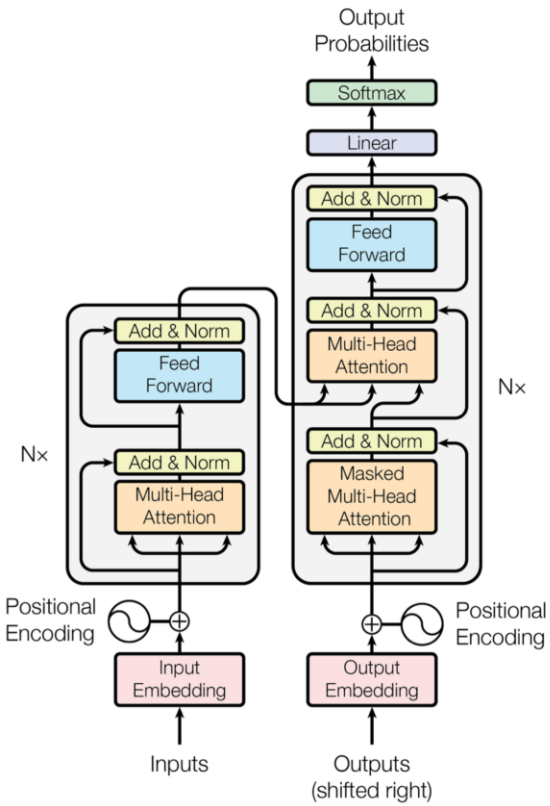
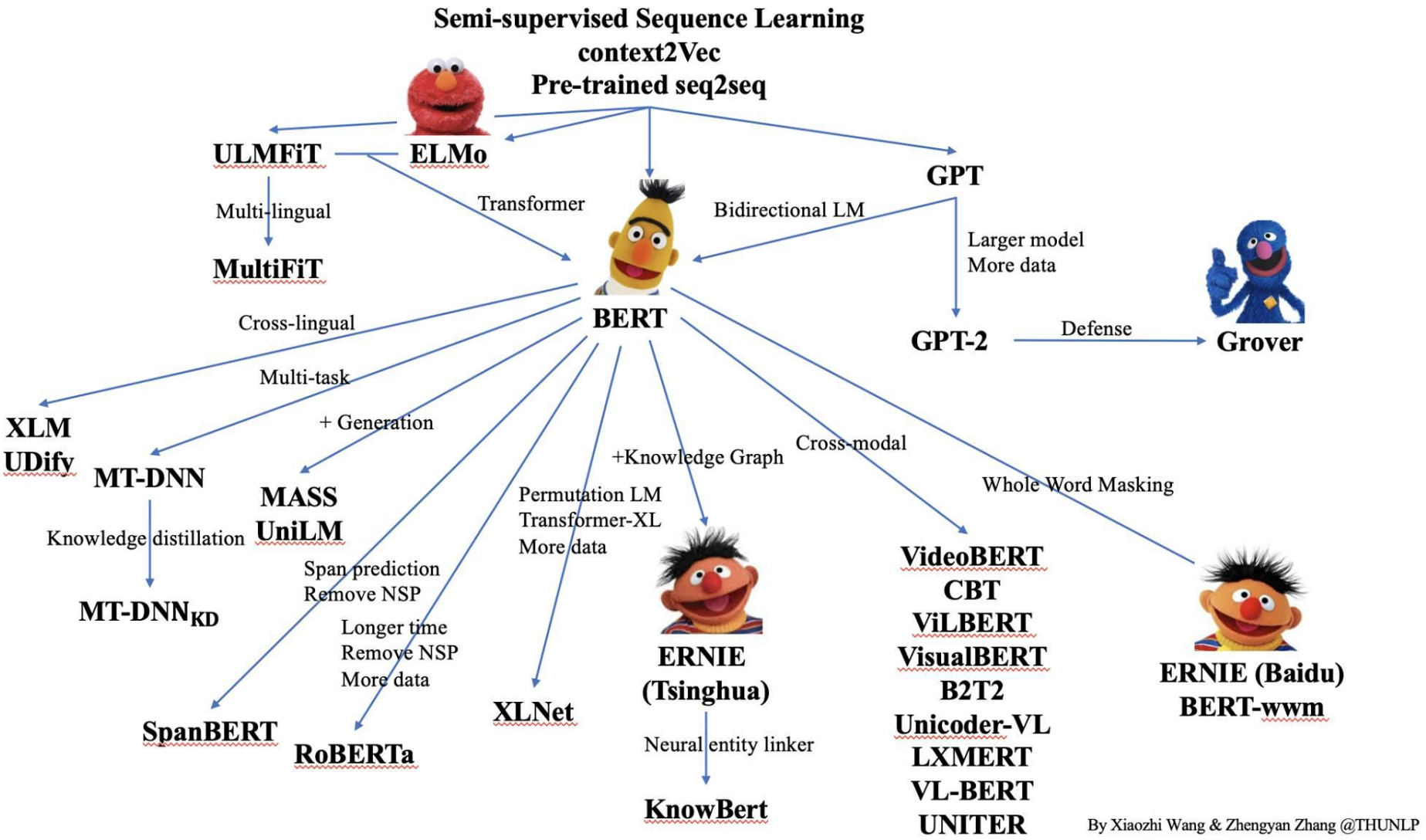


Figure 1: The Transformer - model architecture.

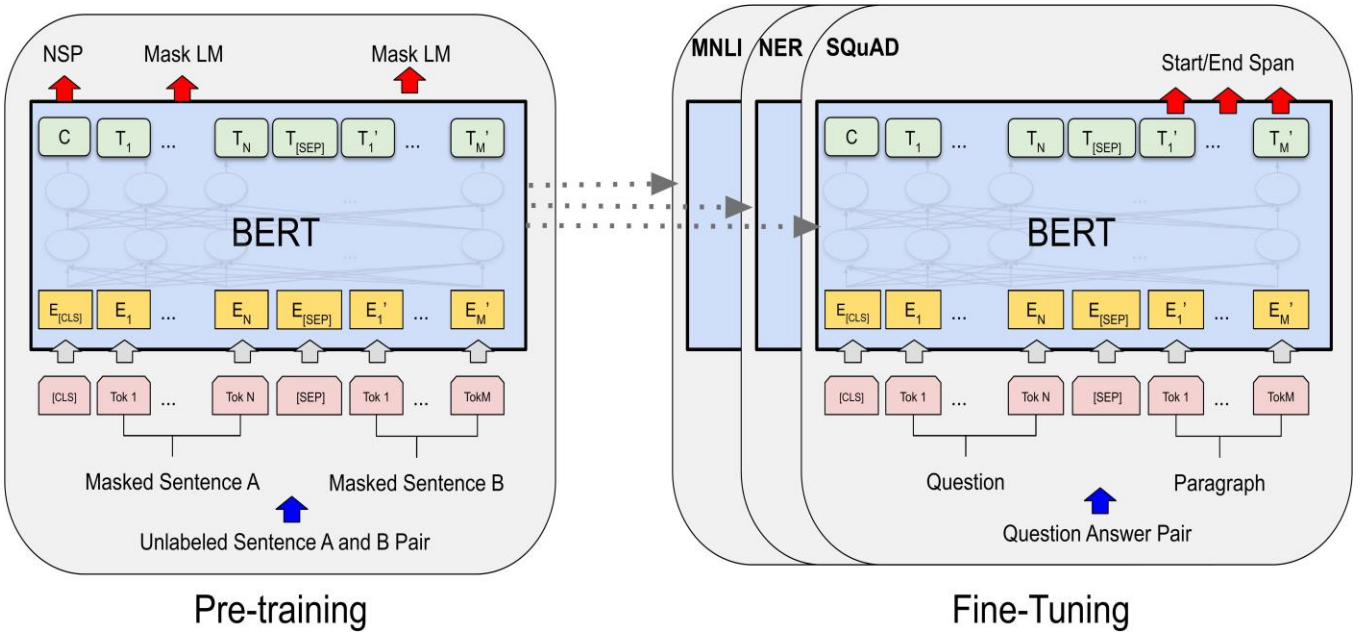
https://medium.com/inside-machine-learning/what-is-a-transformer-d07dd1fbec04

BERT-family

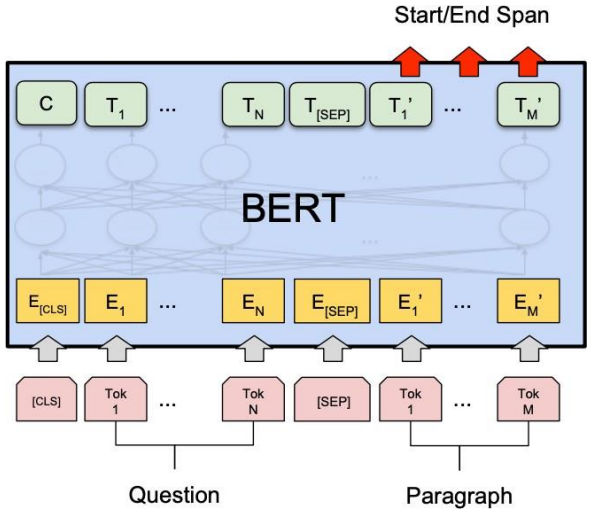

Semi-supervised Sequence Learning
context2Vec
Pre-trained seq2seq
ULMFiT
ELMo
Multi-lingual
MultiFiT
Transformer
BERT
Bidirectional LM
GPT
Larger model
More data
GPT-2
Defense
Grover
Cross-lingual
XLM
UDify
Multi-task
MT-DNN
Knowledge distillation
MT-DNN$_{KD}$
+ Generation
MASS
UniLM
Span prediction
Remove NSP
SpanBERT
Longer time
Remove NSP
More data
RoBERTa
Permutation LM
Transformer-XL
More data
XLNet
+Knowledge Graph
ERNIE
(Tsinghua)
Neural entity linker
KnowBert
Cross-modal
VideoBERT
CBT
ViLBERT
VisualBERT
B2T2
Unicoder-VL
LXMERT
VL-BERT
UNITER
Whole Word Masking
ERNIE (Baidu)
BERT-wwm
By Xiaozhi Wang & Zhengyan Zhang @THUNLP

BERT for reading comprehension

- BERT: mô hình mã hóa Transformer 2 chiều được huấn luyện trước trên tập dữ liệu lớn (Wikipedia + BooksCorpus)
- BERT được huấn luyện trên 2 task:
  - Masked language model (MLM)
  - Next sentence prediction (NSP)
- $BERT_{base}$ có 12 layers , 110M parameters, $BERT_{large}$ có 24 layers , 330M parameters



BERT for reading comprehension



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

$$p_{\text{start}}(i) = \text{softmax}_i(\mathbf{w}_{\text{start}}^\intercal \mathbf{H})$$

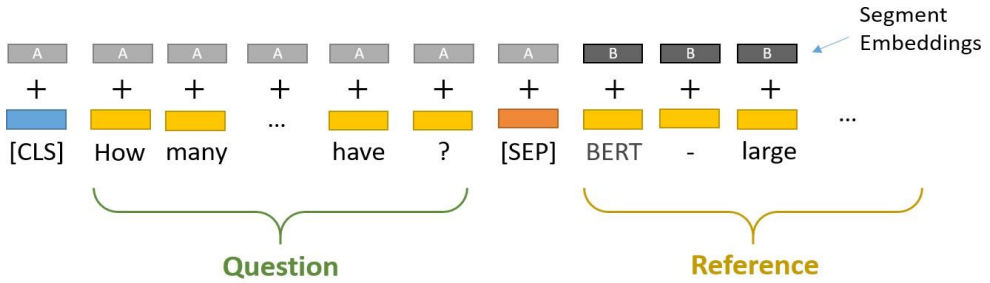
$$p_{\text{end}}(i) = \text{softmax}_i(\mathbf{w}_{\text{end}}^\intercal \mathbf{H})$$

where $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_N]$ are the hidden vectors of the paragraph, returned by BERT

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.
```

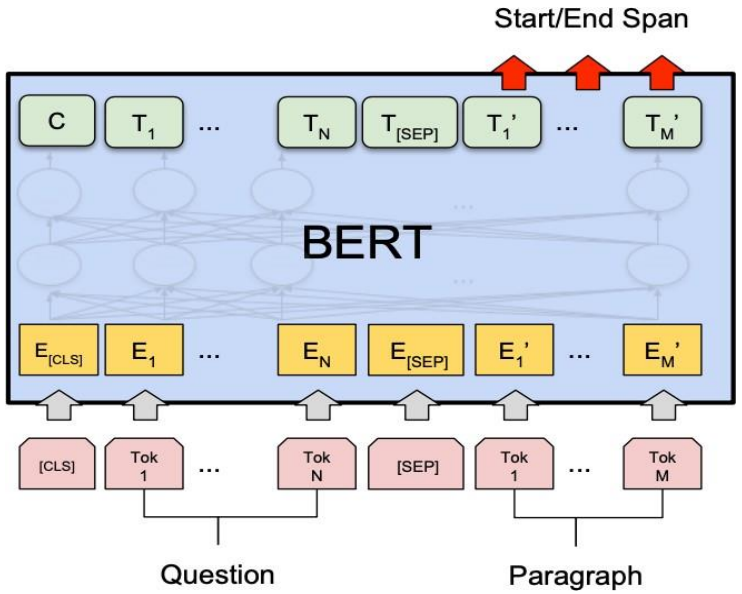
Image credit: https://mccormickml.com/

BERT for reading comprehension

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

	F1	EM
Human performance	91.2	82.3
BiDAF	77.3	67.7
BERT-base	88.5	80.8
BERT-large	90.9	84.1
XLNet	94.5	89.0
RoBERTa	94.6	88.9
ALBERT	94.8	89.3

(dev set, except for human performance)

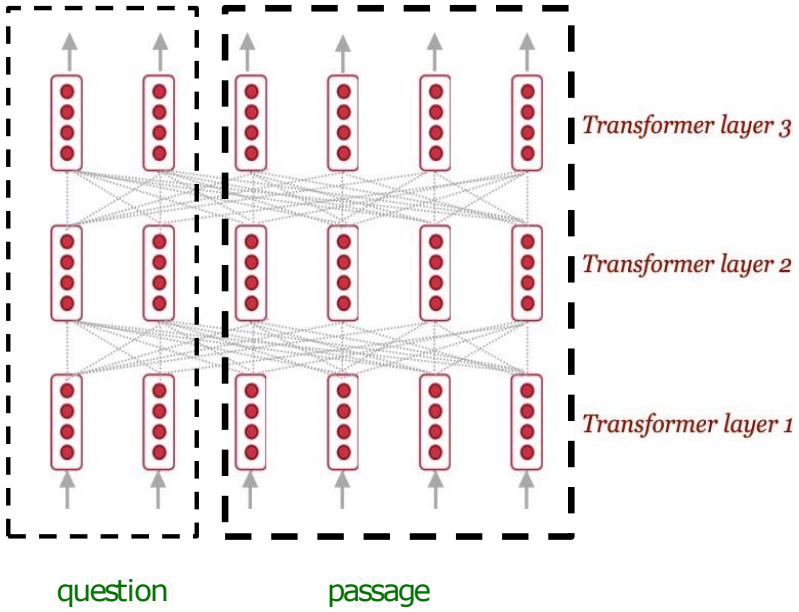


So sánh BiDAF và BERT

- BERT có nhiều tham số hơn (110M or 330M) so với BiDAF (~2.5M).
- BiDAF được tạo bởi vài bidirectional LSTMs; BERT được tạo bởi các Transformers
- BERT được huấn luyện trước; BiDAF chỉ sử dụng bộ word embedding GloVe (tất cả các tham số còn lại phải học từ tập dữ liệu đã gán nhãn).
- Pre-training rõ ràng đã cải thiện được độ chính xác, nhưng tạo ra nó rất đắt đỏ

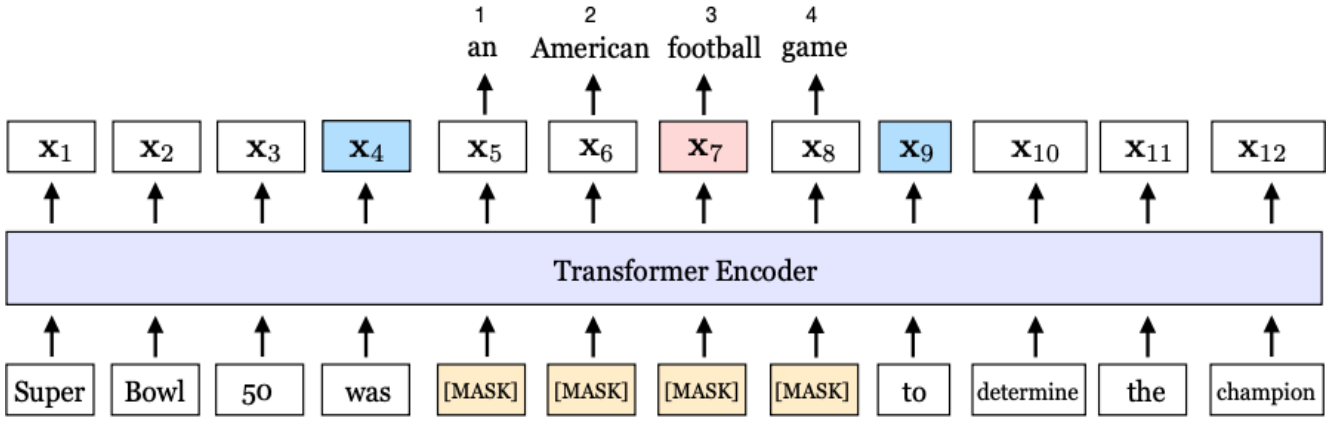
So sánh BiDAF và BERT

- BiDAF mô hình hóa tương tác giữa question và passage.
- BERT sử dụng self-attention giữa kết nối của question và passage = attention(P, P) + attention(P, Q) + attention(Q, P) + attention(Q, Q)
- (Clark and Gardner, 2018) thêm 1 tầng self-attention cho passage attention(P, P) vào BiDAF, cũng cải thiện độ chính xác



Cải tiến hàm mục tiêu cho pre-training

$$\mathcal{L}(\text{football}) = \mathcal{L}_{\text{MLM}}(\text{football}) + \mathcal{L}_{\text{SBO}}(\text{football})$$
$$= -\log P(\text{football} \mid \mathbf{x}_7) - \log P(\text{football} \mid \mathbf{x}_4, \mathbf{x}_9, \mathbf{p}_3)$$



SBO = span boundary objective

Ý tưởng:

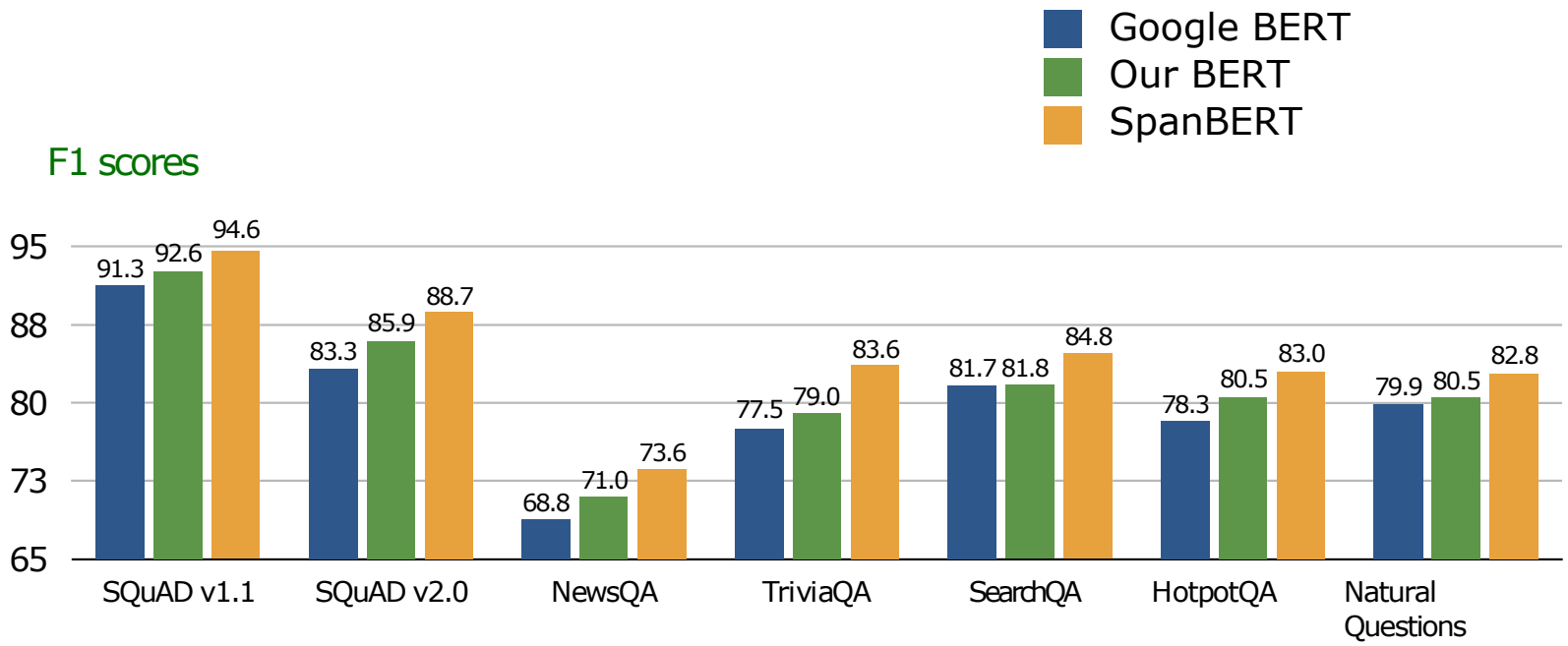
1)Che (mask) các đoạn liên tục các từ thay vì 15% từ ngẫu nhiên

2)Sử dụng 2 điểm kết thúc của span để dự đoán tất cả các từ bị che ở giữa = nén thông tin của spanj vào 2 điểm kết thúc đó

$$\mathbf{y}_i = f(\mathbf{x}_{s-1}, \mathbf{x}_{e+1}, \mathbf{p}_{i-s+1})$$

(Joshi & Chen et al., 2020): SpanBERT: Improving Pre-training by Representing and Predicting Spans

SpanBERT performance



Is reading comprehension solved?

- Các mô hình học sâu đã tốt hơn con người trên bộ SQUAD. Tuy nhiên, khi làm việc với dữ liệu ngoài miền huấn luyện, kết quả vẫn kém

Article: Super Bowl 50
Paragraph: "*Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV.*"
Question: "*What is the name of the quarterback who was 38 in Super Bowl XXXIII?*"
Original Prediction: John Elway
Prediction under adversary: Jeff Dean

	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

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

Is reading comprehension solved?

Fine-tuned on	Evaluated on: SQuAD	TriviaQA	NQ	QuAC	NewsQA
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**

(Sen and Saffari, 2020): What do Models Learn from Question Answering Datasets?

Is reading comprehension solved?

BERT-large model trained on SQuAD

	Test *TYPE* and Description	Failure Rate (🤖)	Example Test cases (with expected behavior and 🤖 prediction)
Vocab	***MFT:*** comparisons	20.0	**C:** Victoria is younger than Dylan. **Q:** Who is less young? **A:** Dylan 🤖: Victoria
	MFT: intensifiers to superlative: most/least	91.3	**C:** Anna is worried about the project. Matthew is extremely worried about the project. **Q:** Who is least worried about the project? **A:** Anna 🤖: Matthew
Taxonomy	***MFT:*** match properties to categories	82.4	**C:** There is a tiny purple box in the room. **Q:** What size is the box? **A:** tiny 🤖: purple
	MFT: nationality vs job	49.4	**C:** Stephanie is an Indian accountant. **Q:** What is Stephanie's job? **A:** accountant 🤖: Indian accountant
	MFT: animal vs vehicles	26.2	**C:** Jonathan bought a truck. Isabella bought a hamster. **Q:** Who bought an animal? **A:** Isabella 🤖: Jonathan
	MFT: comparison to antonym	67.3	**C:** Jacob is shorter than Kimberly. **Q:** Who is taller? **A:** Kimberly 🤖: Jacob
	MFT: more/less in context, more/less antonym in question	100.0	**C:** Jeremy is more optimistic than Taylor. **Q:** Who is more pessimistic? **A:** Taylor 🤖: Jeremy
Robust.	***INV:*** Swap adjacent characters in **Q** (typo)	11.6	**C:** ...Newcomen designs had a duty of about 7 million, but most were closer to 5 million.... **Q:** What was the ideal duty → udty of a Newcomen engine? **A:** INV 🤖: 7 million → 5 million
	INV: add irrelevant sentence to **C**	9.8	(no example)

(Ribeiro et al., 2020): Beyond Accuracy: Behavioral Testing of NLP Models with CheckList

Is reading comprehension solved?

BERT-large model trained on SQuAD

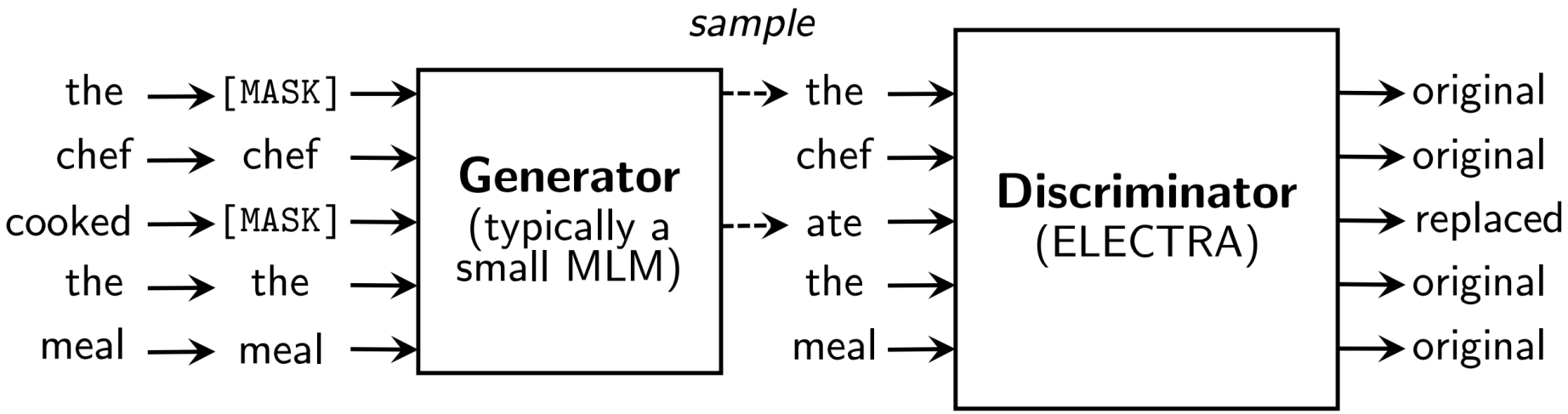
Temporal	***MFT:*** change in one person only	41.5	**C:** Both Luke and Abigail were writers, but there was a change in Abigail, who is now a model. **Q:** Who is a model? **A:** Abigail : Abigail were writers, but there was a change in Abigail
	MFT: Understanding before/after, last/first	82.9	**C:** Logan became a farmer before Danielle did. **Q:** Who became a farmer last? **A:** Danielle : Logan
Neg.	***MFT:*** Context has negation	67.5	**C:** Aaron is not a writer. Rebecca is. **Q:** Who is a writer? **A:** Rebecca : Aaron
	MFT: **Q** has negation, **C** does not	100.0	**C:** Aaron is an editor. Mark is an actor. **Q:** Who is not an actor? **A:** Aaron : Mark
Coref.	***MFT:*** Simple coreference, he/she.	100.0	**C:** Melissa and Antonio are friends. He is a journalist, and she is an adviser. **Q:** Who is a journalist? **A:** Antonio : Melissa
	MFT: Simple coreference, his/her.	100.0	**C:** Victoria and Alex are friends. Her mom is an agent **Q:** Whose mom is an agent? **A:** Victoria : Alex
	MFT: former/latter	100.0	**C:** Kimberly and Jennifer are friends. The former is a teacher **Q:** Who is a teacher? **A:** Kimberly : Jennifer
SRL	***MFT:*** subject/object distinction	60.8	**C:** Richard bothers Elizabeth. **Q:** Who is bothered? **A:** Elizabeth : Richard
	MFT: subj/obj distinction with 3 agents	95.7	**C:** Jose hates Lisa. Kevin is hated by Lisa. **Q:** Who hates Kevin? **A:** Lisa : Jose

(Ribeiro et al., 2020): Beyond Accuracy: Behavioral Testing of NLP Models with CheckList

MÔ HÌNH ELECTRA
(Google AI)

ELECTRA

- Quá trình huấn luyện khác với BERT, tốn ít tài nguyên hơn
- Quá trình huấn luyện là sự dự đoán các từ bị thay thế



https://openreview.net/pdf?id=r1xMH1BtvB

Qúa trình huấn luyện

- Cho một chuỗi, lựa chọn ngẫu nhiên các tokens để dấu [MASKED]
- Bộ sinh (generator) dự đoán các từ gốc cho tất cả các từ bị dấu
- Đầu vào cho bộ tách (discriminator) được tạo bằng cách thay thế các từ [MASKED] với từ dự đoán của bộ sinh
- Với mỗi từ, bộ tách dự đoán từ đó có phải là từ gốc hay từ bị thay thế

Hàm chi phí

- Bộ sinh (generator)
  - Chỉ tính cho các từ bị dấu [MASKED]
- Bộ tách (discriminator)
  - Tính toán cho tất cả các từ trong từ điển
- Bộ tách (discriminator) được dùng cho quá trình “fine-tuning”
- Bộ sinh không được sử dụng

BERT và ELECTRA

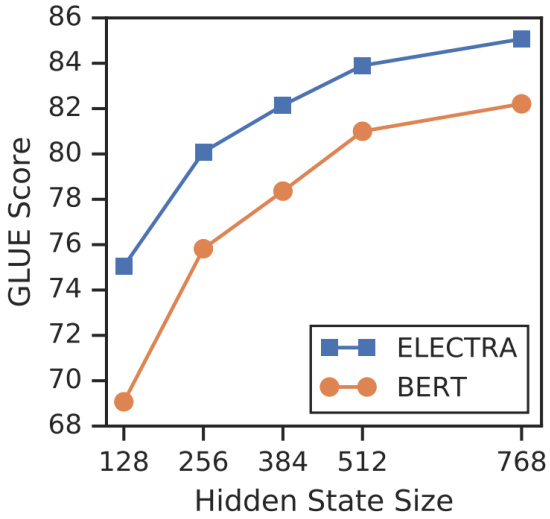
ELECTRA

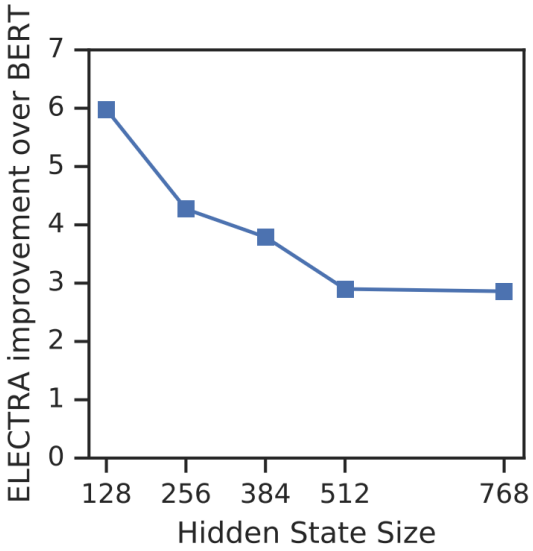
- Hàm chi phí được tính dựa trên toàn bộ từ (discriminator)
- Giải quyết vấn đề out-of-vocab

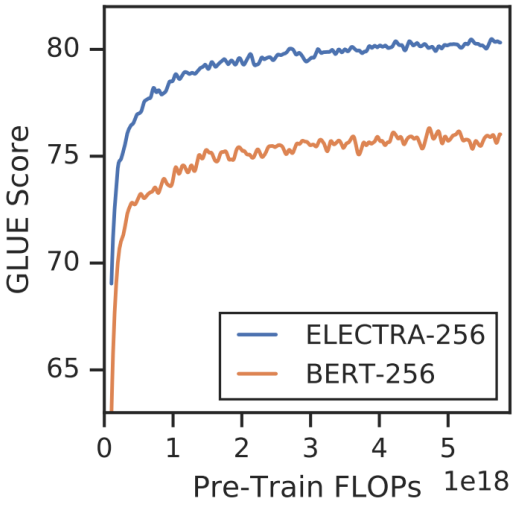
BERT

- Hàm chi phí được tính dựa trên các ký tự bị dấu
- Gặp vấn đề out-of-vocab

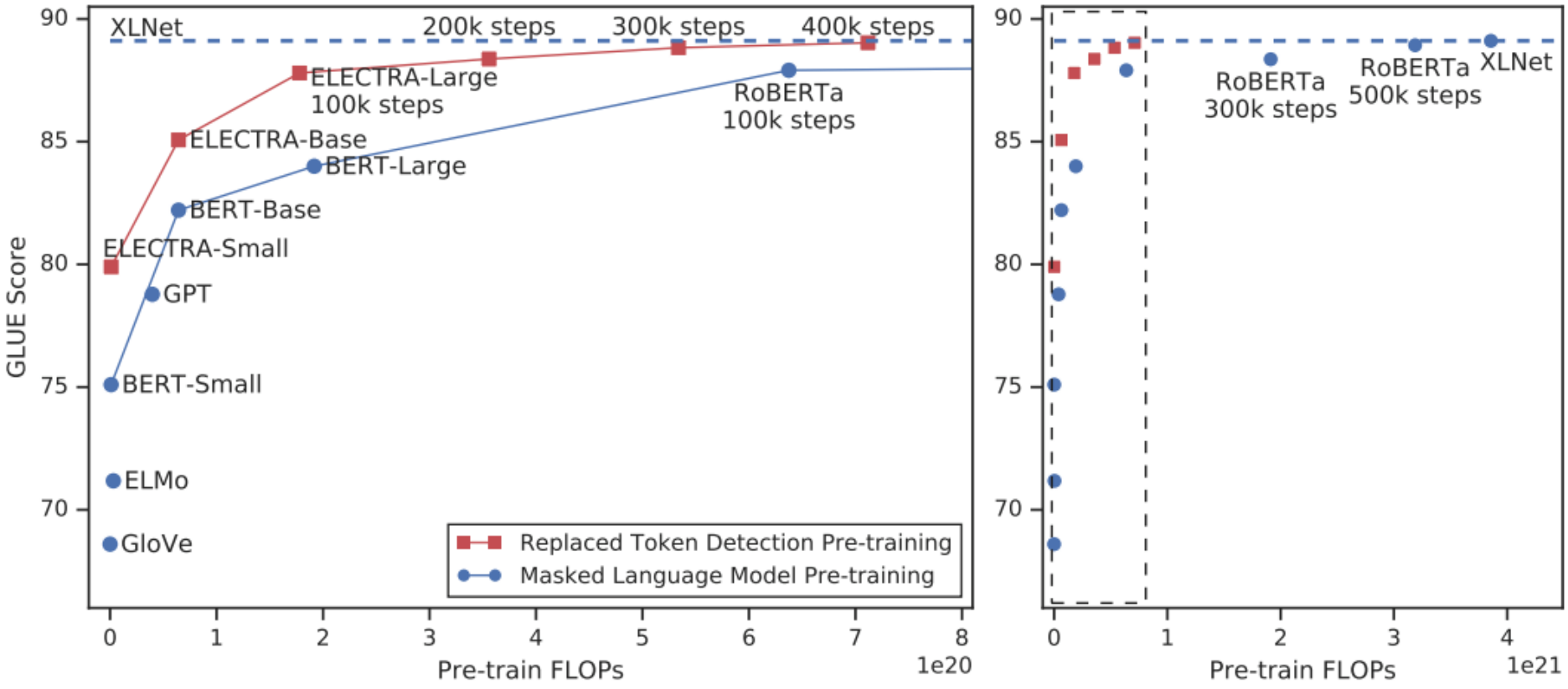
BERT và ELECTRA







Dự đoán từ bị thay thế



Kết quả

Model	Train / Infer FLOPs	Speedup	Params	Train Time + Hardware	GLUE
ELMo	3.3e18 / 2.6e10	19x / 1.2x	96M	14d on 3 GTX 1080 GPUs	71.2
GPT	4.0e19 / 3.0e10	1.6x / 0.97x	117M	25d on 8 P6000 GPUs	78.8
BERT-Small	1.4e18 / 3.7e9	45x / 8x	14M	4d on 1 V100 GPU	75.1
BERT-Base	6.4e19 / 2.9e10	1x / 1x	110M	4d on 16 TPUv3s	82.2
ELECTRA-Small	1.4e18 / 3.7e9	45x / 8x	14M	4d on 1 V100 GPU	79.9
50% trained	7.1e17 / 3.7e9	90x / 8x	14M	2d on 1 V100 GPU	79.0
25% trained	3.6e17 / 3.7e9	181x / 8x	14M	1d on 1 V100 GPU	77.7
12.5% trained	1.8e17 / 3.7e9	361x / 8x	14M	12h on 1 V100 GPU	76.0
6.25% trained	8.9e16 / 3.7e9	722x / 8x	14M	6h on 1 V100 GPU	74.1
ELECTRA-Base	6.4e19 / 2.9e10	1x / 1x	110M	4d on 16 TPUv3s	85.1

So sánh trên bộ dev GLUE với các mô hình nhỏ

Kết quả

Model	Train FLOPs	Params	CoLA	SST	MRPC	STS	QQP	MNLI	QNLI	RTE	Avg.
BERT	1.9e20 (0.27x)	335M	60.6	93.2	88.0	90.0	91.3	86.6	92.3	70.4	84.0
RoBERTa-100K	6.4e20 (0.90x)	356M	66.1	95.6	**91.4**	92.2	92.0	89.3	94.0	82.7	87.9
RoBERTa-500K	3.2e21 (4.5x)	356M	68.0	96.4	90.9	92.1	92.2	90.2	94.7	86.6	88.9
XLNet	3.9e21 (5.4x)	360M	69.0	**97.0**	90.8	92.2	92.3	90.8	94.9	85.9	89.1
BERT (ours)	7.1e20 (1x)	335M	67.0	95.9	89.1	91.2	91.5	89.6	93.5	79.5	87.2
ELECTRA-400K	7.1e20 (1x)	335M	**69.3**	96.0	90.6	92.1	**92.4**	90.5	94.5	86.8	89.0
ELECTRA-1.75M	3.1e21 (4.4x)	335M	69.1	96.9	90.8	**92.6**	**92.4**	**90.9**	**95.0**	**88.0**	**89.5**

So sánh trên bộ dev GLUE với mô hình lớn

Kết quả

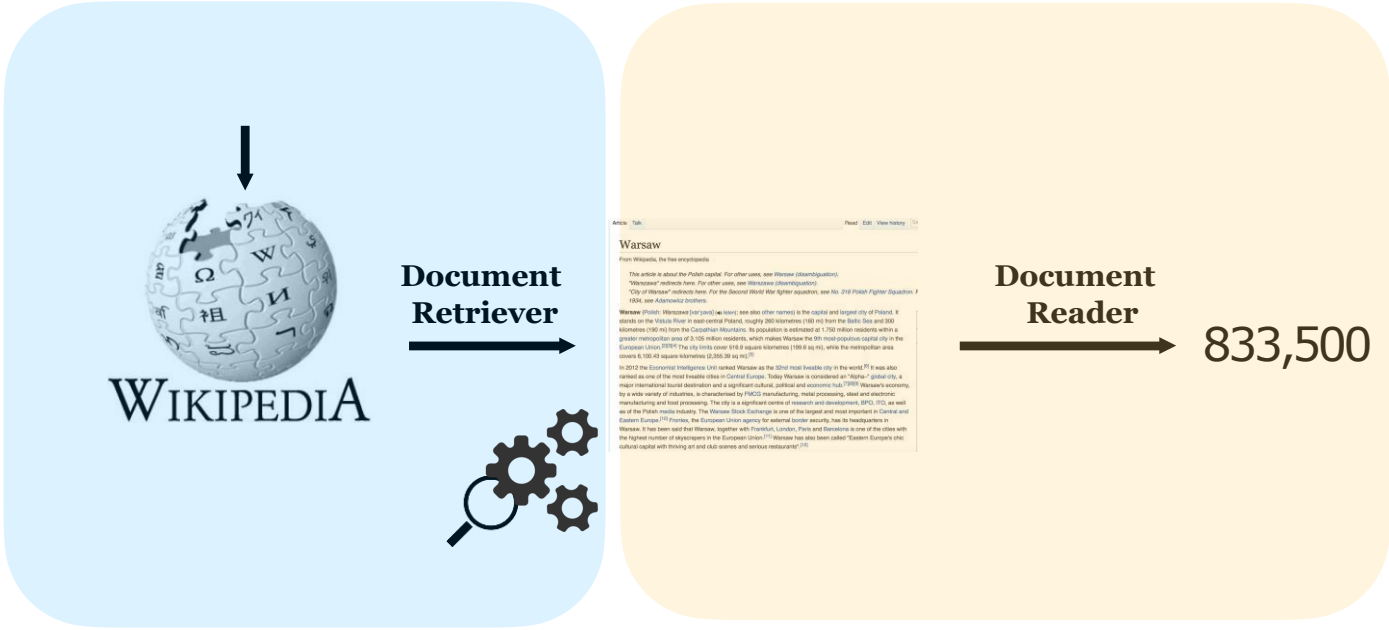
Model	Train FLOPs	Params	SQuAD 1.1 dev		SQuAD 2.0 dev		SQuAD 2.0 test	
			EM	F1	EM	F1	EM	F1
BERT-Base	6.4e19 (0.09x)	110M	80.8	88.5	–	–	–	–
BERT	1.9e20 (0.27x)	335M	84.1	90.9	79.0	81.8	80.0	83.0
SpanBERT	7.1e20 (1x)	335M	88.8	94.6	85.7	88.7	85.7	88.7
XLNet-Base	6.6e19 (0.09x)	117M	81.3	–	78.5	–	–	–
XLNet	3.9e21 (5.4x)	360M	**89.7**	**95.1**	87.9	**90.6**	87.9	90.7
RoBERTa-100K	6.4e20 (0.90x)	356M	–	94.0	–	87.7	–	–
RoBERTa-500K	3.2e21 (4.5x)	356M	88.9	94.6	86.5	89.4	86.8	89.8
ALBERT	3.1e22 (44x)	235M	89.3	94.8	87.4	90.2	88.1	90.9
BERT (ours)	7.1e20 (1x)	335M	88.0	93.7	84.7	87.5	–	–
ELECTRA-Base	6.4e19 (0.09x)	110M	84.5	90.8	80.5	83.3	–	–
ELECTRA-400K	7.1e20 (1x)	335M	88.7	94.2	86.9	89.6	–	–
ELECTRA-1.75M	3.1e21 (4.4x)	335M	**89.7**	94.9	**88.0**	**90.6**	**88.7**	**91.4**

Kết quả trên bộ SQuAD

Hỏi đáp miền mở

- Không cho trước 1 đoạn nội dung
- Tìm kiếm trên tập lớn các tài liệu (như Wikipedia). Cần trả về câu trả lời cho bất kỳ miền dữ liệu nào
- Thách thức hơn nhưng thực tế hơn!

Mô hình Retriever-Reader



https://github.com/facebookresearch/DrQA

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

Mô hình Retriever-Reader

- Input: tập lớn các tài liệu $\mathcal{D} = D_1, D_2, \ldots, D_N$ và Q
- Output: câu trả lời A
- Retriever: $f(\mathcal{D}, Q) \longrightarrow P_1, \ldots, P_K$ K được xác định trước (e.g., 100)
- Reader: $g(Q, \{P_1, \ldots, P_K\}) \longrightarrow A$

Trong DrQA,

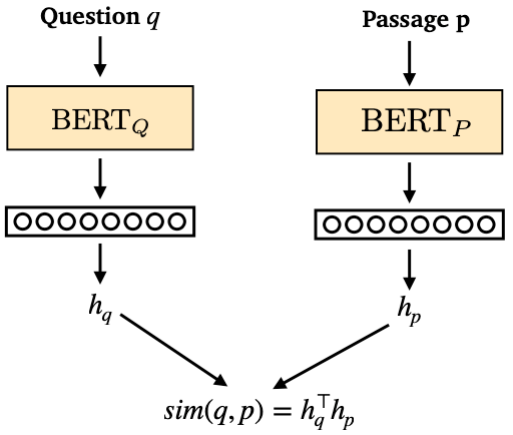
- Retriever = Là mô hình tìm kiếm thưa dựa trên TF-IDF
- Reader = mô hình đọc hiểu sử dụng học sâu
  - Huấn luyện trên SQuAD và các tập QA (CuratedTREC, WebQuestions and WikiMovies) được huấn luyện từ xa (distantly-supervised)

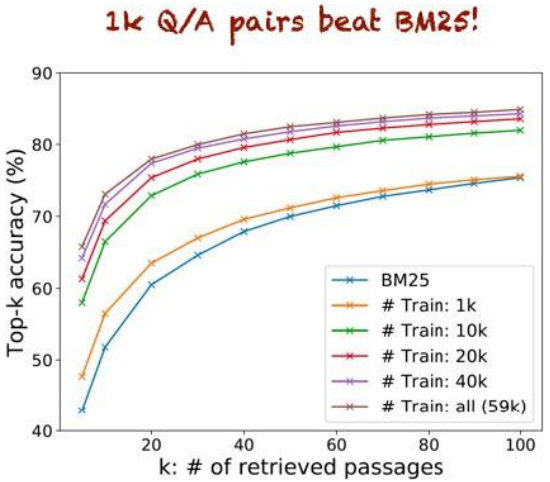
Distantly-supervised examples: (Q, A) (P, Q, A)

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

Huấn luyện bộ tìm kiếm

- Dense passage retrieval (DPR) – chỉ cần huấn luyện công cụ tìm kiếm sử dụng các cặp QA !





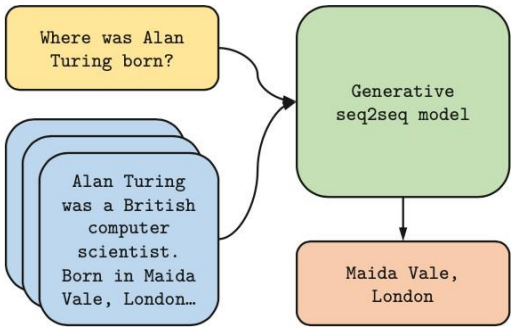
- Các bộ tìm kiếm có thể huấn luyện được (dùng BERT) có kết quả tốt hơn nhiều so với các mô hình IR truyền thống

Karpukhin et al., 2020. Dense Passage Retrieval for Open-Domain Question Answering

Dense retrieval + generative models

Các nghiên cứu hiện nay cho thấy nên sinh câu trả lời hơn là trích xuất câu trả lời

Fusion-in-decoder (FID) = DPR + T5

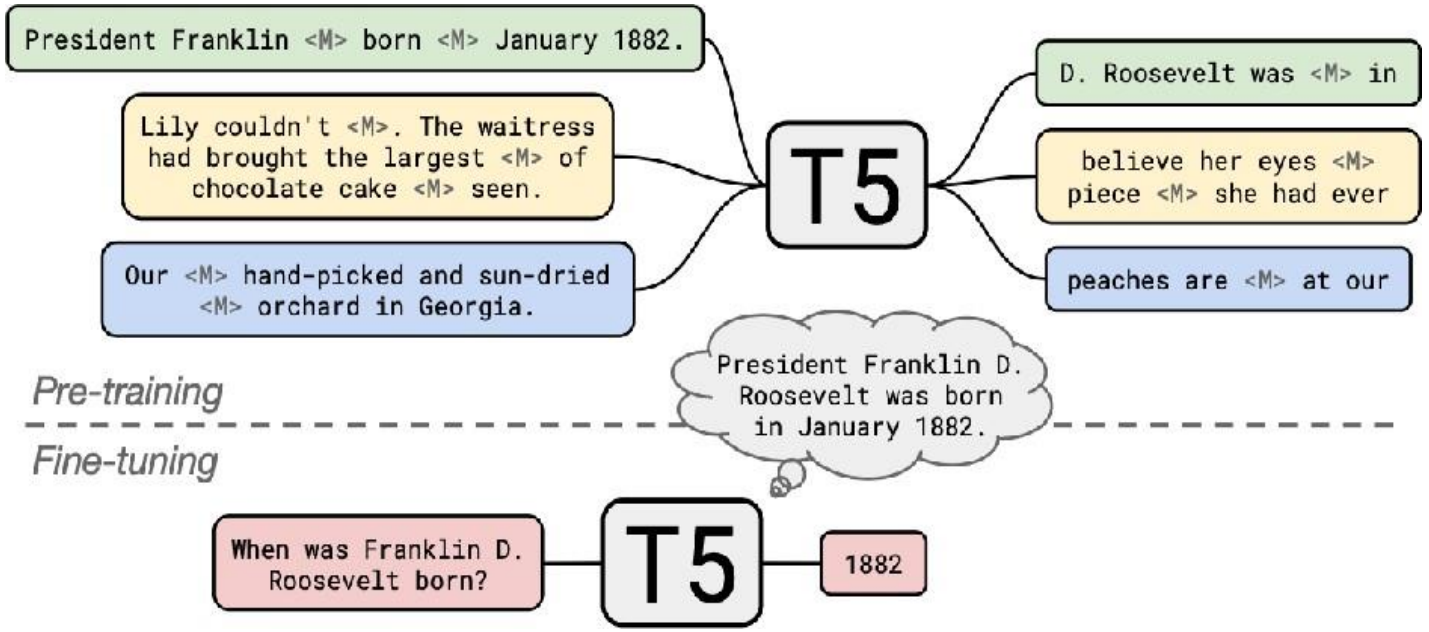


Model	NaturalQuestions	TriviaQA	
ORQA (Lee et al., 2019)	31.3	45.1	-
REALM (Guu et al., 2020)	38.2	-	-
DPR (Karpukhin et al., 2020)	41.5	57.9	-
SpanSeqGen (Min et al., 2020)	42.5	-	-
RAG (Lewis et al., 2020)	44.5	56.1	68.0
T5 (Roberts et al., 2020)	36.6	-	60.5
GPT-3 few shot (Brown et al., 2020)	29.9	-	71.2
Fusion-in-Decoder (base)	48.2	65.0	77.1
Fusion-in-Decoder (large)	**51.4**	**67.6**	**80.1**

Izacard and Grave 2020. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering

Large language models can do open-domain QA well

… không có giai đoạn tìm kiếm



Roberts et al., 2020. How Much Knowledge Can You Pack Into the Parameters of a Language Model?

Q&A

Thank you!