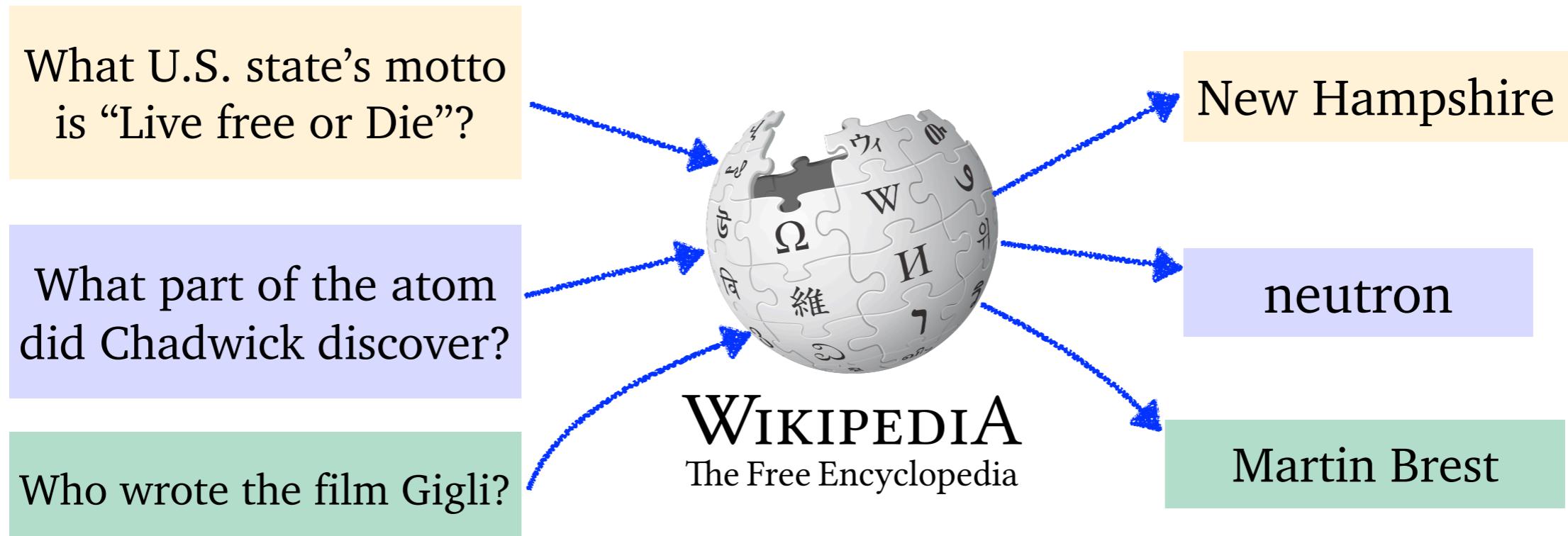


Part IV

Two-stage retriever-reader approaches

Problem setup

- **Input:** question Q , D = English Wikipedia (~ 5 million documents)
- **Output:** answer A



“Machine Reading at Scale”

Why is Wikipedia?



- We treat Wikipedia as a generic collection of articles and don't consider its internal graph structure in this setting \implies easy to extend to any collection of documents.
- The search problem is challenging and realistic while its scale is still manageable, especially for academic research.
- Wikipedia contains a wealth of information of real-world facts. We don't consider the *redundancy* problem too much here (vs. Web documents or news corpora).

DrQA: a first neural open-domain QA system

[Chen et al., 2017]

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

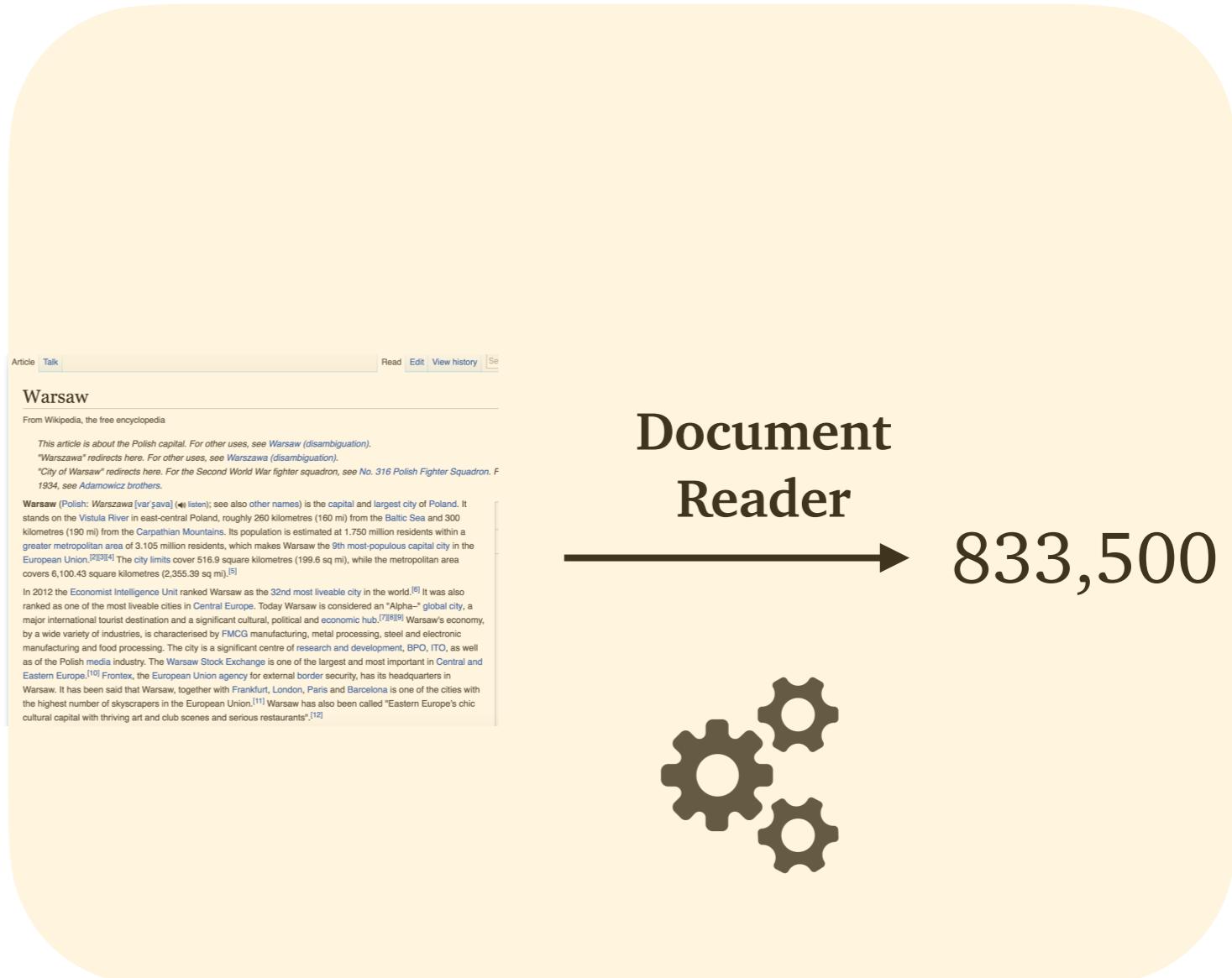


WIKIPEDIA

Document
Retriever



Information Retrieval



Reading Comprehension

Document Retriever

- A TF-IDF weighted term vector model over unigrams/bigrams:

tf = term frequency, idf = inverse document frequency

t : term (uni/bi), d : document (= one Wiki. article), D : corpus (= Wikipedia)

$$\text{tf-idf}(t, d, D) = \text{tf}(t, d) \times \text{idf}(t, D)$$

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

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

- This retriever is not *trainable*.
- Retriever at document level instead of paragraph level.

Document Reader

paragraph, document,
arbitrary-length text blocks..

Cast as a *reading comprehension* problem:

- **Input** is a passage P and a question Q
- **Output** is an answer A

A restricted setting is that A needs
to be a segment of text in $P \Rightarrow$
“extractive question answering”

Stanford Question Answering Dataset
[Rajpurkar et al., 2016]

Passage

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.

Question: Which NFL team won Super Bowl 50?

Answer: Denver Broncos

Question: What does AFC stand for?

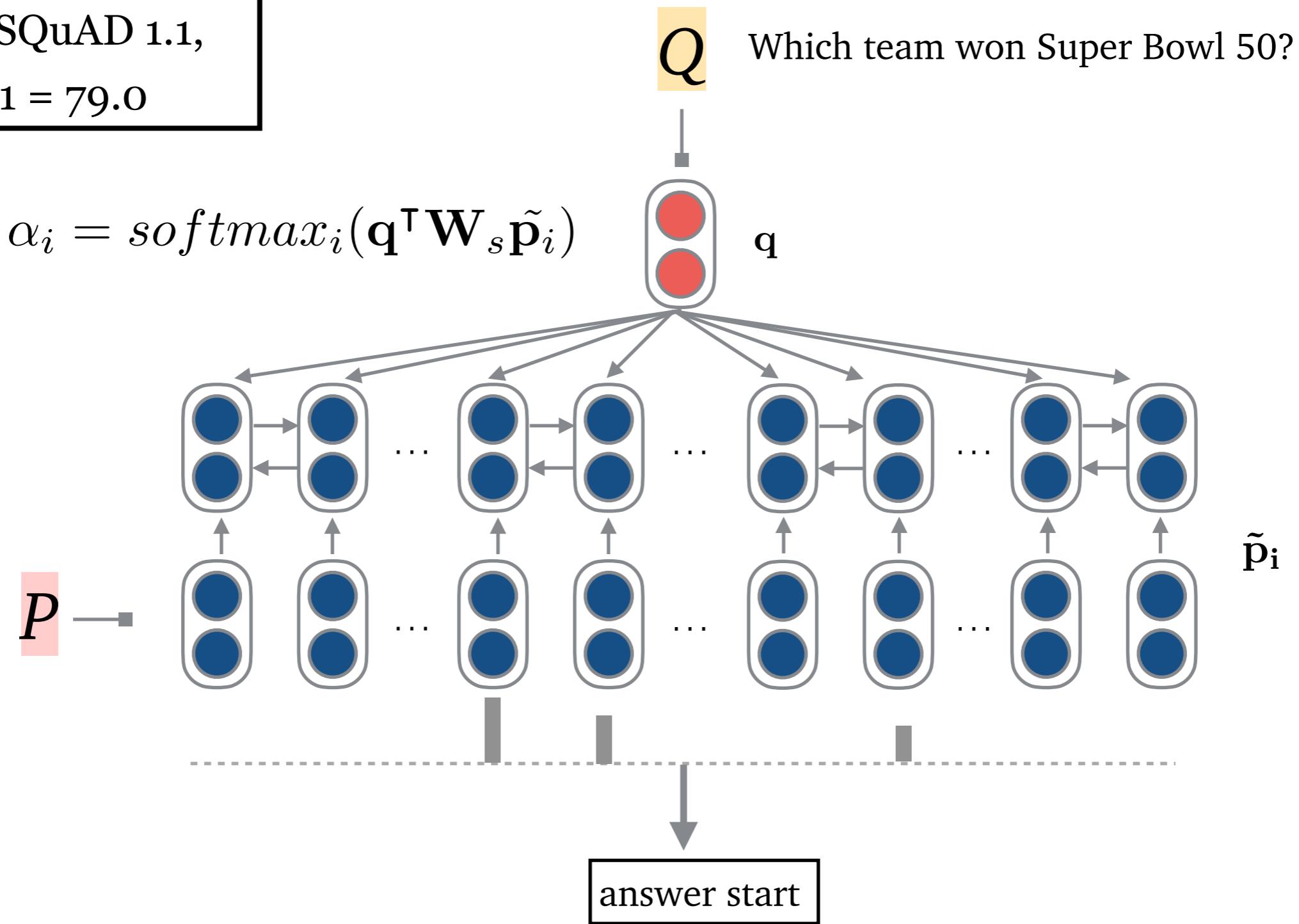
Answer: American Football Conference

Question: What year was Super Bowl 50?

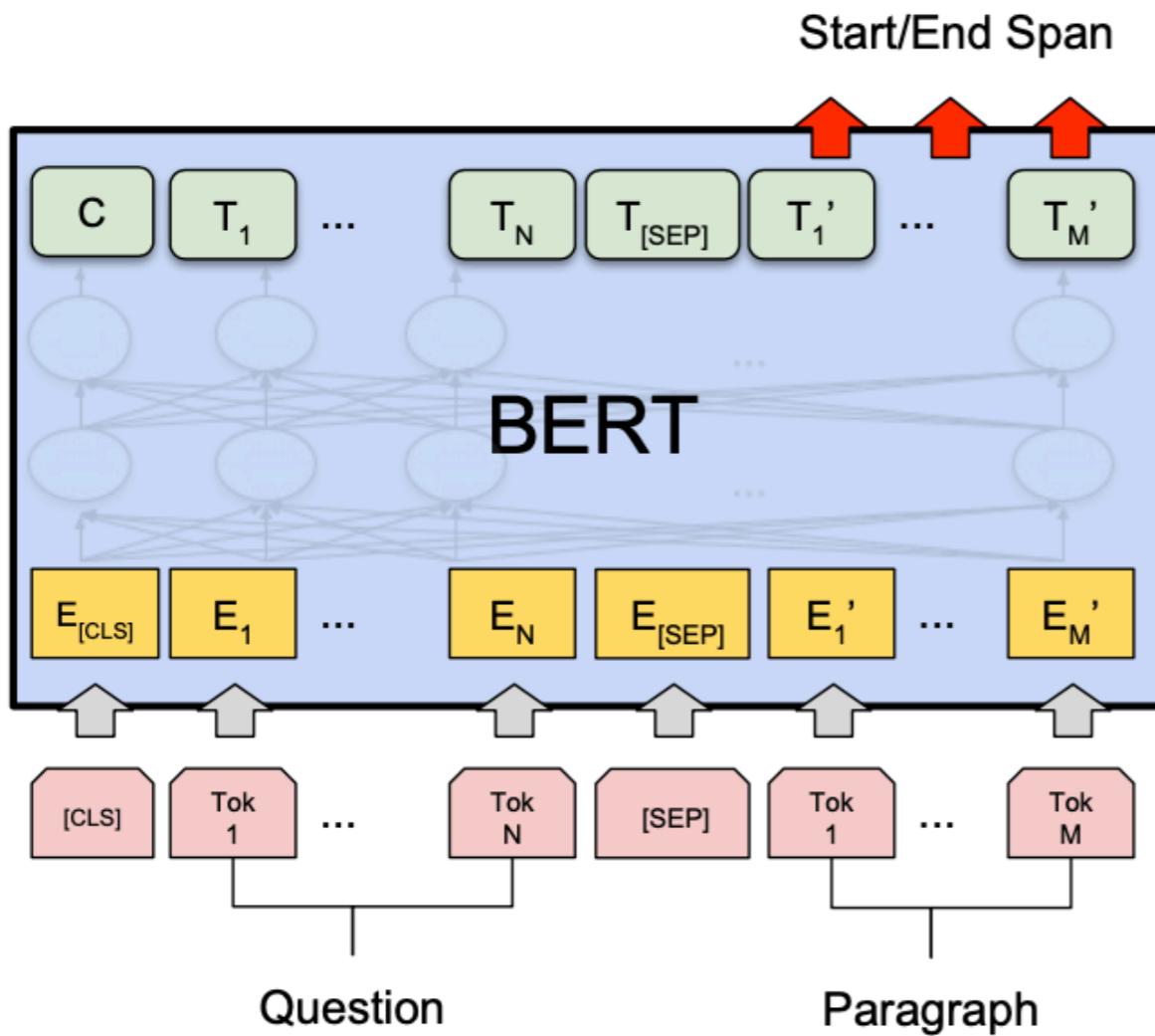
Answer: 2016

Document Reader

On SQuAD 1.1,
• F1 = 79.0



Document Reader



Question = Segment A

Passage = Segment B

Answer = predicting two endpoints in segment B

On SQuAD 1.1,

- Bert : F1 = 90.9
- RoBERTa: F1 = 94.6

How to train the model?

- Using an existing reading comprehension dataset (e.g., SQuAD)!

$$\mathcal{D}_{\text{rc}} = \{(P_i, Q_i, A_i)\}$$

Problem: very different distribution with real-world QA data.

- How about other QA datasets (e.g., WebQuestions, TREC)?

$$\mathcal{D}_{\text{QA}} = \{(Q_i, A_i)\}$$

- Solution: create **distantly-supervised** examples using our **retriever**!

$$(Q, A) \longrightarrow (P, Q, A)$$

if passage is retrieved and answer can be found in passage

Similar to distant supervision
in information extraction [Mintz et al., 2009]

How to train the model?

Question: What U.S. state's motto is "Live free or Die"?

Answer: New Hampshire

Passage

Live Free or Die

From Wikipedia, the free encyclopedia

"**Live Free or Die**" is the official motto of the U.S. state of **New Hampshire**, adopted by the state in 1945.^[1] It is possibly the best-known of all [state mottos](#), partly because it conveys an assertive [independence](#) historically found in [American political philosophy](#) and partly because of its contrast to the milder sentiments found in other state mottos.

Putting it together

Training time:

- Document retriever: not trained
- Document reader: a LSTM-based neural reading comprehension model trained on SQuAD + distantly-supervised data generated from QA datasets

Inference time:

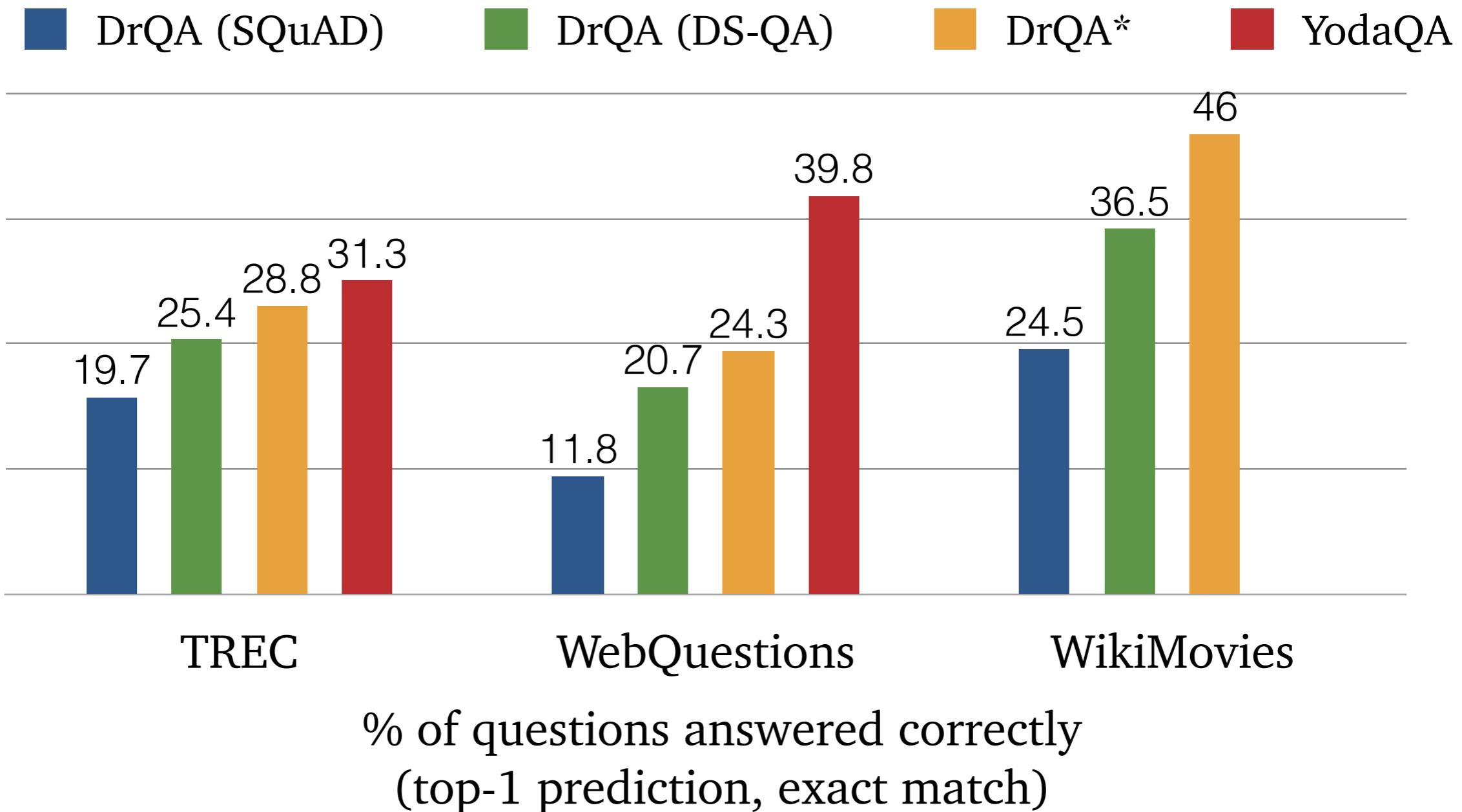
- The document retriever returns *top 5 documents*
- The reader reads every (natural) paragraph in these 5 documents and predicts an answer and its span score.
- The system finally returns the answer with the highest (unnormalized) span score.

Experiments

See [Raison et al., 2018]

Trained reader on
full documents

Hybrid system on KBs, text,
ontologies etc [http://ailao.eu/
yodaqa/](http://ailao.eu/yodaqa/)

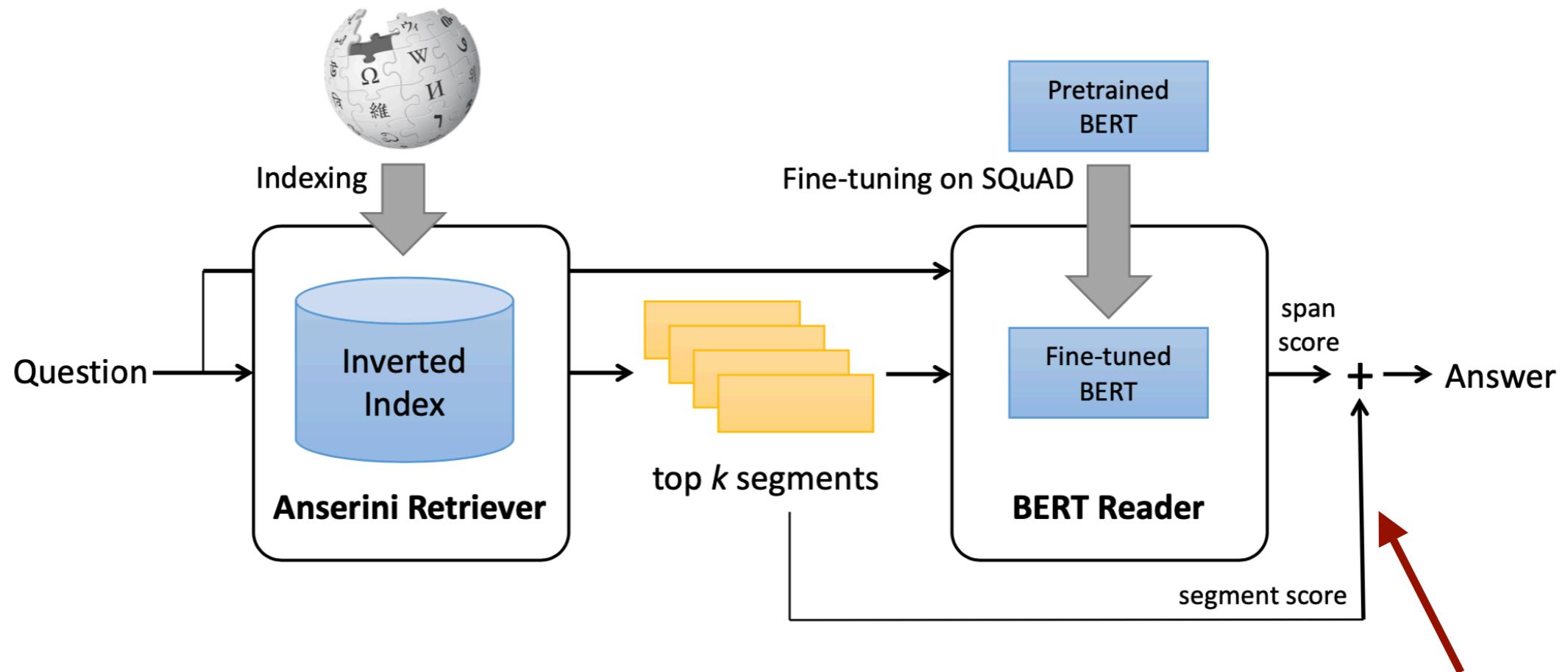


How can we do better?

- DrQA considers retrieval at document level.
Does paragraph-level retriever work better?
- Answers in the retrieved passages might not be directly comparable at inference time.
Does multi-passage training help?
- The importance of each passage has been omitted.
Can we use passage retriever scores or even train a better ranker?
- The retriever is not trained!

The focus of the next part.

BERTserini [Yang et al., 2019]



Anserini Retriever
[Yang et al. 2017]:
Lucene with BM25, operated
on 29.5M paragraphs

BERT Reader:
Trained on SQuAD

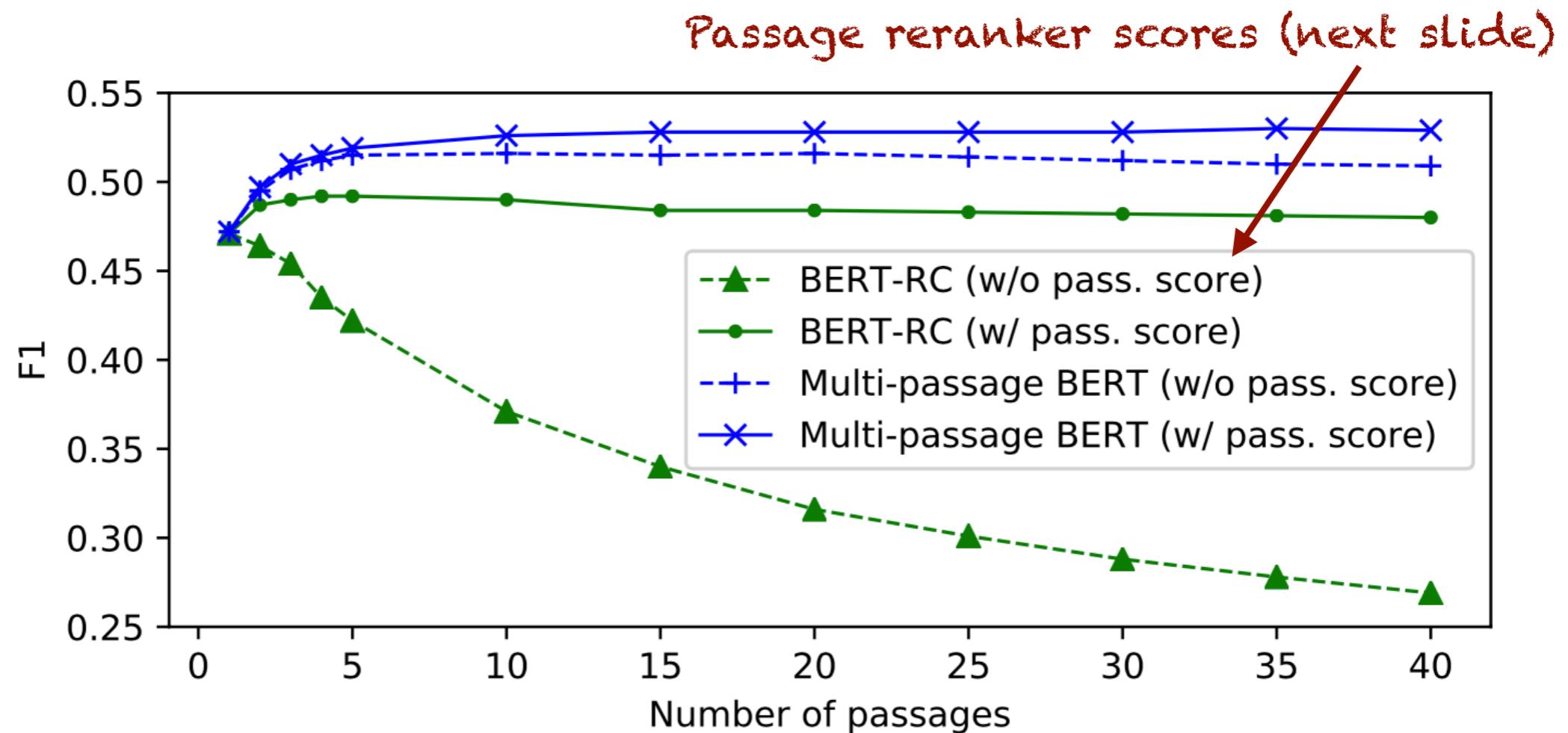
**Scoring from both
retriever and reader:**
$$S = (1 - \mu) \cdot S_{\text{Anserini}} + \mu \cdot S_{\text{BERT}}$$

This system is only evaluated on SQuAD:
27.1 (DrQA, SQuAD only) → 38.6

Multi-passage training

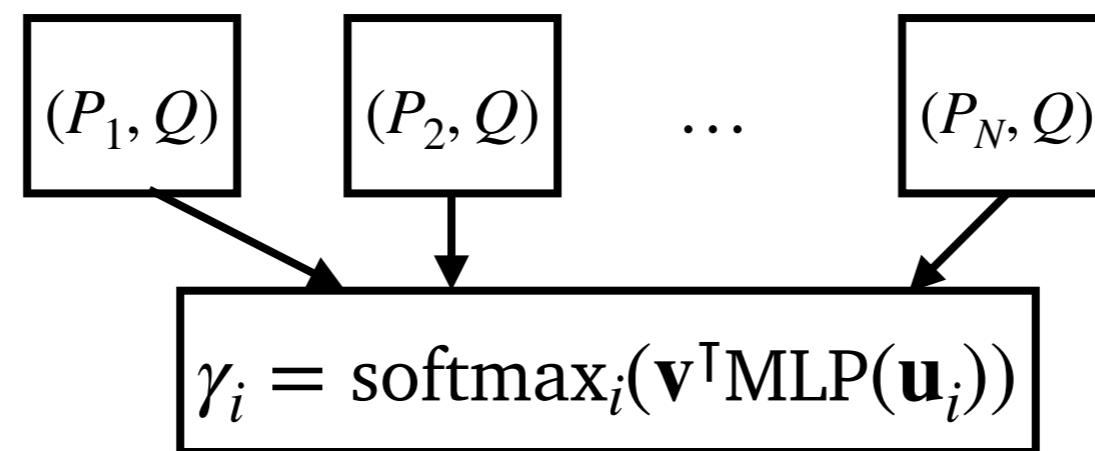
[Clark and Gardner, 2018; Wang et al., 2019]

Shared normalization: process passages independently, but compute the span probability across spans in all passages in every mini-batch



Training a passage re-ranker [Wang et al., 2018]

- Training a “deep” re-ranker model on retrieved passages can help further identify the relevance of the passages.



u_i is a representation of (P_i, Q)

- This reranker can be easily trained using **distant supervision**: whether the passage contains the answer or not.
- A better solution is to use training signal from the reader (next slide).

Wang et al., 2018. R³: Reinforced Ranker-Reader for Open-Domain Question Answering

Lin et al., 2018. Denoising Distantly Supervised Open-Domain Question Answering

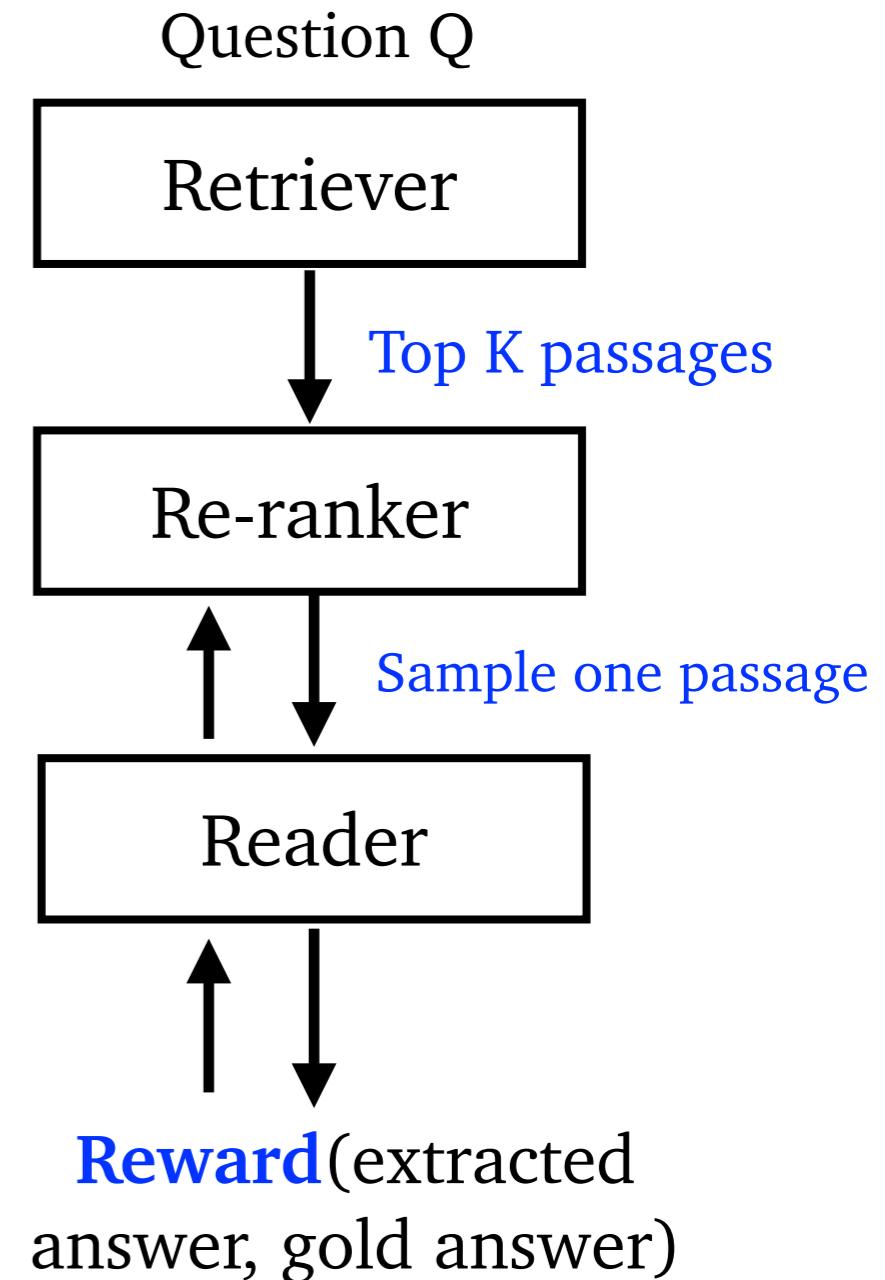
Wang et al., 2019. Multi-passage BERT: A Globally Normalized BERT Model for Open-domain Question Answering

Reinforced ranker-reader

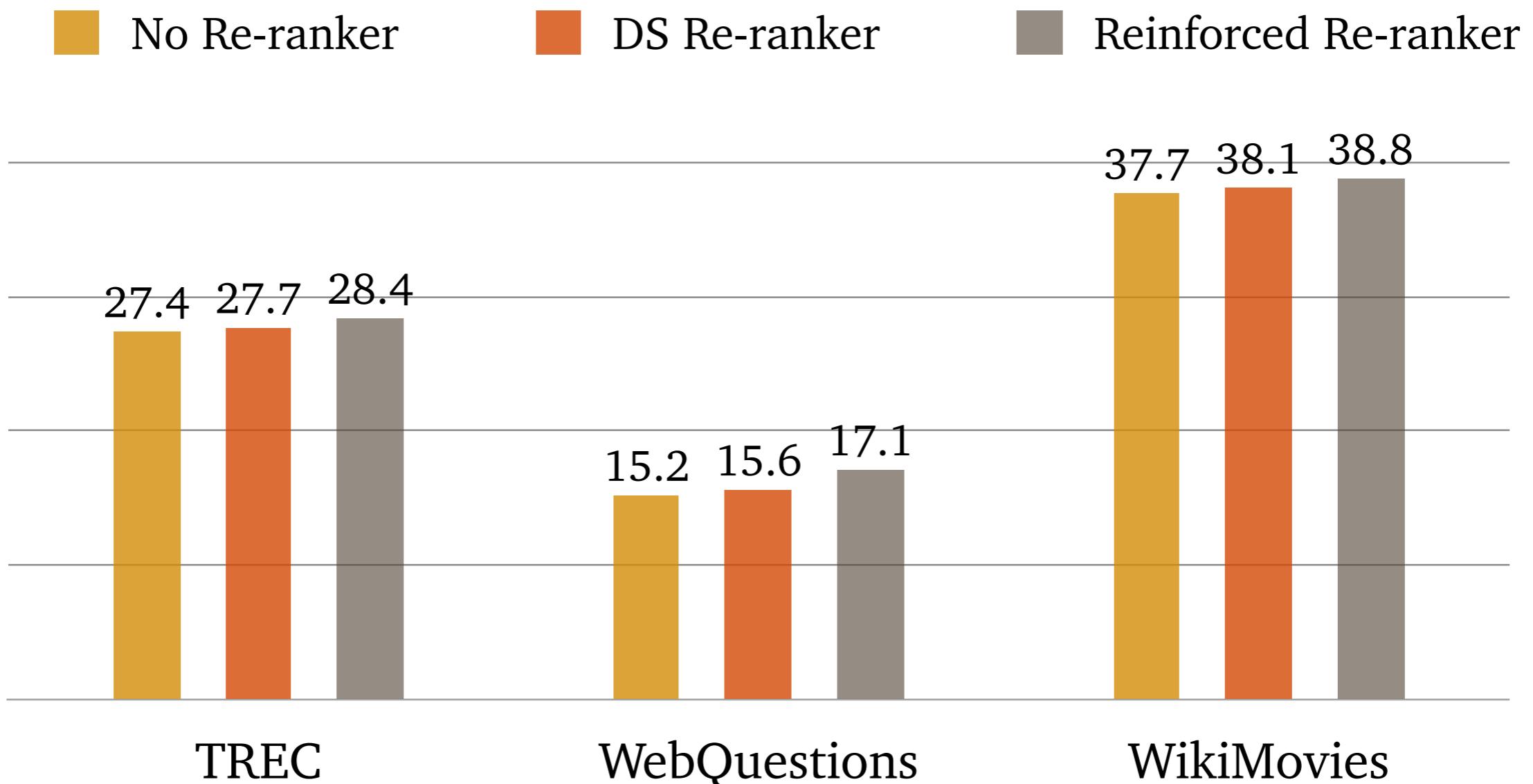
Algorithm 1 Reinforced Ranker-Reader (R^3)

- 1: **Input:** a^g , q , passages from IR
 - 2: **Output:** Θ
 - 3: **Initialize:** $\Theta \leftarrow$ pre-trained Θ with a baseline method⁶
 - 4: **for** each q in dataset **do**
 - 5: For question q , sample K passages from the top N passages retrieved by IR model for training.⁷
 - 6: Randomly sample a positive passage $\tau \sim \pi(\tau|q)$
 - 7: Extract the answer a^{rc} through RC model
 - 8: Get reward r according to $R(a^g, a^{rc}|\tau)$.
 - 9: Updating Ranker (ranking model) through policy gradient $r \frac{\partial}{\partial \Theta} \log(\pi(\tau|q))$
 - 10: Updating Reader (RC model) through supervised gradient $\frac{\partial}{\partial \Theta} L(a^g|\tau, q)$
 - 11: **end for**
-

$$\begin{cases} 2, & \text{if } a^g == a^{rc} \\ f1(a^g, a^{rc}), & \text{else if } a^g \cap a^{rc} \neq \emptyset \\ -1, & \text{else} \end{cases}$$



Experiments



Training an answer re-ranker [Wang et al., 2018]

If every passage returns a candidate answer, can we re-rank the answer candidates based on all their evidence?

Question2: Which physicist , mathematician and astronomer discovered the first 4 moons of Jupiter

A1: Isaac Newton

P1: Sir Isaac Newton was an English physicist , mathematician , astronomer , natural philosopher , alchemist and theologian ...

P2: Sir Isaac Newton was an English mathematician, astronomer, and physicist who is widely recognized as one of the most influential scientists ...

Question2: Which physicist , mathematician and astronomer discovered the first 4 moons of Jupiter

A2: Galileo Galilei

P1: Galileo Galilei was an Italian physicist , mathematician , astronomer , and philosopher who played a major role in the Scientific Revolution .

P2: Galileo Galilei is credited with discovering the first four moons of Jupiter .

Training an answer re-ranker

If every passage returns a candidate answer, can we re-rank the answer candidates based on all their evidence?

- **Strength**-based re-ranker: if an answer candidate is supported by multiple pieces of evidence (with high confidence), the answer is more likely to be correct.
- **Coverage**-based re-ranker: one answer candidate is more likely to be answer if the union of its evidence covers most information in the question.

This works very well for Quasar-T, SearchQA, TriviaQA-unfiltered when there is enough redundancy: 35.3 → 42.3 on Quasar-T, 49.0 → 57.0 on SearchQA.

Not clear in the full Wikipedia setting yet.

How if one passage contains multiple mentions of the answer?

In distant-supervised examples, we don't know which span is the correct one.

Given

Q: Which composer did pianist Clara Wieck marry in 1840?
A: Robert Schumann

Retrieved
passage

Robert Schumann was a German composer and influential music critics of the Romantic era. (...) Robert Schumann himself refers to it as “an affliction of the whole hand” (...) Robert Schumann is mentioned in a 1991 episode of Seinfeld “The Jacket” (...) Clara Schumann was a German musician and composer. Her husband was the composer Robert Schumann. (...) Brahms met Joachim in Hanover, made a very favorable impression on him, and got from him a letter of introduction to Robert Schumann .

Most previous work takes the first or a random mention as the gold label

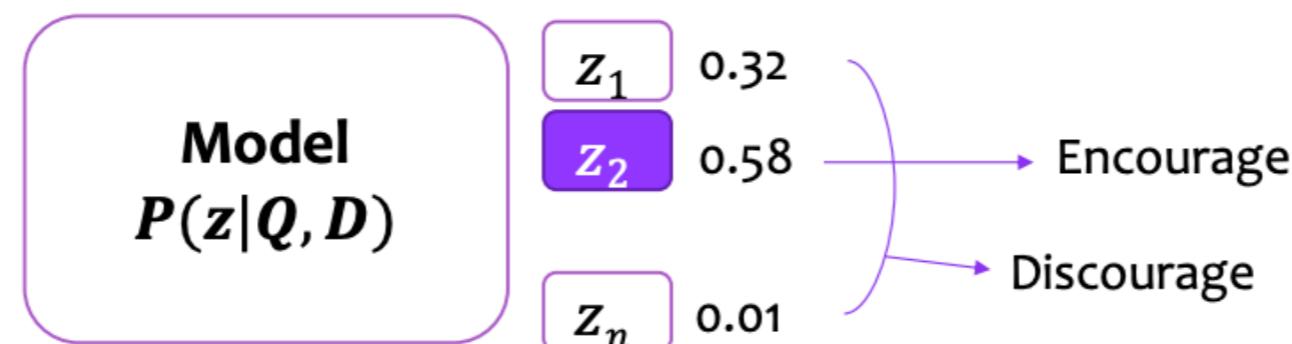
Hard EM Learning

Goal: train $P(z|Q, D)$

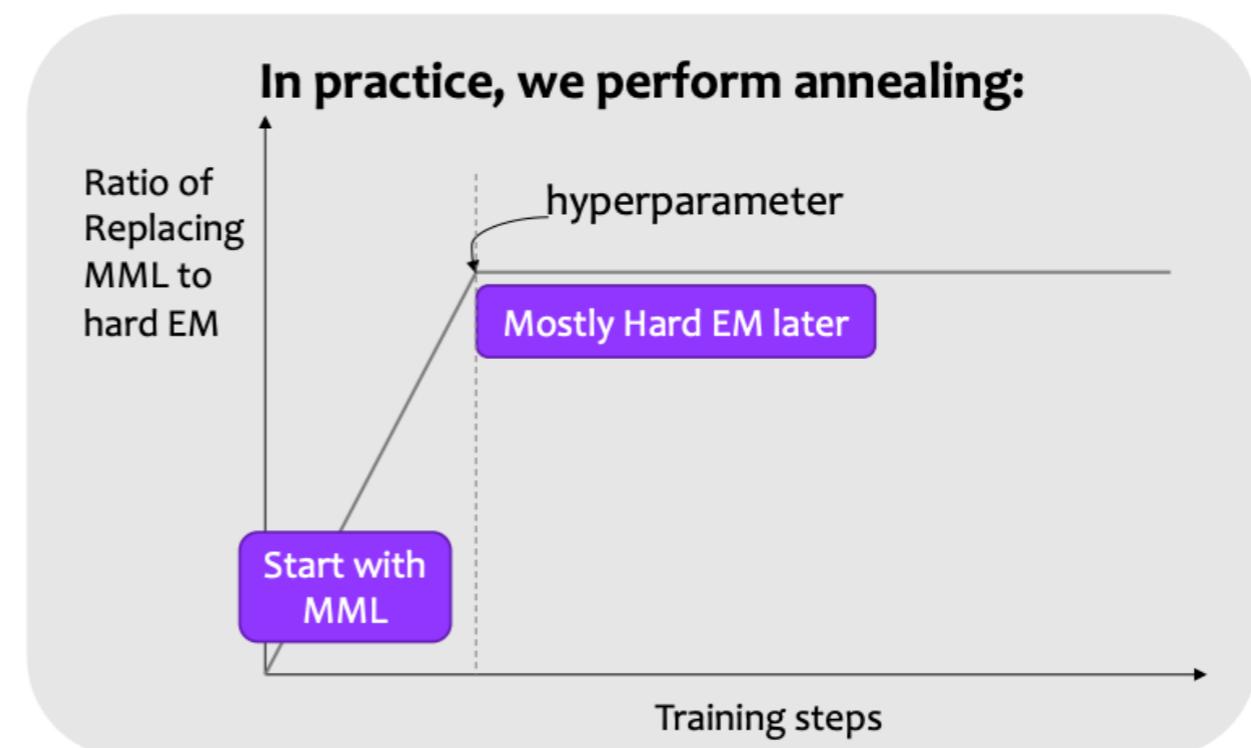
~~Groundtruth solution: $\bar{z} = \{z_1, z_2, \dots, z_n\}$~~ is a solution set executing the correct answer

Ours: Encourage the most likely solution

At each parameter update



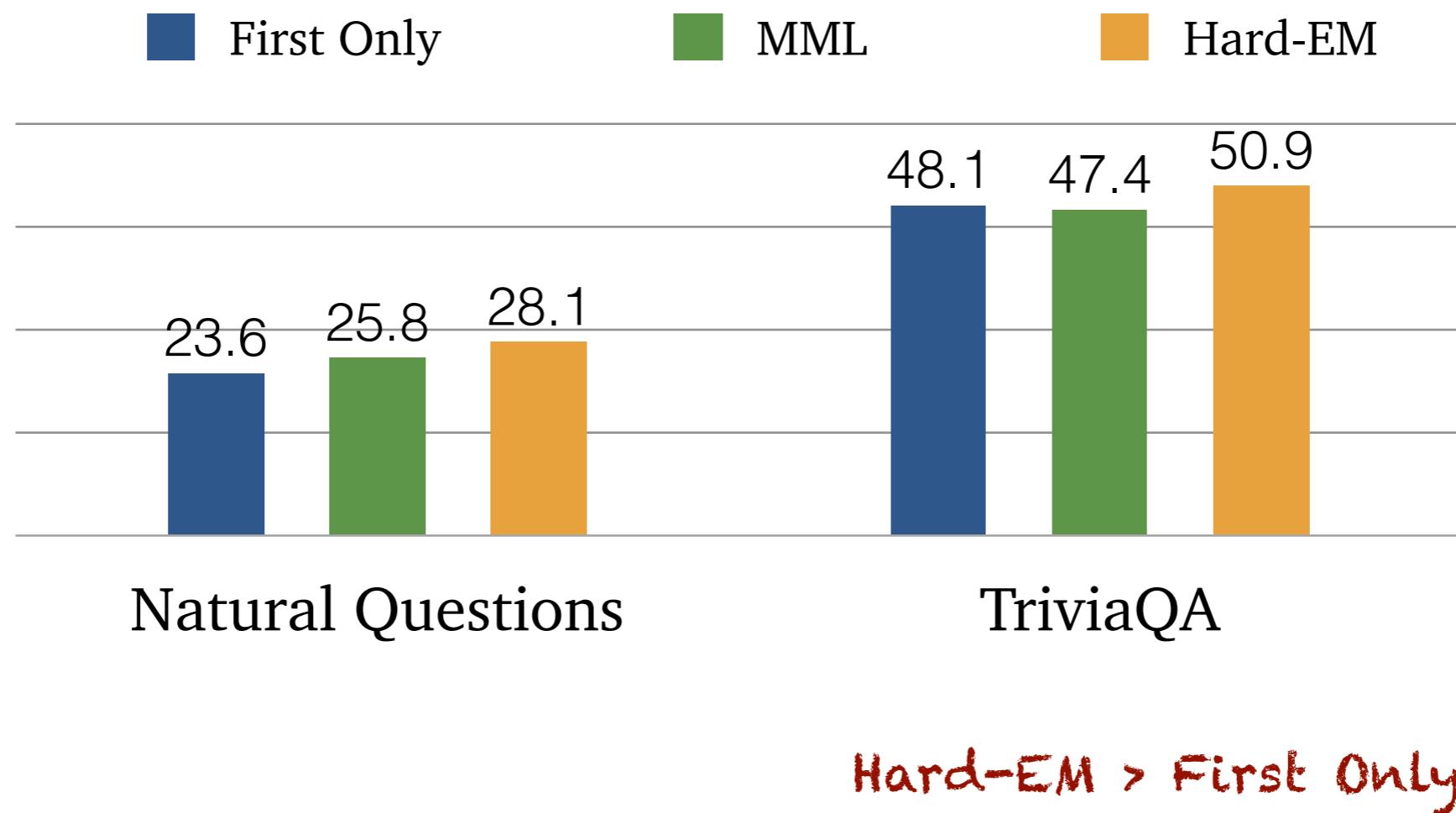
$$MML = \sum_{z \in Z} P(z | Q, D)$$



Experiments

First-only: always take the first span that appears in the passage

MML: marginalizes the likelihood of each possible answer span



Summary

- Document/passage retriever + neural reading comprehension largely simplified the open-domain QA pipeline
- Several ways to further improve performance:
 - Globally normalized multi-passage training
 - Passage re-ranker
 - Answer re-ranker
 - Improved training methods

Questions?

Please come back at 5pm PT!

