TensorFlow 2.0 Question Answering

Yuanhao Wu 吴远皓

wuyhthu@gmail.com

2020/4/23

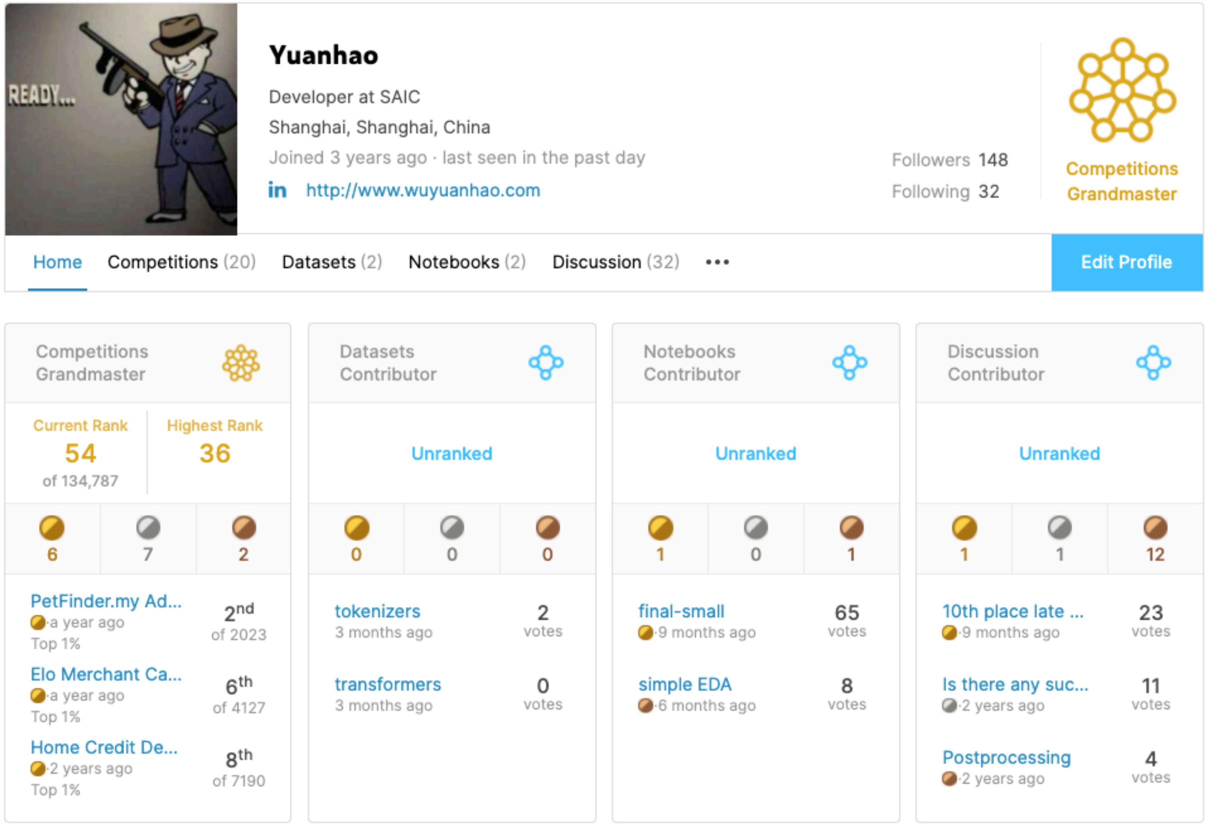
Outline

- Introduction to MRC
    - Overview/Datasets/Models
- TensorFlow 2.0 Question Answering Competition
    - Dataset/Baseline/Tricks
- Other Applications of MRC Algorithm
- What's next

About Me

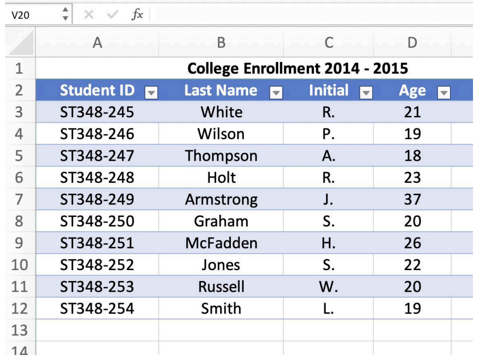
NLP engineer

Kaggle Competitions Grandmaster



Introduction to MRC: Overview

- Machine Reading Comprehension (MRC) is an important **Question Answering** technology

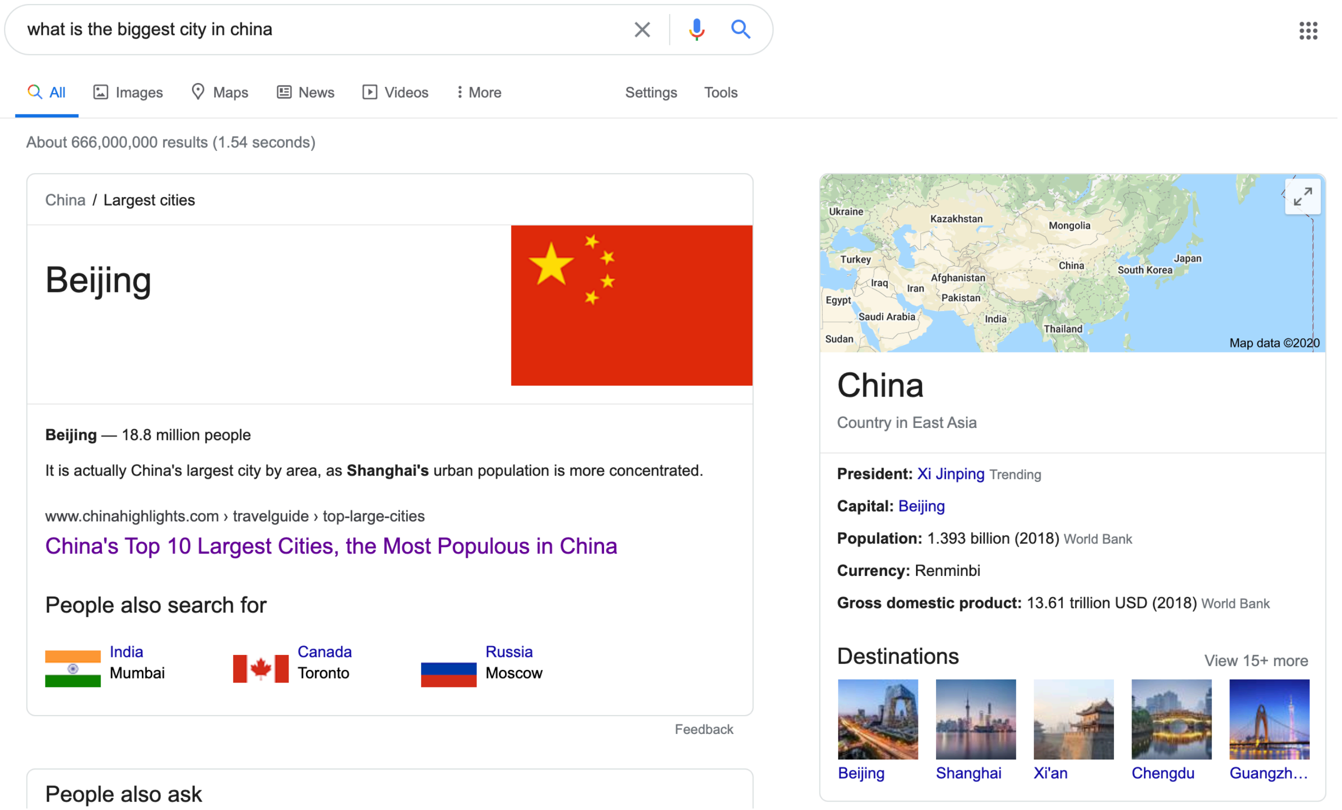


	A	B	C	D
1	College Enrollment 2014 - 2015			
2	Student ID	Last Name	Initial	Age
3	ST348-245	White	R.	21
4	ST348-246	Wilson	P.	19
5	ST348-247	Thompson	A.	18
6	ST348-248	Holt	R.	23
7	ST348-249	Armstrong	J.	37
8	ST348-250	Graham	S.	20
9	ST348-251	McFadden	H.	26
10	ST348-252	Jones	S.	22
11	ST348-253	Russell	W.	20
12	ST348-254	Smith	L.	19
13				
14				

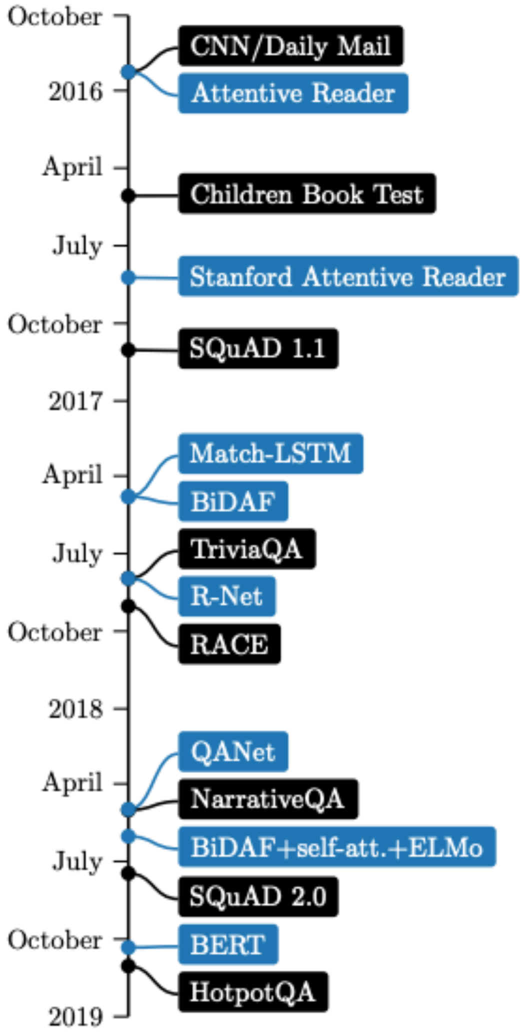
knowledge	question-answer pair	documents	tables	knowledge graph
algorithm	semantic matching	IR & MRC	NL2SQL	?

Introduction to MRC: Overview

- read and understand **unstructured text** and then answer questions about it



Introduction to MRC: Datasets



- extractive QA
- SQuAD 1.1
  - 500 Wikipedia articles
  - 23000 passages
  - 100000 questions
- SQuAD 2.0
  - 50000 unanswerable questions written adversarially by crowdworkers to look like answerable ones

Introduction to MRC: Datasets

- Context is short
- predict only one span (5 annotations)
- most answers are noun phrase

Answer type	Percentage	Example
Date	8.9%	19 October 1512
Other Numeric	10.9%	12
Person	12.9%	Thomas Coke
Location	4.4%	Germany
Other Entity	15.3%	ABC Sports
Common Noun Phrase	31.8%	property damage
Adjective Phrase	3.9%	second-largest
Verb Phrase	5.5%	returned to Earth
Clause	3.7%	to avoid trivialization
Other	2.7%	quietly

Table 2: We automatically partition our answers into the following categories. Our dataset consists of large number of answers beyond proper noun entities.

Oxygen is a chemical element with symbol O and atomic number 8. It is a member of the chalcogen group on the periodic table and is a highly reactive nonmetal and oxidizing agent that readily forms compounds (notably oxides) with most elements. By mass, oxygen is the third-most abundant element in the universe, after hydrogen and helium. At standard temperature and pressure, two atoms of the element bind to form dioxygen, a colorless and odorless diatomic gas with the formula O
2. Diatomic oxygen gas constitutes 20.8% of the Earth's atmosphere. However, monitoring of atmospheric oxygen levels show a global downward trend, because of fossil-fuel burning. Oxygen is the most abundant element by mass in the Earth's crust as part of oxide compounds such as silicon dioxide, making up almost half of the crust's mass.

The atomic number of the periodic table for oxygen?
Ground Truth Answers: 8 | 8 | 8 | 8 | 8

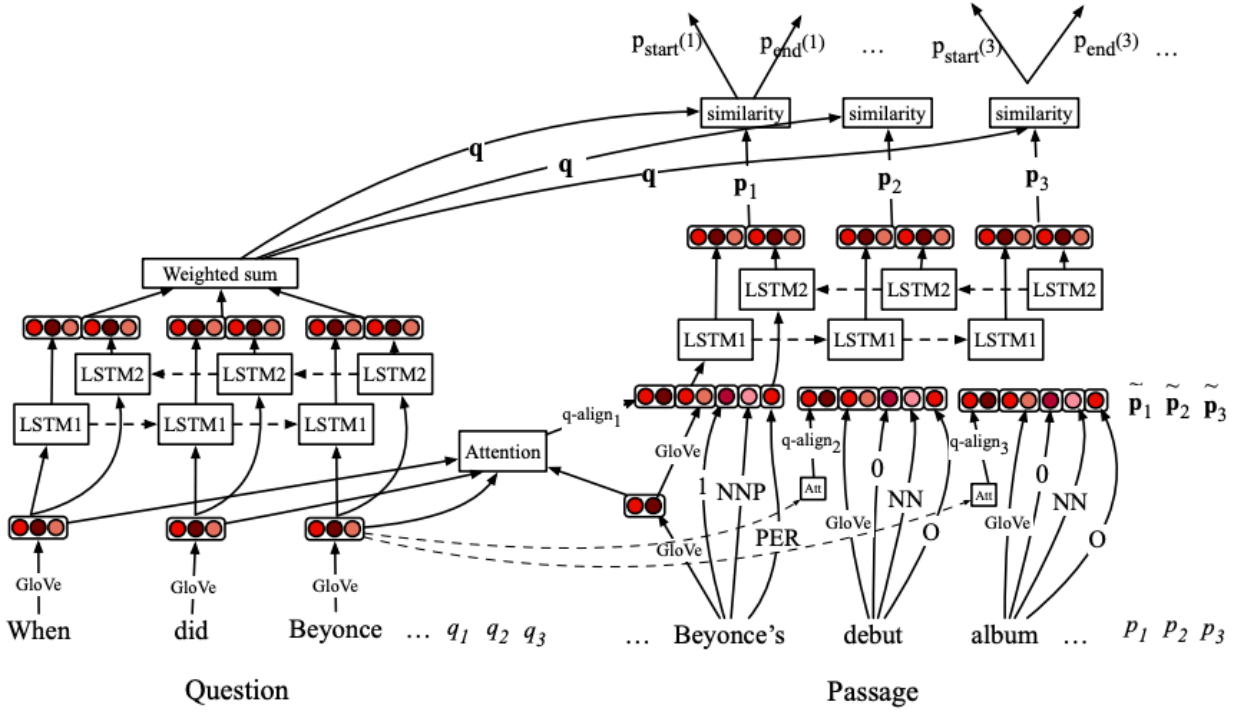
What is the second most abundant element?
Ground Truth Answers: helium | helium | helium | helium | helium

Which gas makes up 20.8% of the Earth's atmosphere?
Ground Truth Answers: Diatomic oxygen | Diatomic oxygen | Diatomic oxygen gas | Diatomic oxygen | Diatomic oxygen gas

How many atoms combine to form dioxygen?
Ground Truth Answers: two atoms | two | two | two | two

Introduction to MRC: Models

- Classical MRC models are complicated
- Pretrained word embedding/question aligned embedding
- RNN-based encoder
- Attention mechanisms
- Lexical features: POS TAG/NER/EM



A full model of STANFORD ATTENTIVE READER

Introduction to MRC: Models

- Pretrained language models make life much easier
- Self-attention transformers + output layer for start/end logits
- No need for problem-specific tricks
- The Devil's in the details
  - Preprocessing: character alignment
  - Post processing

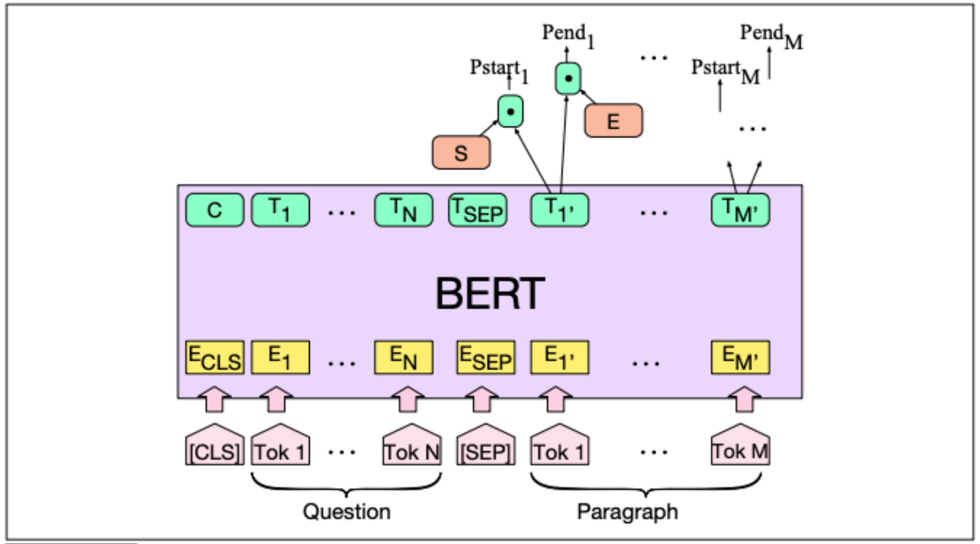
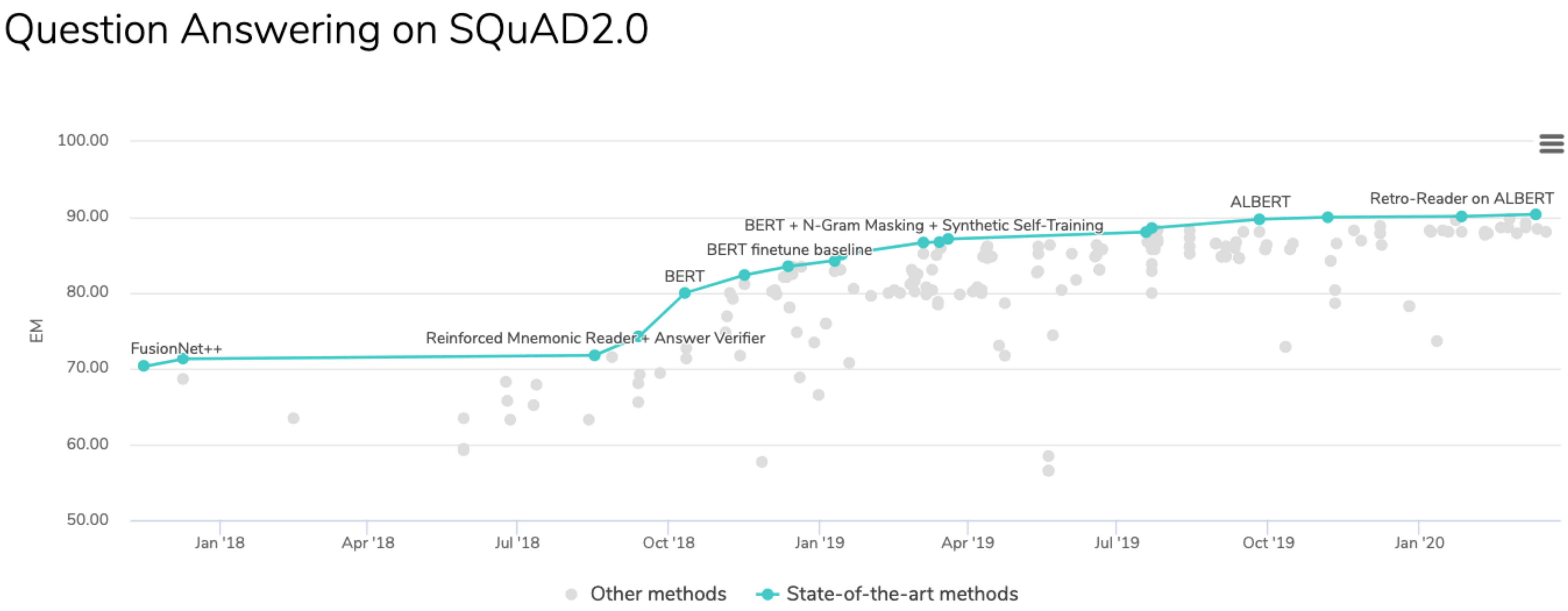


Figure 25.8 The BERT model for span-based question answering from reading-comprehension-based question answering tasks. Figure after Devlin et al. (2019).

Introduction to MRC: Models



https://paperswithcode.com/sota/question-answering-on-squad20

Natural Questions Dataset

- NQ contains 307,372 training examples, 7,830 examples for development, and we withold a further 7,842 examples for testing
- Contexts are **entire** Wikipedia articles (not paragraphs)
- Long answer/short answer
- The NQ training data contains 307,373 examples. 152,148 have a long answer and 110,724 have a short answer. Short answers **can be sets of spans** in the document (106,926), or yes or no (3,798)

Natural Questions Dataset

- Long answers are HTML bounding boxes, and the distribution of NQ long answer types is as follows:

HTML tags	Percent of long answers
<P>	72.9%
<Table>	19.0%
<Tr>	1.5%
<Ul>, <Ol>, <Dl>	3.2%
<Li>, <Dd>, <Dt>	3.4%

Natural Question Dataset

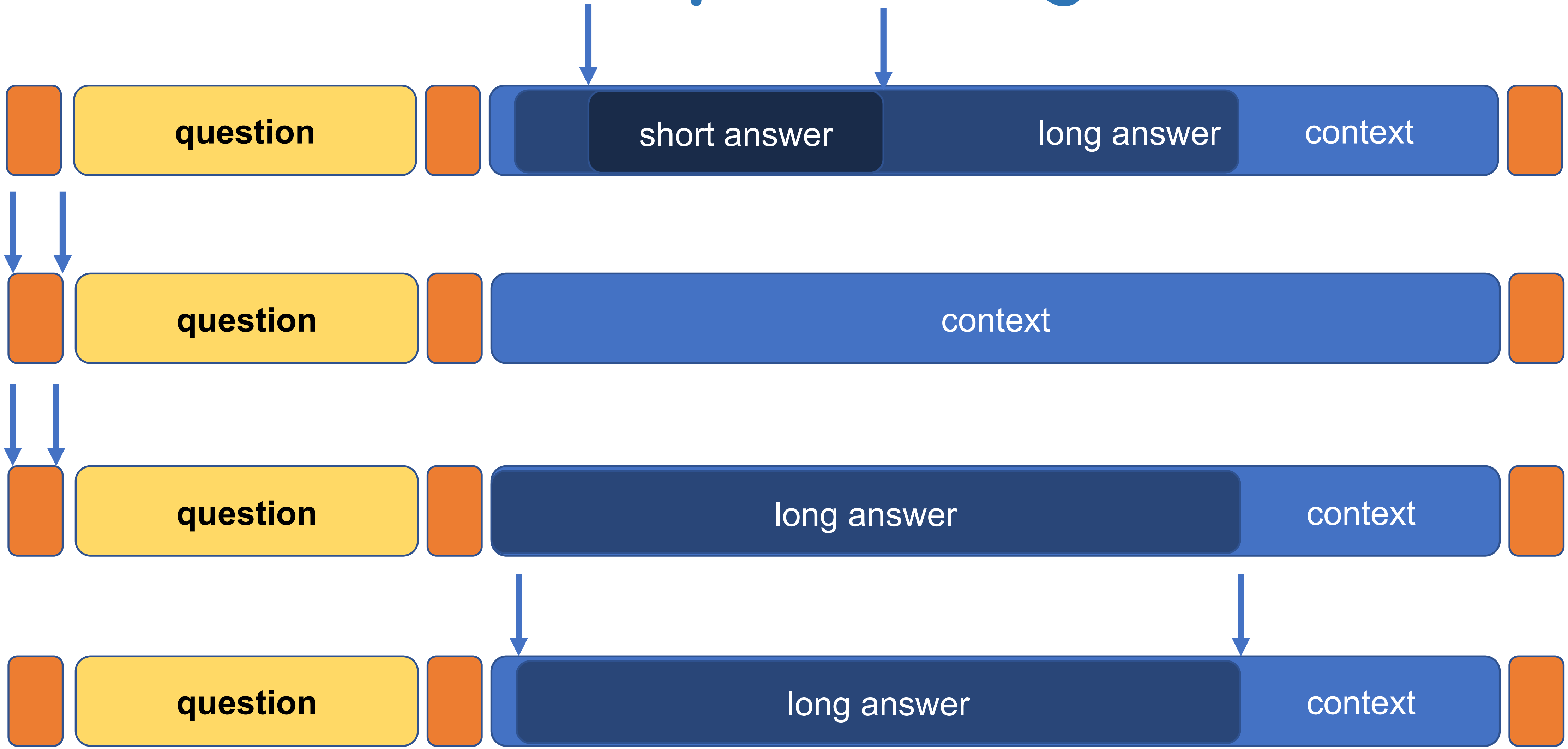
- human upper bound of 87% F1 on the long answer selection task, and 76% on the short answer selection task. (compared to 89.4% F1 of SQuAD 2.0)

#	△pub	Team Name	Notebook	Team Members	Score	Entries	Last
1	▲ 3	Guanshuo Xu			0.71705	8	3mo
2	▼ 1	DeepThought	</> submit_full		0.71684	71	3mo
3	▲ 8	in pytorch we trust			0.70436	71	3mo
4	▲ 3	toxu			0.70025	71	3mo
5	▼ 3	bestfitting			0.69687	36	3mo
6	▼ 1	prvi			0.69474	11	3mo
7	▲ 15	jb			0.69196	31	3mo
8	▲ 2	[ods.ai] Oleg Platonov			0.68916	31	3mo
9	▲ 3	Anastasia Karpovich	</> tfqa-bert-train		0.68878	30	3mo
10	▼ 4	H1kk111111			0.68710	88	3mo

NQ Baseline: Preprocessing

- [cls] question [sep] context [sep]
- sliding window, stride=128. On average, 30 **instances** per NQ **example**.
- use special markup tokens to give the model a notion of which part of the document it is reading. e.g. [Paragraph=N]
- five answer type: null, short, long, yes, no
- **about 98% of generated instances are null**
- **randomly discard null instances**

NQ Baseline: Preprocessing



NQ Baseline: Model

- define a training set instance as a four-tuple, **(c, s, e, t)**
- train with negative log likelihood loss

$$
\begin{aligned}
L &= -\log p(s, e, t|c) \\
&= -\log p_{\text{start}}(s|c) - \log p_{\text{end}}(e|c) \\
&\quad - \log p_{\text{type}}(t|c),
\end{aligned}
$$

$$
p_{\text{start}}(s|c) = \frac{\exp(f_{\text{start}}(s, c; \theta))}{\sum_{s'} \exp(f_{\text{start}}(s', c; \theta))},
$$

$$
p_{\text{end}}(e|c) = \frac{\exp(f_{\text{end}}(e, c; \theta))}{\sum_{e'} \exp(f_{\text{end}}(e', c; \theta))},
$$

$$
p_{\text{type}}(t|c) = \frac{\exp(f_{\text{type}}(t, c; \theta))}{\sum_{t'} \exp(f_{\text{type}}(t', c; \theta))},
$$

NQ Baseline: Post Processing

- Find all valid spans
  - s<=e, e-s<=max length, e&s in document
- Rank all spans by score g

$$\begin{aligned} g(c,s,e) = {} & f_{\text{start}}(s,c;\theta) \\ & + f_{\text{end}}(e,c;\theta) \\ & - f_{\text{start}}(s=[\text{CLS}],c;\theta) \\ & - f_{\text{end}}(e=[\text{CLS}],c;\theta) \end{aligned}$$

important

- Always output one single short answer as prediction

Natural Question Tricks

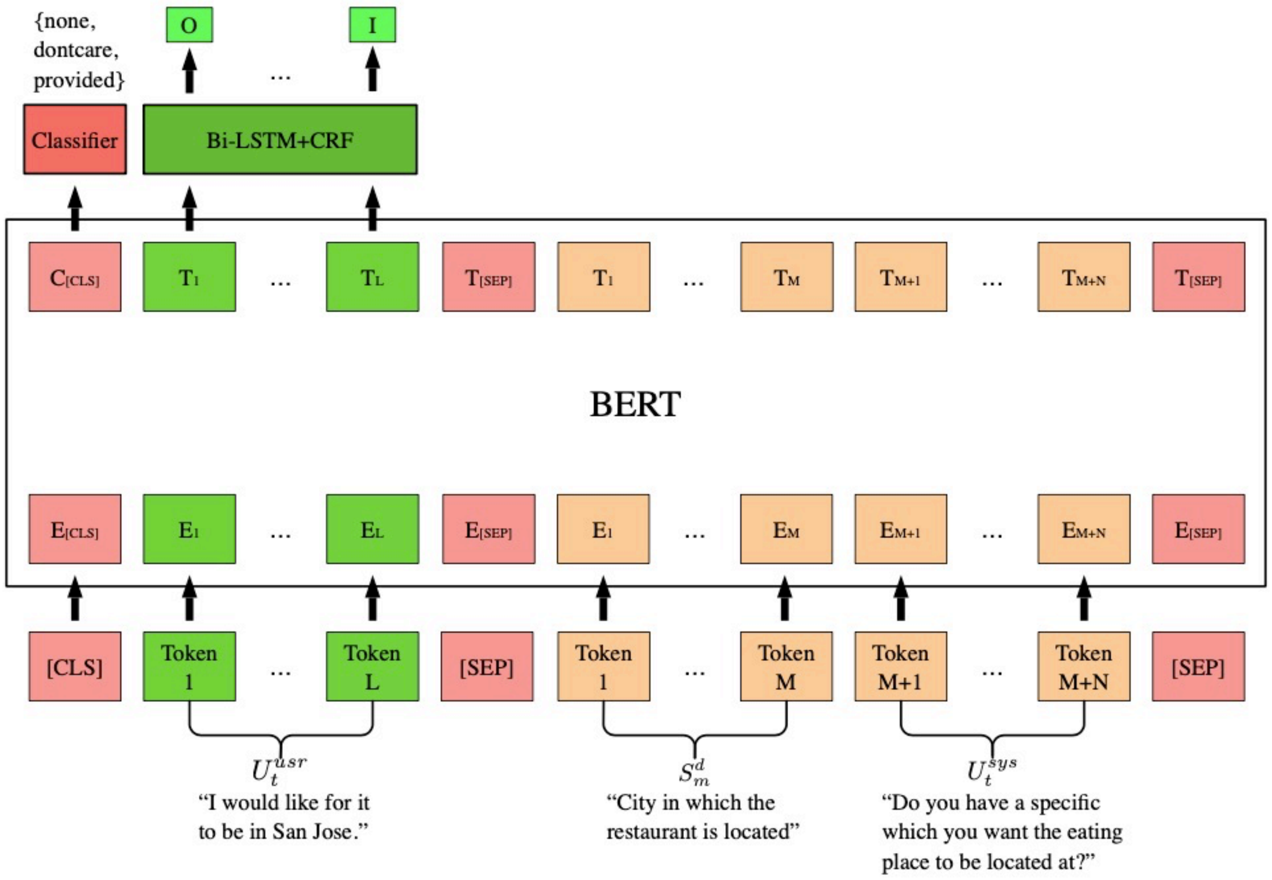
- Don't waste your time playing with model structure
- Pay more attention to data
- Randomly discarding null instances in preprocessing leads to gaps between training and testing
    - can not discard any instance when testing
    - much more data
    - may be harder

Natural Question Tricks

- Ensemble
  - different LMs use different Tokenizers
  - map token logits to word logits
- Hard negative sampling (from 1st solution)
  - firstly trained a model with uniform sampling
  - and predicted on the whole training data, and stored the answer probability for each negative candidate
- 2 stage strategy (from 1st solution)
  - Use Bert-based model to propose long answer candidates

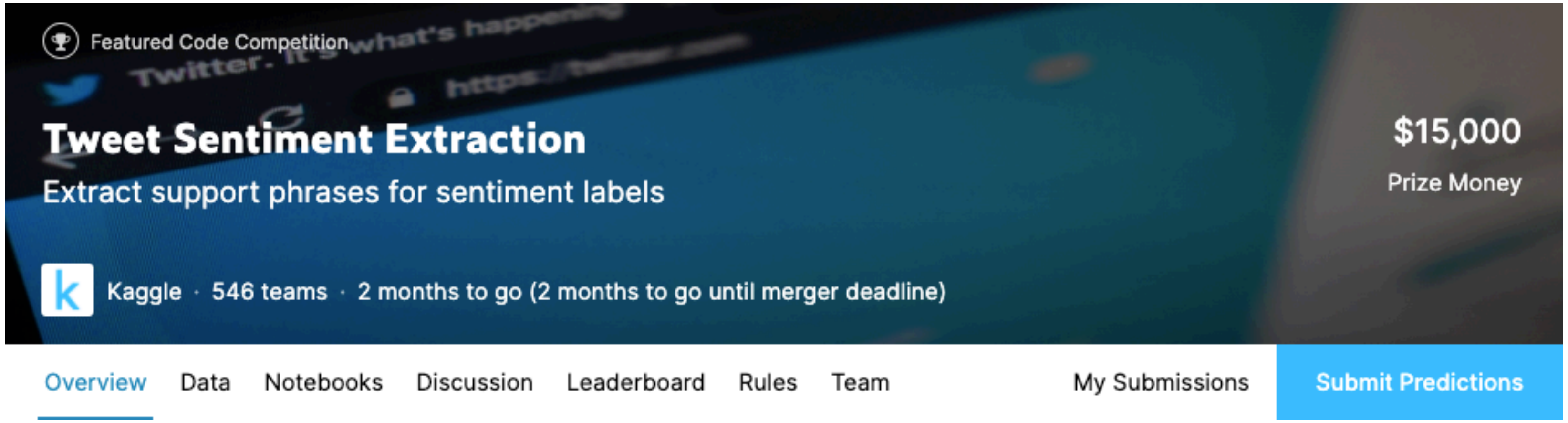
Other Applications

- zero-shot free-form DST
- context: last user utterance
- question: slot desc. and last sys utterance
- classification + span prediction
- BERT + LSTM + CRF



Zero-Shot State Tracking and User Adoption Tracking on Schema-Guided Dialogue, AAAI, 2020

What's next



- ongoing Kaggle competition, https://www.kaggle.com/c/tweet-sentiment-extraction/overview
- You're attempting to predict the word or phrase from the tweet that exemplifies the provided sentiment.

What's next

- small dataset (quite noisy), easy to play with
  - about 27k training samples
  - sentences are all short (they are Tweets!)

	textID	text	selected_text	sentiment
	27481 unique values	27480 unique values	22463 unique values	neutral 40% positive 31% Other (1) 28%
10	fc2cbefa9d	Journey!? Wow... u just became cooler. hehe... (is that possible!?)	Wow... u just became cooler.	positive
11	2339a9b08b	as much as i love to be hopeful, i reckon the chances are minimal =P i`m never gonna get my cake and stuff	as much as i love to be hopeful, i reckon the chances are minimal =P i`m never gonna get my cake and stuff	neutral
12	16fab9f95b	I really really like the song Love Story by Taylor Swift	like	positive

input

target

What's next

- Many great Kernels to start with
  - sentiment as **query**, text as **context**, selected_text as **answer**
- The metric in this competition is the word-level Jaccard score

```
def jaccard(str1, str2):
    a = set(str1.lower().split())
    b = set(str2.lower().split())
    c = a.intersection(b)
    return float(len(c)) / (len(a) + len(b) - len(c))
```

- There may be some metric-related tricks

References

1. Chen, Danqi. "Neural reading comprehension and beyond." PhD diss., Stanford University, 2018.
2. Jurafsky, Dan. *Speech & language processing*. Pearson Education India, 2000.
3. Lei, Shuyu, Shuaipeng Liu, Mengjun Sen, Huixing Jiang, and Xiaojie Wang. "Zero-shot state tracking and user adoption tracking on schema-guided dialogue." In *Dialog System Technology Challenge Workshop at AAAI*. 2020.
4. Alberti, Chris, Kenton Lee, and Michael Collins. "A bert baseline for the natural questions." *arXiv preprint arXiv:1901.08634* (2019).
5. https://github.com/huggingface/transformers/blob/master/src/transformers/data/processors/squad.py
6. https://ai.google.com/research/NaturalQuestions
7. https://github.com/google-research-datasets/natural-questions
8. https://www.kaggle.com/c/tensorflow2-question-answering/discussion/127551

Q&A

Thank you!