Open-domain Question Answering

Danqi Chen

Princeton University

@danqi_chen

Scott Wen-tau Yih

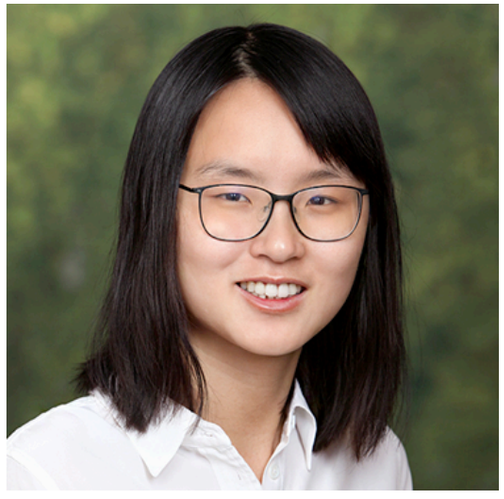
Facebook AI Research

@scottyih

https://github.com/danqi/acl2020-openqa-tutorial

July 5, 2020

Who we are

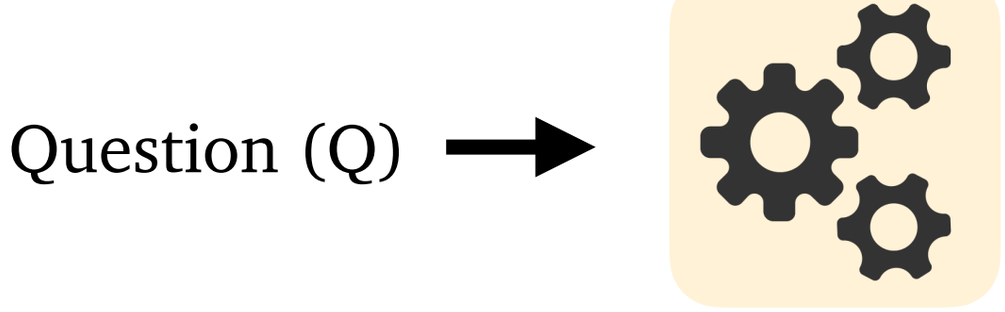


- Assistant prof at Princeton University since 2019 fall
- Working in QA since 2016
- PhD thesis on neural reading comprehension and question answering
- Author of DrQA, CoQA, RoBERTa, dense passage retriever (DPR), Stanford Attentive Reader

- Research scientist at Facebook AI Research
- Working in QA since 2013 (or 2001☺)
- Taught “Question Answering with Knowledge Base, Web and Beyond” in NAACL & SIGIR 2016
- Author of WikiQA, WebQSP, DPR, retrieval-augmented generation (RAG)

Open-domain QA

- Question answering = build computer systems that **automatically** answer questions posed by humans in a **natural language**
- Open-domain = deal with questions about nearly anything, usually rely on **general ontologies** and **world knowledge**



Where does the energy in a nuclear explosion come from?

high-speed nuclear reaction

Where is Einstein's house?

112 Mercer St, Princeton, NJ

How many papers were accepted by ACL 2020?

779 papers

Open-domain QA

Knowledge Bases

Tables

Category	Structure	Country	City	Height (metres)	Height (feet)
Mixed use	Burj Khalifa	United Arab Emirates	Dubai	829.8	2,722
Self-supporting tower	Tokyo Skytree	Japan	Tokyo	634	2,080
Mixed use	Shanghai Tower	China	Shanghai	632	2,073
Clock building	Abraj Al Bait Towers	Saudi Arabia	Mecca	601	1,972
Military structure	Large masts of INS Kattabomman	India	Tirunelveli	471	1,545
Mast radiator	Lualualei VLF transmitter	United States	Lualualei, Hawaii	458	1,503
Twin towers	Petronas Twin Towers	Malaysia	Kuala Lumpur	452	1,482
Residential	432 Park Avenue	United States	New York	425.5	1,396
Chimney	Ekibastuz GRES-2 Power Station	Kazakhstan	Ekibastuz	419.7	1,377
Radar	Dimona Radar Facility	Israel	Dimona	400	1,312
Lattice tower	Kiev TV Tower	Ukraine	Kiev	385	1,263
Electricity pylon	Zhoushan Island Overhead Powerline Tie	China	Zhoushan	370	1,214

Web Documents & Wikipedia

Structured

Semi-structured

Unstructured

- This tutorial mostly focuses on open domain textual QA
- In Part 6, we will discuss hybrid systems using both KBs and text

Open-domain QA

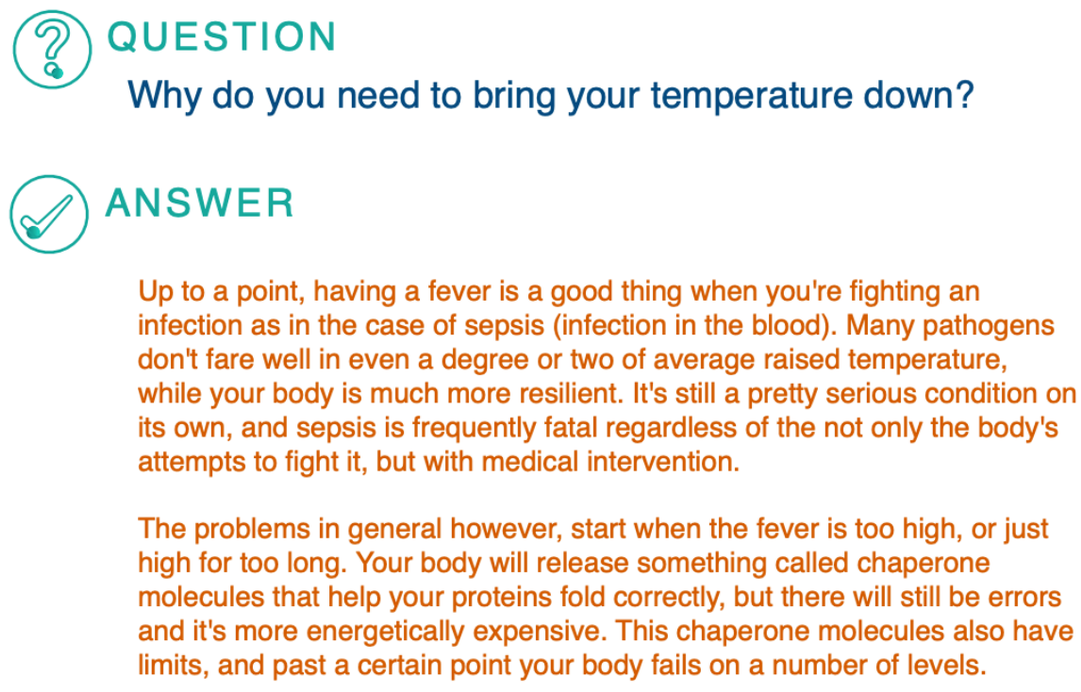
We mostly focus on **factoid question answering**:

- Require systems to return a *short* and *concise* answer to these questions
- Focus more on *retrieval* and *reading instead of answer generation*
- In contrast to other QA problems: community question answering, non-factoid question answering

How do Jellyfish function without brains or nervous systems?

Why can't humans see in the dark?

How to protect yourself from COVID-19?



Prager, 2007. Open-Domain Question Answering

Fan et al., 2019. ELI5: Long Form Question Answering

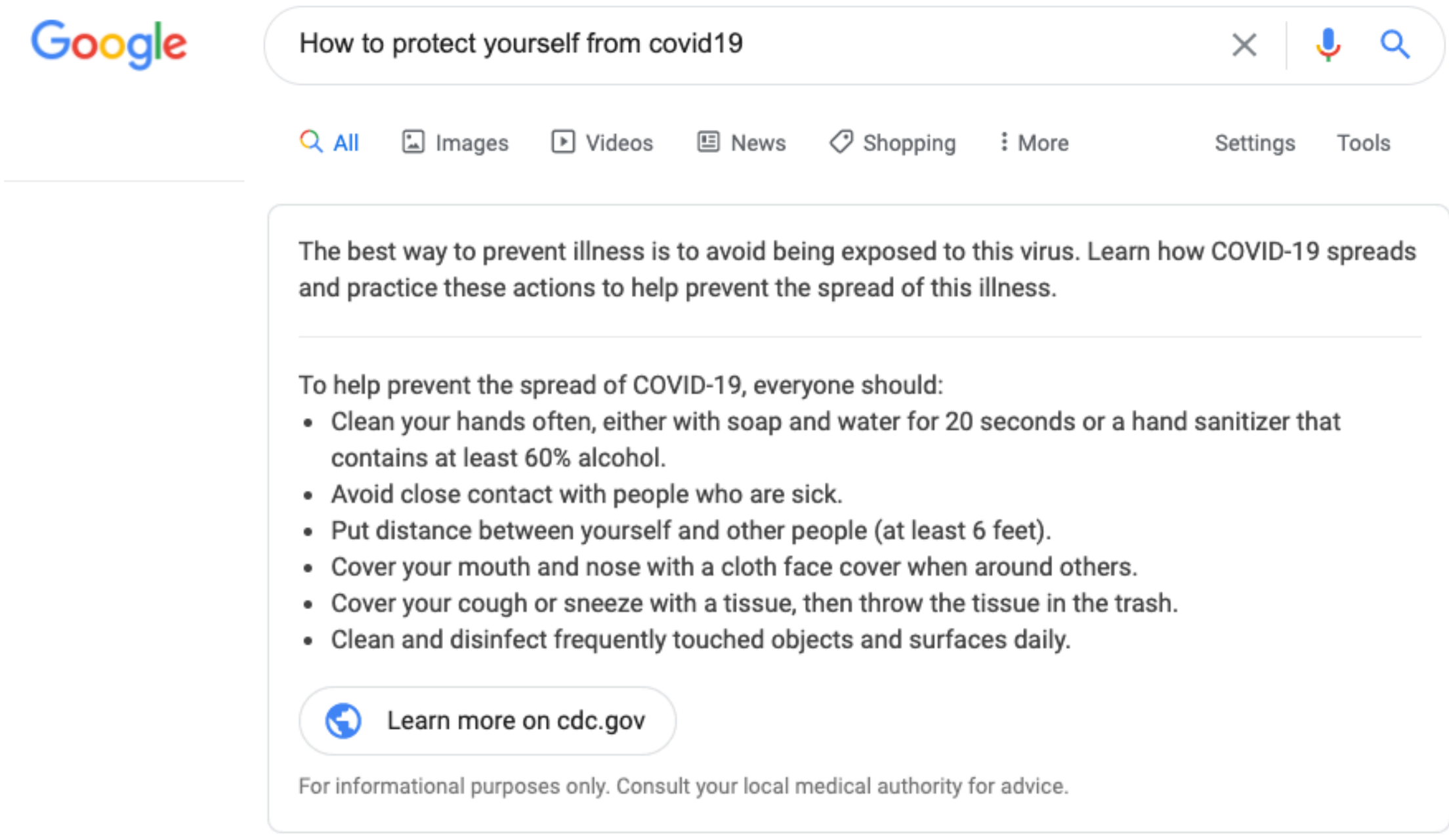
Search engines: from keyword matching to question answering

Search needs a shake-up

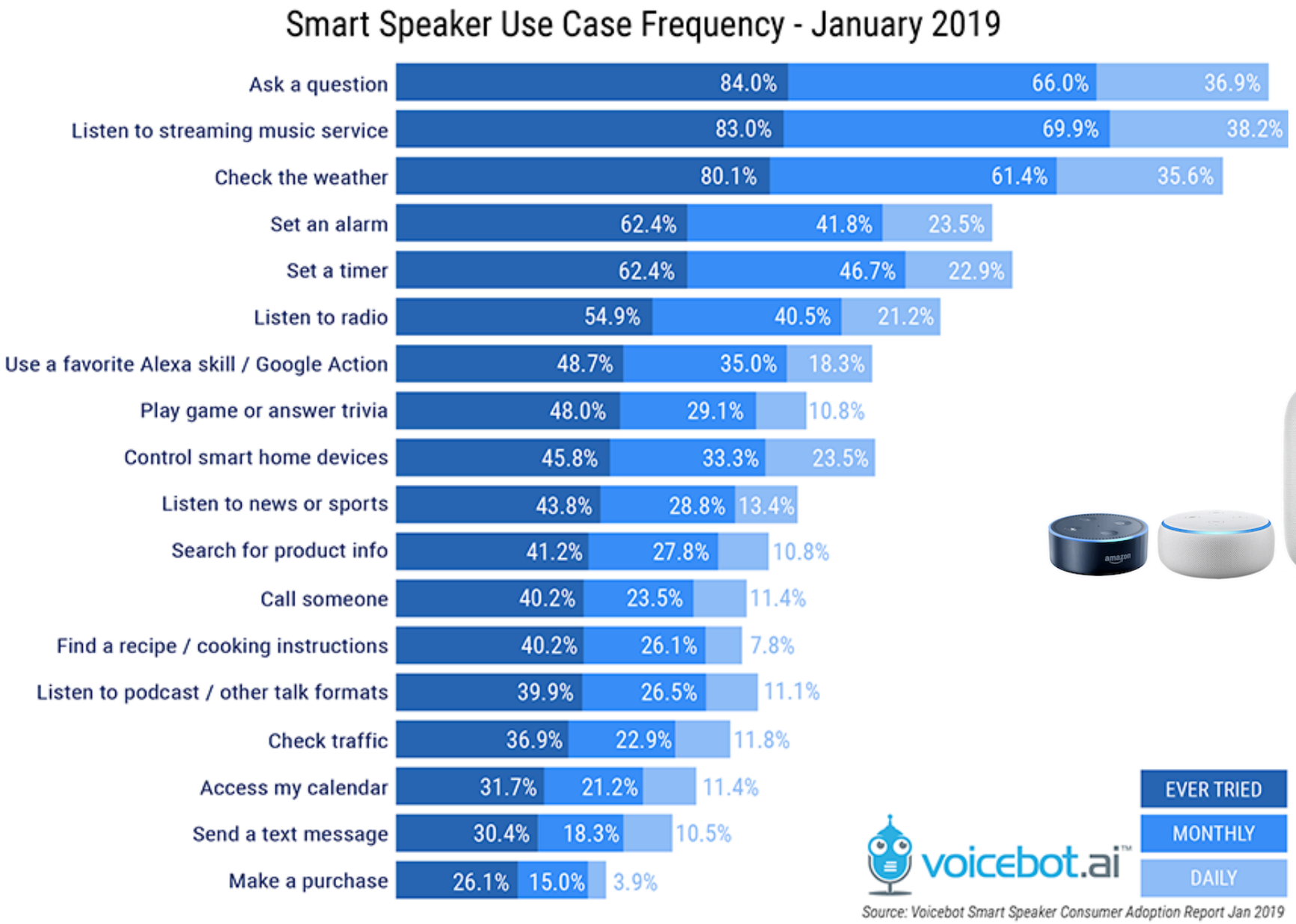
Etzioni, 2011. Search needs a shake-up

Search engines: from keyword matching to question answering

Search engines: from keyword matching to question answering



People ask lots of questions on digital personal assistants



https://voicebot.ai/2019/03/12/smart-speaker-owners-agree-that-questions-music-and-weather-are-killer-apps-what-comes-next/

Why give this tutorial today?

NAACL 2001

Open-Domain Textual Question Answering

Sanda Harabagiu and Dan Moldovan
Department of Computer Science and Engineering, Southern Methodist University

Brief Description

Question Answering (QA) is a fast growing area of research and commercial interest. The problem of QA is to find answers to open-domain questions by searching a large collection of documents. Unlike Internet search engines, QA systems provide short, relevant answers to questions. The recent explosion of information available on the World Wide Web makes question answering a compelling framework for finding information that closely matches user needs. The success of QA services, like AskJeeves serves as proof of the popularity of this technique. Due to the fact that both questions and answers are expressed in natural language, QA methodologies deal with language ambiguities and incorporate NLP techniques. Several current NLP-based technologies are able to provide the framework that approximates the complex problem of answering questions from large collections of texts. Ideal QA systems should have good dialog understanding, rich knowledge bases and quality text mining methods. They will certainly incorporate common sense reasoning methods and use good approximations of world knowledge. Until we have these more advanced tools, we can approximate QA with NLP enhancements of IR and IE techniques. The tutorial presents the recent results in QA research and system implementations.

TREC QA 1999-2001 competitions and participant systems

NAACL 2012

ACL Anthology FAQ Corrections Submissions

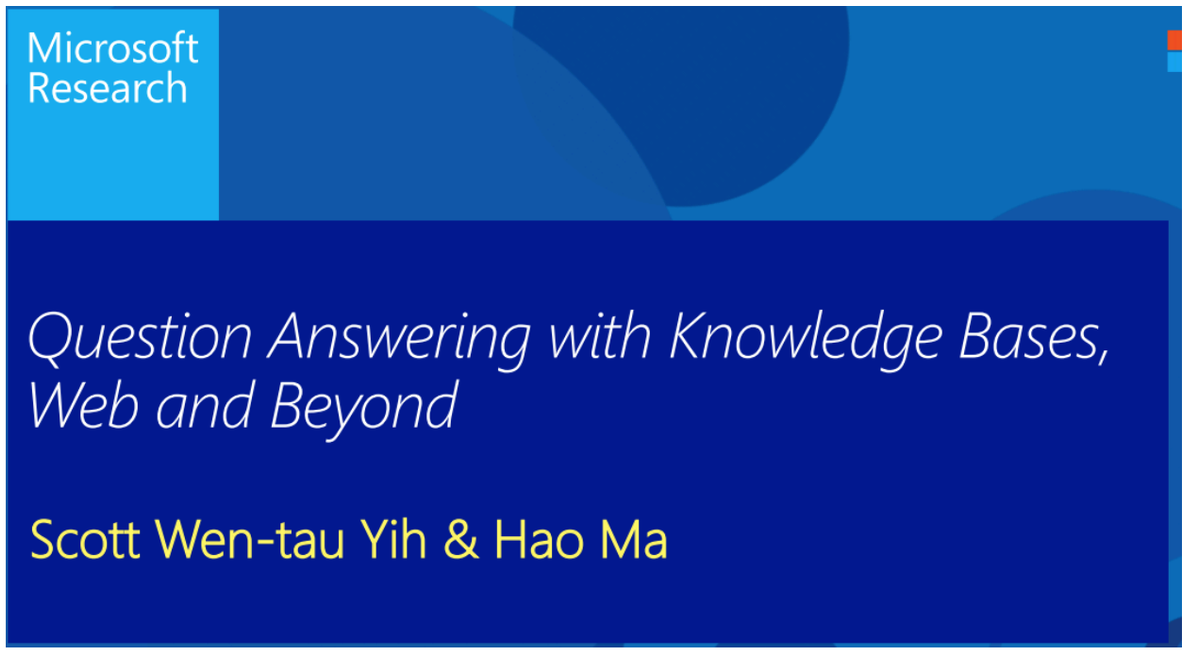
Natural Language Processing in Watson

Alfio M. Gliozzo, Aditya Kalyanpur, James Fan

IBM Watson system on Jeopardy questions

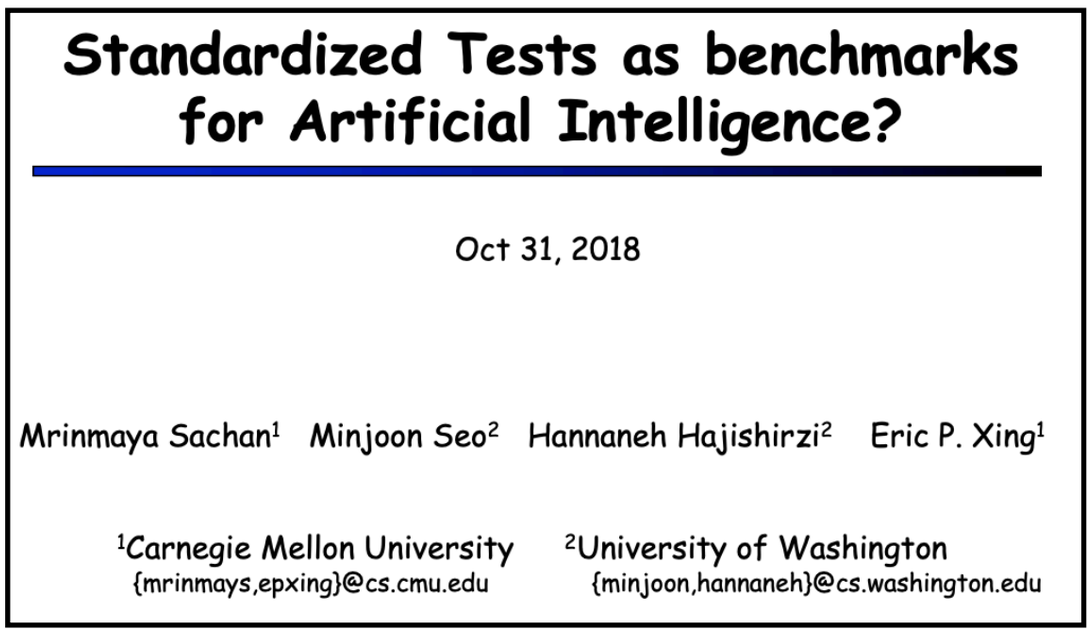
Why give this tutorial today?

NAACL 2016



QA with KBs and tables

EMNLP 2018

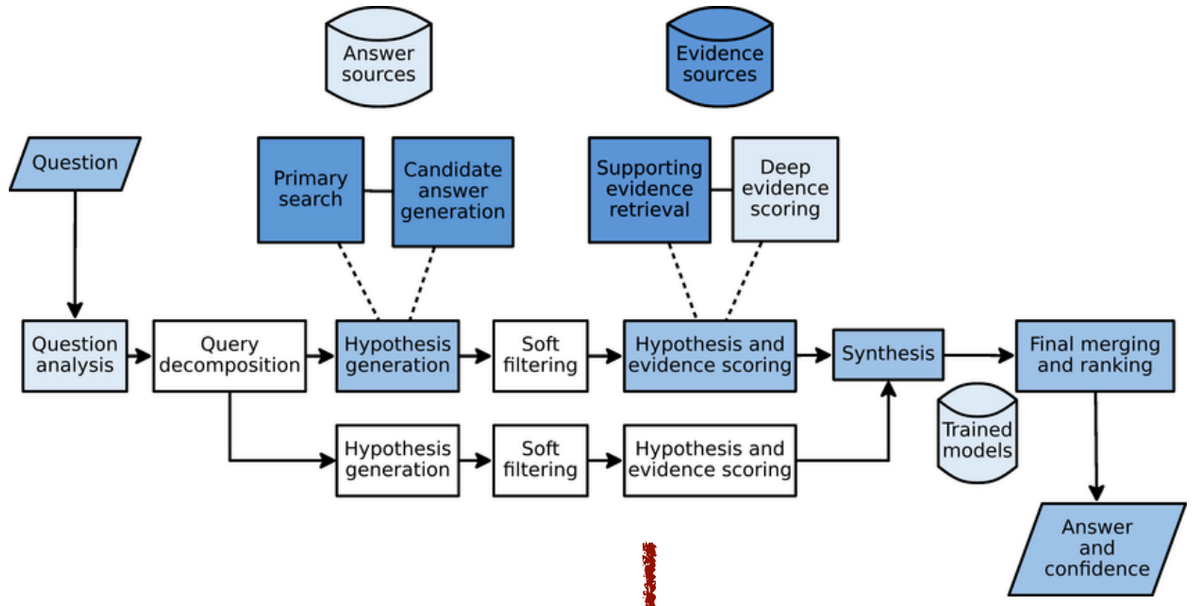


QA as standardized tests: reading comprehension and closed domain problems (science, geometry, algebra word problems etc)

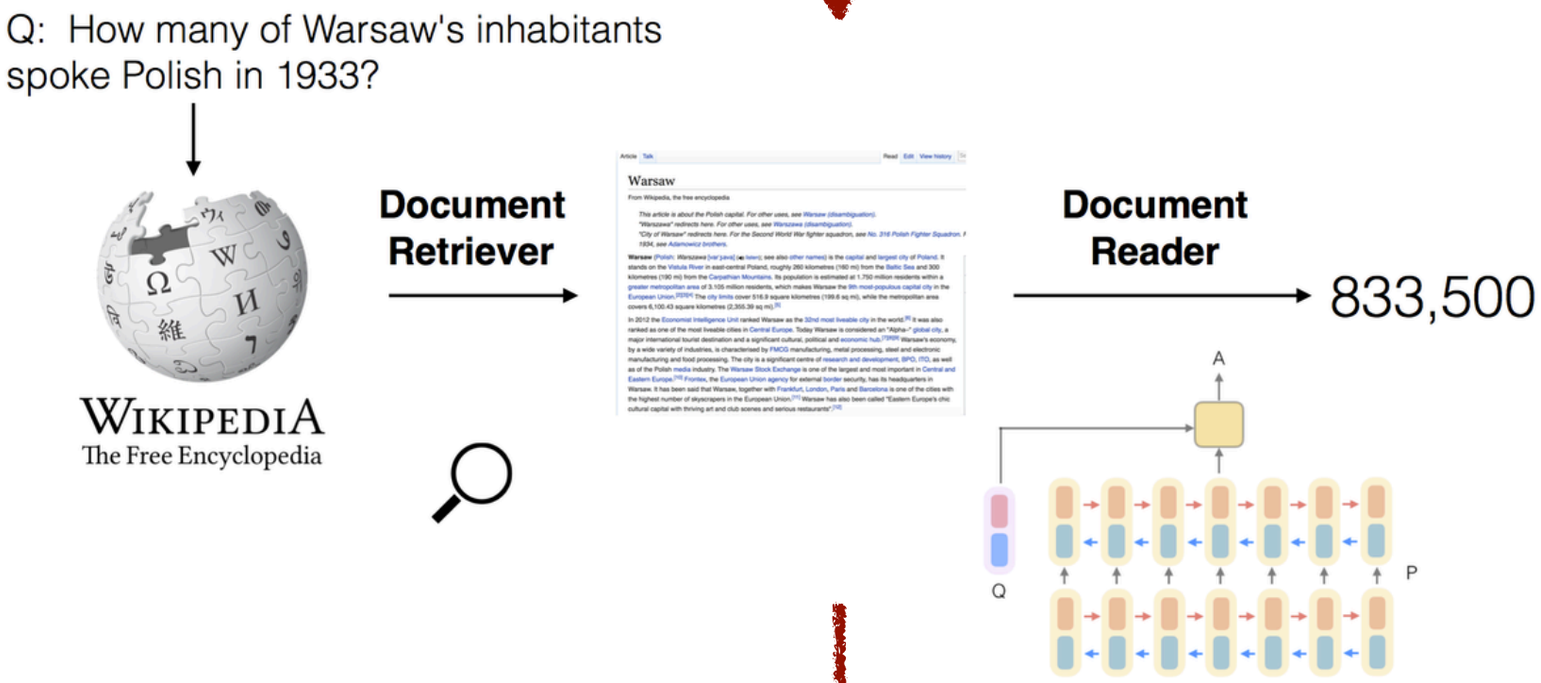
This tutorial: open-domain question answering over a large collection of unstructured documents; mostly the new generation and paradigm of the NLP technologies (2017-2020)

Why give this tutorial today?

Classical QA pipeline

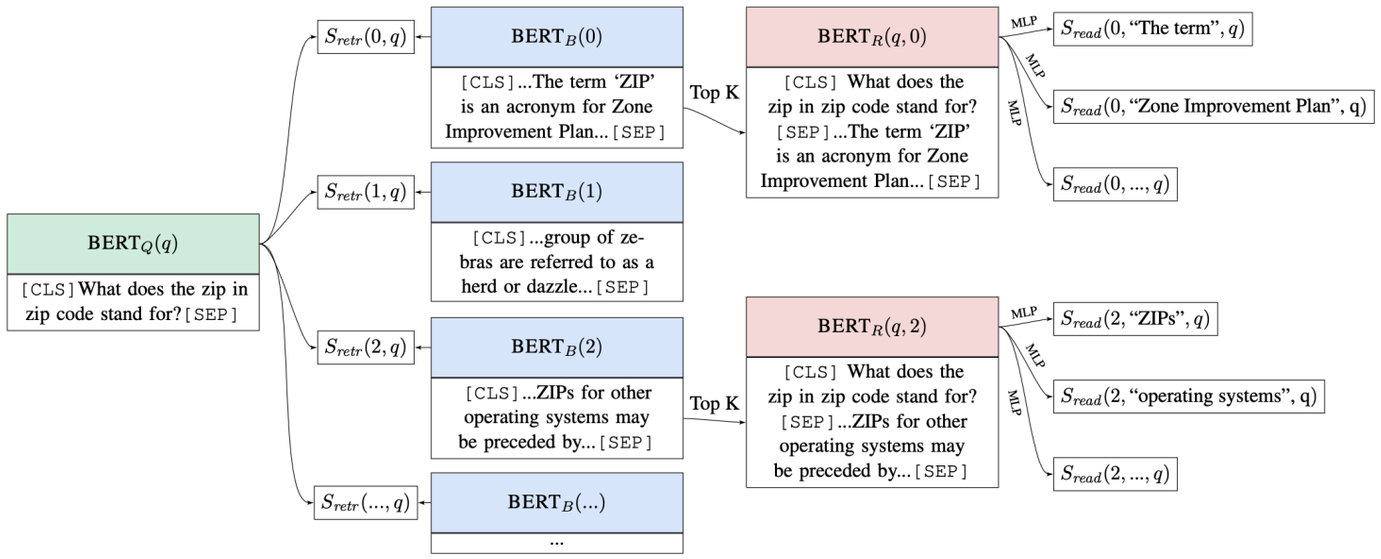


Two-stage Retriever-reader approaches

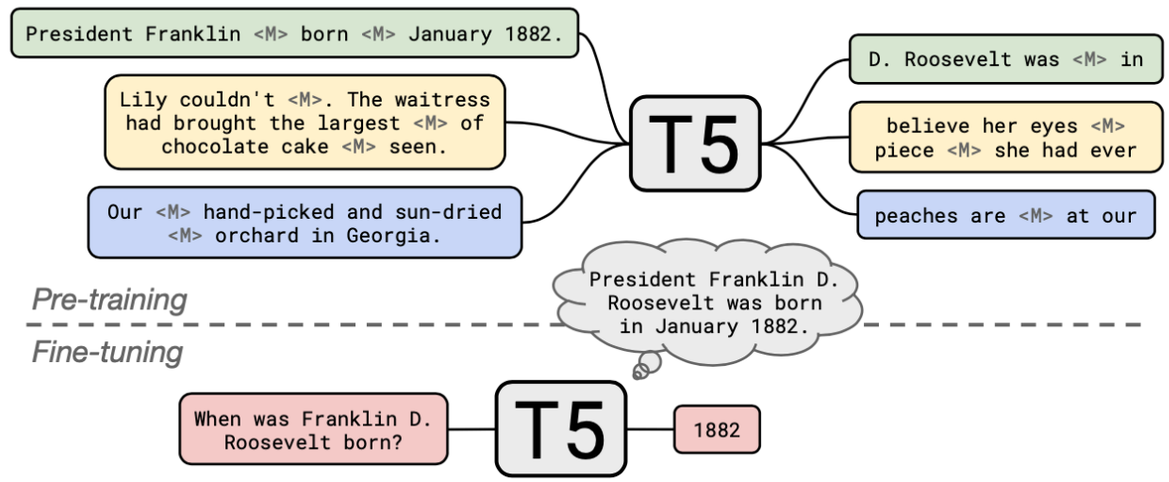


Why give this tutorial today?

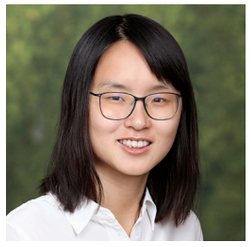
End-to-end learning



Retrieval-free models

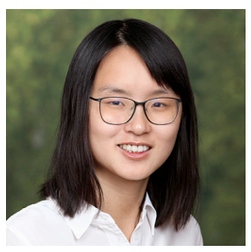


Outline



- Part 1. Introduction <- We are here!

- Part 2. A history of open-domain QA
- Part 3. Datasets & evaluation



- Part 4. Two-stage retriever-reader approaches

30min coffee break

- Part 5. Dense retriever and end-to-end training
- Part 6. Retrieval-free approaches

- Part 7. Open-domain QA using KBs and text
- Part 8. Open problems and future directions

Part II

A Brief History of Open-Domain Question Answering

Natural language understanding: early QA systems

- Question-answering machine [Simmons, 1965]
  - General-purpose language processors that communicate with users in natural language (e.g., English)
  - Deal with statements and/or questions

http://csunplugged.org/turing-test

Simmons, 1965. Answering English Questions by Computer: a Survey

Categories of (early) QA systems

- **List-structured database systems**
  - Organizing knowledge (e.g., kinship) in list DB
- **Graphic database systems**
  - Map text and graphic data (e.g., pictures, diagrams) to the same logical representations
- **Text-based systems**
  - Matching questions and text in a corpus to find answers
- **Logical inference systems**
  - Textual entailment, answering science text book questions & algebra word problems

Simmons, 1965. Answering English Questions by Computer: a Survey

Baseball [Green et al., 1963]

Q: How many games did the Yankees play in July?

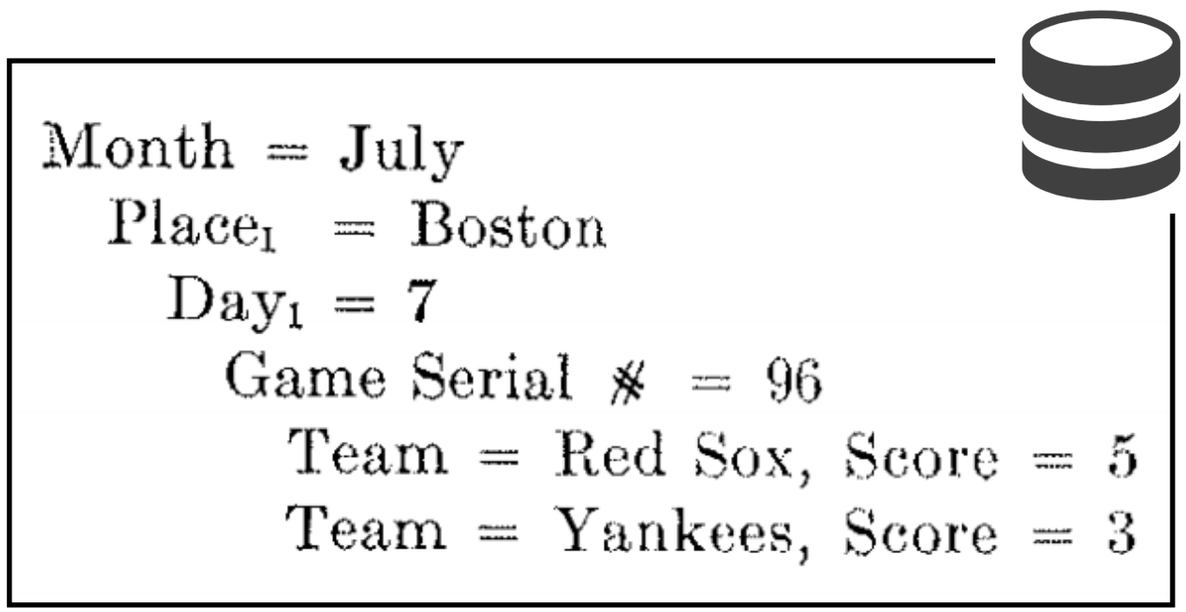
- Step 1: Simple dictionary-based syntactic analysis

  (How many games) did (the Yankees) play (in (July))?

- Step 2: Semantic analysis that builds "spec"

  "Who" ⟶ ("team" = ?)

  Conditions (e.g., "winning", "how many") ⟶ routines



```
Month = July
   Place1  = Boston
     Day1  = 7
        Game Serial #  = 96
           Team = Red Sox, Score = 5
           Team = Yankees, Score = 3
```

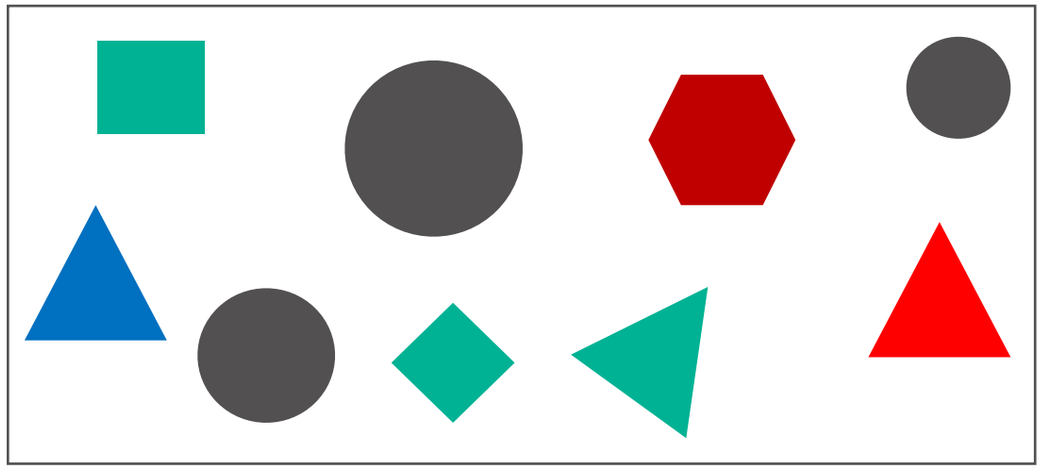
- Step 3: Execution

Green et al.,1963. Baseball: An Automatic Question-Answerer

The Picture Language Machine [Krisch, 1964]

Is the statement true?

"All circles are black circles."



Both pictures and text are translated into logical language:

- Circle(a), Black(a), Bigger(a, b), Between(a, b, c)
- $(\forall x)[\text{Circle}(x) \supset (\exists y)[\text{Circle}(y) \wedge \text{Black}(y) \wedge (x = y)]]$

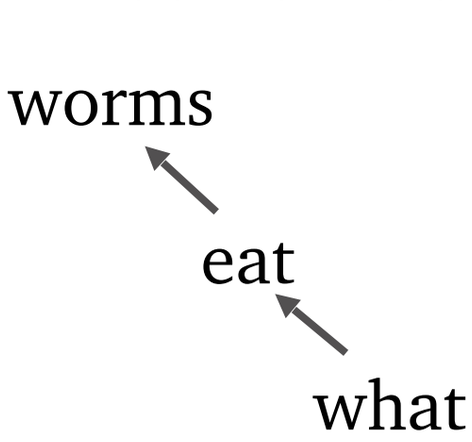
Krisch, 1964. Computer Interpretation of English Text and Picture Patterns

Protosynthex [Simmons+ 1963]

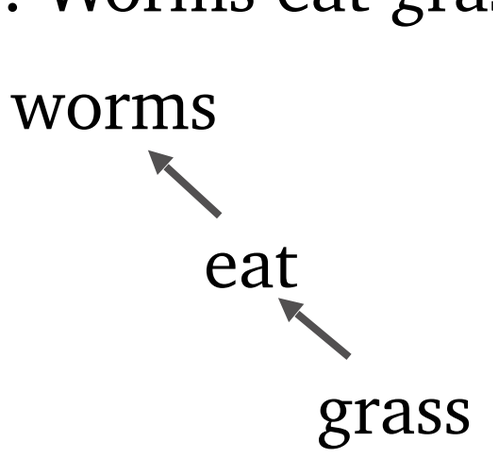
Answer questions from an Encyclopedia

- Matching questions & text in dependency logic [Hays, 1962]

Q: What do worms eat?

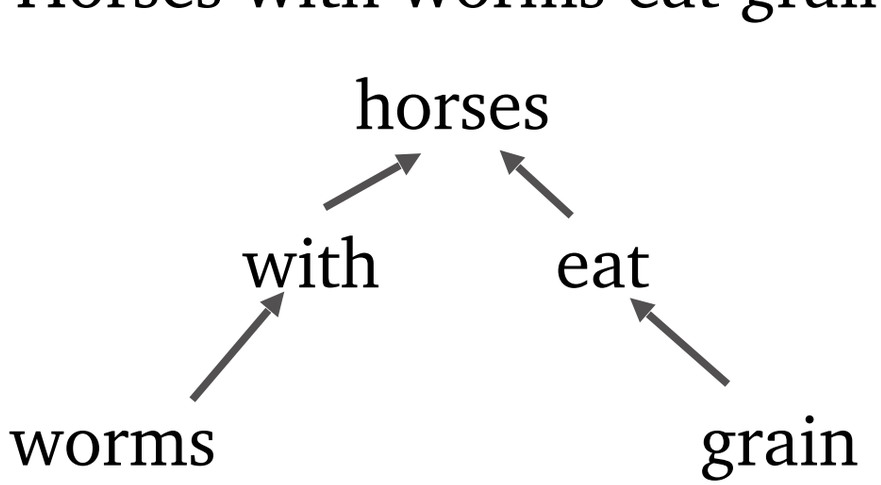


A1: Worms eat grass



Complete Agreement

A2: Horses with worms eat grain



Partial Agreement

Student [Bobrow, 1964]

The first algebra problem solver

- Translate a set of English statements to mathematical equations

- Step 1: Simplify text and annotate operators

  “twice” →“two times”, “the square of” →“square”

  Tag operators like “plus”, “percent”, “times”

- Step 2: Heuristics to break problem into simple sentences

- Step 3: Mapping sentences to equations

  Rules based on dictionary of words and numbers

Bobrow, 1964. Natural Language Input for a Computer Problem Solving System

Lessons from old QA systems

Limited success

- Small & limited domains and scopes
  - Often work only on well-controlled, specialized subset of English
- Not data-driven (e.g., machine learning approaches)
  - Mostly rule-based, potentially brittle
- Lacks rigorous evaluation

Open questions [Simmons, 1965]

- Meaning representation & the need of formal languages
- Syntactic and semantic disambiguation
- Combine partial answers from various sources

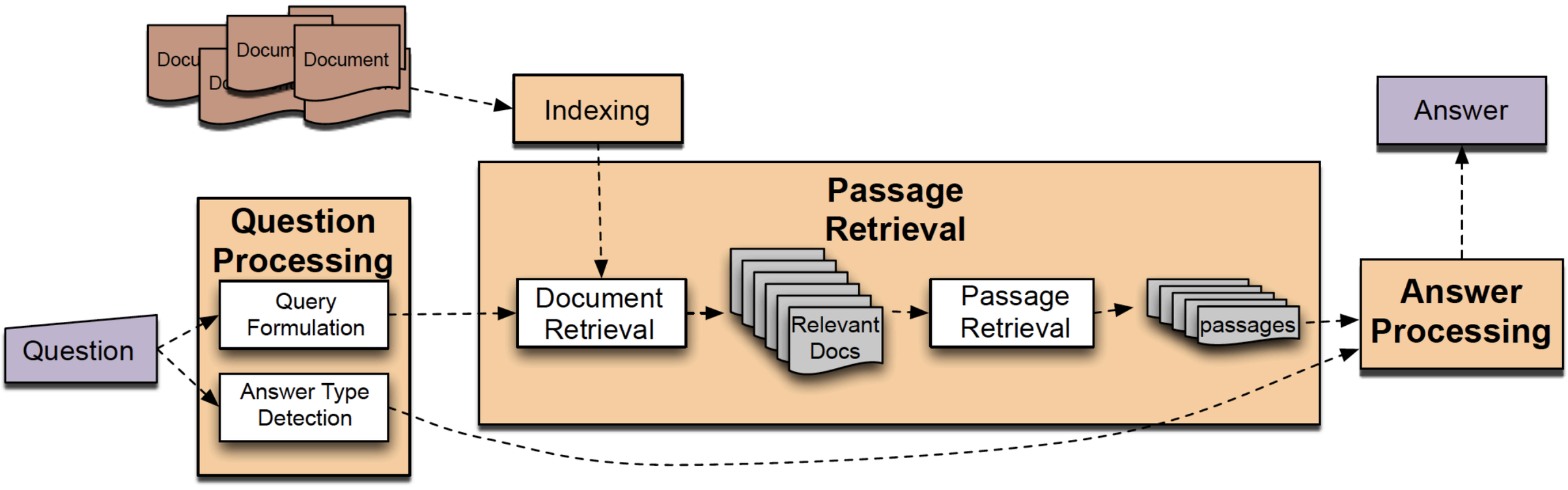
Simmons, 1965. Answering English Questions by Computer: a Survey

Text Retrieval Conference (TREC)
QA Tracks (1999 - 2007)

- Originating from the IR community as the next version of search
  - Relevant documents → short answer with support

- Shared tasks & competitions
  - Corpus: newswire (AP, WSJ, LA Times, etc.); 979k articles, 3GB
  - Question sources: Excite, Encarta, MSNSearch, AskJeeves
  - Test set: 500 questions
  - Human judges decide the correctness of the answers from QA systems

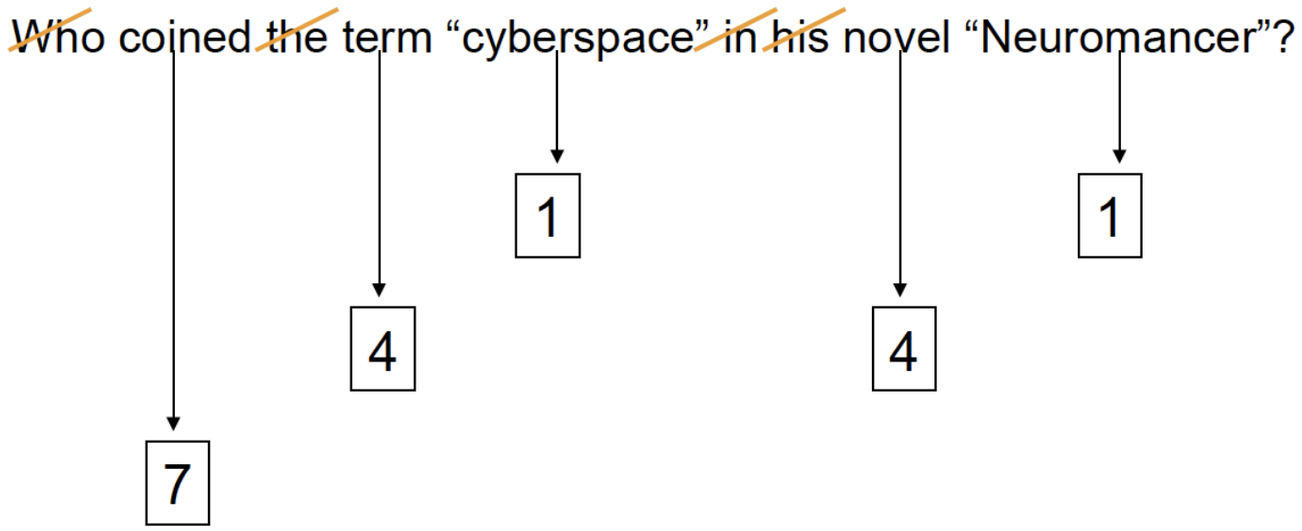
Typical pipeline of TREC-QA systems

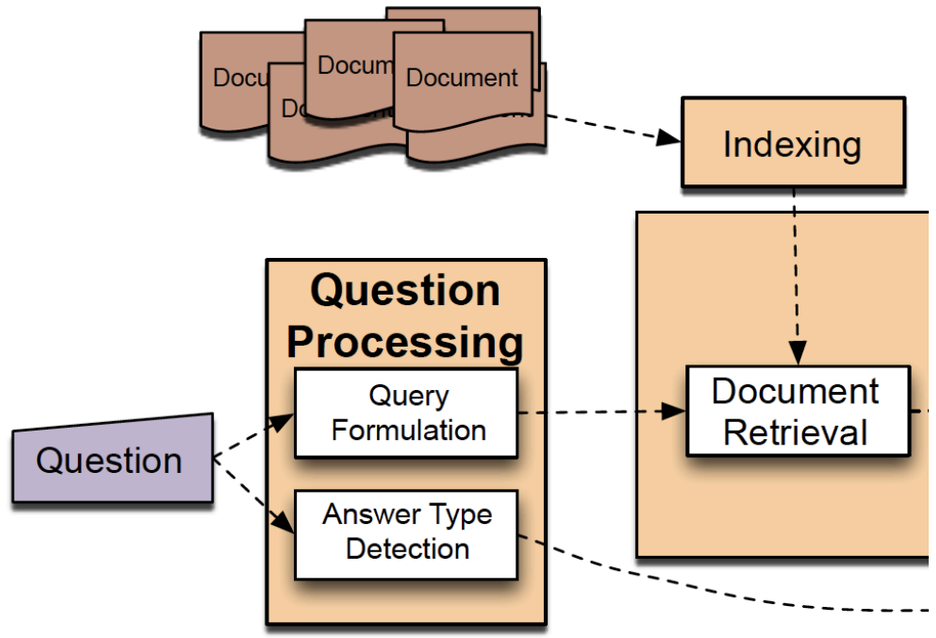


Jurafsky and Martin. Speech and Language Processing. 3rd edition.

Query formulation

Choosing keywords from the question:



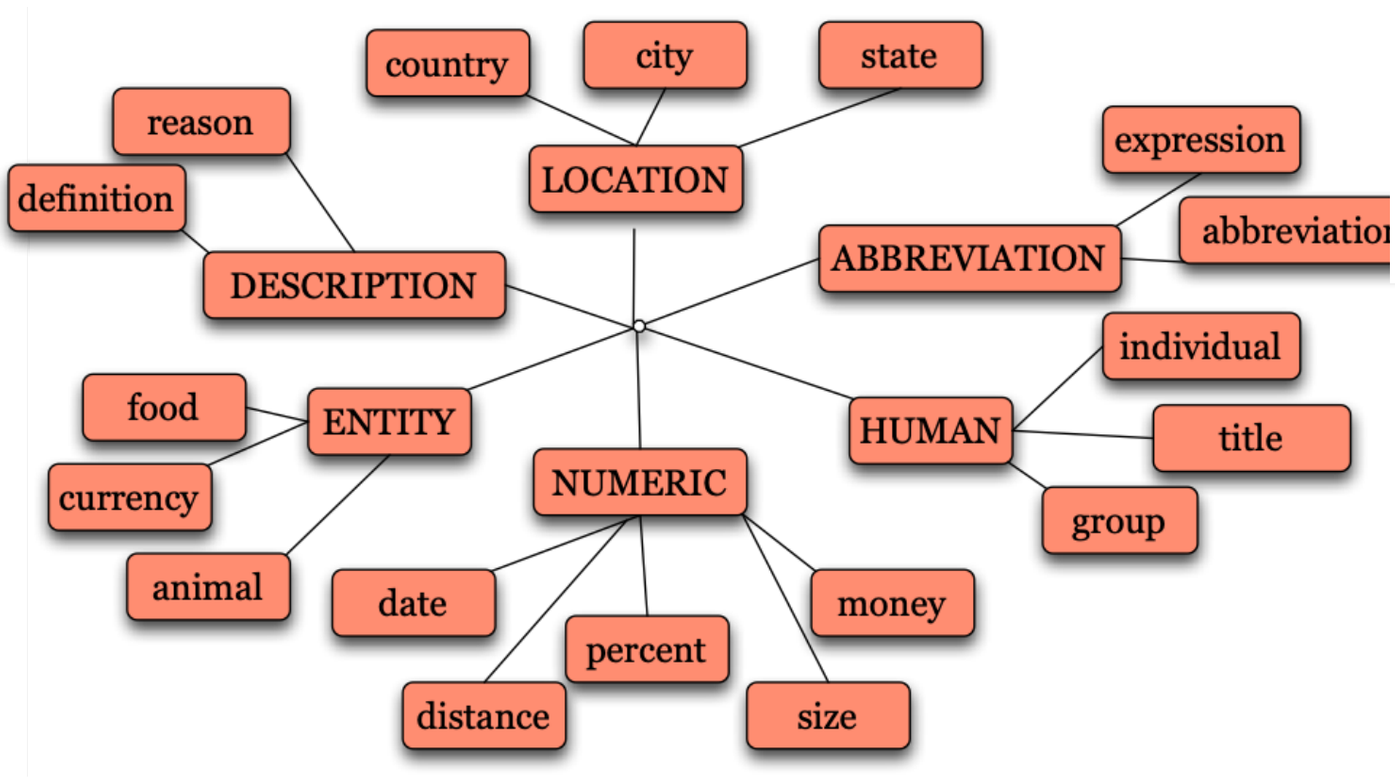


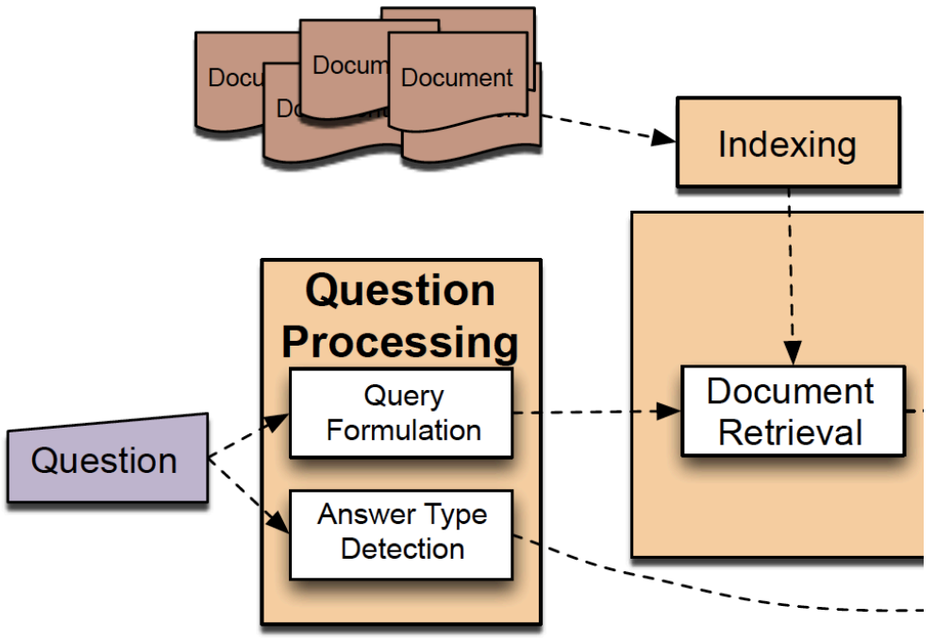
Example from Mihai Surdeanu

Answer type detection

Answer type taxonomy [Li & Roth, 2002]

- 6 coarse classes, 50 fine classes





Li and Roth, 2002. Learning Question Classifiers

Example questions

Answer type: entity

Type	Questions
ENTY:animal	What was the first domesticated bird?
ENTY:letter	What's the second-most-used vowel in English?
ENTY:food	What rum is so "mixable" it is a one-brand bar?
ENTY:color	What's the only color Johnny Cash wears on stage?
ENTY:product	Which two products use a tiger as their symbol?
ENTY:religion	In what religion was Isis the nature goddess?
ENTY:other	What does a spermologer collect?

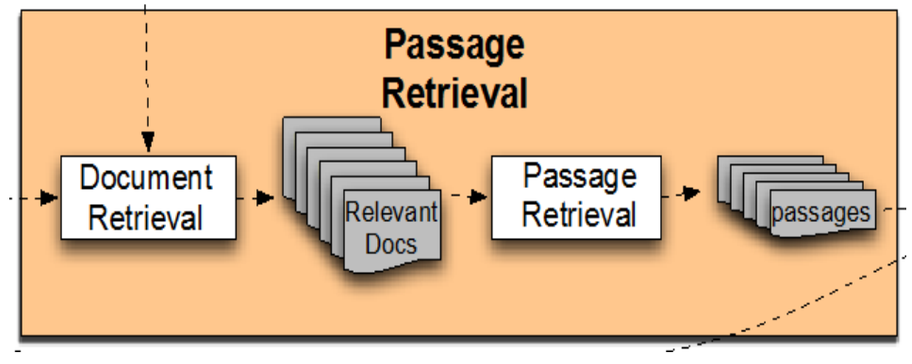
Example questions

Answer type: human, location, numeric

Type	Questions
HUM:ind	What crooner joined The Andrews Sisters for Pistol Packin Mama?
HUM:gr	Who has won the most Super Bowls?
LOC:city	What city did the Flintstones live in?
LOC:other	What stadium do the Miami Dolphins play their home games in?
LOC:state	What U.S. state lived under six flags?
NUM:date	When was Ozzy Osbourne born?
NUM:count	How many people in the world speak French?

Passage retrieval

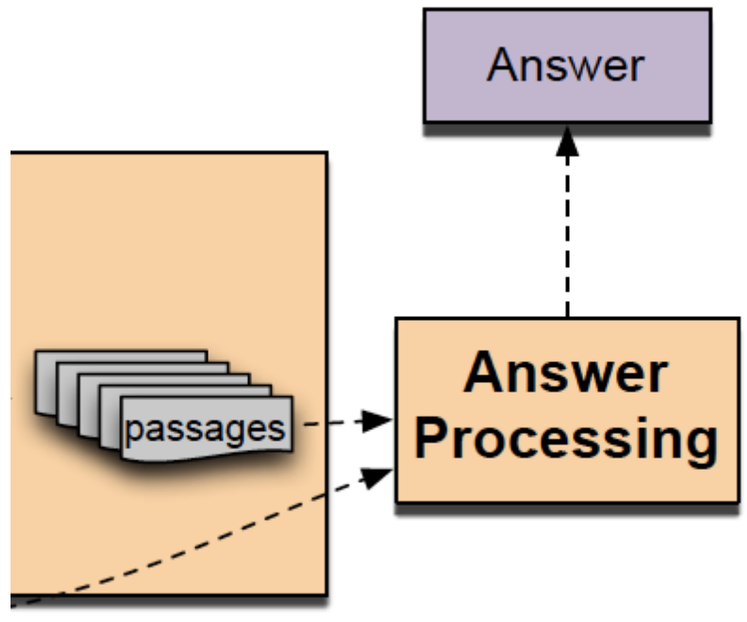
- Document retrieval via standard information retrieval methods
- Retrieved documents segmented into shorter units as paragraphs
  - Only a small chunk of text is assumed relevant
  - Answer extraction/processing is more computation intensive
- Passage ranking/selection
  - Linear ML models based on hand-crafted features
  - Example features
    - Number of Named Entities of the right type in passage
    - Number of query words in passage
    - Rank of the document containing passage



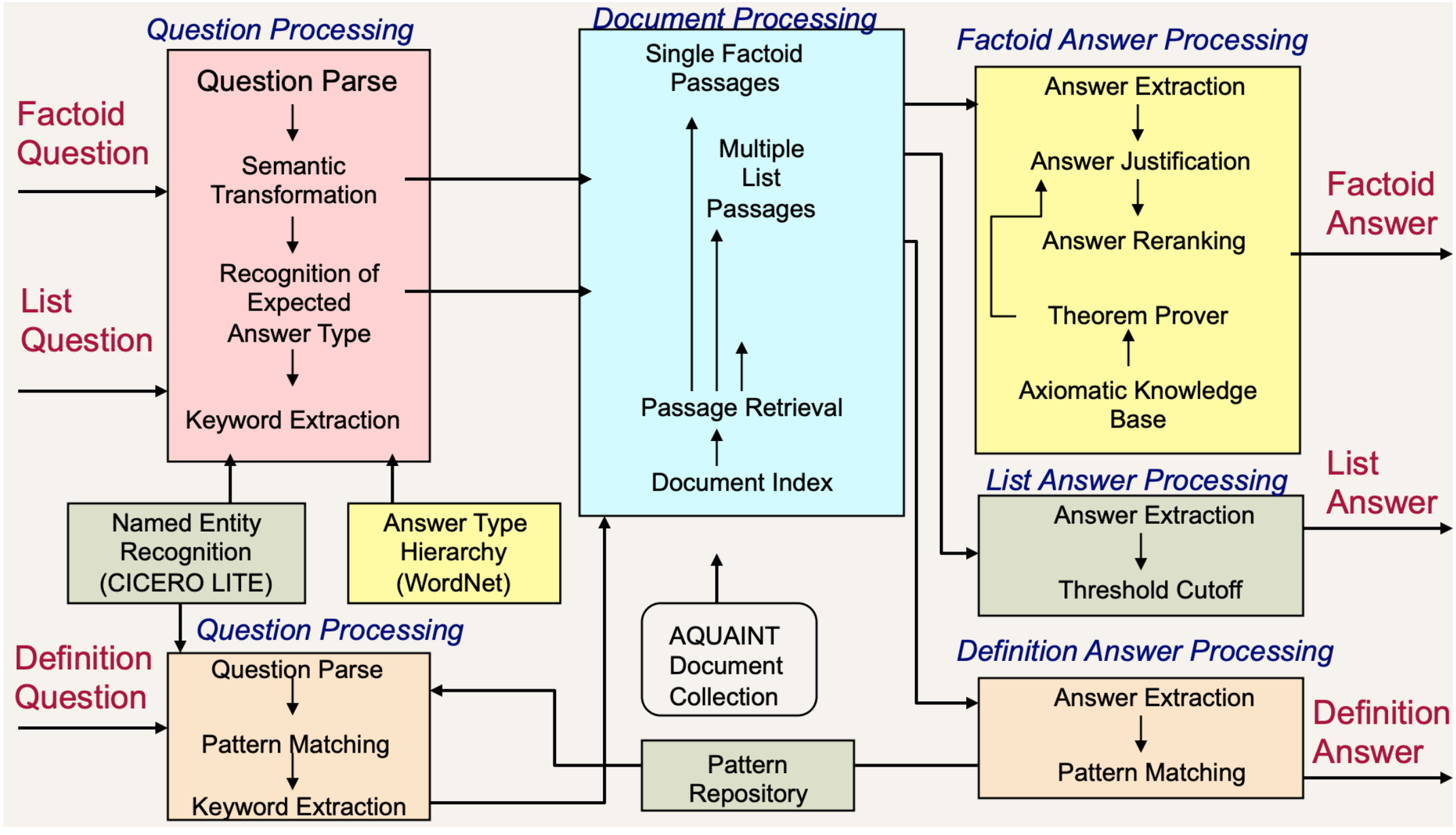
Answer extraction/processing

- Typically another classifier with hand-crafted features plus heuristics
- Run an answer-type named-entity tagger on the passages
  - Each answer type requires a named-entity tagger that detects it
  - If answer type is CITY, tagger has to tag CITY
  - Return the string with the right type

How many bones in a human body? (Number)
The human skeleton is the internal framework of the body. It is composed of 270 bones at birth — this total decreases to 206 bones by adulthood after some bones have fused together.



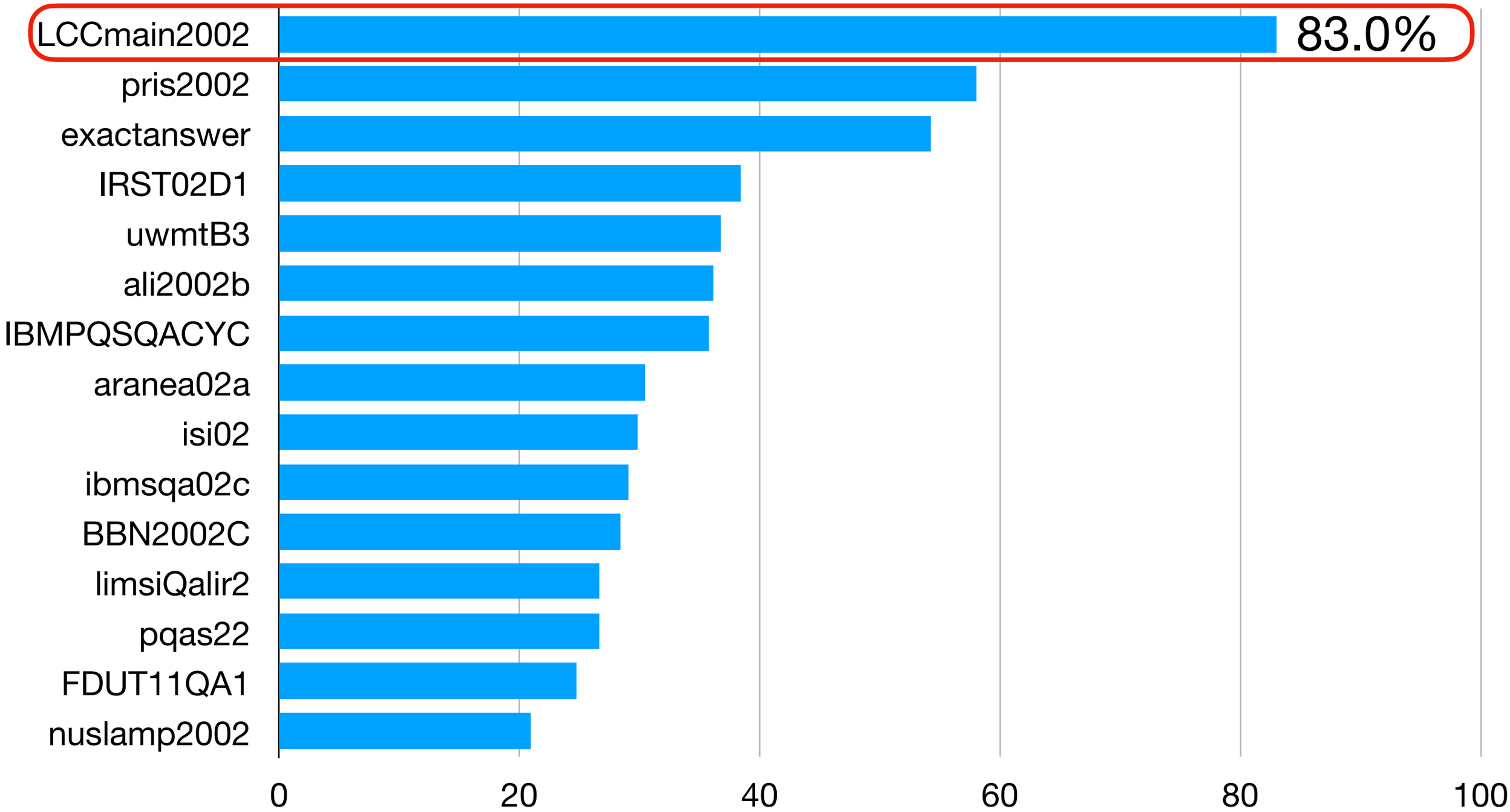
Top TREC-QA system (circa 2003): LCC



Harabagiu and Moldovan, 2003. Question Answering.

https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1086/handouts/cs224n-QA-2008-1up.pdf

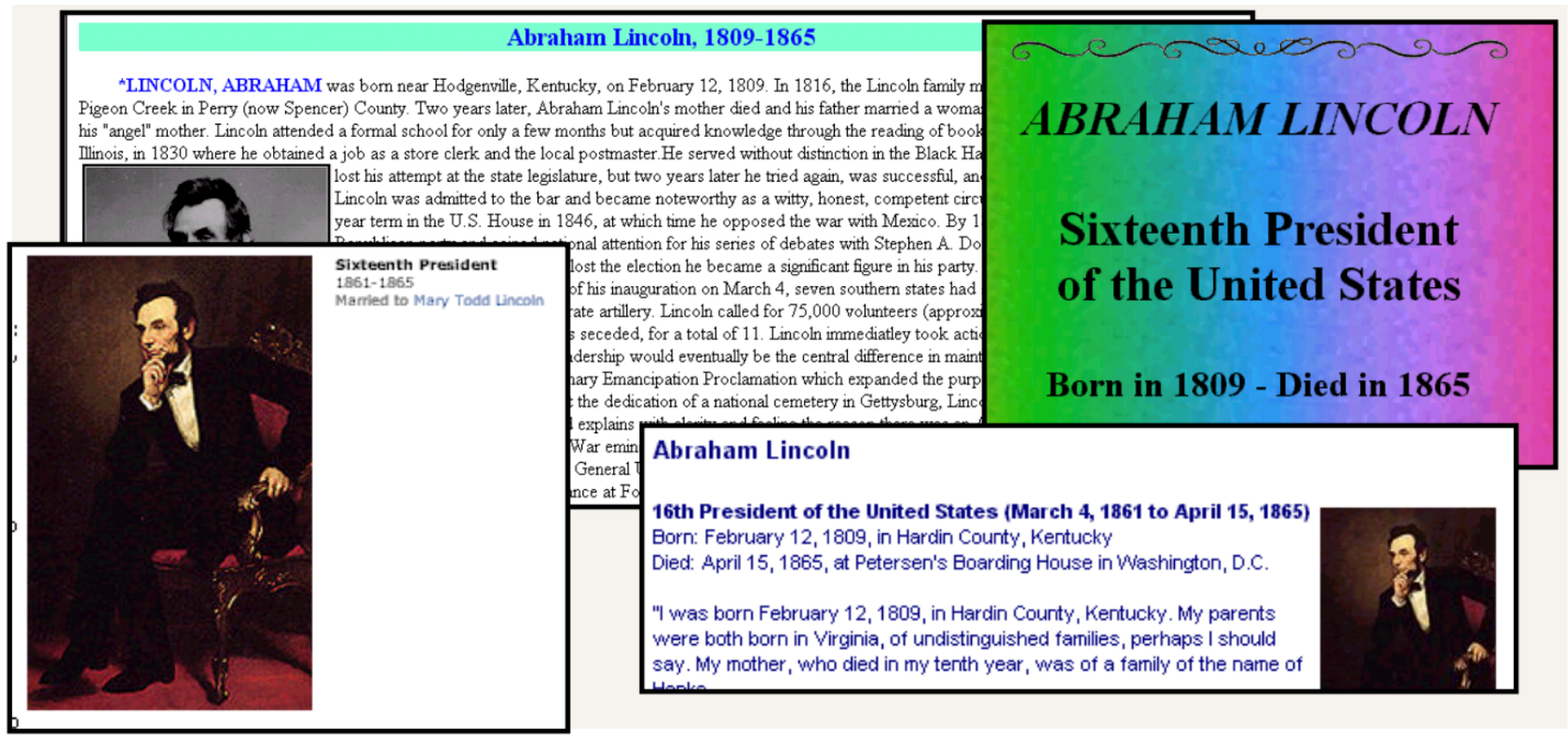
TREC-2002 QA main track results



https://trec.nist.gov/pubs/trec11/papers/QA11.pdf

AskMSR: data-intensive QA

- *"In what year did Abraham Lincoln die?"*
- Leverage "Web Redundancy"
  - Many mentions of the fact on the Web
  - Use patterns to find the "easy" ones



Brill et al., 2002. An Analysis of the AskMSR Question-Answering System
https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1086/handouts/cs224n-QA-2008-1up.pdf

AskMSR: data-intensive QA

- Query patterns of *"In what year did Abraham Lincoln die?"*

  "Abraham Lincoln died in XXXX"

  "Abraham Lincoln (YYYY -- XXXX)"
- Use the most frequent *n*-gram in the documents as answer

- **Observations**
  - Search engine as language model
  - Pattern formulation is no longer needed [Tsai et al., 2015]
  - Works great for head questions, but poorly on tails
  - Cannot provide support evidence (e.g., excerpt, snippet) reliably

Brill et al., 2002. An Analysis of the AskMSR Question-Answering System

Tsai et al., 2015. Web-based Question Answering: Revisiting AskMSR

IBM Watson

The Deep QA Project

IBM Watson defeated two of Jeopardy's greatest champions in 2011.

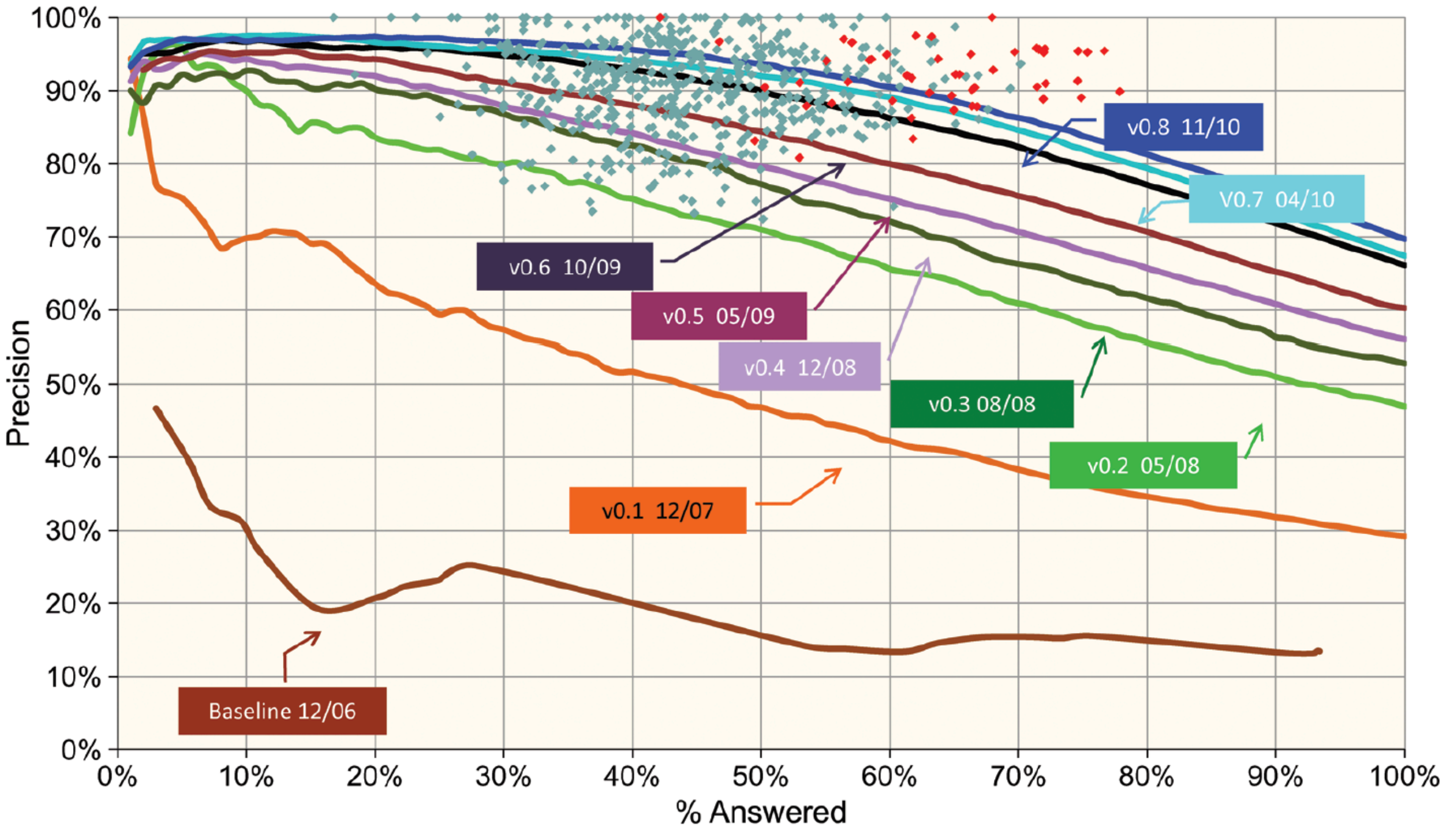
What is Jeopardy?

- Jeopardy! is an American TV quiz show (1964 - present)
  - Question: clues in the form of answers
  - Answer: phrase in the form of question

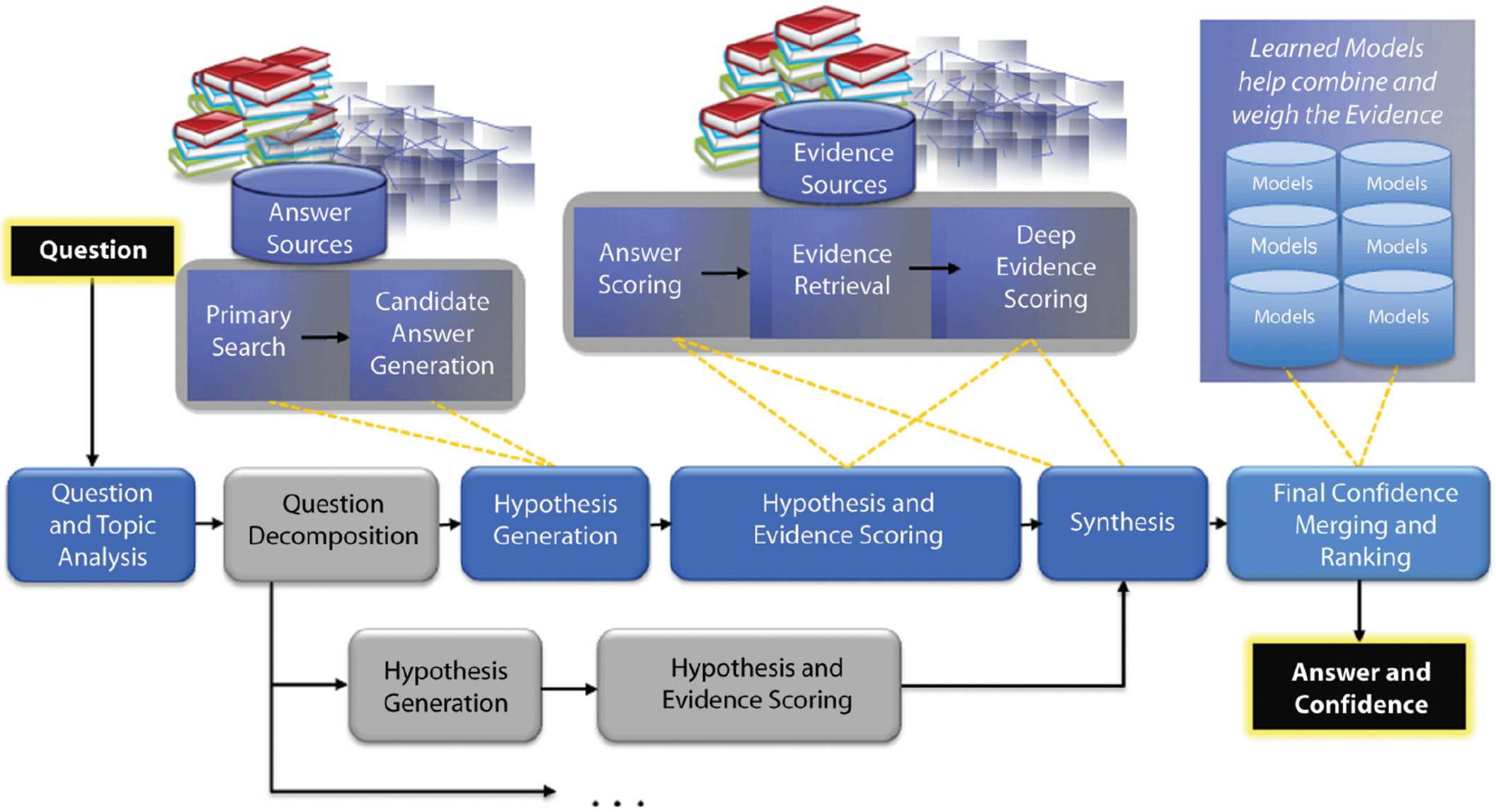
> Category: Michigan Mania
> Clue: In 1894 C.W. Post created his warm cereal drink Possum in this Michigan city
> Answer: Where is Battle Creek?

- Jeopardy! questions are "trivia" or "test" questions
  - Typically long and with detailed information
  - Question askers know the answer (not information seeking)

Progress: June-2007 to Nov-2011



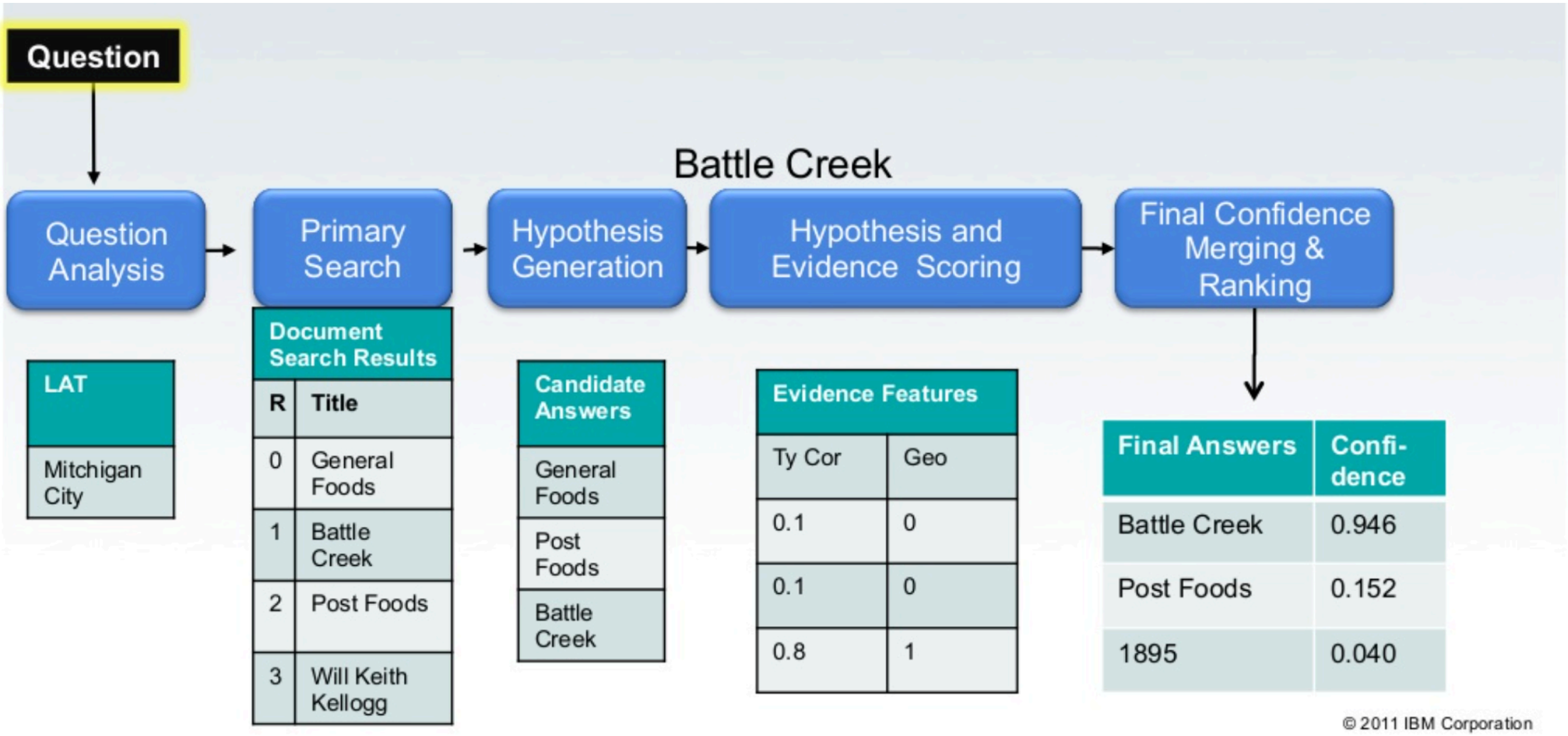
IBM watson DeepQA architecture



Ferrucci, 2012. Introduction to “This is Watson”

“Minimal” DeepQA pipeline

Category: Michigan Mania
Clue: In 1894 C.W. Post created his warm cereal drink Possum in this Michigan city
Answer: Where is Battle Creek?



Success of IBM Watson DeepQA

Possible reasons

- Jeopardy! questions may in fact be easier to answer
- Wikipedia is an important resource for trivia questions
- Large-scale team work with strong engineering support

Implications

- Perceived as an important milestone of AI
- Rekindle the research interest in QA

Recent developments 2013+

- Trend: Macro-reading ⟶ Micro-reading
- General problem setting
  - Given a question and a "context", answer the question using the "context"
  - Two different goals
    - Test machine's intelligence AI (machine reading comprehension)
    - Fulfill user's information need (answer extraction/processing stage)
- Research directions guided by development of new tasks/datasets
- Rapid progress made by new deep learning models

Machine Comprehension Test

[Richardson et al., 2013]

Timmy liked to play games and play sports but more than anything he liked to collect things. He collected bottle caps. He collected sea shells. He collected baseball cards. He has collected baseball cards the longest. He likes to collect the thing that he has collected the longest the most. He once thought about collecting stamps but never did. His most expensive collection was not his favorite collection. Timmy spent the most money on his bottle cap collection.

1) Timmy liked to do which of these things the most?
   A) Collect things
   B) Collect stamps
   C) Play games
   D) Play sports

2) Which is Timmy's most expensive collection?
   A) Stamps
   B) Baseball Cards
   C) Bottle Cap
   D) Sea Shells

3) Which item did Timmy not collect?
   A) Bottle caps
   B) Baseball cards
   C) Stamps
   D) Sea shells

4) Which item did Timmy like to collect the most?
   A) Stamps
   B) Baseball cards
   C) Bottle caps
   D) Sea shells

Richardson et al., 2013. MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text

Stanford Question Answering Dataset

[Rajpurkar et al., 2016]

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

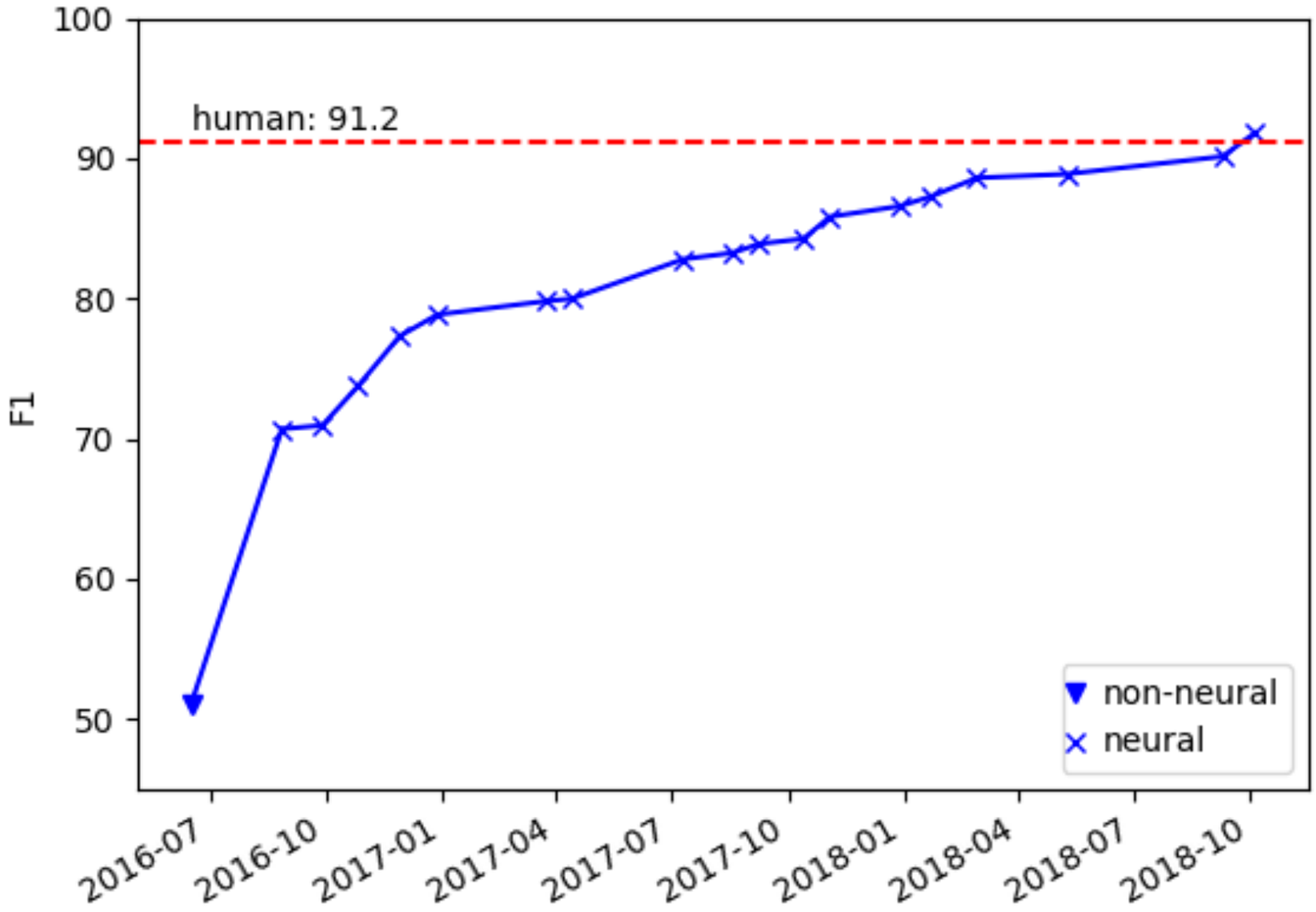
What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?
graupel

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

- (passage, question, answer) triples
- Passage is from Wikipedia, question is crowd-sourced
- Answer must be a span of text in the passage (aka. "extractive question answering")
- SQuAD 1.1: 100k answerable questions, SQuAD 2.0: another 50k unanswerable questions

Rajpurkar et al., 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text

Stanford Question Answering Dataset



Rajpurkar et al., 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text

WikiQA

Given a factoid question, find the sentence in the candidate set that

- Contains the answer
- Can sufficiently support the answer

> Q: Who won the best actor Oscar in 1973?
> S1: Jack Lemmon was awarded the Best Actor Oscar for Save the Tiger (1973).
> S2: Academy award winner Kevin Spacey said that Jack Lemmon is remembered as always making time for others.

Properties

- Real questions (from Bing query logs)
- Candidate sentences from Wikipedia description paragraphs
- Including questions without no answers
  - Answer Triggering: whether answer exists in contexts

Yang et al., 2015. WikiQA: A Challenge Dataset for Open-Domain Question Answering

And many more…

- Reading comprehension
  - RACE [Lai et al., 2017], DuoRC [Saha et al., 2018]
- Fill-in-the-blank questions
  - DeepMind Q&A Dataset [Hermann et al., 2015], Facebook Children Stories [Hill et al., 2016]
- Reasoning challenges
  - Facebook bAbI [Weston et al., 2015], AI2 ARC [Clark et al., 2018], Multi-RC [Khashabi et al., 2018]
- Multi-turn questions
  - SQA [Iyyer et al., 2017], QuAC [Choi et al., 2018], CoQA [Reddy et al., 2019]
- Multi-hop questions
  - HotpotQA [Yang et al., 2018], OBQA [Mihaylov et al., 2018], QASC [Khot et al., 2020]

Extended Reading

TREC Open-domain Question Answering

- Prager, 2007. Open-Domain Question-Answering
- Lin, 2007. An Exploration of the Principles Underlying Redundancy-Based Factoid Question Answering

IBM Watson: The Deep QA project

- Ferrucci et al., 2010. Building Watson: An Overview of the DeepQA Project
- Ferrucci, 2012. This is Watson (IBM Journal of Research and Development)
- Boytsov, 2018. Demystifying IBM Watson

Part III

Datasets & Evaluation

Datasets & Evaluation

- Datasets popular for open-domain QA
  - TriviaQA [Joshi et al., 2017], SearchQA [Dunn et al., 2017], Quasar-T [Dhingra et al., 2017], Natural Questions [Kwiatkowski et al., 2019]
- Datasets repurposed for open-domain QA
  - SQuAD, CuratedTREC, WebQuestions

- Properties to check
  - Motivation: targeted "task" or "scenario"
  - Source of questions, answers and documents/passages
  - Evaluation metric & methods
  - Limitations when used for evaluating open-domain QA

TriviaQA [Joshi et al., 2017]

Motivation

- Large-scale reading comprehension dataset
- Complex, compositional questions

Question: The Dodecanese Campaign of WWII that was an attempt by the Allied forces to capture islands in the Aegean Sea was the inspiration for which acclaimed 1961 commando film?
Answer: The Guns of Navarone
Excerpt: The Dodecanese Campaign of World War II was an attempt by Allied forces to capture the Italian-held Dodecanese islands in the Aegean Sea following the surrender of Italy in September 1943, and use them as bases against the German-controlled Balkans. The failed campaign, and in particular the Battle of Leros, inspired the 1957 novel **The Guns of Navarone** and the successful 1961 movie of the same name.

Joshi et al., 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension

TriviaQA: Collection

- Question-answer pairs
  - 14 trivia and quiz-league Websites
- Textual evidence
  - Web search (Bing)
    - Search query: question
    - Top-10 Web pages (excl. trivia, question, answer, etc.)
  - Wikipedia
    - Identify entities in questions (via TAGME)
    - Add corresponding Wikipedia pages as evidence document
- Filter documents that do **not** contain the correct answer string

Joshi et al., 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension

TriviaQA: Statistics

- Filter documents that do **not** contain the correct answer string

Total number of QA pairs	95,956
Number of unique answers	40,478
Number of evidence documents	662,659
Avg. question length (word)	14
Avg. document length (word)	2,895

- Full unfiltered dataset
  - 110,495 QA pairs
  - 740k evidence documents

Open-domain Setting

Joshi et al., 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension

TriviaQA: Distribution

- Analysis based on 200 randomly sampled questions
- Questions

Property	Example annotation	Statistics
Avg. entities / question	Which politician won the **Nobel Peace Prize** in 2009?	1.77 per question
Fine grained answer type	What **fragrant essential oil** is obtained from Damask Rose?	73.5% of questions
Coarse grained answer type	**Who** won the Nobel Peace Prize in 2009?	15.5% of questions
Time frame	What was photographed for the first time in **October 1959**	34% of questions
Comparisons	What is the appropriate name of the **largest** type of frog?	9% of questions

- Answers
  - Wikipedia: Contains answers for 79.7% questions
  - Web: Contains answers for 75.4% questions

Type	Percentage
Numerical	4.17
Free text	2.98
Wikipedia title	92.85
Person	32
Location	23
Organization	5
Misc.	40

Joshi et al., 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension

TriviaQA: Evaluation

- SQuAD metrics
  - Exact match (EM)
  - F1 over words in the answer(s).

- Questions that have numerical and free-form answers
  - The given answer

- Questions that have Wikipedia entities as answers
  - The given answer plus Wikipedia aliases

Joshi et al., 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension

SearchQA [Dunn et al., 2017]

Motivation

- A general question-answering system should be open-domain
- Use search snippets as the context

Question: Guinness says that by number of users this language, devised by fictional language
Answer: Klingon
Snippet: The Klingons are a fictional extraterrestrial humanoid warriors ... A dictionary, a book of sayings, and a cultural guide to the language have portrayed Montgomery Scott, devised the ... of Guinness World Records, Klingon language by...

Dunn et al., 2017. SearchQA: A New Q&A Dataset Augmented with Context from a Search Engine

SearchQA: Collection

Question-Answer Pairs

- Jeopardy! (J! Archive)

Textual evidence

- Web search (Google)
  - Search query: question-answer pair
- Snippets after some post-processing: removing Jeopardy! related
  - The air-date of the Jeopardy! episode
  - Exact copy of question
  - Terms "Jeopardy!", "quiz" or "trivia"

Dunn et al., 2017. SearchQA: A New Q&A Dataset Augmented with Context from a Search Engine

SearchQA: Statistics

140,461 question-answer pairs

- Each pair is with 49.6 ± 2.10 snippets
- Each snippet is 37.3 ± 11.7 tokens

No learning from the future!

- Training, Validation, Test sets from non-overlapping years.
- The validation and test question-answer pairs are from years later than the training set's pairs.

Split	# Examples
Training	99,820
Validation	13,393
Test	27,248

Dunn et al., 2017. SearchQA: A New Q&A Dataset Augmented with Context from a Search Engine

SearchQA: Evaluation

- Single-word (unigram) answers
  - Top-1 & Top-5 accuracies
- Multi-word (*n*-gram) answers
  - F1 scores

- Human performance

Answer	Unigram	n-gram
Per-question Average	66.97%	42.86%
Per-user Average	64.85%	43.85%
Per-user Std. Dev.	8.16%	10.43%
F1 score (for n-gram)	-	57.62 %

Dunn et al., 2017. SearchQA: A New Q&A Dataset Augmented with Context from a Search Engine

Quasar-T [Dhingra et al., 2017]

Motivation

- Large-scale datasets for evaluating end-to-end QA systems
  - Search, aggregate information from multiple passages, extract answers
- Question answer by search and reading
- Quasar-T is based on trivia questions

Question	7-Eleven stores were temporarily converted into Kwik E-marts to promote the release of what movie?
Answer	**the simpsons movie**
Context excerpt	In July 2007 , 7-Eleven redesigned some stores to look like Kwik-E-Marts in select cities to promote **The Simpsons Movie** . Tie-in promotions were made with several companies , including 7-Eleven , which transformed selected stores into Kwik-E-Marts . " 7-Eleven Becomes Kwik-E-Mart for ' **Simpsons Movie** ' Promotion " .

Dhingra et al., 2017. Quasar: Datasets for Question Answering by Search and Reading

Quasar-T: Collection

Question-Answer Pairs

- Collected by Reddit user 007craft and released in Dec 2015
- Remove True/False and multi-choice questions
- Most answers are noun phrases

Textual Evidence

- Source: ClueWeb09 [Callan et al., 2009]
  - 1 billion web pages collected between Jan. and Feb. 2009
- Phase 1: 100 documents from ClueWeb09 batch query service
  - Query: Question + Answer
  - Long context: 2048 characters, short context: 200 characters
- Phase 2: Top pseudo-documents that contain the answer using Lucene
  - Query: Question
  - 20 long context & 100 short context per question

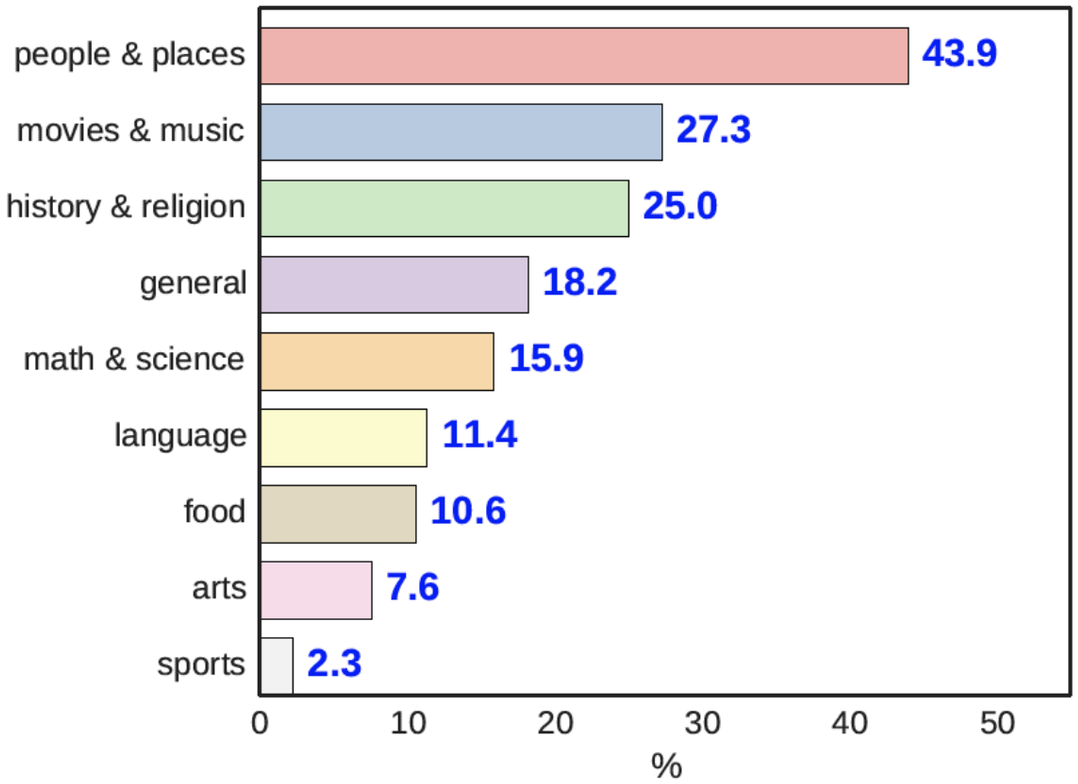
Dhingra et al., 2017. Quasar: Datasets for Question Answering by Search and Reading

Quasar-T: Statistics

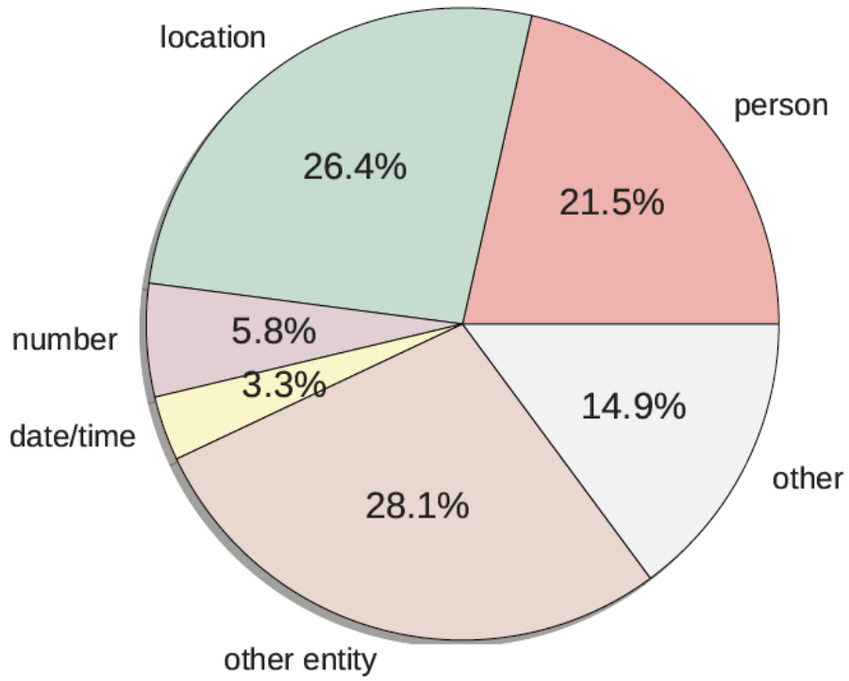
	Total	Single-Word Answer	Answer in Short Context	Answer in Long Context
Train	37,012	18,726	25,465	26,318
Validation	3,000	1,507	2,068	2,129
Test	3,000	1,508	2,043	2,102

Dhingra et al., 2017. Quasar: Datasets for Question Answering by Search and Reading

Quasar-T: Distribution



Question genres



Answer categories

Dhingra et al., 2017. Quasar: Datasets for Question Answering by Search and Reading

Quasar-T: Evaluation

- SQuAD Metrics
  - Exact match (EM)
  - F1 over words in the answer(s).

- Exact match measures whether the two strings, after preprocessing, are equal or not.

- F1 measures the overlap between the two bags of tokens in answers, after preprocessing

Dhingra et al., 2017. Quasar: Datasets for Question Answering by Search and Reading

Natural Questions [Kwiatkowski et al., 2019]

Motivation

- Large-scale end-to-end training data for QA
- "Natural" questions from search engine query logs

Example 1

Question: what color was john wilkes booth's hair

Wikipedia Page: John_Wilkes_Booth

Long answer: Some critics called Booth "the handsomest man in America" and a "natural genius", and noted his having an "astonishing memory"; others were mixed in their estimation of his acting. He stood 5 feet 8 inches (1.73 m) tall, had jet-black hair , and was lean and athletic. Noted Civil War reporter George Alfred Townsend described him as a "muscular, perfect man" with "curling hair, like a Corinthian capital".

Short answer: jet-black

Kwiatkowski et al., 2019. Natural Questions: A Benchmark for Question Answering Research

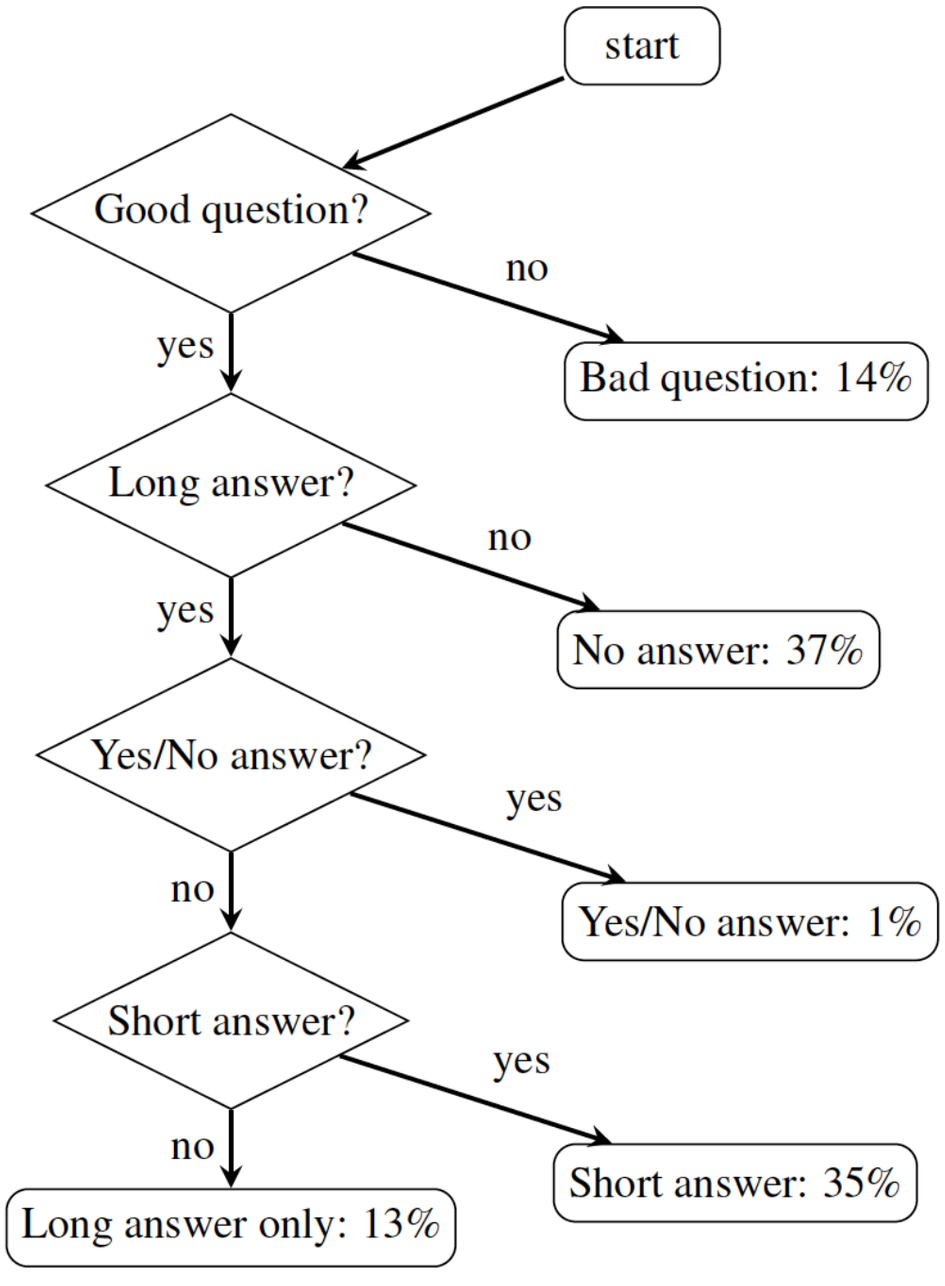
Natural Questions: Collection

- Question source: Google search queries
  - Queries of 8 words or more, by multiple users in a short period of time
- Answer source: Wikipedia page from top 5 search results
  - Long answer: paragraph, table, list (HTML bounding box)
  - Short answer: span(s), yes/no, NULL
- Annotation: a pool of ~50 annotators

Kwiatkowski et al., 2019. Natural Questions: A Benchmark for Question Answering Research

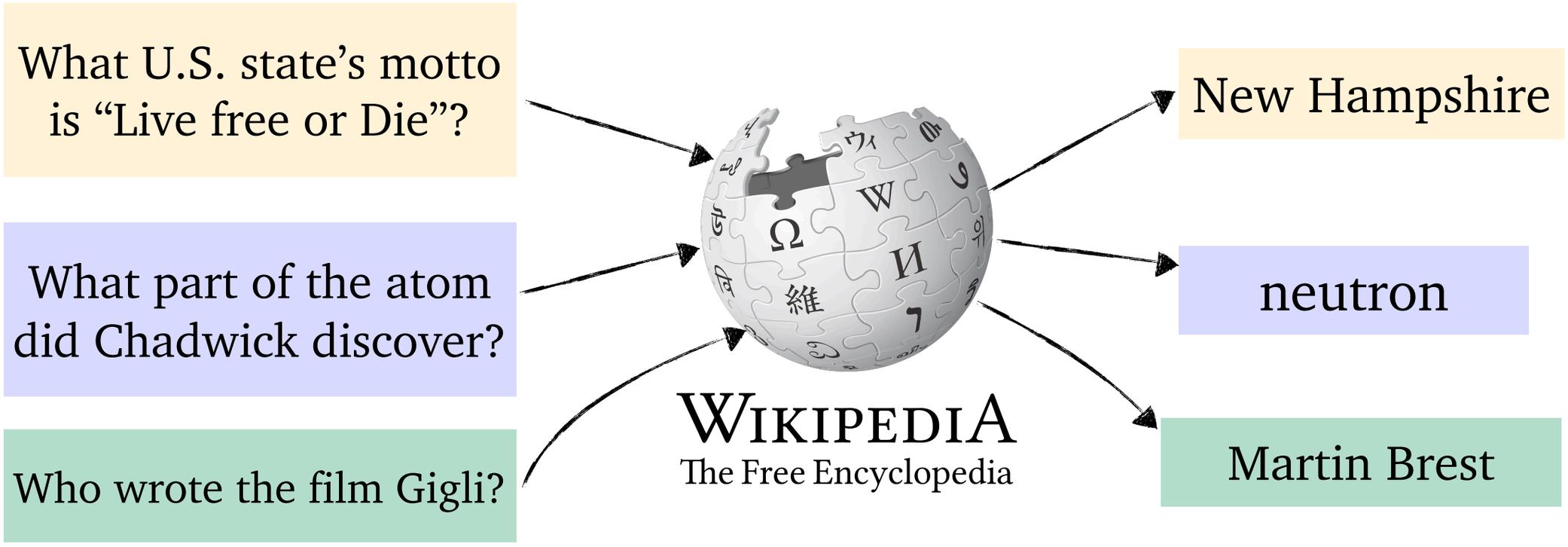
Natural Questions: Statistics

- Train: 307,373 examples with single annotations
- Dev: 7,830 examples with 5-way annotations
- Test: 7,842 examples with 5-way annotations (sequestered)



Kwiatkowski et al., 2019. Natural Questions: A Benchmark for Question Answering Research

Open-Domain QA Evaluation



[Chen et al., 2017; Lee et al., 2019]

- The correctness of the supporting evidence is not evaluated
- Dataset and Wikipedia dump may not be created at the same time

Open-Domain QA Datasets

Used in ORQA [Lee et al., 2019]

- Natural Questions
  - Questions with short answers (<5 tokens)
- WebQuestions [Berant et al., 2013]
  - Questions sampled using Google Suggest API
  - Answers are Freebase entities
- CuratedTREC [Baudis & Sedivy, 2015]
  - Questions from TREC-QA; askers do not observe evidence doc.
- TriviaQA
  - Questions from the unfiltered set (i.e., all questions)
- OpenSQuAD [Rajpurkar et al., 2016]
  - Questions from SQuAD v1.1; askers do see the context (Wikipedia paragraph)

Dataset	Train	Val	Test
NQ	79,168	8,757	3,610
WebQ	3,417	361	2,032
TREC	1,353	133	694
TriviaQA	78,785	8,837	11,313
SQuAD	78,713	8,886	10,570

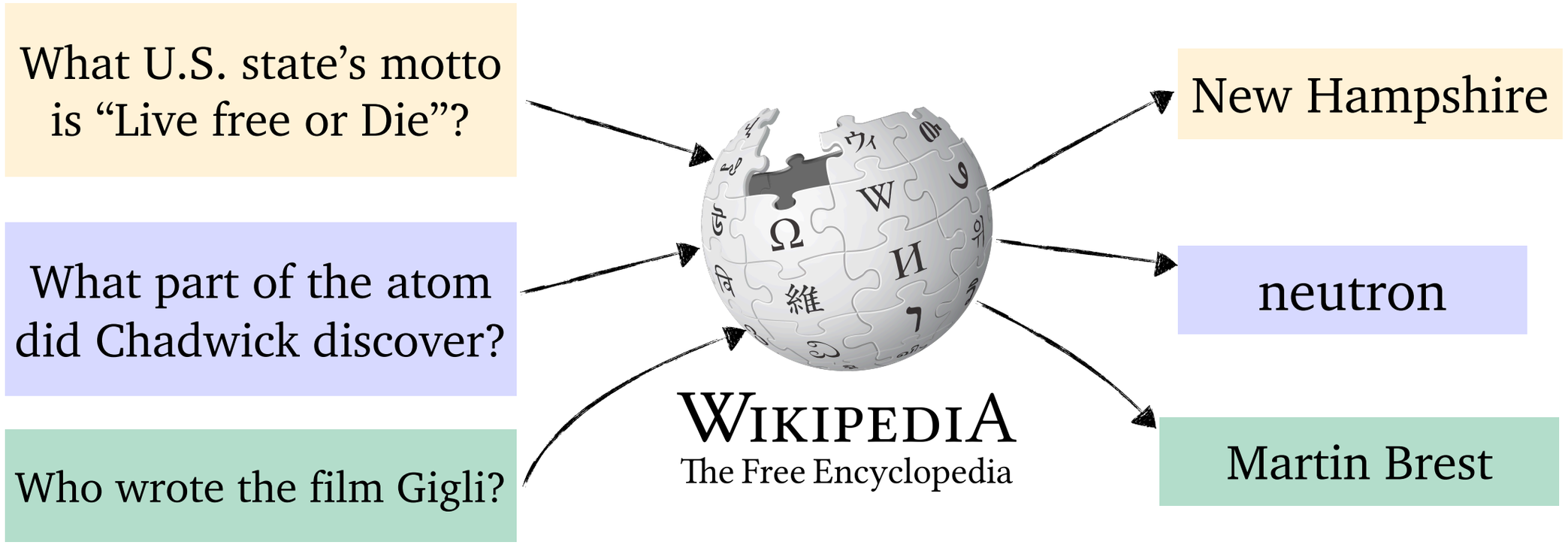
Lee et al., 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering

Part IV

Two-stage **retriever-reader** approaches

Problem setup

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



"Machine Reading at Scale"

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

Why is Wikipedia?

- Wikipedia is treated as a generic collection of articles and internal graph structured is not considered in this setting $\Longrightarrow$ 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 need to consider the *redundancy* problem too much here.

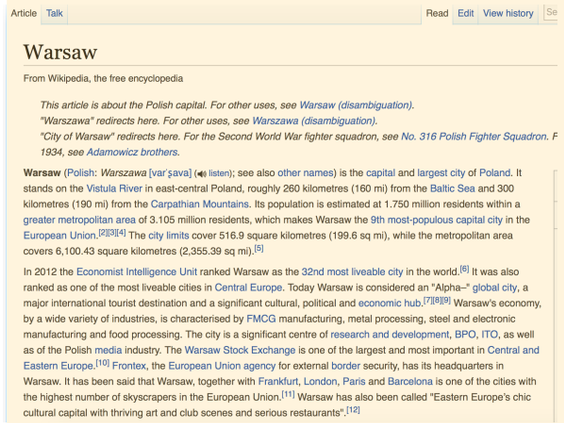
Chen et al., 2017. Reading Wikipedia to Answer Open-domain Questions

DrQA: a first neural open-domain QA system

[Chen et al., 2017]

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

Document Retriever



Article Talk Read Edit View history

Warsaw

From Wikipedia, the free encyclopedia

This article is about the Polish capital. For other uses, see Warsaw (disambiguation).
"Warszawa" redirects here. For other uses, see Warszawa (disambiguation).
"City of Warsaw" redirects here. For the Second World War fighter squadron, see No. 316 Polish Fighter Squadron. F
1934, see Adamowicz brothers.

Warsaw (Polish: *Warszawa* [varˈʂava] (listen); see also other names) is the capital and largest city of Poland. It stands on the Vistula River in east-central Poland, roughly 260 kilometres (160 mi) from the Baltic Sea and 300 kilometres (190 mi) from the Carpathian Mountains. Its population is estimated at 1.750 million residents within a greater metropolitan area of 3.105 million residents, which makes Warsaw the 9th most-populous capital city in the European Union.[2][3][4] The city limits cover 516.9 square kilometres (199.6 sq mi), while the metropolitan area covers 6,100.43 square kilometres (2,355.39 sq mi).[5]

In 2012 the Economist Intelligence Unit ranked Warsaw as the 32nd most liveable city in the world.[6] It was also ranked as one of the most liveable cities in Central Europe. Today Warsaw is considered an "Alpha–" global city, a major international tourist destination and a significant cultural, political and economic hub.[7][8][9] Warsaw's economy, by a wide variety of industries, is characterised by FMCG manufacturing, metal processing, steel and electronic manufacturing and food processing. The city is a significant centre of research and development, BPO, ITO, as well as of the Polish media industry. The Warsaw Stock Exchange is one of the largest and most important in Central and Eastern Europe.[10] Frontex, the European Union agency for external border security, has its headquarters in Warsaw. It has been said that Warsaw, together with Frankfurt, London, Paris and Barcelona is one of the cities with the highest number of skyscrapers in the European Union.[11] Warsaw has also been called "Eastern Europe's chic cultural capital with thriving art and club scenes and serious restaurants".[12]

Document Reader

833,500

Information Retrieval

Reading Comprehension

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

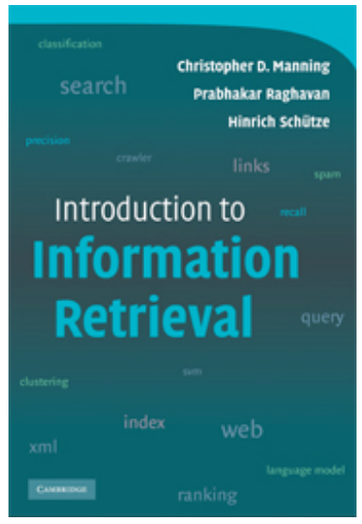
Document Retriever

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

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

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

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



tf =term frequency, idf = inverse document frequency
t: term, d: document (= one Wiki. article), D: corpus (= Wikipedia)

- Important - This retriever is not *trainable*.
- DrQA considers the retrieval problem at document level instead of paragraph level.

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

Document Reader

- It casts as a *reading comprehension* problem:
  - Input is a passage *P* and a question *Q*
  - Output is answer *A*. When the problem is restricted as a segment of text in the passage, the problem is also called as “extractive question answering”.

(Rajpurkar et al, 2016):
Stanford Question Answering Dataset

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

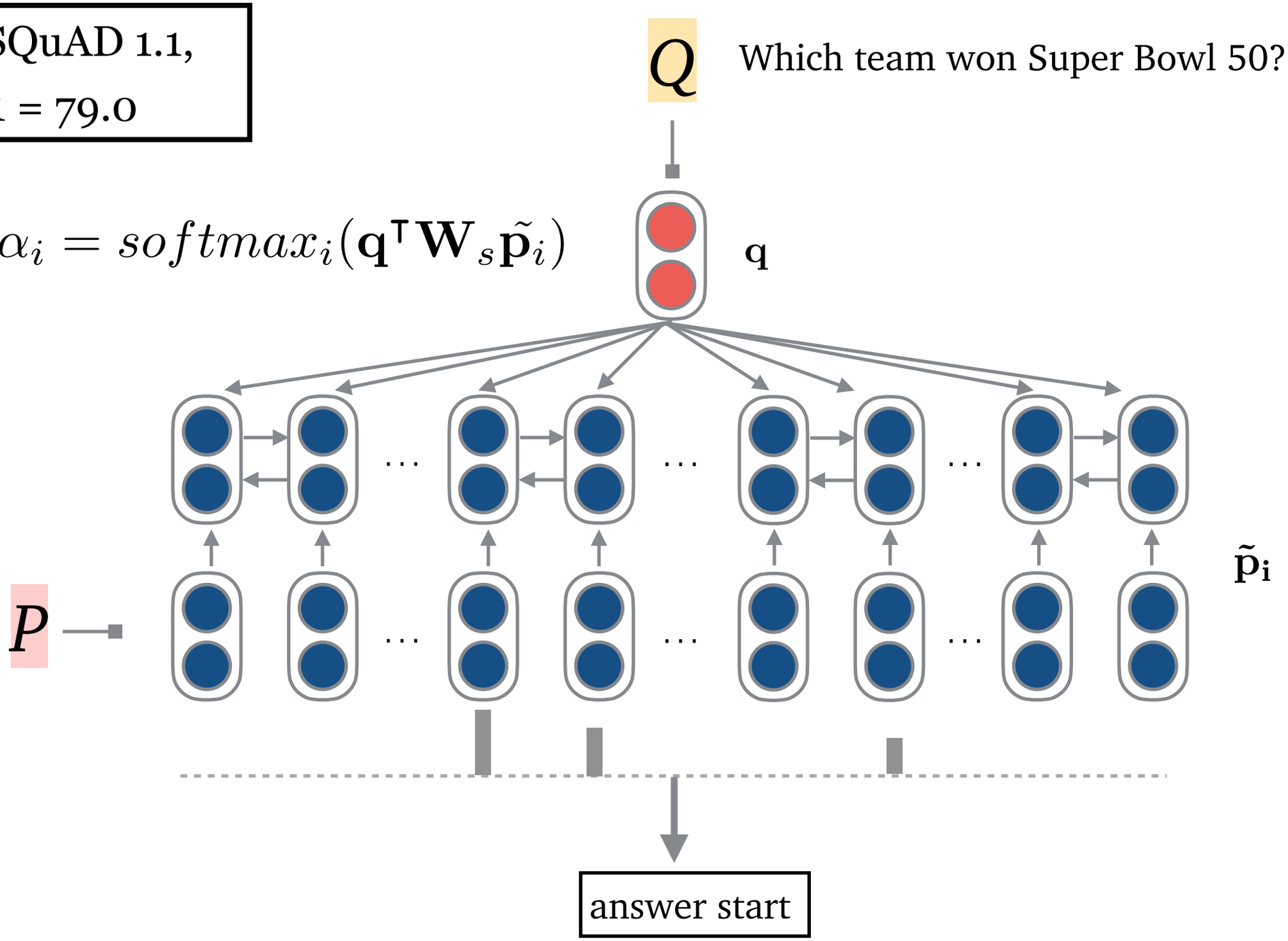
Chen et al., 2017. Reading Wikipedia to Answer Open-domain Questions

Document Reader

On SQuAD 1.1,

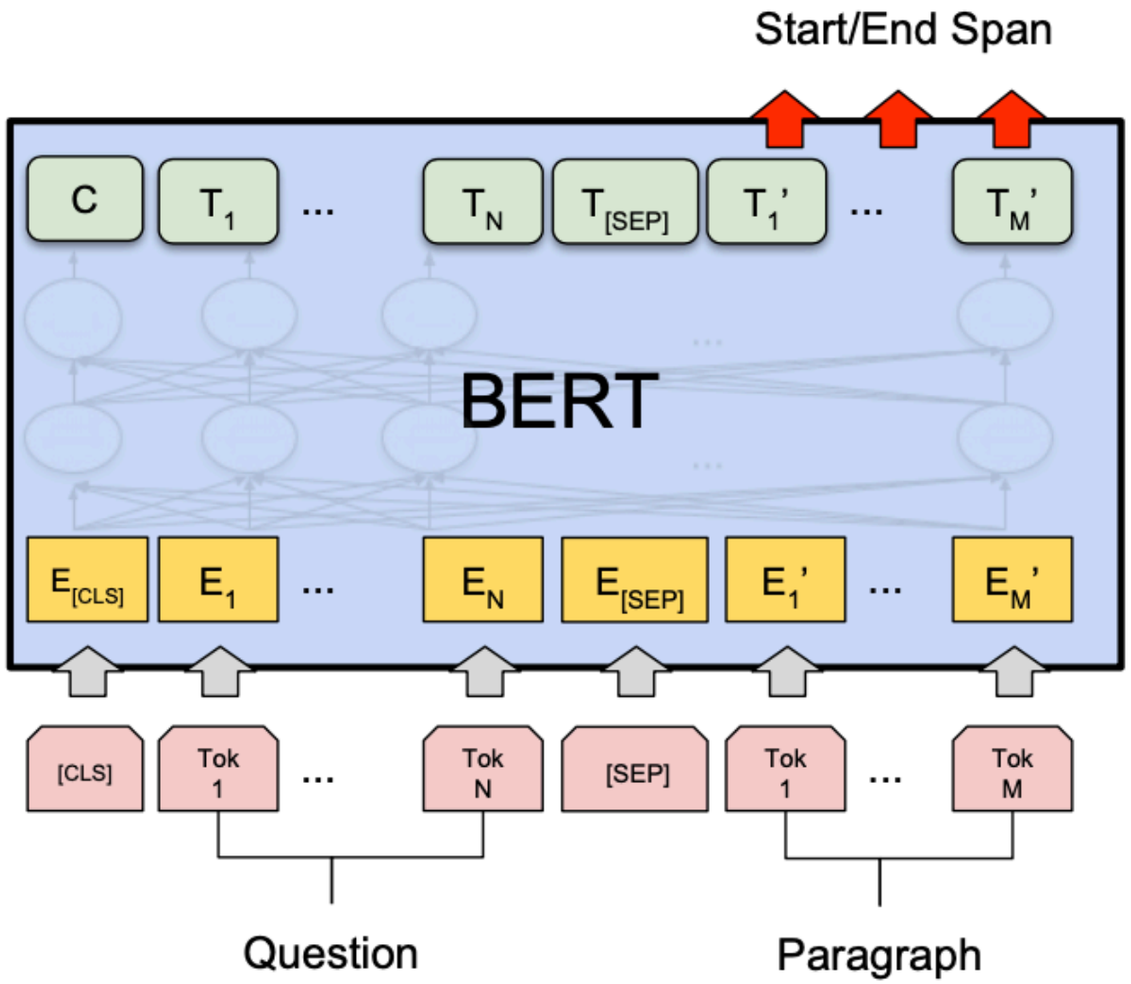
- F1 = 79.0

$$\alpha_i = softmax_i(\mathbf{q}^\intercal \mathbf{W}_s \tilde{\mathbf{p}}_i)$$



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

Document Reader



Question = Segment A

Passage = Segment B

Answer = predicting two endpoints in segment B

On SQuAD 1.1,

- BiDAF +  : F1 = 85.8
-  : F1 = 90.9
- RoBERTa: F1 = 94.6

Devlin et al., 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

How to train the reader?

- 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-QA)?

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

- Solution: create new training examples using our **retriever**!

(Q, A) ⟶ (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)

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

How to train the reader?

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.

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

Putting it together

Training time:

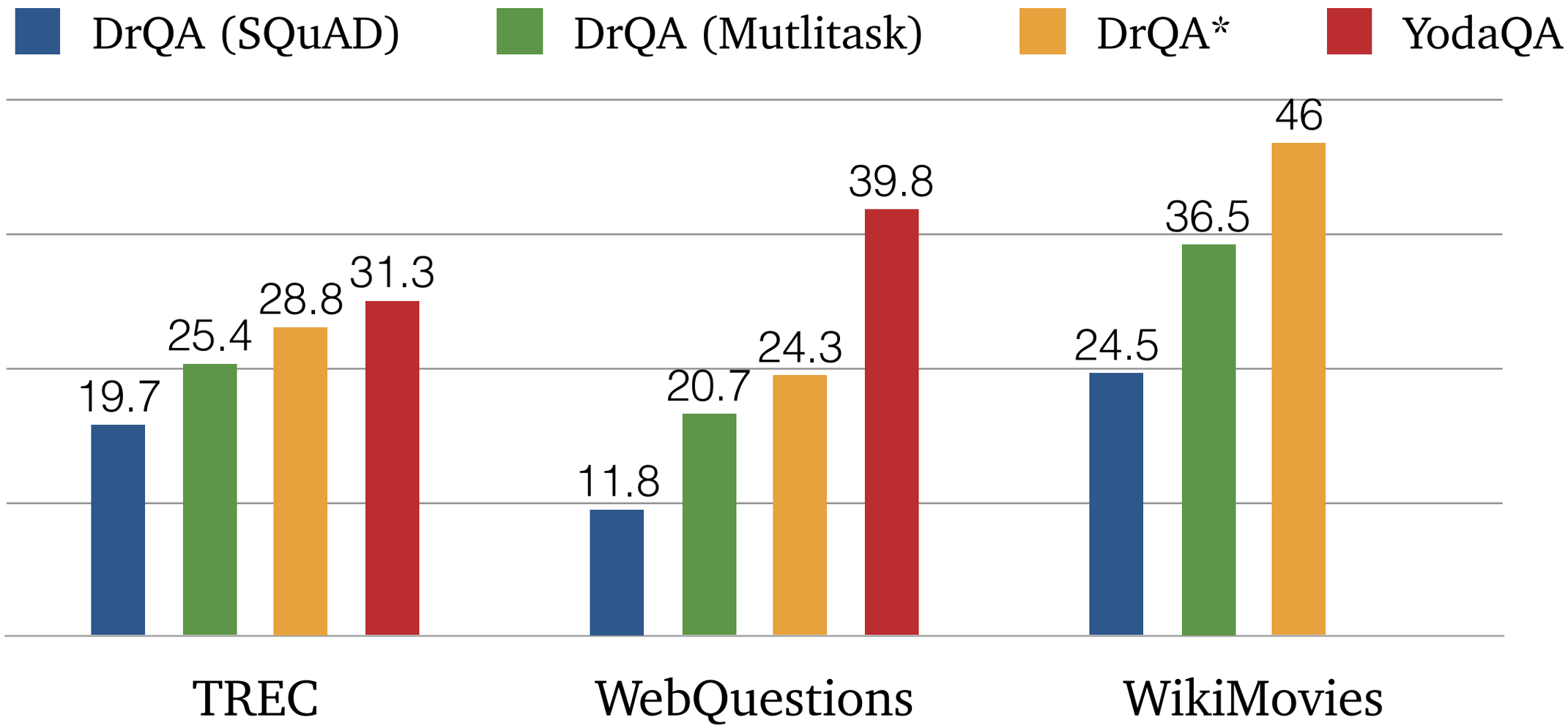
- Document retriever: not trained
- Document reader: a reading comprehension model trained on SQuAD + distantly-supervised data generated from QA datasets

Inference time:

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

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

Experiments



% of questions answered correctly
(top-1 prediction, exact match)

DrQA*: see (Raison et al, 2018)

YodaQA: http://ailao.eu/yodaqa/

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

How can we do better?

- DrQA considers retrieval at document level.

  Does passage-level retriever work better?

- Answers in the retrieved passages might not be directly comparable at inference time.

  Does multi-passage training help?

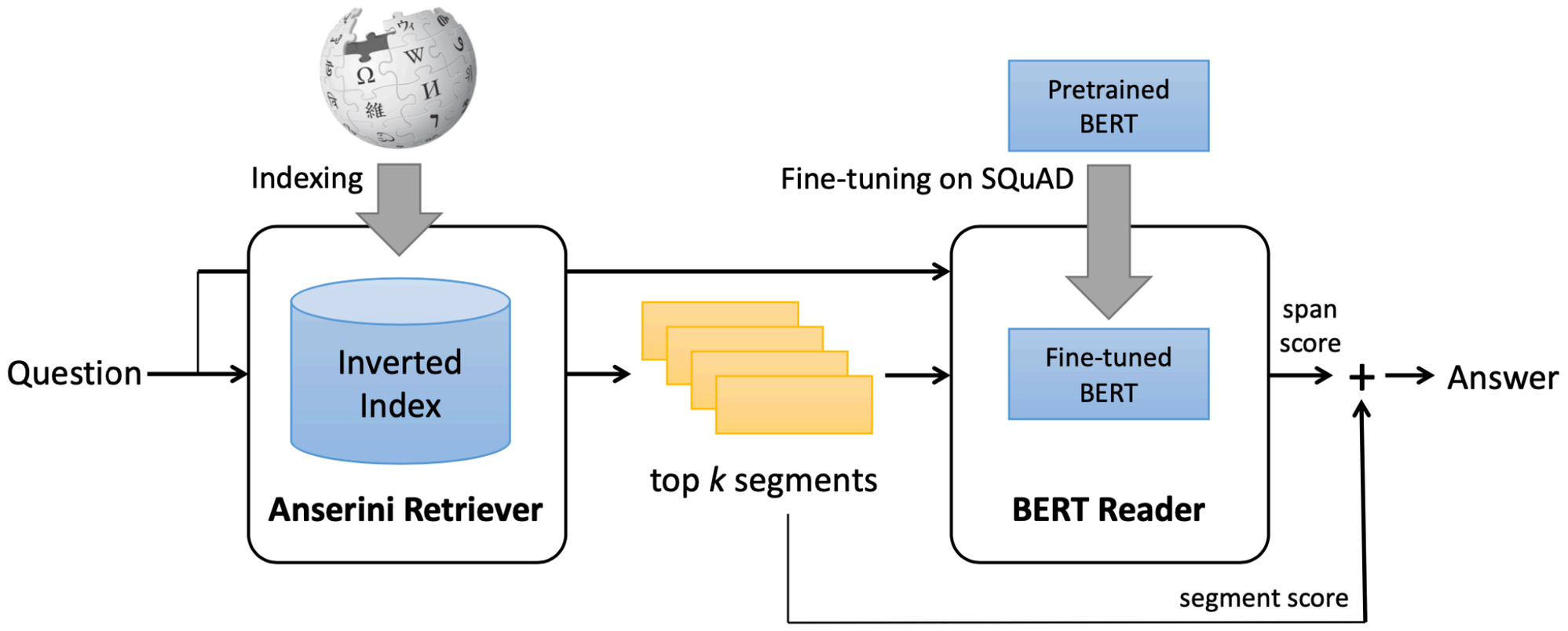
- The importance of each passage has been omitted.

  Can we train a ranker on the retrieved passages?

- The retriever is not trained!

  The focus of the next part.

BERTserini [Yang et al., 2019]



Anserini Retriever:
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 though:

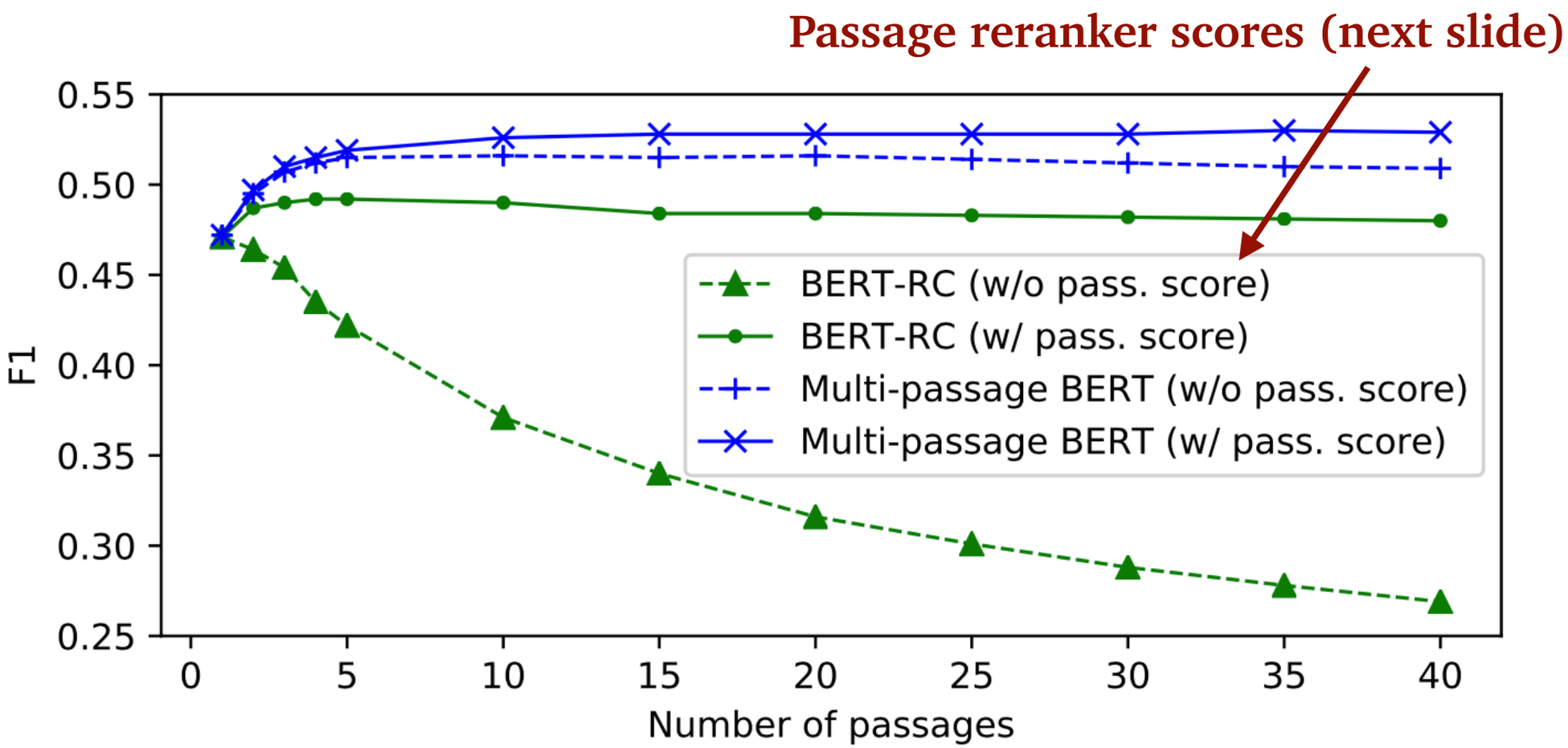
27.1 (DrQA, SQuAD only) $\longrightarrow$ **38.6**

Yang et al., 2019. End-to-End Open-Domain Question Answering with BERTserini

Multi-passage training

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

- **Shared normalization**: process paragraphs independently, but compute the span probability across spans in all paragraphs in each mini-batch



Clark and Gardner, 2018. Simple and Effective Multi-Paragraph Reading Comprehension
Wang et al., 2019. Multi-passage BERT: A Globally Normalized BERT Model for Open-domain Question Answering

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.
- Let the retriever return the top N passages and we can train a ranker component to select the best passage.

$$
\begin{aligned}
\mathbf{u}_i &= \text{MaxPooling}(\mathbf{H}_i^{\text{Rank}}), \\
\mathbf{C} &= \text{Tanh}\left(\mathbf{W}^c[\mathbf{u}_1; \mathbf{u}_2; ...; \mathbf{u}_N] + \mathbf{b}^c \otimes \mathbf{e}_N\right), \\
\gamma &= \text{Softmax}(\mathbf{w}^c\mathbf{C}),
\end{aligned}
$$

$\mathbf{u}_i$ 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)

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

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

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

Passage re-ranker: results

	Quasar-T		$\text{SQuAD}_{\text{OPEN}}$		WikiMovies		CuratedTREC		WebQuestions	
	F1	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**
GA (Dhingra et al. 2017)	26.4	26.4	-	-	-	-	-	-	-	-
BiDAF (Seo et al. 2017)	28.5	25.9	-	-	-	-	-	-	-	-
DrQA (Chen et al. 2017a)	-	-	-	28.4	-	34.3	-	25.7	-	**19.5**
Single Reader (SR)	$38.5^{.2}$	$31.5^{.2}$	$35.4^{.2}$	$26.9^{.2}$	$38.8^{.1}$	$37.7^{.1}$	$33.6^{.6}$	$27.4^{.4}$	$22.0^{.2}$	$15.2^{.3}$
Simple Ranker-Reader (SR^2)	$38.8^{.2}$	$31.9^{.2}$	$35.8^{.2}$	$27.2^{.2}$	$39.3^{.1}$	$38.1^{.1}$	$33.4^{.6}$	$27.7^{.5}$	$22.5^{.3}$	$15.6^{.4}$
Reinforced Ranker-Reader (R^3)	$\mathbf{40.9}^{.3}$	$\mathbf{34.2}^{.3}$	$\mathbf{37.5}^{.2}$	$\mathbf{29.1}^{.2}$	$\mathbf{39.9}^{.1}$	$\mathbf{38.8}^{.1}$	$\mathbf{34.3}^{.6}$	$\mathbf{28.4}^{.6}$	$\mathbf{24.6}^{.3}$	$17.1^{.3}$
DrQA-MTL (Chen et al. 2017a)	-	-	-	29.8	-	36.5	-	25.4	-	20.7
YodaQA (Baudiš and Šedivỳ 2015)	-	-	-	-	-	-	-	31.3	-	39.8

Single Reader < Simple Ranker-Reader < Reinforced Ranker-Reader

The improvement is relatively limited though.

Wang et al., 2018. R^3: Reinforced Ranker-Reader for Open-Domain Question Answering

Hard EM Learning [Min et al., 2019]

When a retrieved passage contains multiple mentions of the answer, 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.

Min et al., 2019. A Discrete Hard EM Approach for Weakly Supervised Question Answering

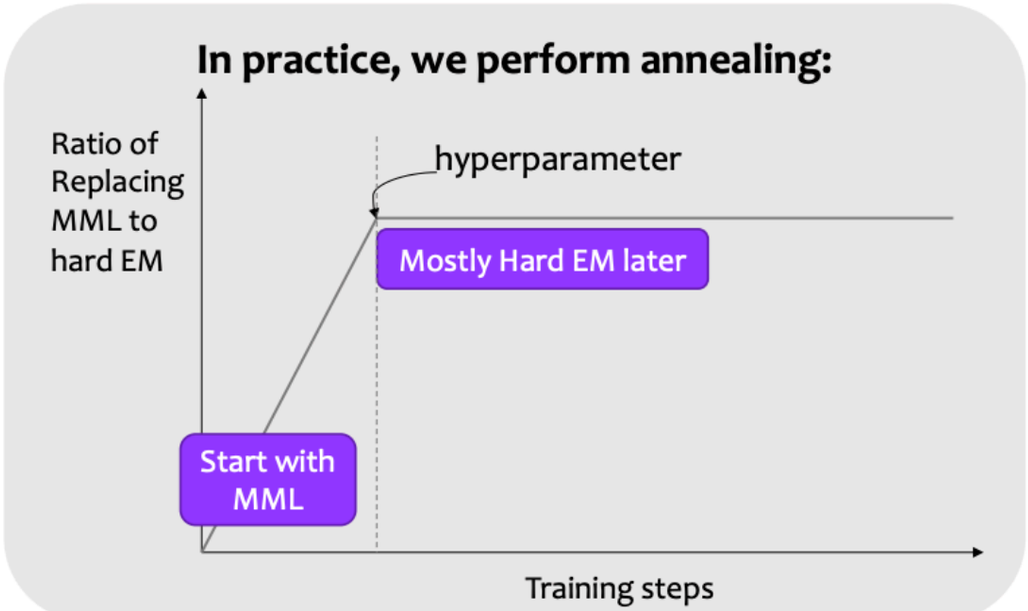
Hard EM Learning

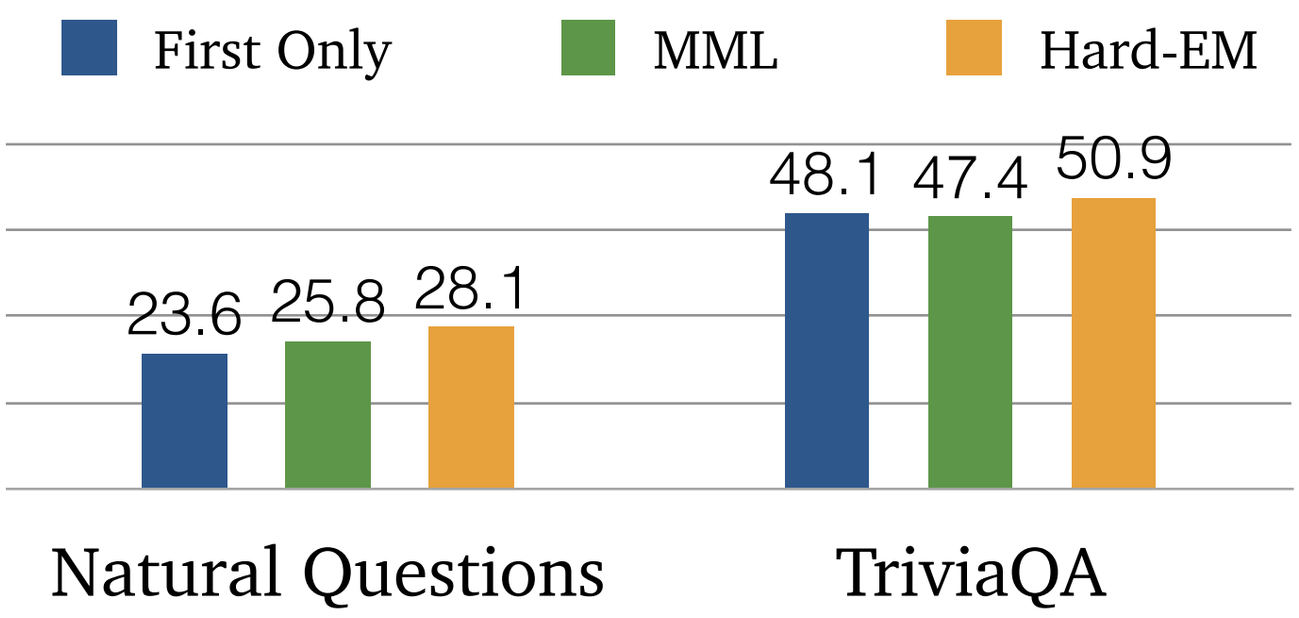
Hard-EM approach

First Only: $J(\theta) = -\log \mathbb{P}(z_1|x;\theta)$, where z_1 appears first in the given document among all $z_i \in Z$.

MML: $J(\theta) = -\log \Sigma_{i=1}^{n} \mathbb{P}(z_i|x;\theta)$.

Ours: $J(\theta) = -\log \max_{1 \le i \le n} \mathbb{P}(z_i|x;\theta)$.





Hard-EM > First Only, MML

Min et al., 2019. A Discrete Hard EM Approach for Weakly Supervised Question Answering

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 .

Wang et al., 2018. Evidence Aggregation for Answer Re-Ranking in Open-Domain Question Answering

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 for Quasar-T, SearchQA, TriviaQA when there is enough redundancy but is not evaluated on the Wikipedia setting yet.

Wang et al., 2018. Evidence Aggregation for Answer Re-Ranking in Open-Domain Question Answering

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

Part V

Dense Retriever and **End-to-end** Training

Key questions

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

Document Retriever

Warsaw

From Wikipedia, the free encyclopedia

This article is about the Polish capital. For other uses, see Warsaw (disambiguation).
"Warszawa" redirects here. For other uses, see Warszawa (disambiguation).
"City of Warsaw" redirects here. For the Second World War fighter squadron, see No. 316 Polish Fighter Squadron. F
1934, see Adamowicz brothers.

Warsaw (Polish: *Warszawa* [varˈʂava] (listen); see also other names) is the capital and largest city of Poland. It stands on the Vistula River in east-central Poland, roughly 260 kilometres (160 mi) from the Baltic Sea and 300 kilometres (190 mi) from the Carpathian Mountains. Its population is estimated at 1.750 million residents within a greater metropolitan area of 3.105 million residents, which makes Warsaw the 9th most-populous capital city in the European Union.[2][3][4] The city limits cover 516.9 square kilometres (199.6 sq mi), while the metropolitan area covers 6,100.43 square kilometres (2,355.39 sq mi).[5]

In 2012 the Economist Intelligence Unit ranked Warsaw as the 32nd most liveable city in the world.[6] It was also ranked as one of the most liveable cities in Central Europe. Today Warsaw is considered an "Alpha–" global city, a major international tourist destination and a significant cultural, political and economic hub.[7][8][9] Warsaw's economy, by a wide variety of industries, is characterised by FMCG manufacturing, metal processing, steel and electronic manufacturing and food processing. The city is a significant centre of research and development, BPO, ITO, as well as of the Polish media industry. The Warsaw Stock Exchange is one of the largest and most important in Central and Eastern Europe.[10] Frontex, the European Union agency for external border security, has its headquarters in Warsaw. It has been said that Warsaw, together with Frankfurt, London, Paris and Barcelona is one of the cities with the highest number of skyscrapers in the European Union.[11] Warsaw has also been called "Eastern Europe's chic cultural capital with thriving art and club scenes and serious restaurants".[12]

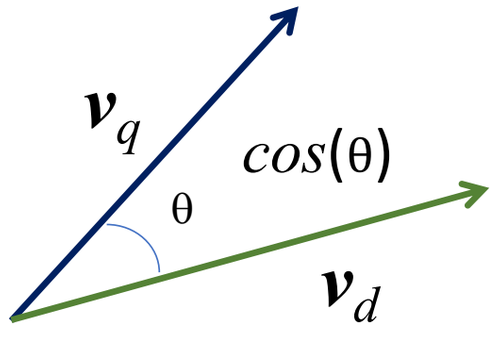
Document Reader

833,500

*Is the retriever **trainable** or not?*

*Can we train dense **representations** for the retriever?*

Sparse vs dense representations for retrieval



$d_1 >> d_2$

sparse repr: [0… 1 …. 1 …0..1] $\in \mathbb{R}^{d_1}$

dense repr: [1.03, -5.72, 6.42, .., 9.91] $\in \mathbb{R}^{d_2}$

sparse

"How many provinces did the Ottoman empire contain in the 17th century?"

"What part of the atom did Chadwick discover?"

dense

"Who is the **bad guy** in lord of the rings?"

*"Sala Baker is an actor and stuntman from New Zealand. He is best known for portraying the **villain** Sauron in the Lord of the Rings trilogy by Peter Jackson."*

They capture complementary information; Dense representations have never been shown to outperform sparse representations in open-domain QA before 2019...

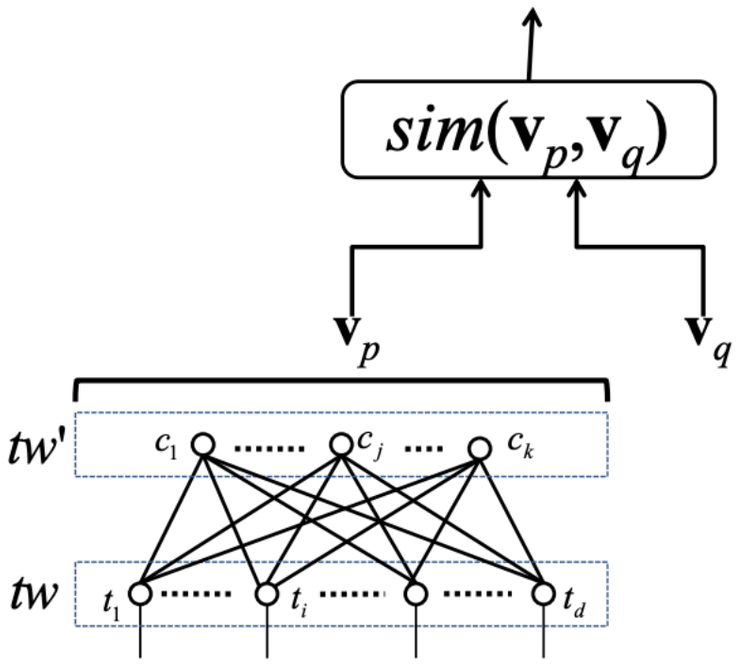
Why **dense** retrieval now?

- It is a difficult problem: we need to *encode*, *index* and *search* from 5M documents or 30M paragraphs or 60B phrases.

Learning Discriminative Projections for Text Similarity Measures

CoNLL 2011

Wen-tau Yih **Kristina Toutanova** **John C. Platt** **Christopher Meek**
Microsoft Research
One Microsoft Way
Redmond, WA 98052, USA



- Cross-lingual document retrieval
- Ad relevance prediction
- Web search ranking

Why **dense** retrieval now?

- It is actually not easy to make these dense models "work".
  - Needs large enough labeled data (e.g., 82M query-doc pairs from user clicks).

We have pre-trained models now..

- Now we have much better techniques and tools to support fast maximum inner product search (MIPS):
  - In-memory data structure and indexing schemes

e.g., FAISS (Johnson et al, 2017)

ORQA: Open-Retriever Question Answering

[Lee et al., 2019]

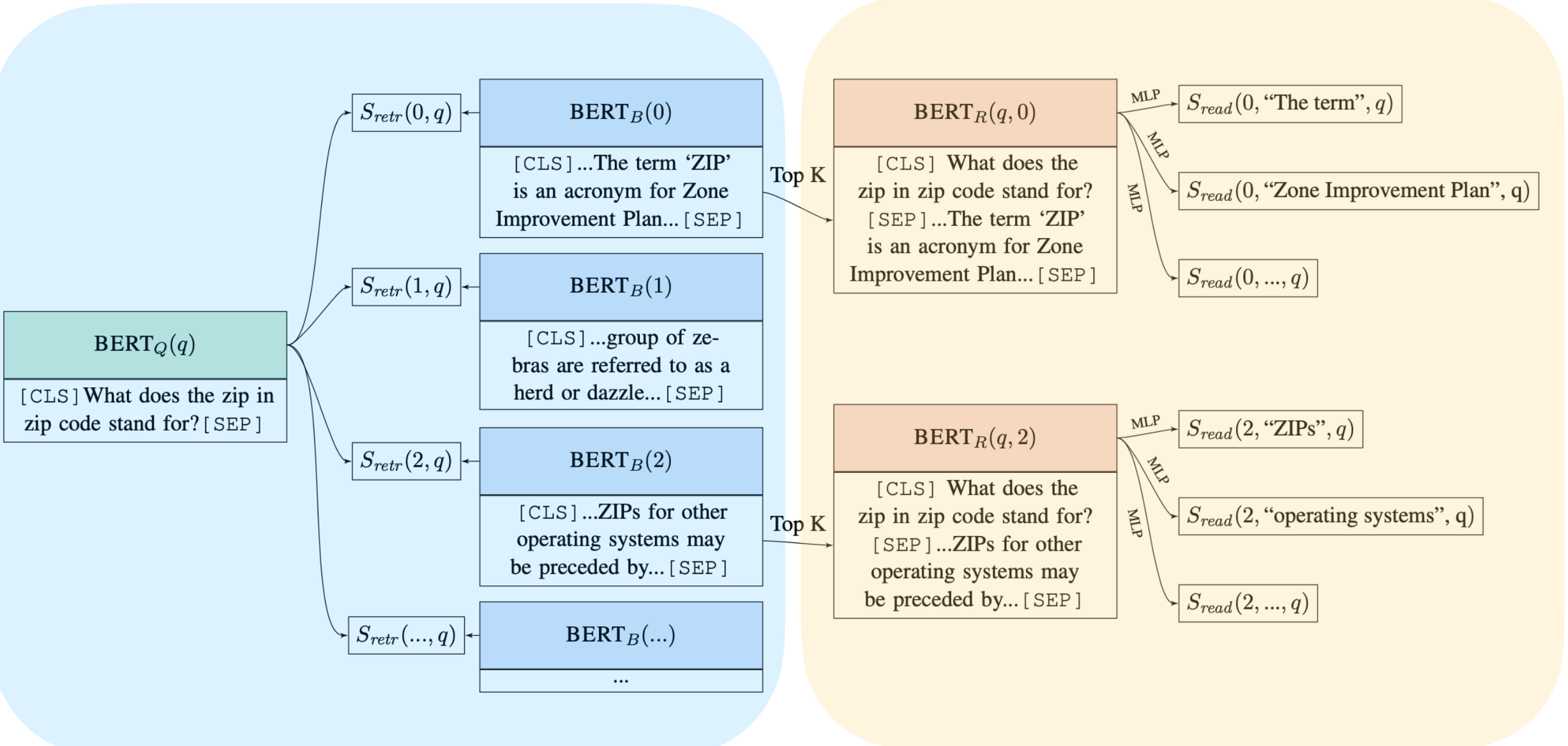
First model to learn retriever/reader jointly

Key insights:

- Both retriever and reader are learnable with NNs (= BERT)
- Only learned from question-answer pairs; No reading comprehension datasets! $\mathcal{D}_{\text{QA}} = \{(Q_i, A_i)\}$
- A new pre-training task called *Inverse Cloze Task (ICT)* to address the challenging retrieval problem.

Lee et al., 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering

ORQA: Open-Retriever Question Answering



Information Retrieval

Reading Comprehension

Lee et al., 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering

ORQA: Overview

Notations: b - block, s = a span of text within b, q = question

Fixed-length blocks as "passages": 288 wordpieces $\Longrightarrow$13M in total

Each block has 2000 possible answer spans

Modeling

$$S(b, s, q) = S_{retr}(b, q) + S_{read}(b, s, q)$$

Inference

$$a^* = \text{TEXT}(\underset{b,s}{\text{argmax}}\, S(b, s, q))$$

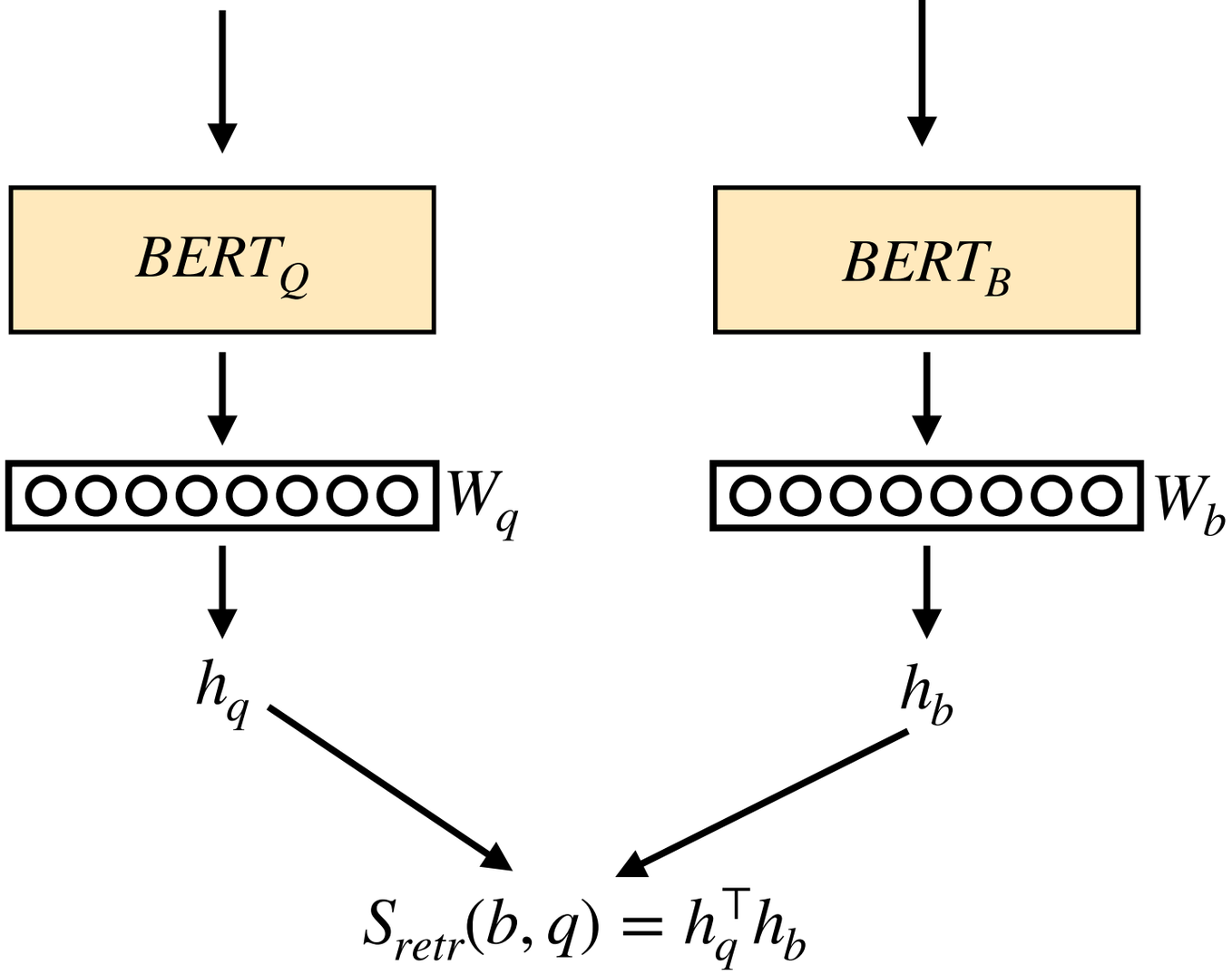
Lee et al., 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering

$$h_q = \mathbf{W_q}\mathbf{BERT}_Q(q)[\texttt{CLS}]$$
$$h_b = \mathbf{W_b}\mathbf{BERT}_B(b)[\texttt{CLS}]$$
$$S_{retr}(b,q) = h_q^\top h_b$$

Retriever score: $S_{retr}(b, q)$

Question q

What does the zip in zip code stand for?



All of Wikipedia: select top K

Each evidence block b

Reader score: $S_{read}(b, s, q)$

$$h_{start} = \text{BERT}_R(q, b)[\text{START}(s)]$$
$$h_{end} = \text{BERT}_R(q, b)[\text{END}(s)]$$
$$S_{read}(b, s, q) = \text{MLP}([h_{start}; h_{end}])$$

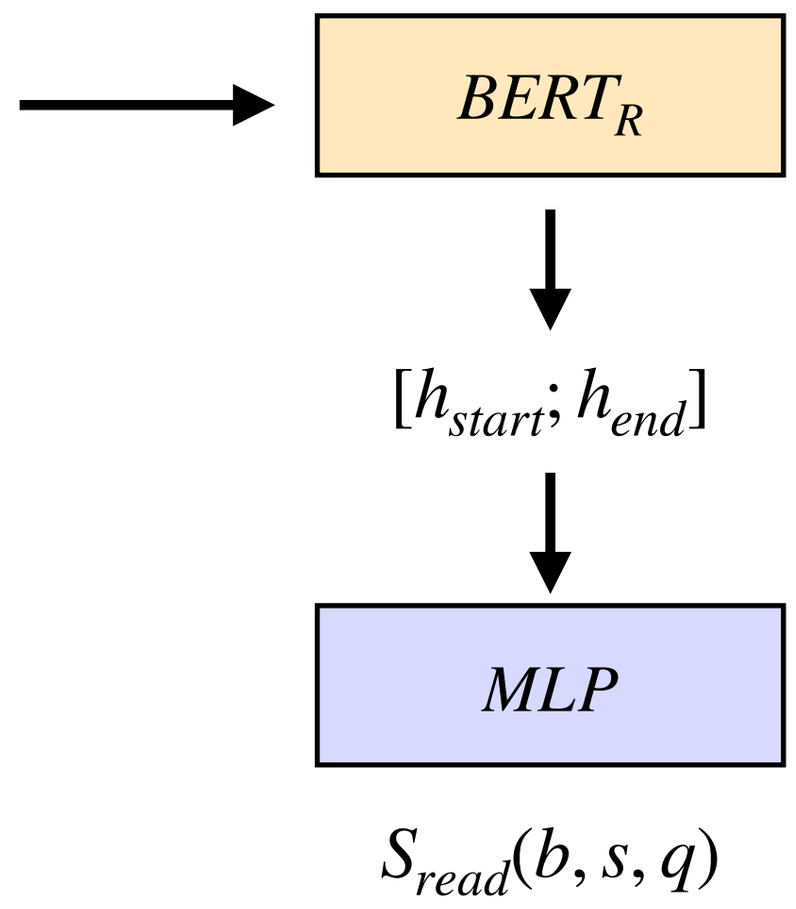
Top-K evidence blocks concatenated with question

Question q

Top Evidence Block 1

Question q

Top Evidence Block 2



Lee et al., 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering

How is this model learned?

Loss function

$$P(b, s|q) = \frac{\exp(S(b, s, q))}{\sum_{b' \in \text{TOP}(k)} \sum_{s' \in b'} \exp(S(b', s', q))}$$

How if top 5 (out of 29M) blocks don't contain the answer at all?

$$L_{\text{full}}(q, a) = -\log \sum_{b \in \text{TOP(k)}} \sum_{s \in b, a = \text{TEXT}(s)} P(b, s \mid q)$$

k=5, TOP(k) = the top k retrieved blocks according to $S_{(retr)}(b, q)$

How to update BERT_B for all the blocks??

Early learning: to consider a larger set of evidence blocks

$c = 5000$

$$P_{\text{early}}(b|q) = \frac{\exp(S_{retr}(b, q))}{\sum_{b' \in \text{TOP}(c)} \exp(S_{retr}(b', q))}$$

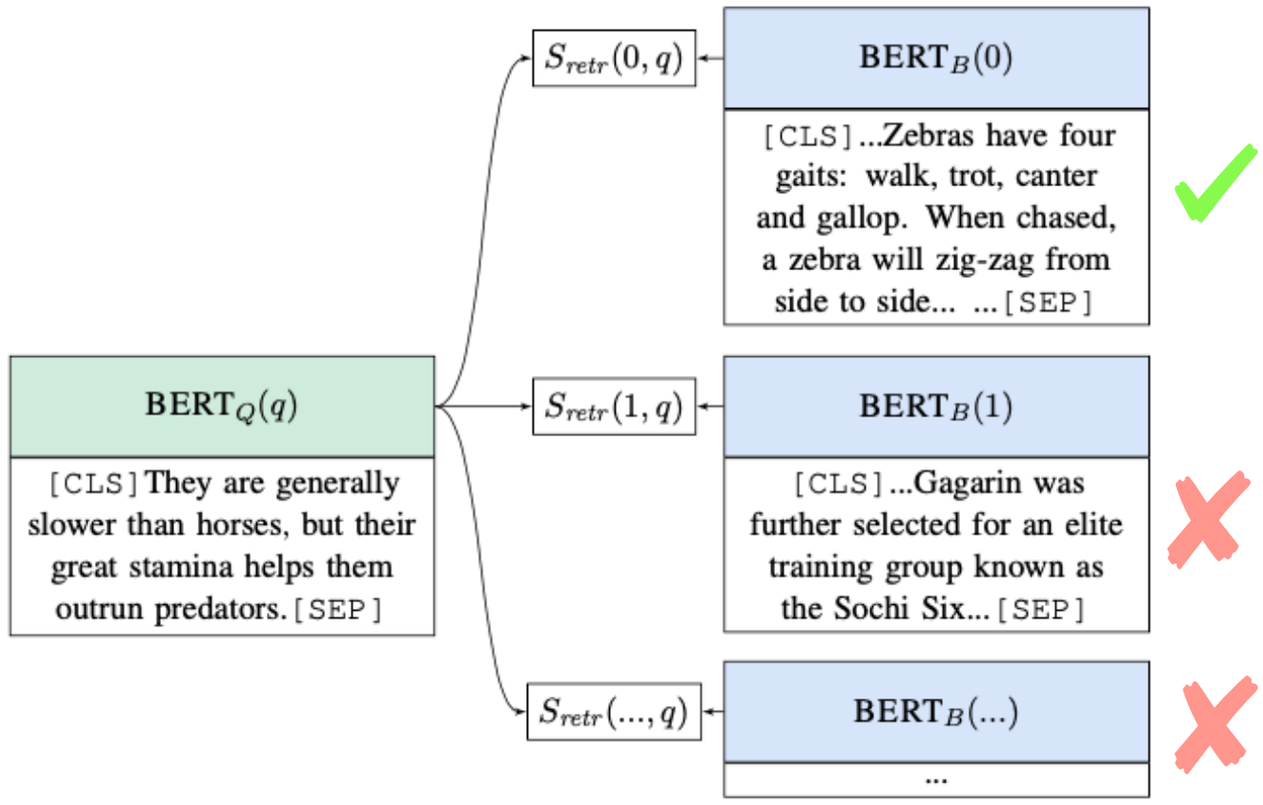
$$L_{\text{early}}(q, a) = -\log \sum_{b \in \text{TOP}(c),\ a \in \text{TEXT}(b)} P_{\text{early}}(b|q)$$

Lee et al., 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering

Pre-training: Inverse Cloze Task (ICT)

Key idea: a sentence is treated as a *pseudo-question*, and its context is treated as *pseudo-evidence*. The goal is to predict the correct context among a set of other random options.

...Zebras have four gaits: walk, trot, canter and gallop. **They are generally slower than horses, but their great stamina helps them outrun predators.** When chased, a zebra will zig- zag from side to side...



After pre-training, **BERT_B is fixed** and all the block representations can be pre-computed and search efficiently using existing tools (e.g., Locality Sensitive Hashing).

Lee et al., 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering

ORQA: experimental results

	Model	BM25 +BERT	ORQA
Test	Natural Questions	26.5	**33.3**
	WebQuestions	17.7	**36.4**
	CuratedTrec	21.3	**30.1**
	TriviaQA	**47.1**	45.0
	SQuAD	**33.2**	20.2

ORQA wins!

BM25 + BERT wins!

TriviaQA/SQuAD — question writer knows the answer

NQ/WebQ/TREC — “genuine information-seeking questions”

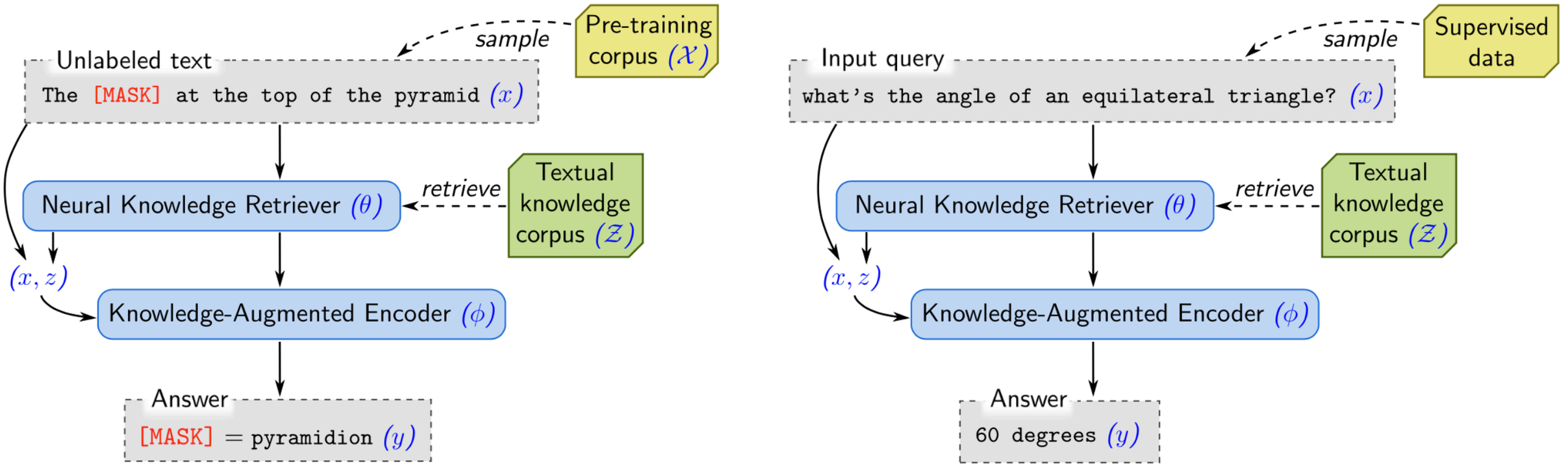
Dataset bias? SQuAD: only 536 documents, violating IID assumption?

Lee et al., 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering

REALM: improved pre-training [Guu et al., 2020]

Pre-training: masked language model (MLM)

Fine-tuning: QA



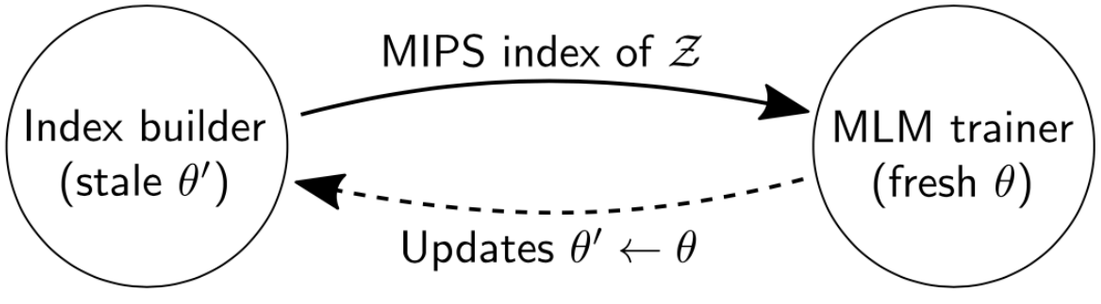
$$p(y \mid x) = \sum_{z \in \mathcal{Z}} p(y \mid z, x)\, p(z \mid x)$$

Top K documents

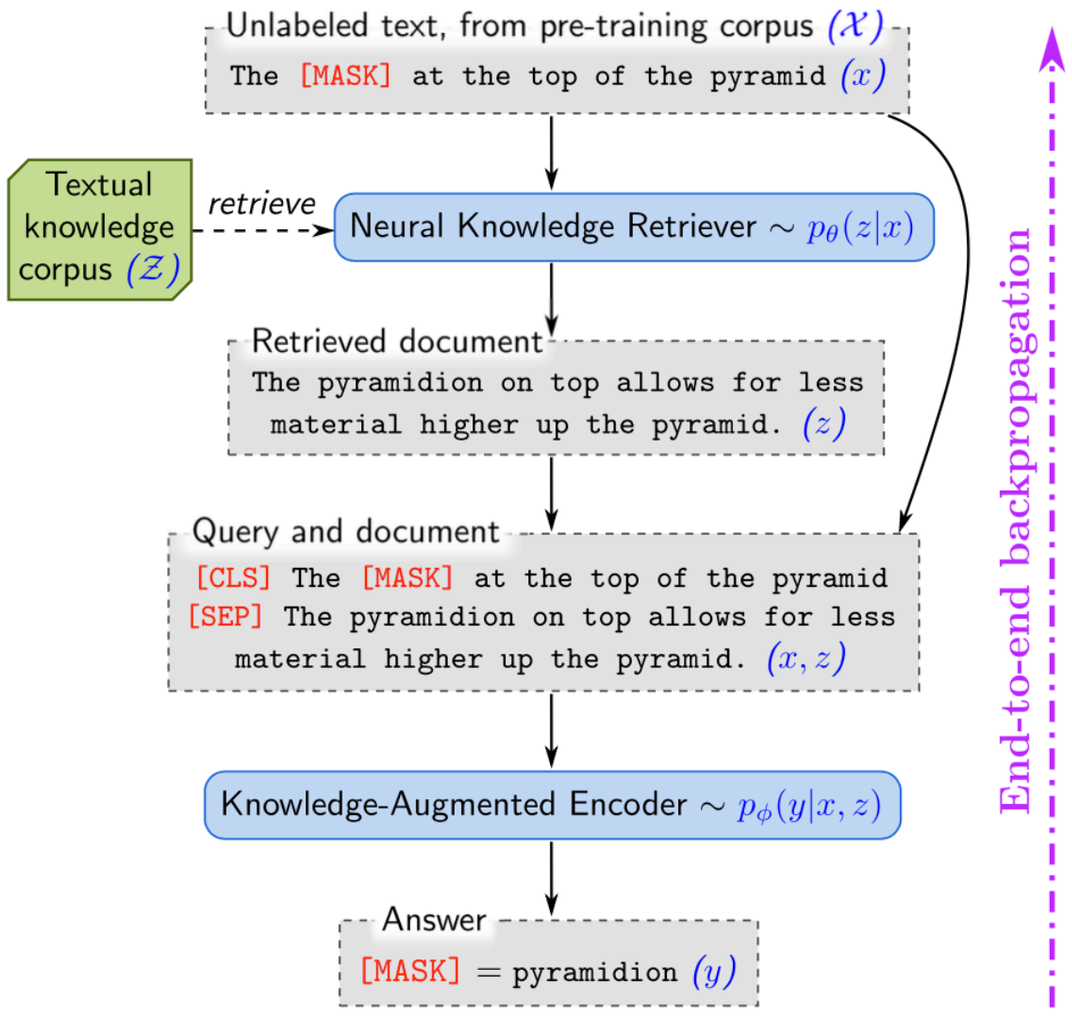
Guu et al., 2020. REALM: Retrieval-Augmented Language Model Pre-Training

REALM: improved pre-training

- Use ICT as a first-stage pre-training
- **Salient span masking**: similar to span masking [Joshi et al., 2020] but only mask named entities + dates.
- Can use corpora even larger than Wikipedia for pre-training.
- Can allow updating evidence encoder (BERT_B) asynchronously



MLM pre-training



ORQA vs REALM: 33.3 vs 40.4 on NQ

Guu et al., 2020. REALM: Retrieval-Augmented Language Model Pre-Training

Take-aways from ORQA/REALM

- It is possible to jointly train the retriever and reader, without any sparse IR components.

- The pre-training makes this retrieval process feasible and pre-training strategies matter.

- However, pre-training is very expensive.

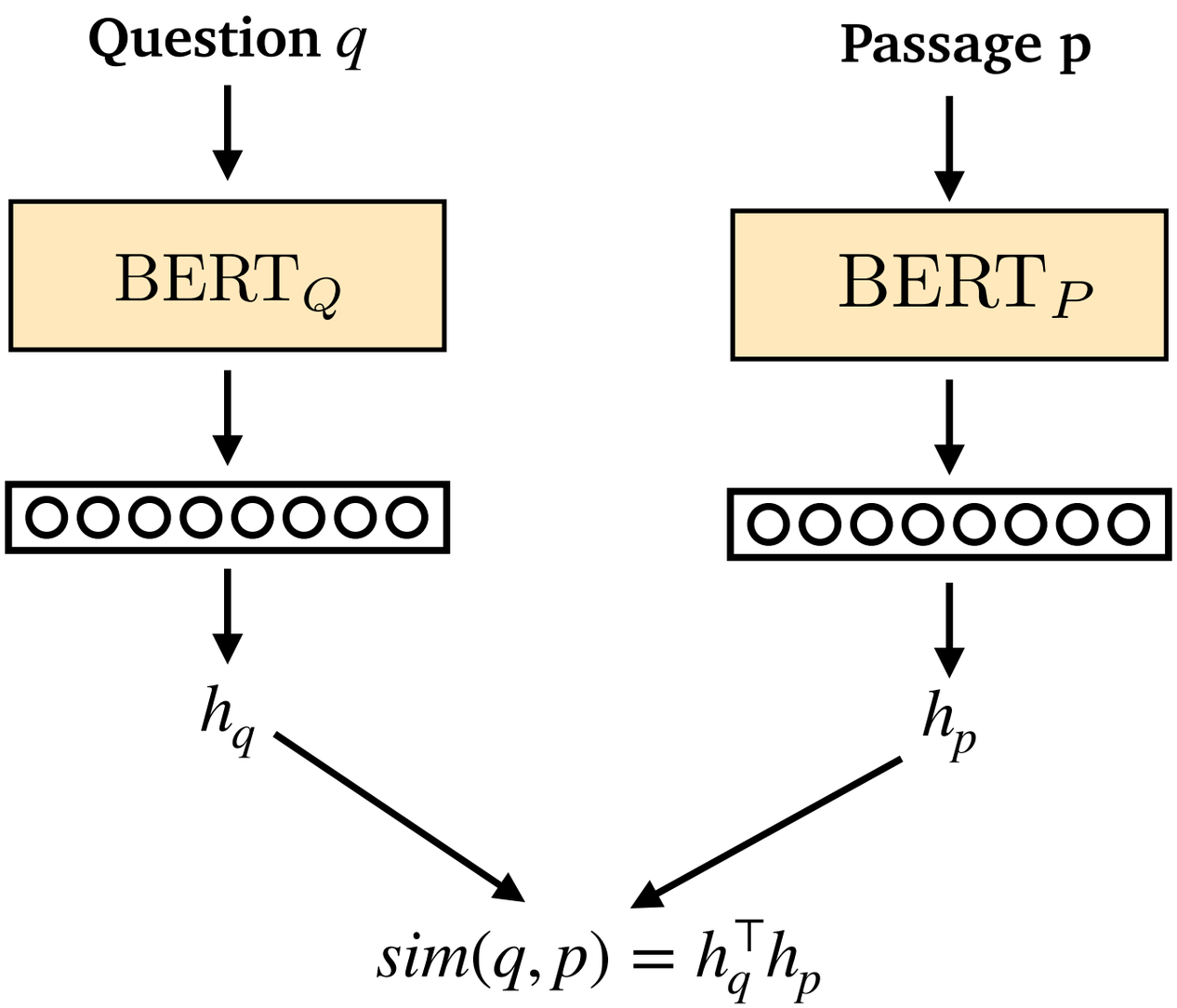
  "We pre-train for 200k steps on 64 Google Cloud TPUs, with a batch size of 512"

Next question: Is pre-training really necessary?

Dense Passage Retrieval (DPR)

[Karpukhin et al., 2020]

- **Key message**: you can train the dense-only retrieval only from a small number of Q/A pairs, without any pre-training!



How to get positives and negatives?

$$\mathcal{D} = \{\langle q_i, p_i^+, p_{i,1}^-, \cdots, p_{i,n}^- \rangle\}_{i=1}^m$$

$$L(q_i, p_i^+, p_{i,1}^-, \cdots, p_{i,n}^-) = -\log \frac{e^{\text{sim}(q_i, p_i^+)}}{e^{\text{sim}(q_i, p_i^+)} + \sum_{j=1}^n e^{\text{sim}(q_i, p_{i,j}^-)}}$$

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

DPR: training examples

Positives

(1) Provided in the reading comprehension datasets

(2) Passages of high BM25 that contain the answer string

Negatives

(1) Random passages from the corpus

(2) Passages of high BM25 scores that DO NOT contain the answer string

(3) Positive passages of OTHER questions

The best model uses (3) from the same mini-batch [in-batch negatives] and one passages from (2) [hard negatives].

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

DPR: in-batch negatives

- A small trick to effective generate more training pairs
- Suppose we have n pairs of relevant questions and passages. Let $Q_{d\times n}$ and $P_{d\times n}$ be the question and passage embeddings.
- $S = Q^\intercal P$ is a n-by-n matrix of the similarity scores. Scores of n^2 pairs of questions and passages. For each question, 1 positive passage and $n-1$ negative passages

in-batch negatives

Type	#N	IB	Top-5	Top-20	Top-100
Random	7	✗	47.0	64.3	77.8
BM25	7	✗	50.0	63.3	74.8
Gold	7	✗	42.6	63.1	78.3
Gold	7	✓	51.1	69.1	80.8
Gold	31	✓	52.1	70.8	82.1
Gold	127	✓	55.8	73.0	83.1
G.+BM25$^{(1)}$	31	✓	65.0	77.3	84.4
G.+BM25$^{(2)}$	31	✓	64.5	76.4	84.0
G.+BM25$^{(1)}$	127	✓	**65.8**	**78.0**	**84.9**

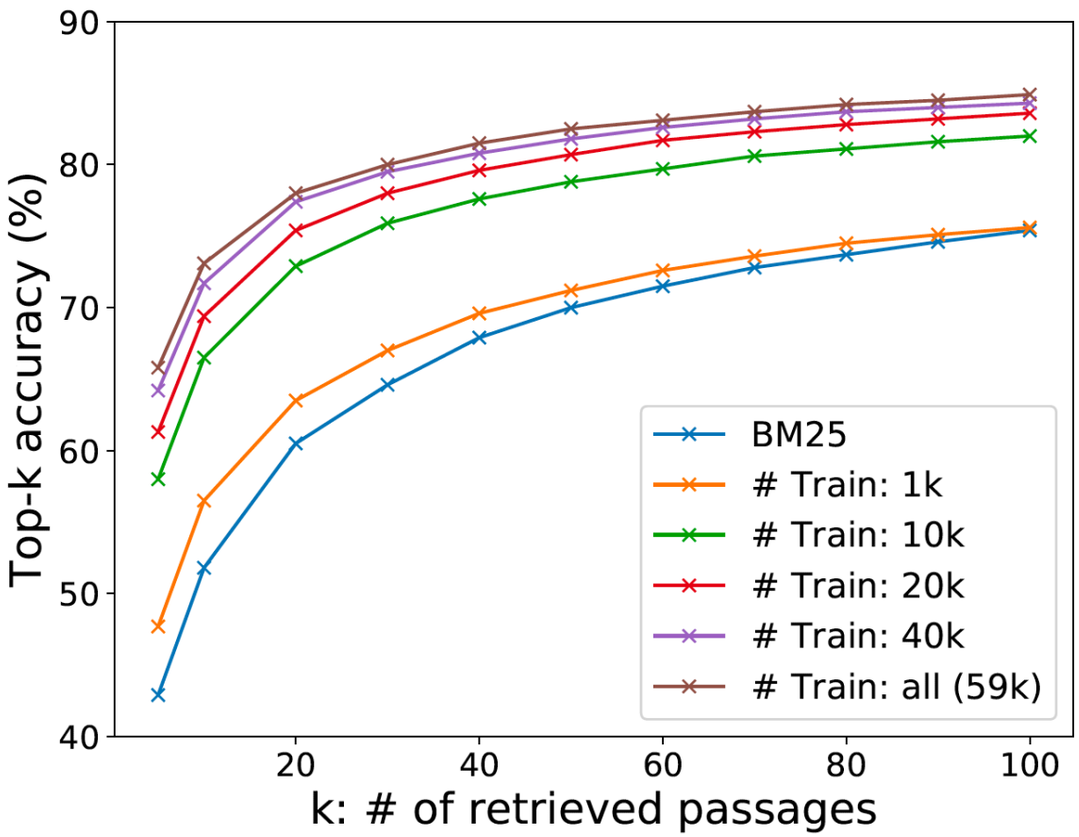
Retriever performance

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

DPR: experimental results

Retriever performance on NQ

1k Q/A pairs beat BM25!



End-to-end QA performance

With a multi-passage BERT reader trained on the retrieved passages from the retriever:

- DPR is better than BM25 on NaturalQuestions, WebQuestions, TREC, TriviaQA but not SQuAD.
- DPR is better than REALM on bigger datasets (41.5 vs 39.2 on NQ). However, for smaller datasets (WebQ, TREC), it needs a mixed training with bigger datasets to outperform REALM.

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

Take-aways from DPR

- No need of expensive pre-training?
  - At least for modest-sized QA datasets

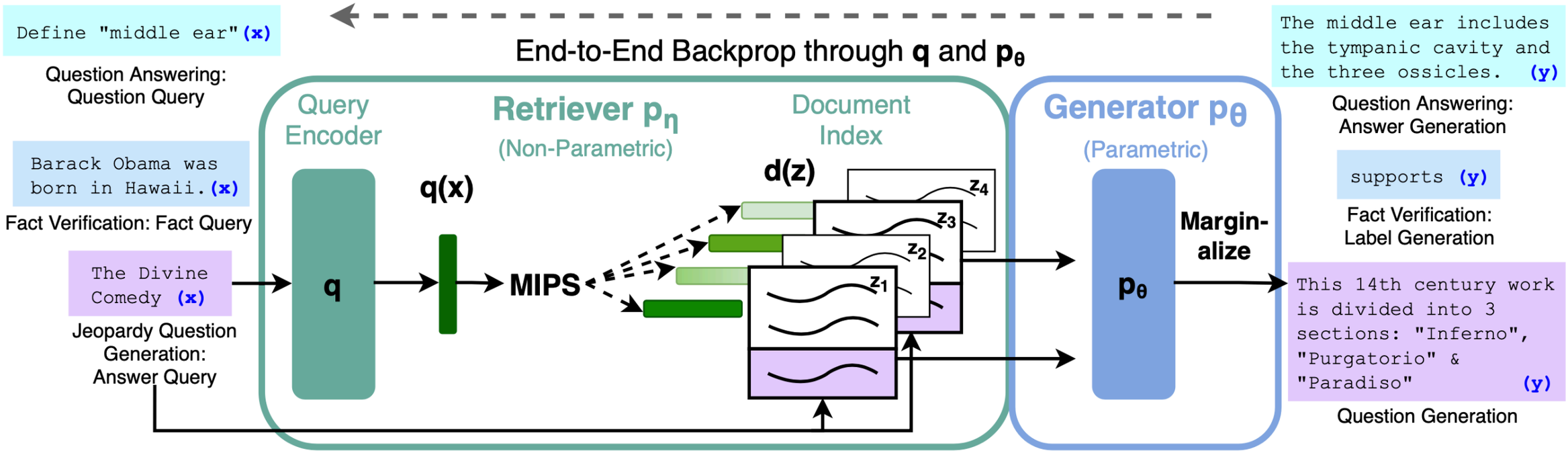
- **Joint** training vs **pipeline** training of retriever and reader?
  - Joint training didn't help DPR (41.5 → 39.8 on NQ)
  - Pipeline training is much simpler. You only build the indexing once!

- **Reading comprehension** vs **QA** datasets
  - Doesn't make a much difference (41.5 vs 41.0 on NQ).
  - You can train the system using only Q/A pairs!

- BM25 + DPR is slightly better than DPR but the difference is small.

Retrieval-Augmented Generation (RAG)

[Lewis et al., 2020]



$$p_{\text{RAG-Sequence}}(y|x) = \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) \prod_i^N p_\theta(y_i|x, z, y_{1:i-1})$$

retrieval model

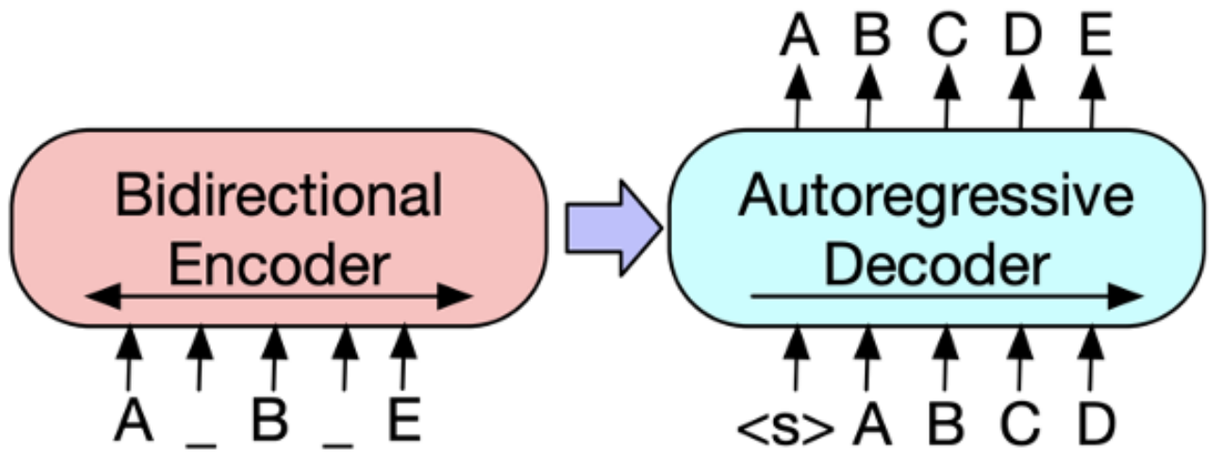
seq2seq model

Lewis et al., 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

Retrieval-Augmented Generation (RAG)

- Retrieval model: $p(z \mid x)$ = Dense passage retrieval (DPR)
- seq2seq model: $p(y \mid x, z)$ = BART [Lewis et al., 2020]

  Pre-trained on large text corpora, simply concatenates z to question x



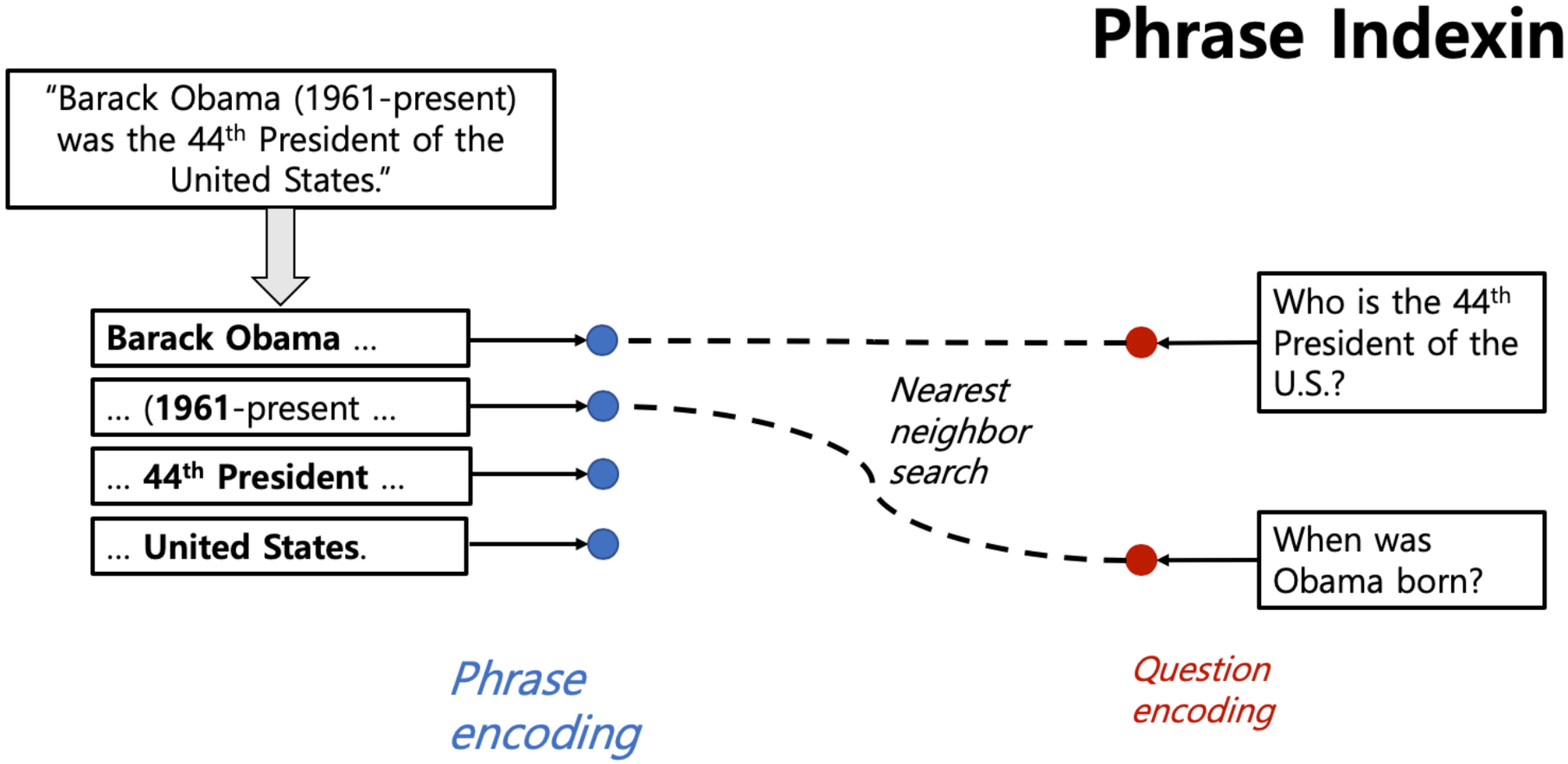
- Joint training of the two components
- Improved performance on open-domain QA evaluation and easy to extend to other generation tasks

Lewis et al., 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

DenSPI: Dense-Sparse Phrase Index

[Seo et al., 2019]

Key question: is it possible to encode/index **at phrase level** instead of paragraphs or documents so retriever + reader will be reduced to a harder "retriever" problem?



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

DenSPI: Challenges

Decomposability gap: question and passage/answer have to be encoded independently

Model phrase question document

$$\hat{a} = \underset{a}{\text{argmax}}\, F_\theta(a, q, d)$$

We know modeling "attention" between question and passage is important..

Query-Agnostic Decomposition

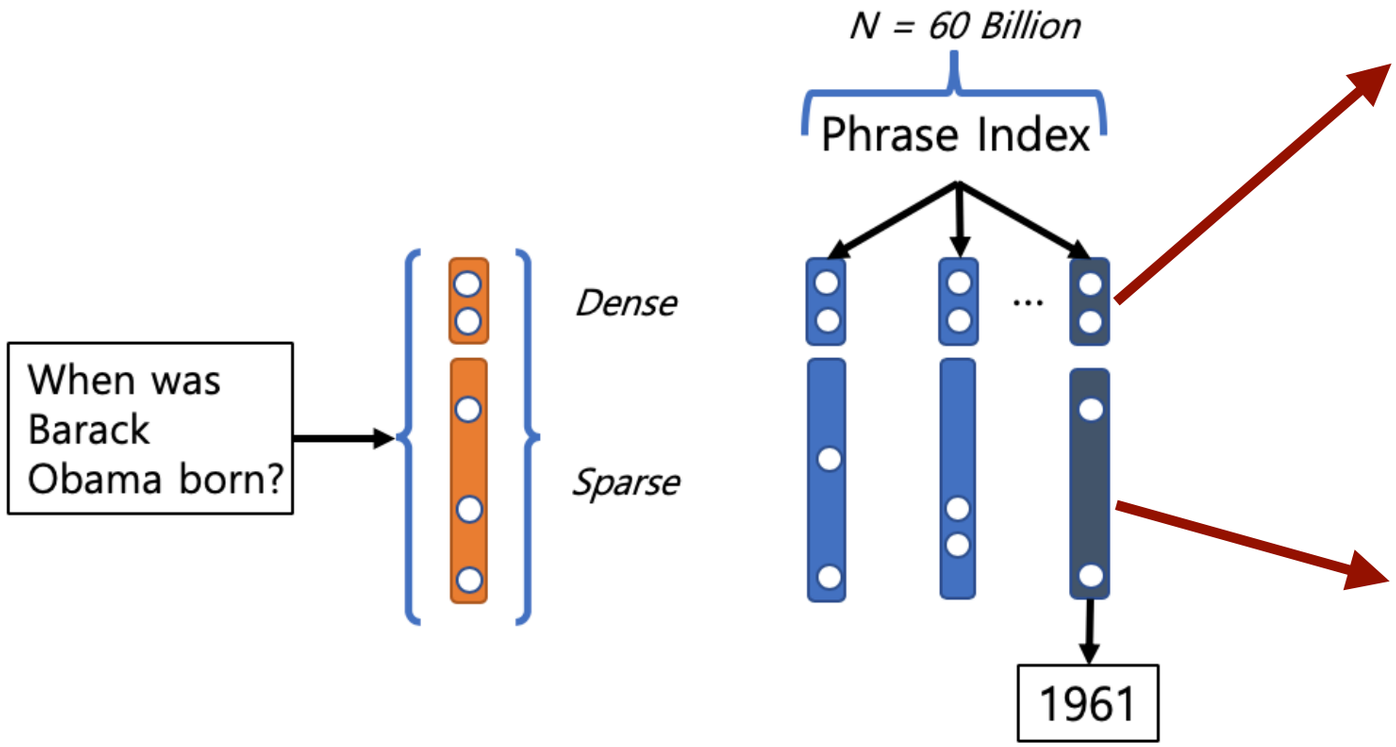
$$\hat{a} = \underset{a}{\text{argmax}}\, G_\theta(q) \cdot H_\theta(a, d)$$

Question encoder Phrase encoder

Scalability: *60 billion phrases* in Wikipedia! Storage, indexing and search all remain challenging.

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

DenSPI: dense & sparse representations



Dense representation

Start and end vectors from BERT + coherency scalar

Sparse representation

TF-IDF vector for the corresponding document & paragraph vector

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

DenSPI: experimental results

Storage: pointer, filter, scalar quantization: 240T → 1.5T

Search: Dense-first search (LSH), sparse-first search (TF-IDF search), Hybrid

Performance on SQuAD

	F1	EM	s/Q
DrQA	-	29.8	35
R^3	37.5	-	-
Paragraph ranker	-	30.2	-
Multi-step reasoner	39.2	31.9	-
MINIMAL	42.5	34.7	-
BERTserini	46.1	38.6	115
Weaver	-	42.3	-
DENSPI-SFS	42.5	33.3	0.60
DENSPI-DFS	35.9	28.5	0.51
–sparse scale=0	16.3	11.2	0.40
DENSPI-Hybrid	44.4	36.2	0.81

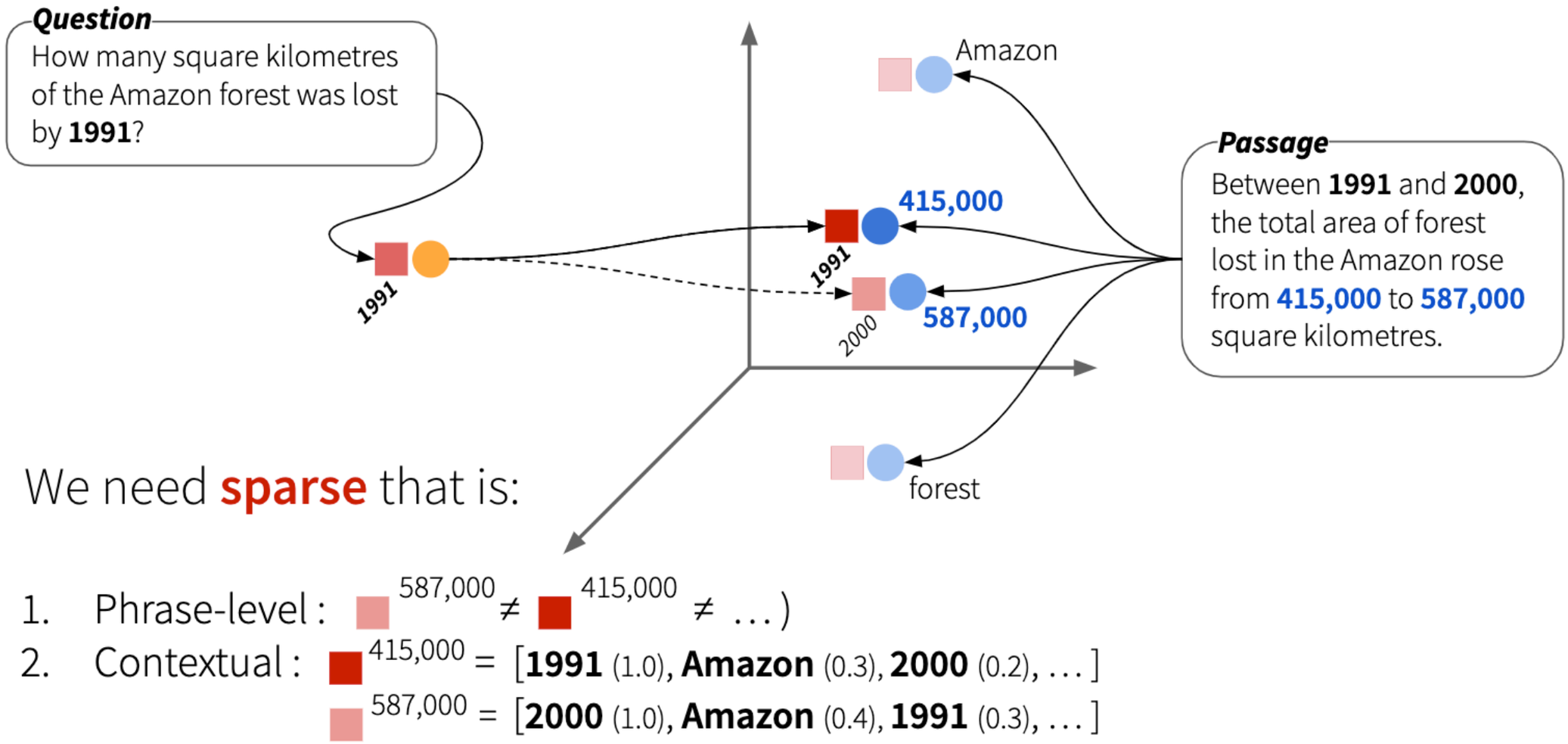
Removing sparse vector performs much worse

Very fast inference time

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

DenSPI: contextualized sparse representations

[Lee et al., 2020]



- **Sparc:** use different weights for different sparse terms
- Improved DenSPI by +4.1% on TREC and +4.5% on SQuAD

Lee et al., 2020. Contextualized Sparse Representations for Real-Time Open-Domain Question Answering

Comparison of all models

		SQuAD	TriviaQA	TREC	WebQ	NaturalQ
2017/04	DrQA	28.4	-	25.7	19.5	-
2017/08	R^3	29.1	-	28.4	17.1	-
2018/04	DrQA*	40.4	-	28.8	24.3	-
2019/02	BERTserini	38.6	-	-	-	-
2019/08	Multi-passage BERT@	**53.0**	-	-	-	-
2019/09	Hard-EM	-	50.9	-	-	28.1
2019/06	ORQA	20.2	45.0	30.1	36.4	33.3
2020/02	REALM (Wiki)	-	-	46.8	40.2	39.2
2020/02	REALM (CC-News)	-	-	42.9	40.7	40.4
2020/04	DPR	29.8	56.8	49.4*	42.4*	41.5
2020/05	RAG@	-	56.1	**52.2***	**45.2***	**44.5**
2019/06	DenSPI	36.2	-	31.6#	-	-
2019/11	DenSPI + Sparc	40.7	-	35.7#	-	-
2020/04	BM25 + DPR	36.7	**57.0**	50.6*	41.1*	39.0

#: no supervision using target training data / *: jointly trained with bigger datasets (e.g., NQ) / @: BERT-large models

Summary

- ORQA/REALM: First demonstrate that it is possible jointly train the retriever and reader without any sparse IR components; requires novel ways of pre-training

- DPR/RAG: It is possible to learn the dense retrieval only from Q/A pairs without pre-training!

- DenSPI /SParc: It is possible to index and retrieve at phrase level without requiring an explicit “reader”. Very fast inference time but slightly worse performance. Sparse features still matter.

Part VI

Retrieval-free Approaches

No explicit retriever?

- Key question: can we use **pre-trained language models** to act as "knowledge storage"?

- Instead of explicitly storing all the text and searching among their *dense* or *sparse* representations, can we query the LMs to obtain the answer directly?

- The LMs were pre-trained on Wikipedia (and other textual corpora) so they should be able to memorize a fair amount of information.

LMs as KBs?

Barack Obama was born in Honolulu.

Petroni et al., 2019. Language Models as Knowledge Bases?

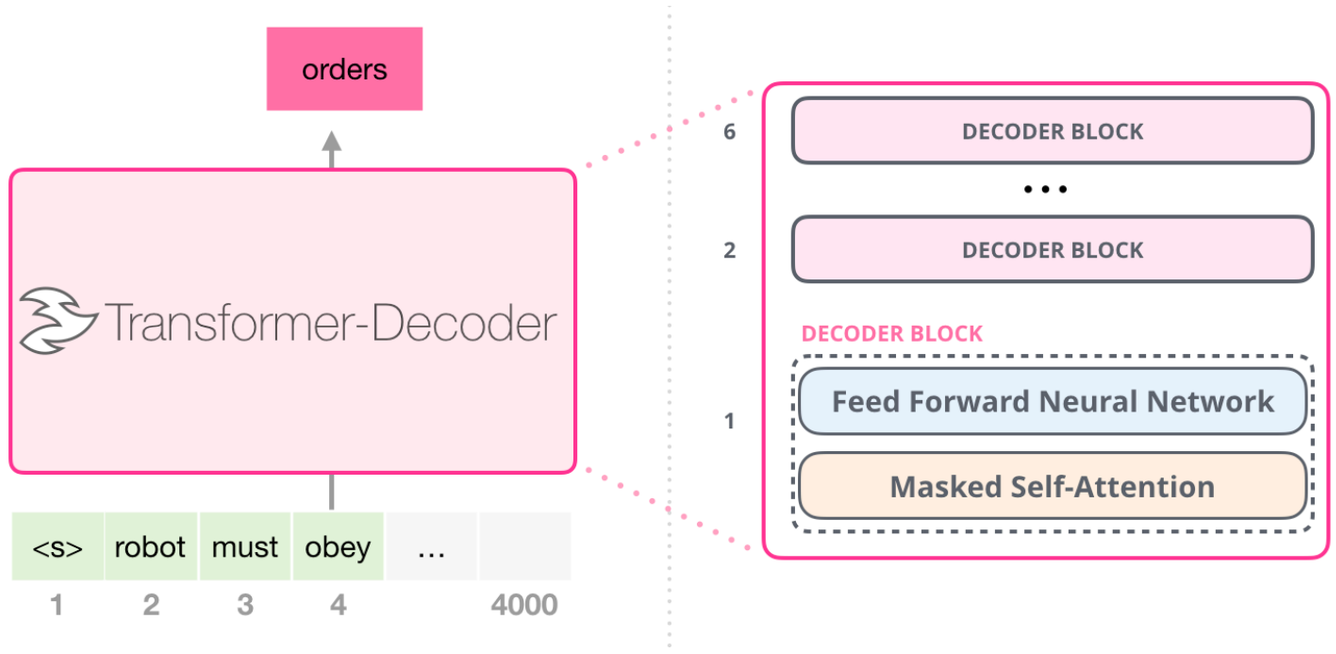
GPT-2 [Rardford et al., 2019]

- GPT-2 is a *very large*, transformer-based language model trained on *a massive dataset*.

48 layers, hidden size 1600, 1.5B parameters

WebText: 8 million documents, excluding Wikipedia (!)

$$p(x) = \prod_{i=1}^{n} p(s_n | s_1, \ldots, s_{n-1})$$



Radford et al., 2019. Language Models are Unsupervised Multitask Learners
http://jalammar.github.io/illustrated-gpt2/

GPT-2: zero-shot QA

Evaluated on Natural Questions and **no training at all**

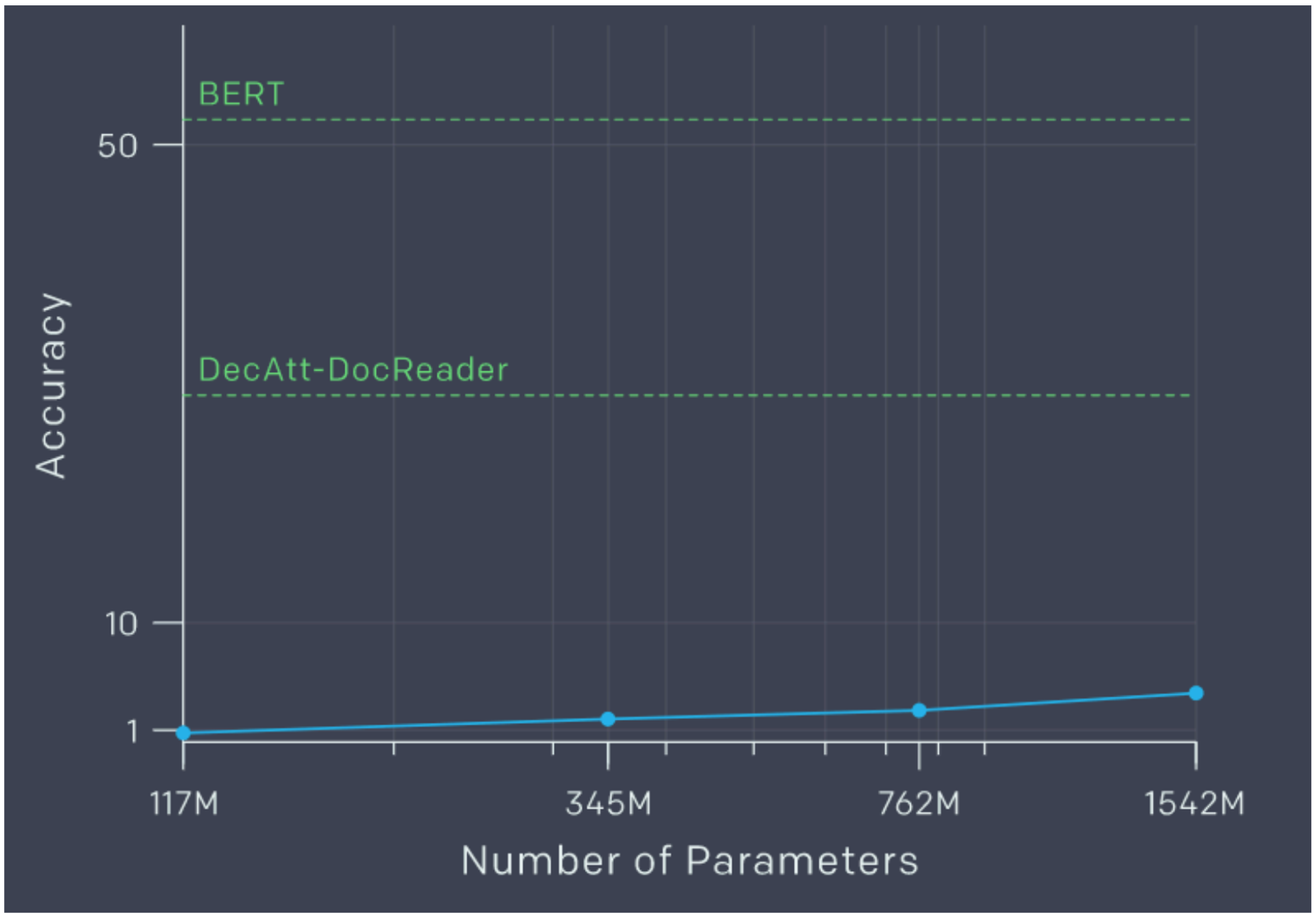
63.1% on the **1%** of questions it is most confident in

Question	Generated Answer	Correct	Probability
Who wrote the book the origin of species?	Charles Darwin	✓	83.4%
Who is the founder of the ubuntu project?	Mark Shuttleworth	✓	82.0%
Who is the quarterback for the green bay packers?	Aaron Rodgers	✓	81.1%
Panda is a national animal of which country?	China	✓	76.8%
Who came up with the theory of relativity?	Albert Einstein	✓	76.4%
When was the first star wars film released?	1977	✓	71.4%
What is the most common blood type in sweden?	A	✗	70.6%
Who is regarded as the founder of psychoanalysis?	Sigmund Freud	✓	69.3%
Who took the first steps on the moon in 1969?	Neil Armstrong	✓	66.8%
Who is the largest supermarket chain in the uk?	Tesco	✓	65.3%
What is the meaning of shalom in english?	peace	✓	64.0%
Who was the author of the art of war?	Sun Tzu	✓	59.6%
Largest state in the us by land mass?	California	✗	59.2%
Green algae is an example of which type of reproduction?	parthenogenesis	✗	56.5%
Vikram samvat calender is official in which country?	India	✓	55.6%
Who is mostly responsible for writing the declaration of independence?	Thomas Jefferson	✓	53.3%
What us state forms the western boundary of montana?	Montana	✗	52.3%
Who plays ser davos in game of thrones?	Peter Dinklage	✗	52.1%
Who appoints the chair of the federal reserve system?	Janet Yellen	✗	51.5%
State the process that divides one nucleus into two genetically identical nuclei?	mitosis	✓	50.7%
Who won the most mvp awards in the nba?	Michael Jordan	✗	50.2%
What river is associated with the city of rome?	the Tiber	✓	48.6%
Who is the first president to be impeached?	Andrew Johnson	✓	48.3%
Who is the head of the department of homeland security 2017?	John Kelly	✓	47.0%
What is the name given to the common currency to the european union?	Euro	✓	46.8%
What was the emperor name in star wars?	Palpatine	✓	46.5%
Do you have to have a gun permit to shoot at a range?	No	✓	46.4%
Who proposed evolution in 1859 as the basis of biological development?	Charles Darwin	✓	45.7%
Nuclear power plant that blew up in russia?	Chernobyl	✓	45.7%
Who played john connor in the original terminator?	Arnold Schwarzenegger	✗	45.2%

Radford et al., 2019. Language Models are Unsupervised Multitask Learners

GPT-2: zero-shot QA

Evaluated on Natural Questions and **no training at all**



4% accuracy:
Much much worse than supervised systems

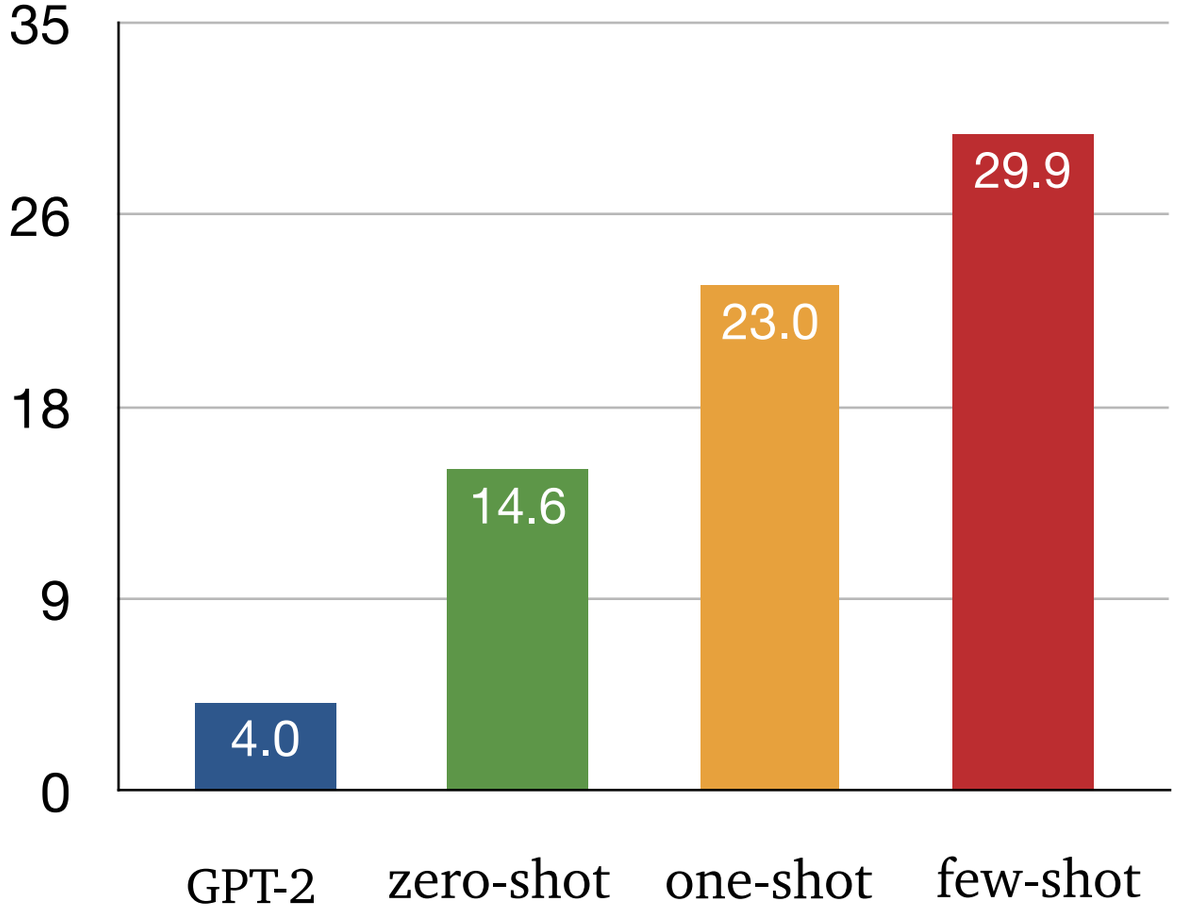
Radford et al., 2019. Language Models are Unsupervised Multitask Learners

GPT-3: Few-shot Learner [Brown et al., 2020]

96 layers, hidden size 12288, **175B** parameters

Larger corpora: Common Crawl + WebText + Books + English Wikipedia

Evaluated on Natural Questions:



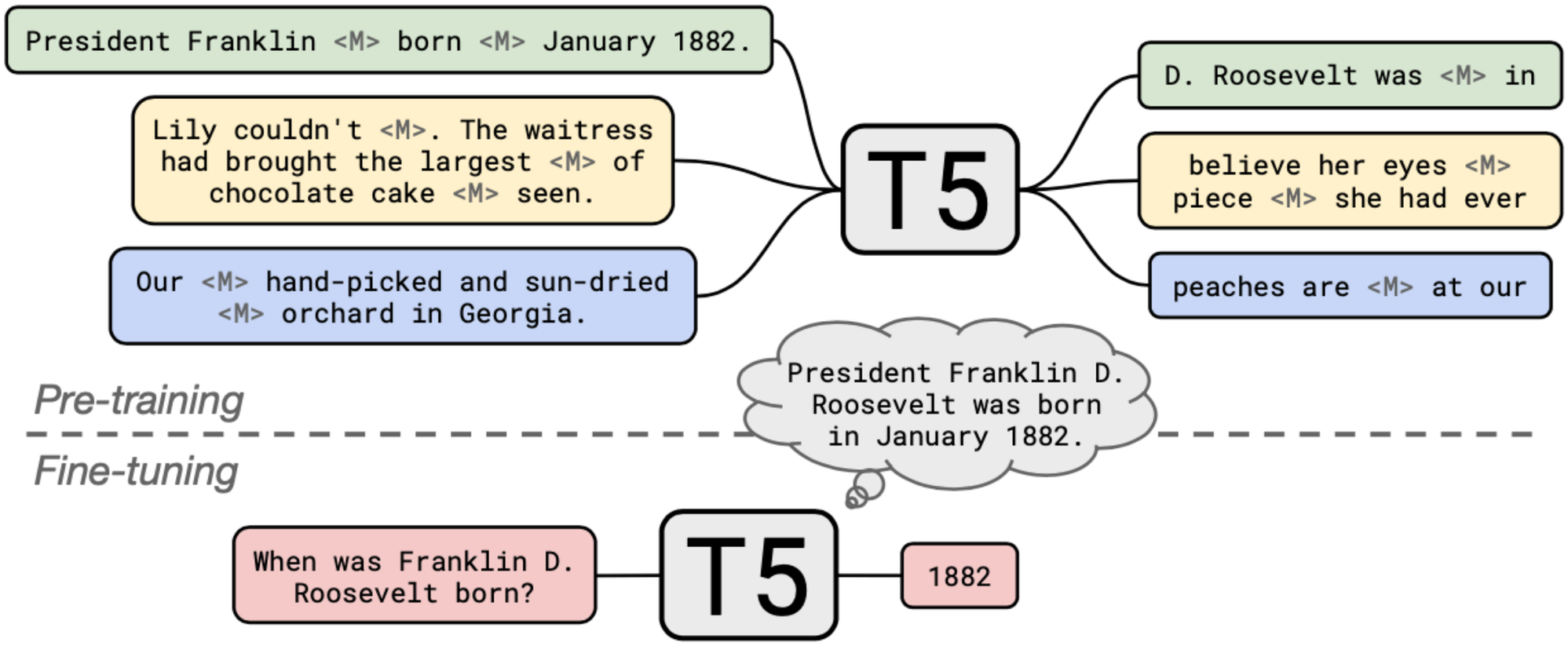
Few-shot learner

- No weight updates
- Only augment the prompt with **K** (question, answer) pairs as "demonstration"
- One-shot setting is a special case when only **one** example is given.

Brown et al., 2020. Language Models are Few-Shot Learners

T5: Fine-tuning leads to improved performance
[Roberts et al., 2020]

Text-to-Text Transfer Transformer [Raffel et al., 2019]



*: Pre-trained on a multitask mixture including an **unsupervised "span corruption" task** on unlabeled text as well as supervised translation, summarization, classification, and reading comprehension tasks

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

T5: Fine-tuning leads to improved performance

		Natural Questions	WebQuestions	TriviaQA
		NQ	WQ	TQA
	Chen et al. (2017)	–	20.7	–
	Lee et al. (2019)	33.3	36.4	47.1
	Min et al. (2019a)	28.1	–	50.9
	Min et al. (2019b)	31.8	31.6	55.4
	Asai et al. (2019)	32.6	–	–
	Ling et al. (2020)	–	–	35.7
	Guu et al. (2020)	40.4	40.7	–
	Févry et al. (2020)	–	–	53.4
	Karpukhin et al. (2020)	**41.5**	42.4	57.9
220M	T5-Base	27.0	29.1	29.1
770M	T5-Large	29.8	32.2	35.9
3B	T5-3B	32.1	34.9	43.4
11B	T5-11B	34.5	37.4	50.1
	T5-11B + SSM	36.6	**44.7**	**60.5**

BERT-base = 110M parameters

SSM: salient span masking, pre-training data proposed in REALM

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

Summary

- Large language models pre-trained on unstructured text can attain competitive results in open-domain QA without accessing external knowledge.
- The performance is largely impacted by the model size. A 11B T5 model is able to match the performance with DPR with 3 BERT-base models (220M parameters each).

NeurIPS'20 EfficientQA Competition

How should we store the "knowledge" for use by our open-domain QA system?

Passages, databases or parameters of NNs?

We are looking for the systems (evaluated on Natural Questions):

- Most accurate self-contained QA system under 6Gb
- Most accurate self-contained QA system under 500Mb
- Smallest self-contained QA system that achieves 25% accuracy
- Most accurate QA system with no constraints

Baselines: TF-IDF, DPR and T5

Important Dates

July, 2020	Leaderboard launched.
October 14, 2020	Leaderboard frozen.
November 14, 2020	Human evaluation completed and winners announced.
December 11-12, 2020	NeurIPS workshop and human-computer competition (held virtually).

https://efficientqa.github.io/

Part VII

Open-Domain Question Answering using Text & **Knowledge Bases**

Large-scale Factual Knowledge Bases

Google Knowledge Graph

Microsoft 悟 SATORI

NELL

Freebase

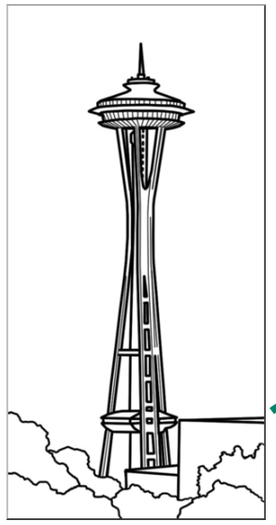
DBpedia

yago select knowledge

WIKIDATA

- Database that stores a large number of facts in an organized way
  - Freebase: 46m entities, 2.6b facts
  - WikiData: 87m items
- Most knowledge bases are curated

Entity-centric Knowledge Graph



Address: 400 Broad St, Seattle, 98109
Phone: (800) 937-9582
Opened: Apr 21, 1962
Height: 605 feet (184.41 m)
Floors: 6

Super Bowl championships: 2013
Head coach: Pete Carroll
Founded: 1976
Division: NFC West

Location

Home Field

Seattle

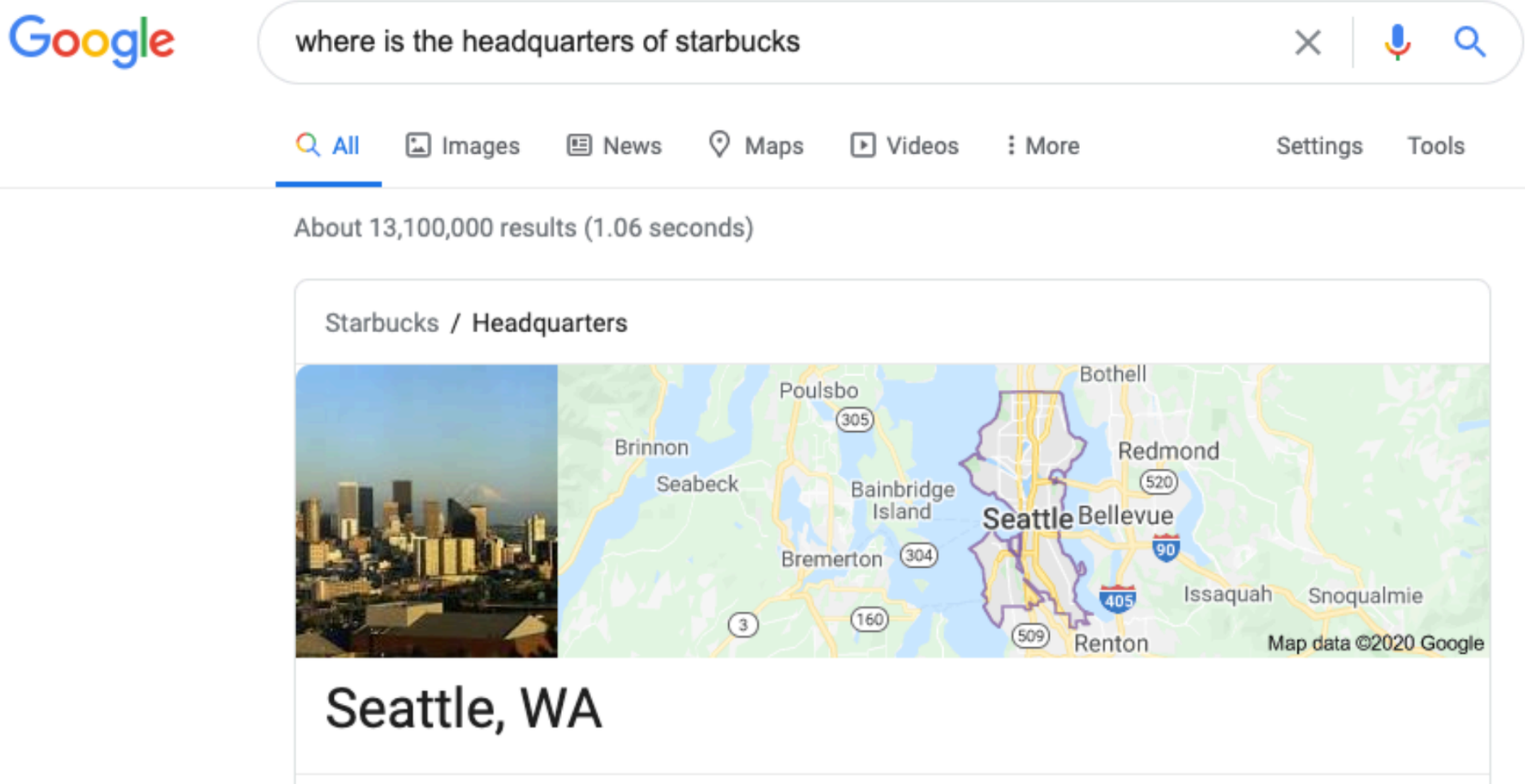
Population: 744,955 (2018)
Area: 142.07 sq miles (369.20 km²)
Mayor: Jenny Durkan

Headquarters

Founded: Mar 30, 1971 · Pike Place Market
Customer service: +1 800-782-7282
CEO: Kevin Johnson
Founders: Jerry Baldwin · Zev Siegl · Gordon Bowker

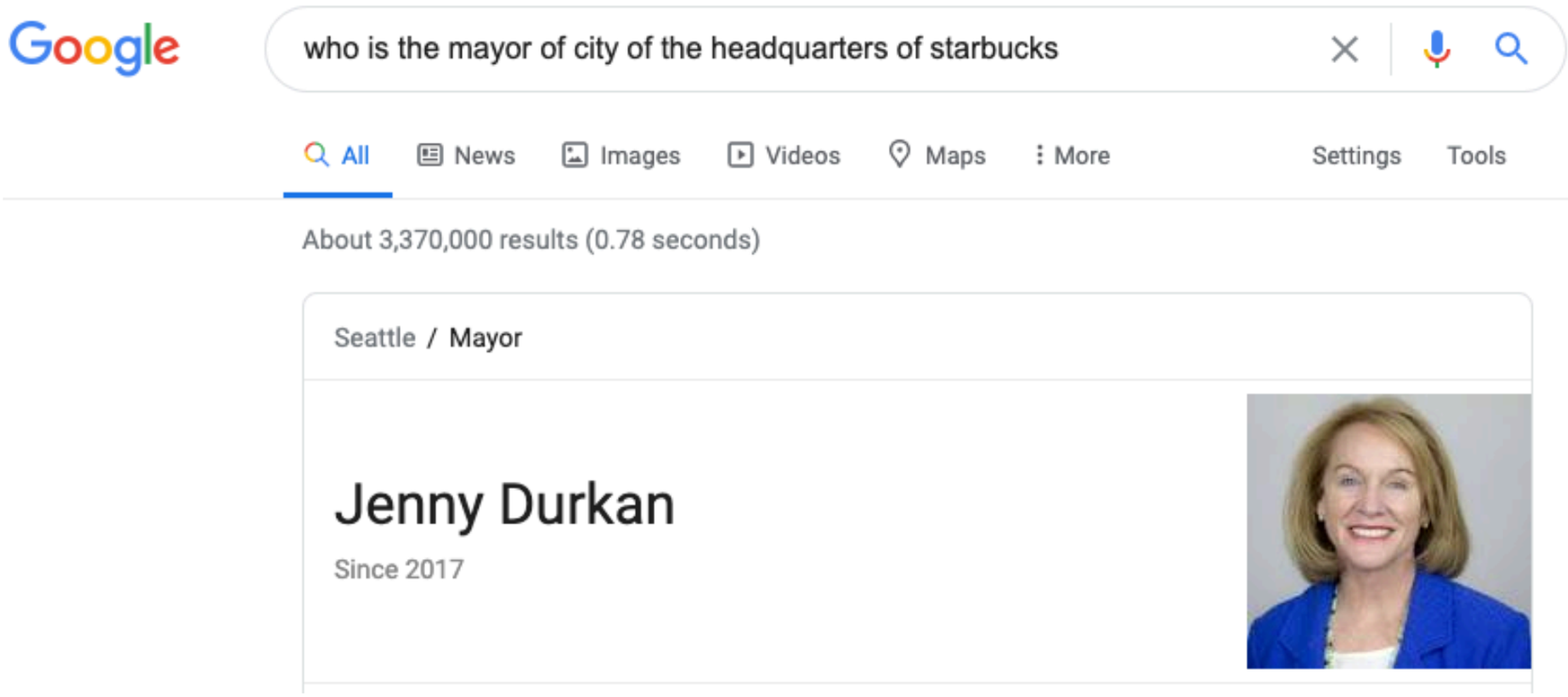
Open-Domain Question Answering using Knowledge Base

- Curated KB ensures the correctness of the information (answer)
- Common or "simple" questions can be answered easily
  - If semantic parsing (question → query) is done correctly



Open-Domain Question Answering using Knowledge Base

- Curated KB ensures the correctness of the information (answer)
- Common or "simple" questions can be answered easily
  - If semantic parsing (question → query) is done correctly
  - Graph structure enables multi-hop question answering



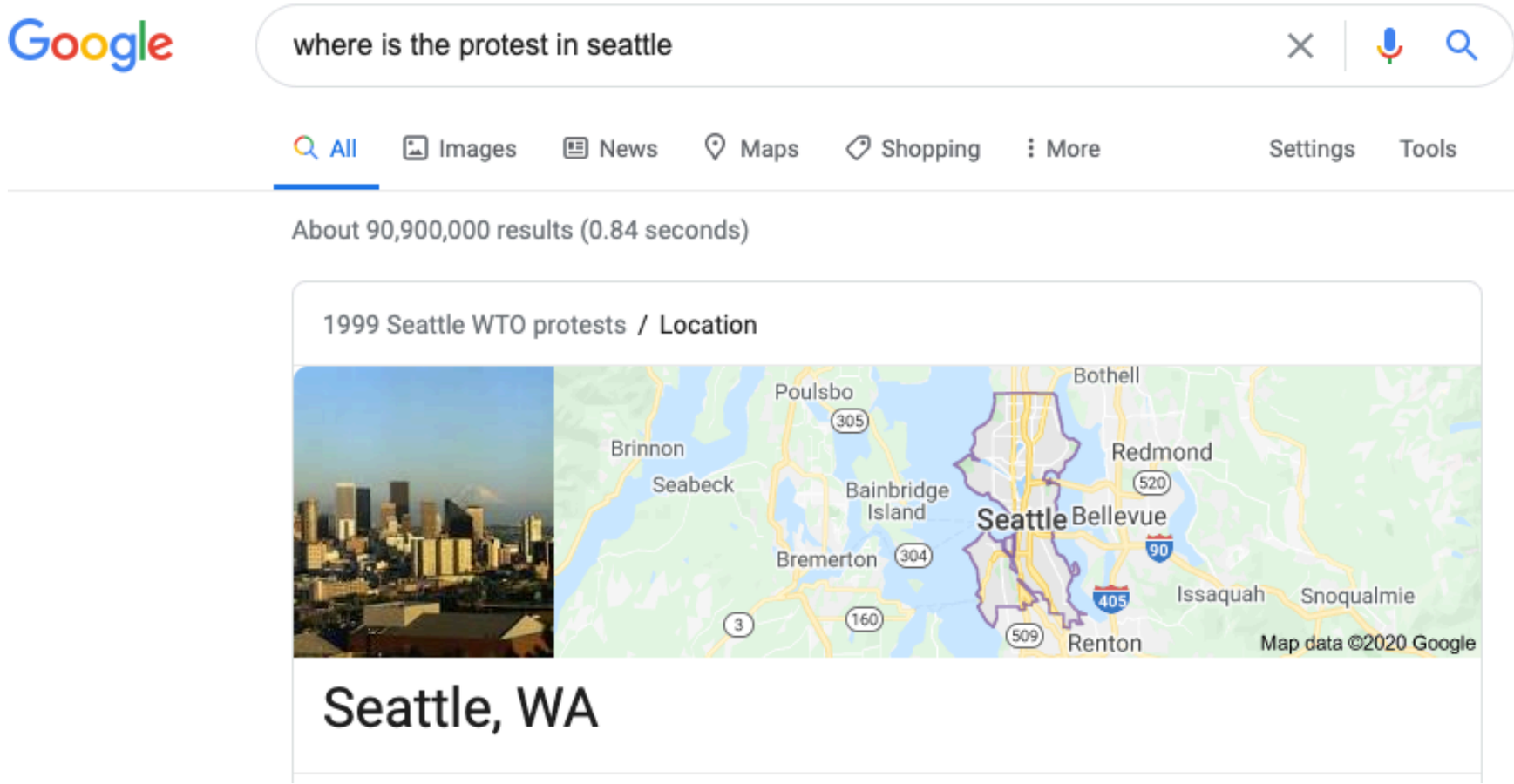
Open-Domain Question Answering using Knowledge Base

- Curated KB ensures the correctness of the information (answer)
- Common or "simple" questions can be answered easily
  - If semantic parsing (question → query) is done correctly
  - Graph structure enables multi-hop question answering

Open-Domain Question Answering using Knowledge Base

Limited coverage of questions that can be answered

- Common relationships & entities
- KB may not be timely updated



Hybrid Approach: QA using Both Text and KB

Question: Who did Cam Newton sign with?

Knowledge graph

Cam Newton *plays* football
Cam Newton *career_starts* 2011
Cam Newton *date_of_birth* 19890511
…

Wikipedia Text

Cameron Newton (born May 11, 1989) players for the Carolina Panthers of the National Football League (NFL)
…

Answer: Carolina Panthers

Xiong et al., 2019. Improving Question Answering over Incomplete KBs with Knowledge-Aware Reader

Hybrid Approach:
QA using Both Text and KB

- General Approach
  - Entity linking that connect mentions in question/text to KB
  - Graph representations for answer classification

- Papers to discuss
  - GRAFT-Net [Sun & Dhingra et al., 2018]
  - PullNet [Sun et al., 2019]
  - Knowledge-aware Reader [Xiong et al., 2019]
  - Knowledge-Guided Text Retrieval [Min et al., 2020]

WebQuestionsSP (WebQSP) [Yih et al., 2016]

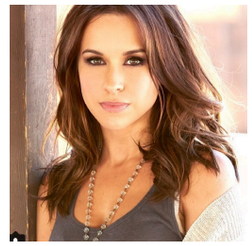
- Answers to the questions are Freebase entities
- An enhanced version of WebQuestions
  - Remove ill-formed, ambiguous questions
  - Include full semantic parses (SPARQL) with entity annotations

***Question*: Who voiced Meg on Family Guy?**

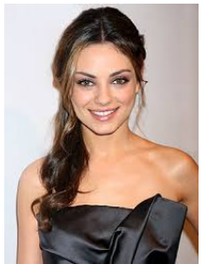
Parse:

```
PREFIX ns: <http://rdf.freebase.com/ns/>
SELECT ?x
WHERE {
  ns:m.035szd ns:tv.tv_character.appeared_in_tv_program ?y0 .
  ?y0 ns:tv.regular_tv_appearance.actor ?x ;
        ns:tv.regular_tv_appearance.series ns:m.019nnl ;
        ns:tv.regular_tv_appearance.special_performance_type
                                                ns:m.02nsjvf .
}
```

Answer:



Lacey Chabert

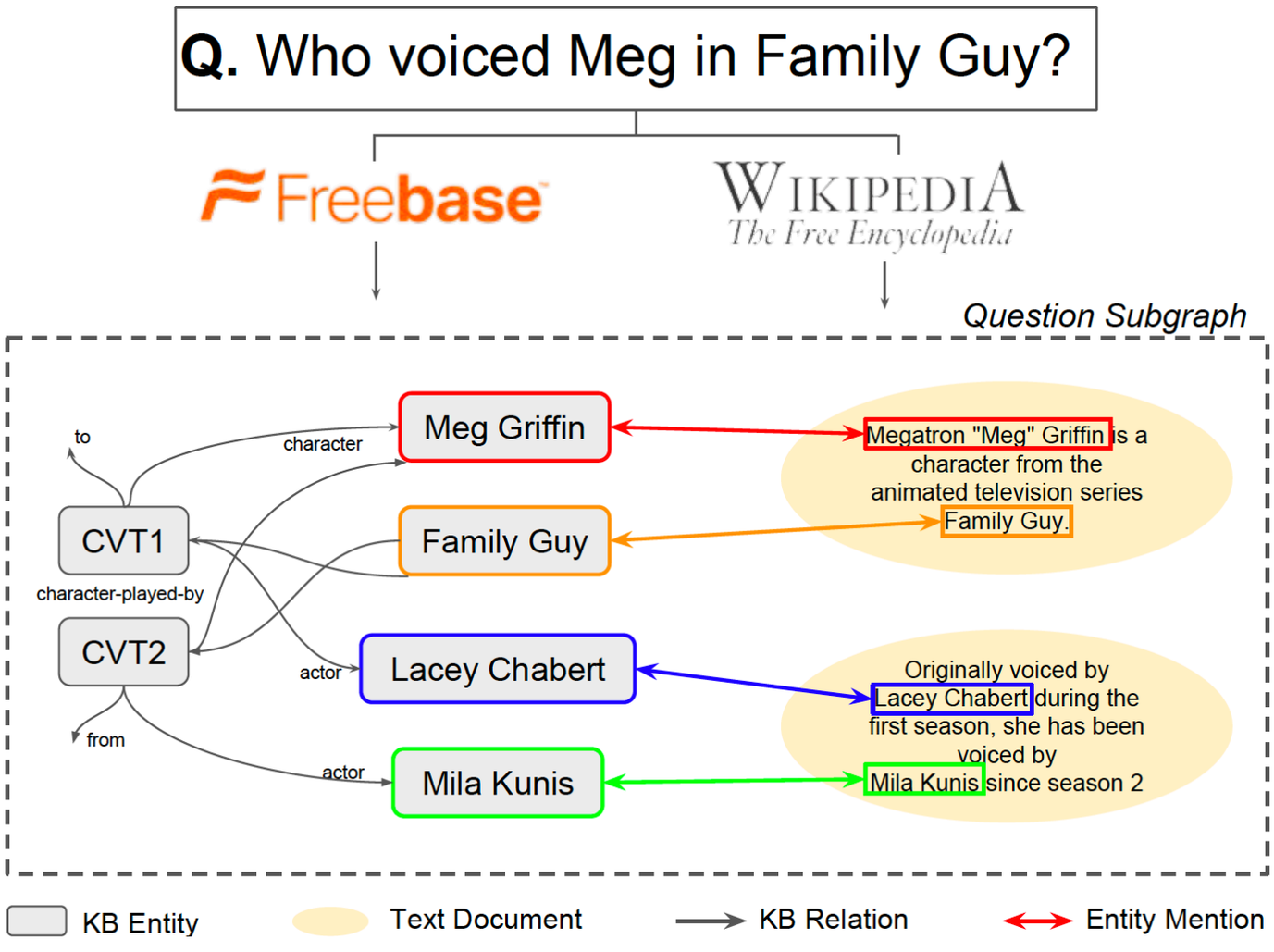


Mila Kunis

Yih et al., 2016. The Value of Semantic Parse Labeling for Knowledge Base Question Answering

GRAFT-Net [Sun & Dhingra et al., 2018]

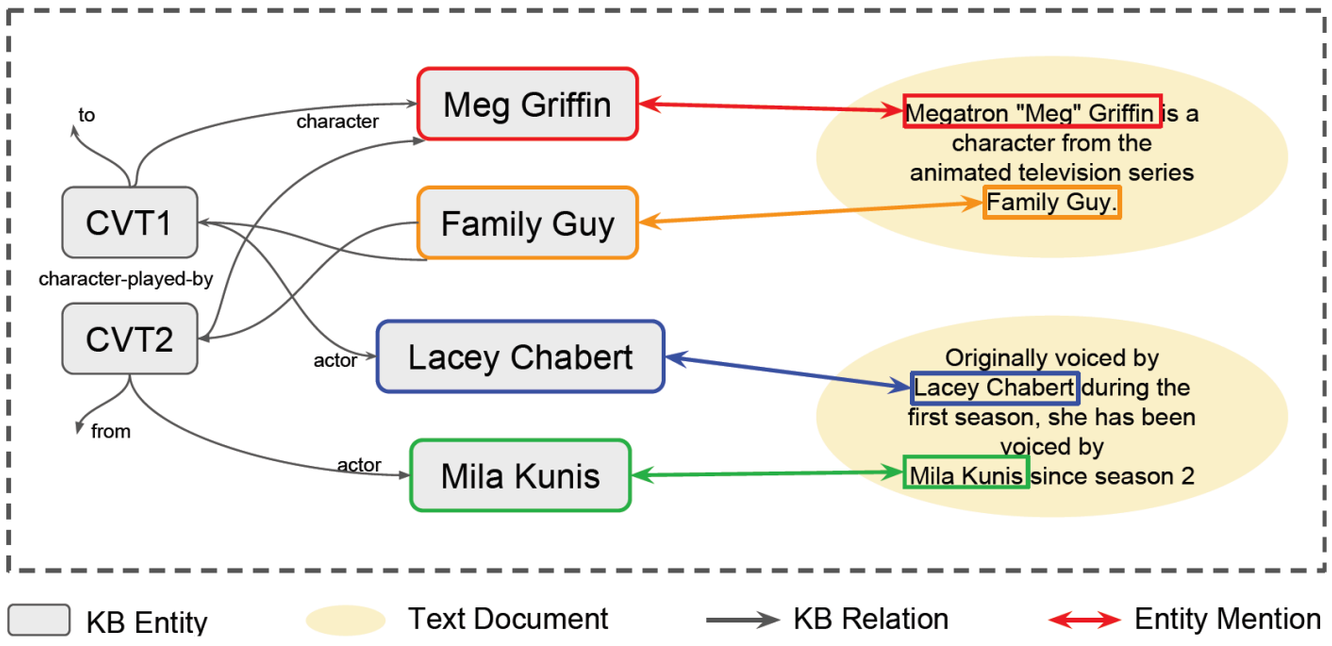
- Construct a heterogeneous question graph using both KB and text
- Use a variant of graph CNN to select an entity (node) as answer



Sun & Dhingra et al., 2018. Open Domain Question Answering Using Early Fusion of Knowledge Bases and Text

Question Subgraph Retrieval

- KB Retrieval
  - Seed entities: entities in the question
  - Use Personalized PageRank (PPR) to find top entities around seed entities, weighted by similarity of question and relation (edge type)
- Text Retrieval
  - Top sentences from Top 5 relevant Wikipedia articles



Sun & Dhingra et al., 2018. Open Domain Question Answering Using Early Fusion of Knowledge Bases and Text

Graph Propagation

- Basic recipe
  1. Initialize node representations $h_v^{(0)}$
  2. For $l = 1, \cdots, L$, update node representations

$$h_v^{(l)} = \phi\left(h_v^{(l-1)}, \sum_{v' \in N_r(v)} h_{v'}^{(l-1)}\right)$$

- Different update rules for KB and text entities in GRAFT-Net
- Answer selection: binary classification on nodes (entities)

$$Pr(v \in \{a\}_q \,|\, G_q, q) = \sigma(w^T h_v^{(L)} + b)$$

Sun & Dhingra et al., 2018. Open Domain Question Answering Using Early Fusion of Knowledge Bases and Text

PullNet [Sun et al., 2019]

- Built on GRAFT-Net
- Incrementally construct the question graph using a graph CNN

- Algorithm
  1. Initial graph G^0 with question entities $V^0 = \{e_q\}$
  2. For $t = 1, \cdots, T$ do
     1) Select entity nodes from V^{t-1}: $\{v_e\}$
     2) Find documents that contain entity e, and all other entities in the same documents
     3) Find all neighboring entities of entity e in the KB
     4) Add all entities and edges to G^t
  3. Select an entity node of V^T as answer

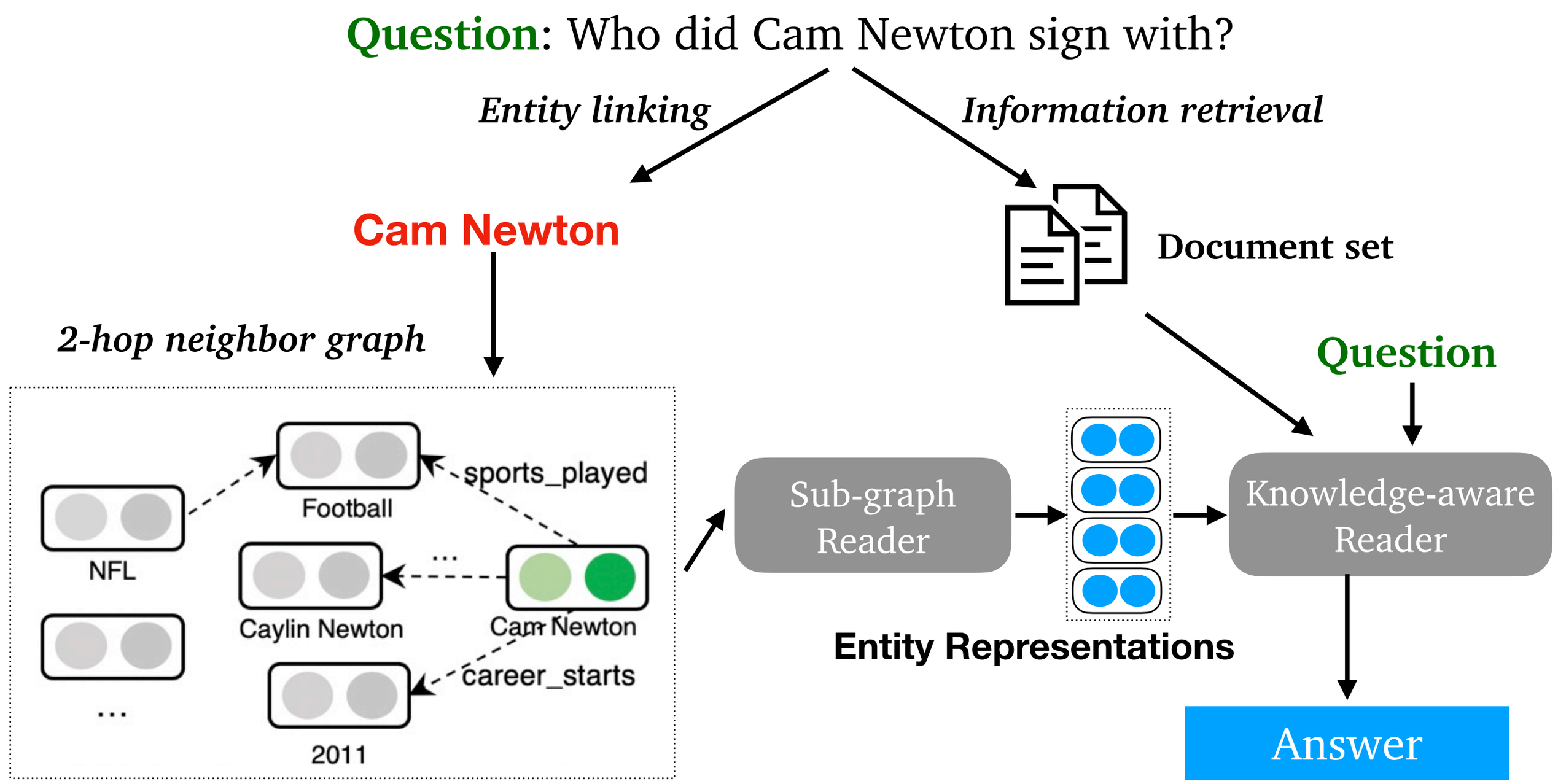
Sun et al., 2019. PullNet: Open Domain Question Answering with Iterative Retrieval on Knowledge Bases and Text

PullNet [Sun et al., 2019]

- The Select operation is determined by the same binary classifier as in Graft-Net.
- Training data: Approximate an ideal question graph using weak supervision (questions and their gold answers)
    - Find all shortest paths in the KB between question entities and answer entities
    - For each entity in the paths, include its neighboring entities up to a pre-determined length

Sun et al., 2019. PullNet: Open Domain Question Answering with Iterative Retrieval on Knowledge Bases and Text

Knowledge-aware Reader [Xiong et al., 2019]



Xiong et al., 2019. Improving Question Answering over Incomplete KBs with Knowledge-Aware Reader

Sub-graph Reader

- Start with the sub-graph embeddings from Graft-Net
- Reweigh the entity embeddings using neighboring entities and relations:

$$\overrightarrow{e_i'} = \sum_{e_j \in N(e_i)} \alpha_{ij} W \cdot [\vec{r}_{ij}; \overrightarrow{e}_j]$$

Similarity between question and r_{ij}

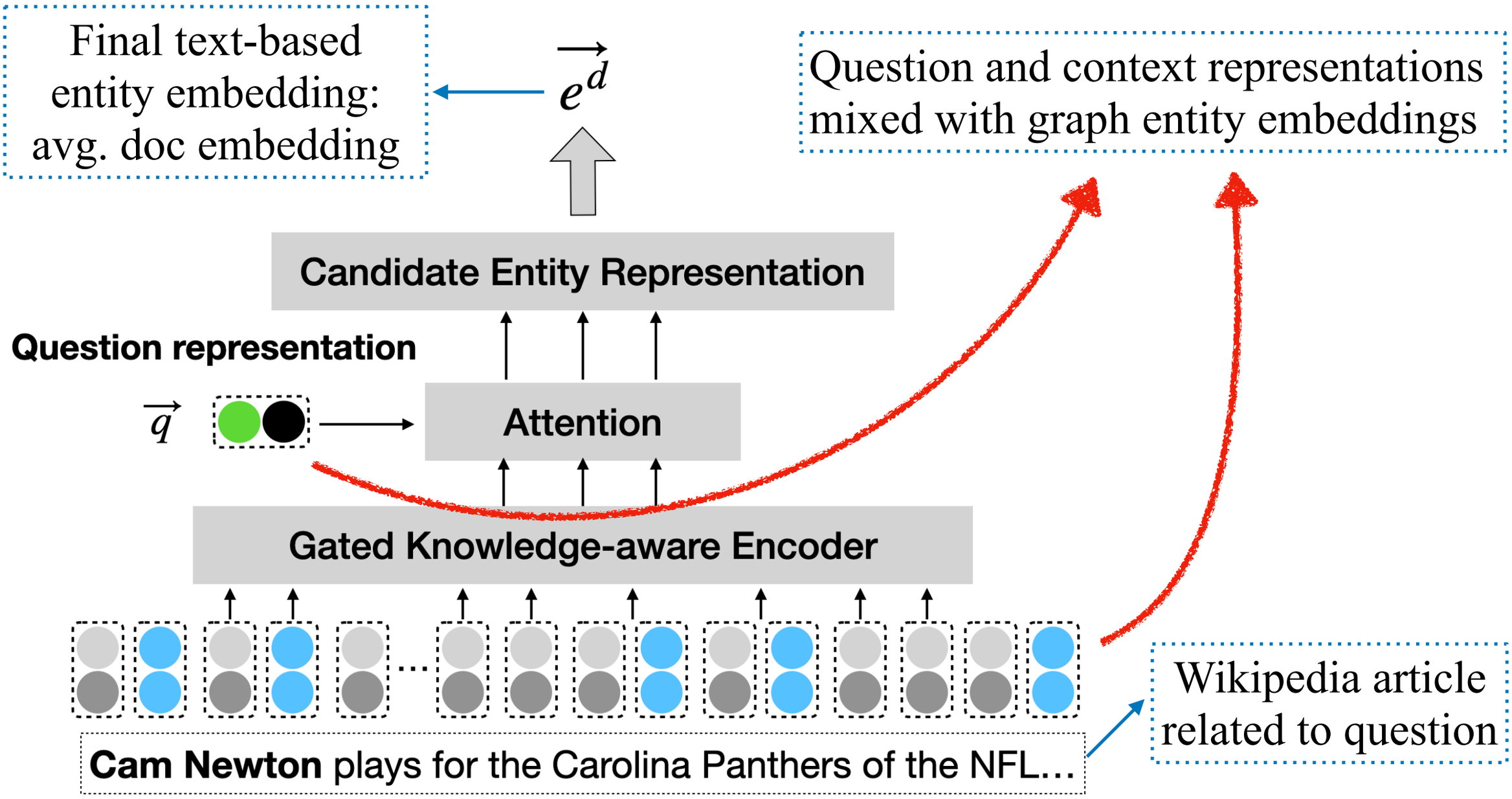
Learned weighting parameters

Relationship between e_i and e_j

Neighboring node

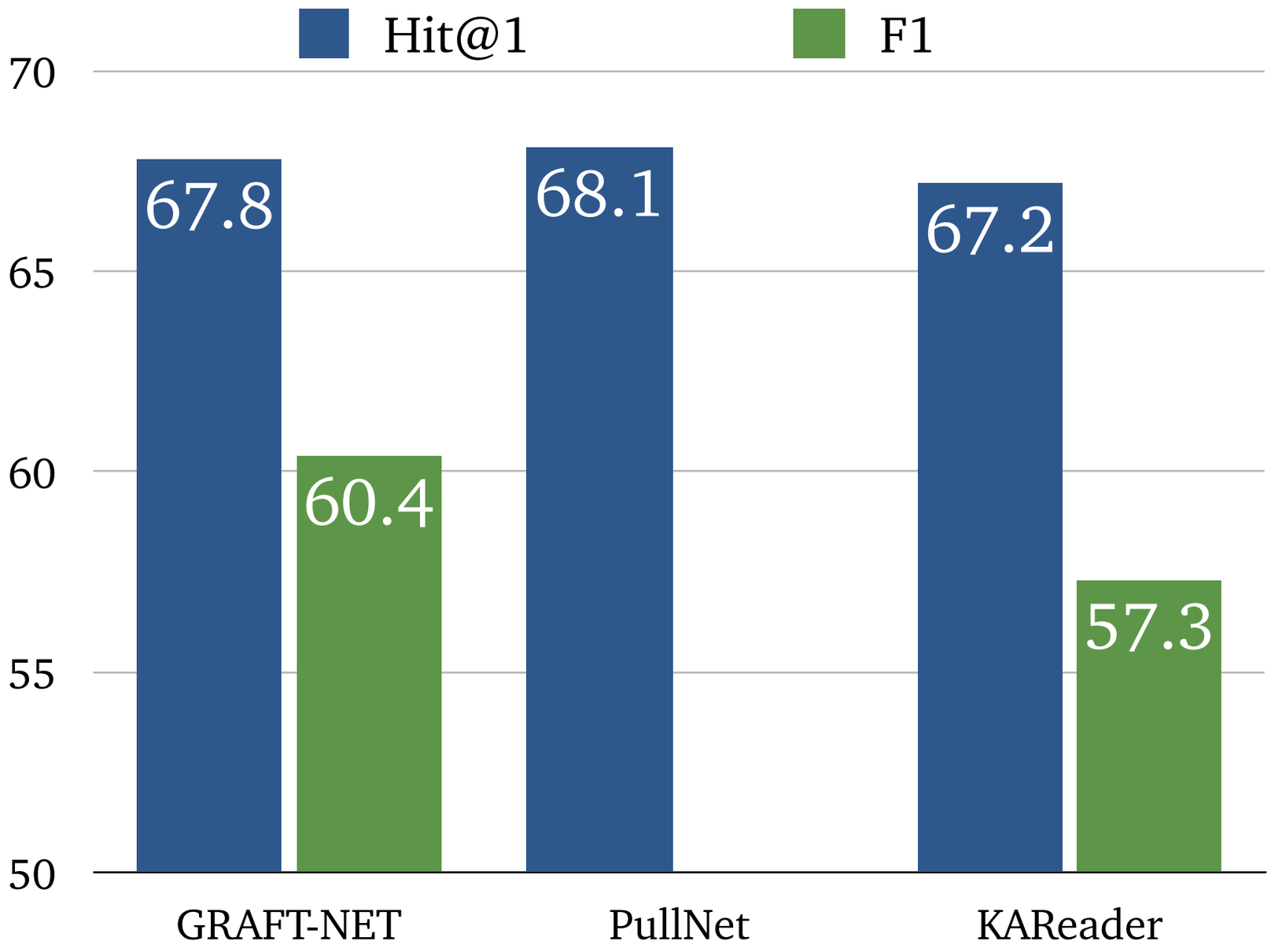
Xiong et al., 2019. Improving Question Answering over Incomplete KBs with Knowledge-Aware Reader

Knowledge-Aware Text Reader

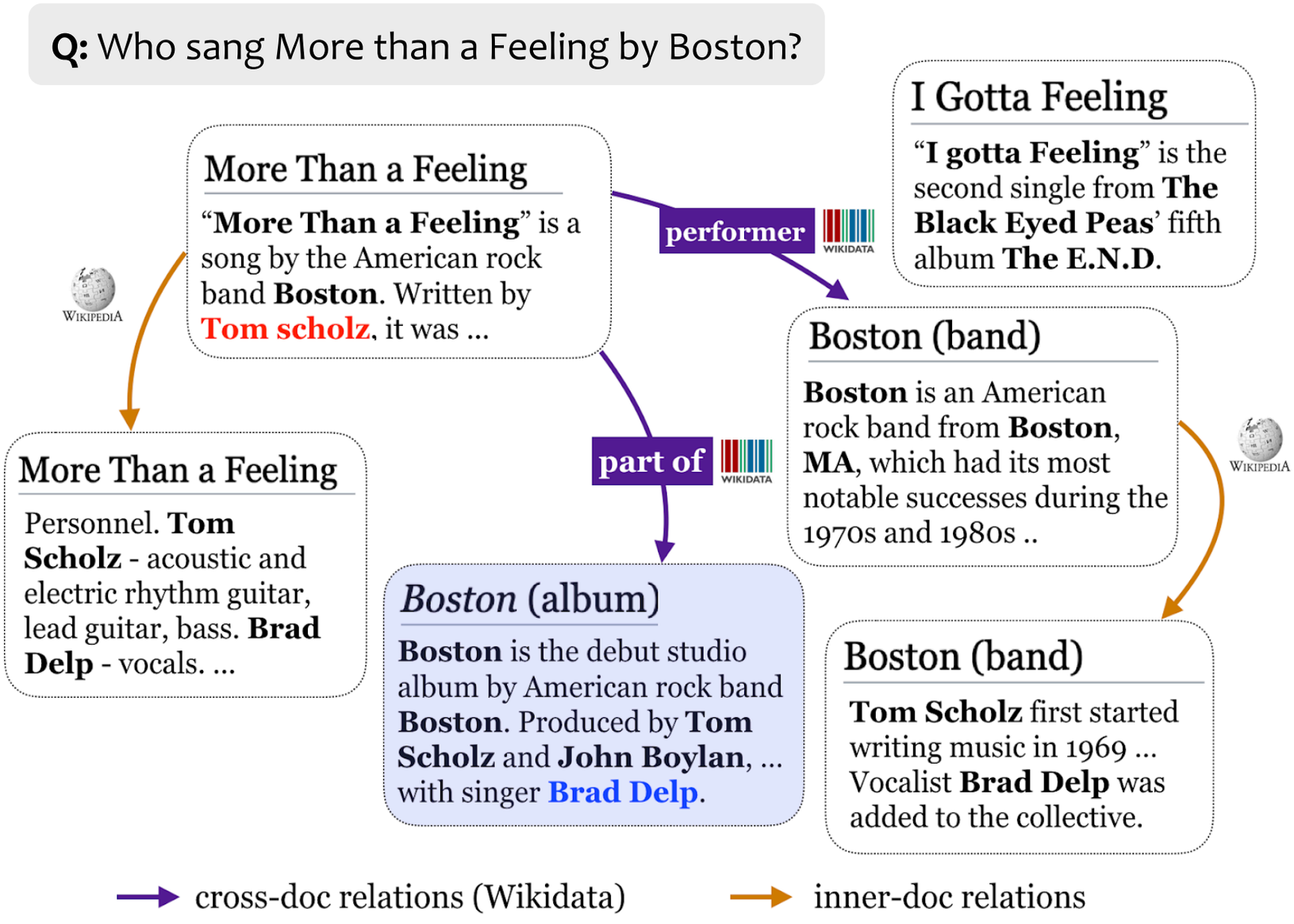


Xiong et al., 2019. Improving Question Answering over Incomplete KBs with Knowledge-Aware Reader

Results on WebQSP

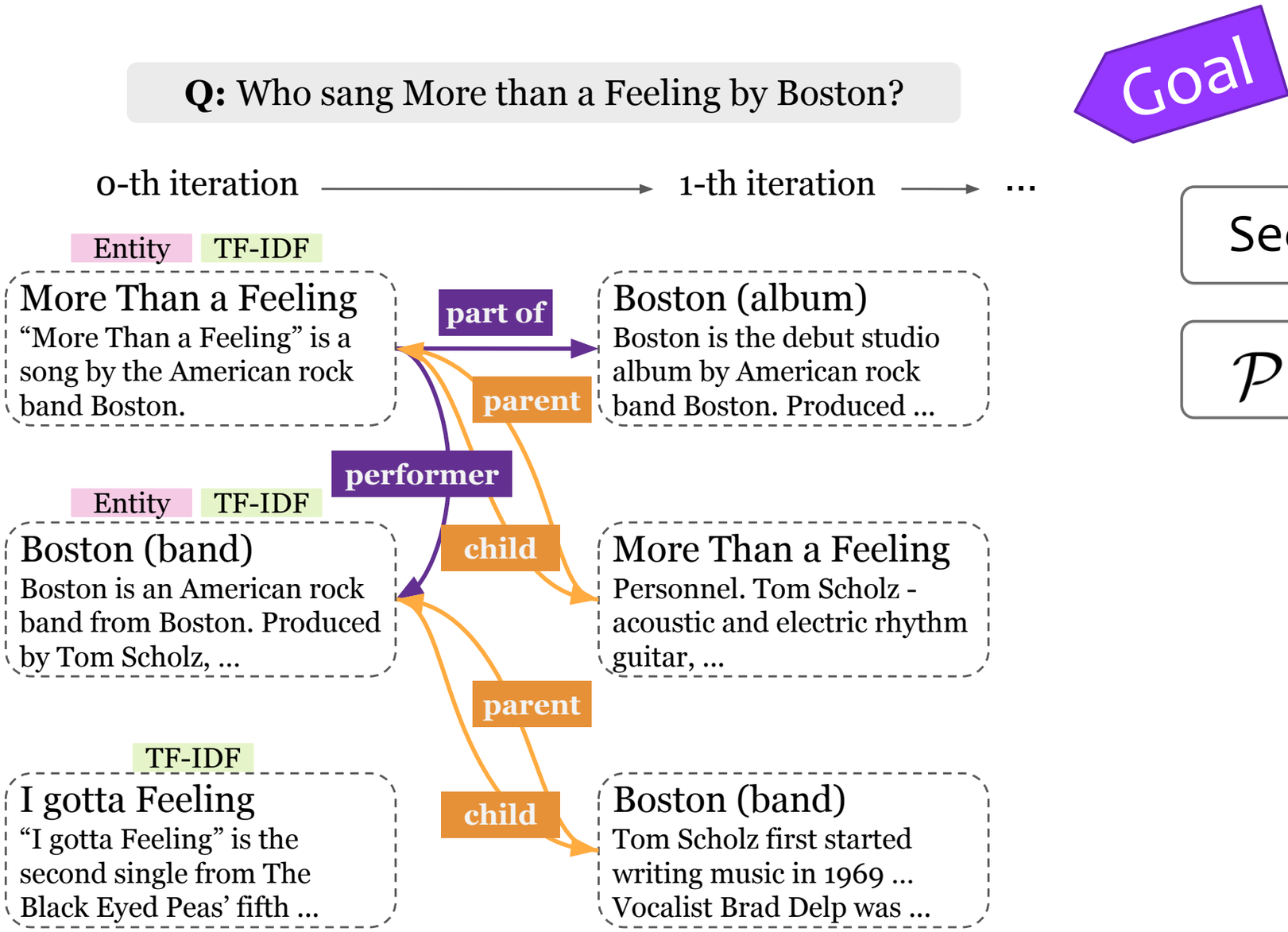


Knowledge-Guided Text Retrieval and Reading
[Min et al., 2020]



Min et al., 2020. Knowledge Guided Text Retrieval and Reading for Open Domain Question Answering

Graph Retriever



Seed passages $\mathcal{P}^{(0)}$

$\mathcal{P}^{(m-1)} \rightarrow \mathcal{P}^{(m)}$

Min et al., 2020. Knowledge Guided Text Retrieval and Reading for Open Domain Question Answering

Graph Retriever

Seed passages $\mathcal{P}^{(0)}$

Q: Who sang More than a Feeling by Boston?

0-th iteration

Entity TF-IDF

More Than a Feeling
"More Than a Feeling" is a song by the American rock band Boston.

Entity TF-IDF

Boston (band)
Boston is an American rock band from Boston. Produced by Tom Scholz, ...

TF-IDF

I gotta Feeling
"I gotta Feeling" is the second single from The Black Eyed Peas' fifth ...

Q: Who sang More than a Feeling by Boston?

Entity linking

More than a Feeling | Boston (band)

Q: Who sang More than a Feeling by Boston?

TF-IDF article retrieval

More than a Feeling | Boston (band)

I gotta Feeling | Boston, MA

Min et al., 2020. Knowledge Guided Text Retrieval and Reading for Open Domain Question Answering

Graph Retriever

$$\mathcal{P}^{(m-1)} \rightarrow \mathcal{P}^{(m)}$$

Q: Who sang More than a Feeling by Boston?

0-th iteration → 1-th iteration → ...

Entity TF-IDF

More Than a Feeling

"More Than a Feeling" is a song by the American rock band Boston.

Entity TF-IDF

Boston (band)

Boston is an American rock band from Boston. Produced by Tom Scholz, ...

TF-IDF

I gotta Feeling

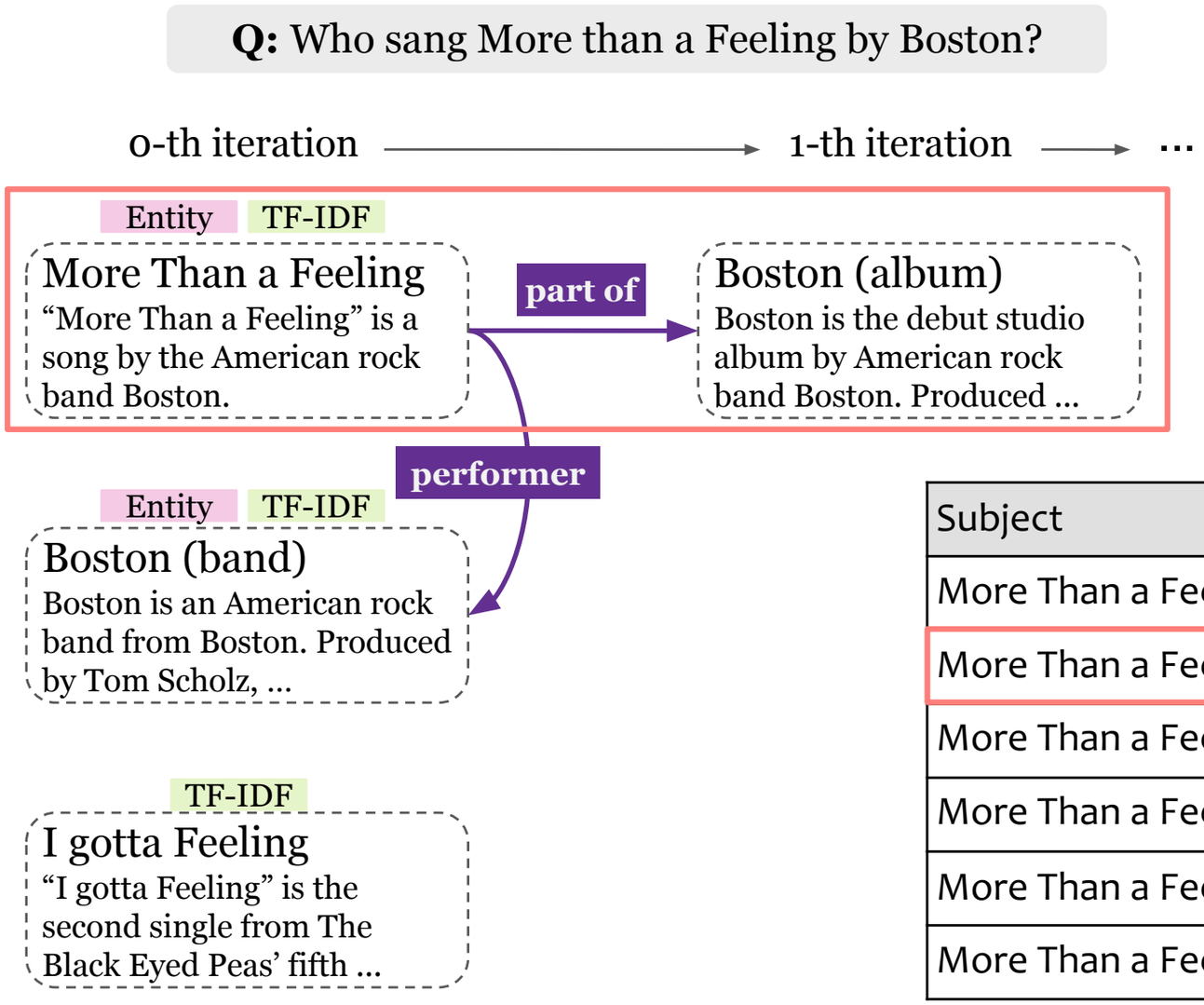
"I gotta Feeling" is the second single from The Black Eyed Peas' fifth ...

Subject	Relation	Object
More Than a Feeling	performer	Boston (band)
More Than a Feeling	part of	Boston (album)
More Than a Feeling	genre	Hard rock
More Than a Feeling	country of origin	USA
More Than a Feeling	record label	Epic
More Than a Feeling	followed by	Foreplay/Long Time

Min et al., 2020. Knowledge Guided Text Retrieval and Reading for Open Domain Question Answering

Graph Retriever

$$\mathcal{P}^{(m-1)} \rightarrow \mathcal{P}^{(m)}$$



WIKIDATA

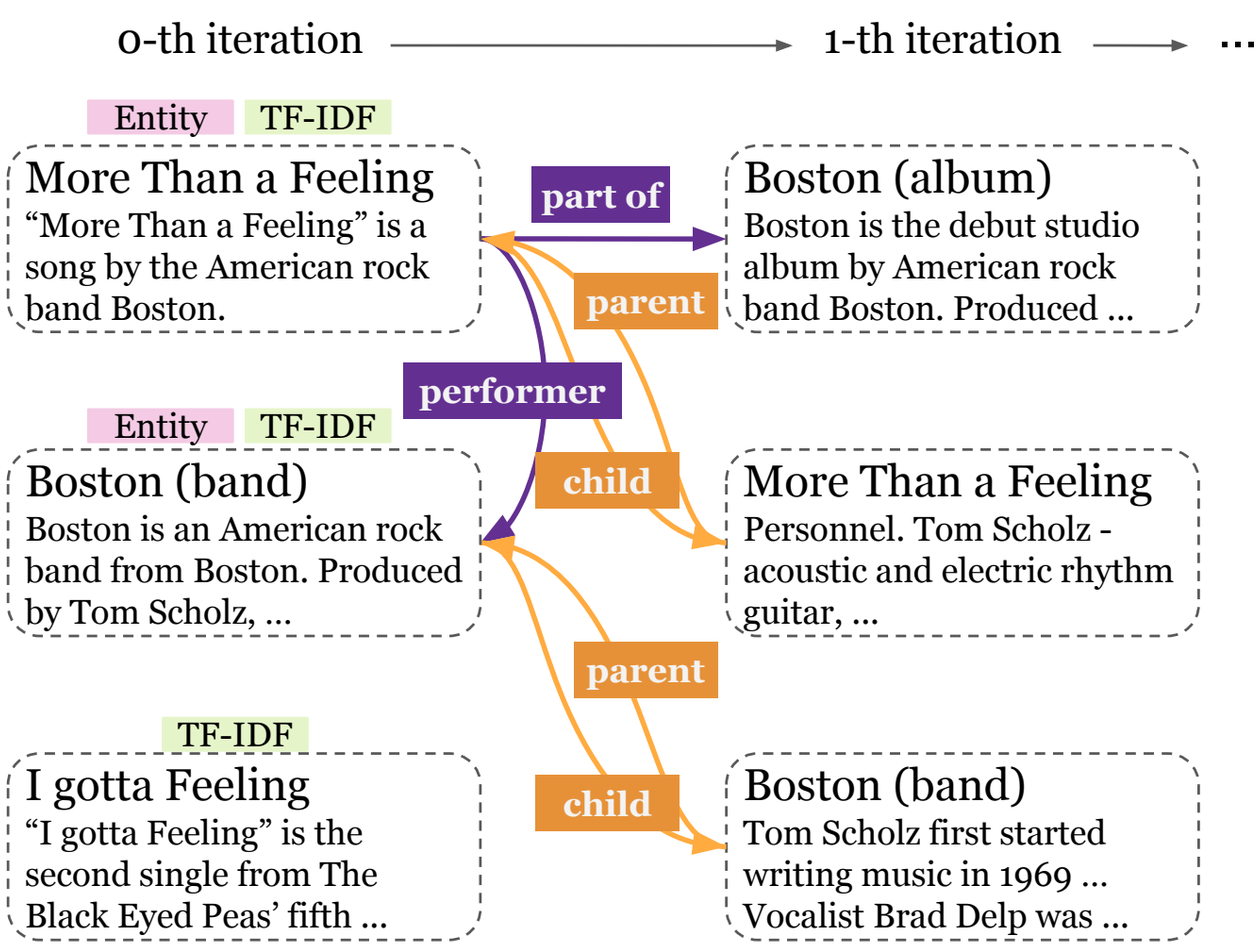
Subject	Relation	Object
More Than a Feeling	performer	Boston (band)
More Than a Feeling	part of	Boston (album)
More Than a Feeling	genre	Hard rock
More Than a Feeling	country of origin	USA
More Than a Feeling	record label	Epic
More Than a Feeling	followed by	Foreplay/Long Time

Min et al., 2020. Knowledge Guided Text Retrieval and Reading for Open Domain Question Answering

Graph Retriever

$$\mathcal{P}^{(m-1)} \rightarrow \mathcal{P}^{(m)}$$

Q: Who sang More than a Feeling by Boston?



More Than a Feeling

From Wikipedia, the free encyclopedia

"More Than a Feeling" is a song by the American rock band Boston. Written by Tom Scholz, it was released ...

Background and writing

The song took Tom Scholz five years to complete. It is one of six songs (five of which eventually appeared ...)

Content

The Book of Rick Lists suggests that the chorus ...

Reception

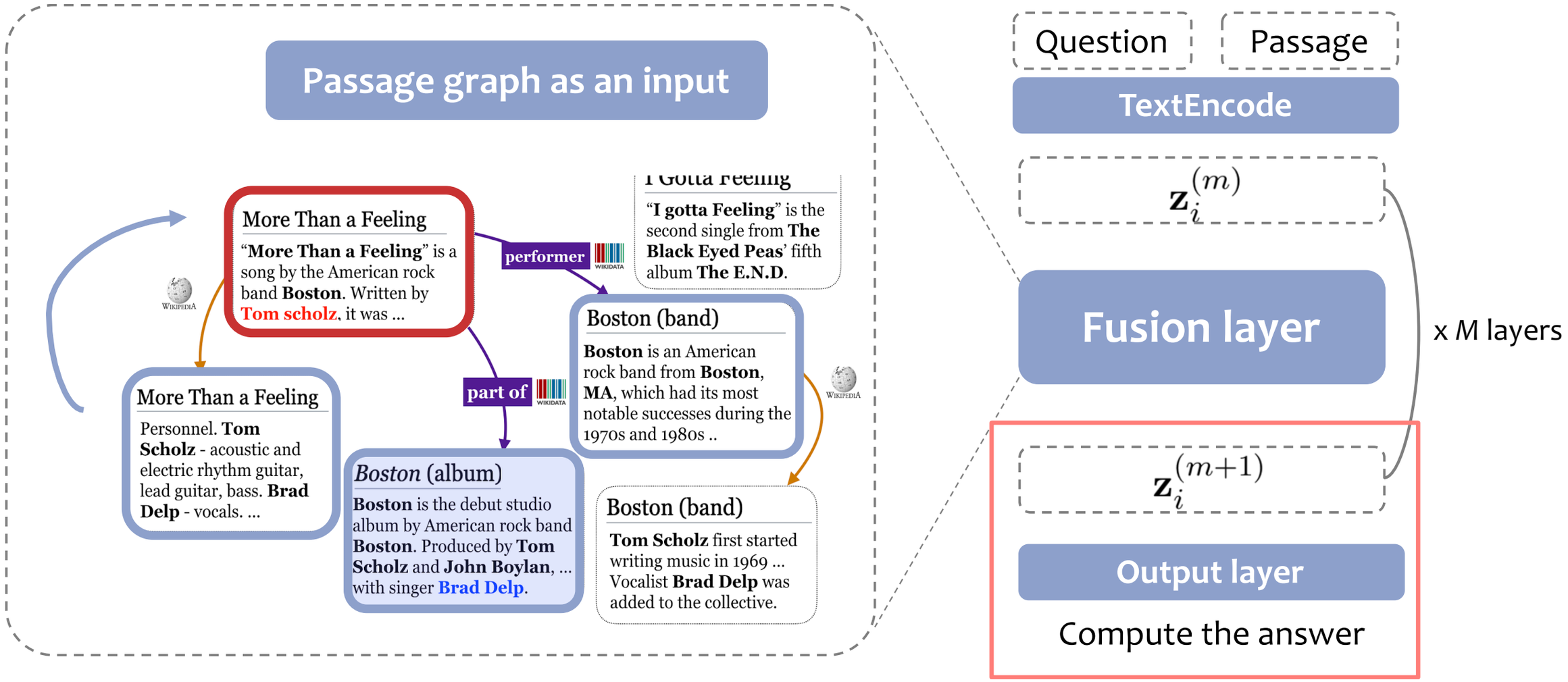
Guitar World states that when the radio plays ...

Single version

Epic released an edited version of the song for the ...

Min et al., 2020. Knowledge Guided Text Retrieval and Reading for Open Domain Question Answering

Graph Reader



Min et al., 2020. Knowledge Guided Text Retrieval and Reading for Open Domain Question Answering

Part VIII

Conclusion

Tutorial Summary

Introduction to Open-domain Question Answering

- Problem definition, motivation, applications
- Brief historical review
  - One of the earliest AI challenges
  - TREC QA tracks
  - IBM Watson Deep QA
  - Machine reading comprehension

Tutorial Summary
Latest development on QA

- Popular datasets for training and evaluating open-domain QA
- Retriever + Reader
  - DrQA - Chen et al. 2017. *Reading Wikipedia to Answer Open-domain Questions.*
- Dense retriever and end-to-end training
  - ORQA - Lee et al. 2019. *Latent Retrieval for Weakly Supervised Open Domain Question Answering.*
- Retrieval-free
  - T5 - Roberts et al. 2020. *How Much Knowledge Can You Pack Into the Parameters of a Language Model?*

Tutorial Summary
Open-domain QA using KBs and text

- Introduction to open-domain QA using KBs
  - Properties of entity-centric knowledge bases
  - Pros and cons of using KBs for open-domain QA

- Recent work on using both text and KBs
  - Constructing graphs of heterogeneous nodes & edges
    - Sun & Dhingra et al. 2018. *Open Domain Question Answering Using Early Fusion of Knowledge Bases and Text.*
  - Leverage graphs for passage retrieval
    - Min et al. 2020. *Knowledge Guided Text Retrieval and Reading for Open Domain Question Answering.*

Open problems and future directions

- Hot topic: the two open-domain QA paradigms:
  Explicit context retrieval vs. knowledge encoded in models
  - Pros/Cons & Trade-off
  - Impact of large pre-trained models
  - Efficiency & accuracy

- Complete user experience
  - Rationale and evidence to support answers
  - Answer triggering: knowing when it doesn't know

- User interaction and grounding
  - Multi-turn, conversational QA
  - Multi-modal interactions (e.g., VQA, virtual tour guide)

References

1. Danqi Chen, Adam Fisch, Jason Weston, Antoine Bordes. Reading Wikipedia to Answer Open-Domain Questions. ACL 2017.
2. Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerald Tesauro, Bowen Zhou, Jing Jiang. R^3: Reinforced Reader-Ranker for Open-Domain Question Answering. AAAI 2018.
3. Shuohang Wang, Mo Yu, Jing Jiang, Wei Zhang, Xiaoxiao Guo, Shiyu Chang, Zhiguo Wang, Tim Klinger, Gerald Tesauro, Murray Campbell. Evidence Aggregation for Answer Re-Ranking in Open-Domain Question Answering. ICLR 2018.
4. Yankai Lin, Haozhe Ji, Zhiyuan Liu, Maosong Sun. Denoising Distantly Supervised Open-domain Question Answering. ACL 2018.
5. Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, William Cohen. Open Domain Question Answering Using Early Fusion of Knowledge Bases and Text. EMNLP 2018.
6. Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever. Language Models are Unsupervised Multitask Learners. OpenAI 2019.
7. Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, Jimmy Lin. End-to-end Open-domain Question Answering with BERTserini. NAACL 2019 (demonstration).
8. Kenton Lee, Ming-Wei Chang, Kristina Toutanova. Latent Retrieval for Weakly Supervised Open Domain Question Answering. ACL 2019.
9. Minjoon Seo, Jinhyuk Lee, Tom Kwiatkowski, Ankur P. Parikh, Ali Farhadi, Hannaneh Hajishirzi. Real-Time Open-Domain Question Answering with Dense-Sparse Phrase Index. ACL 2019.
10. Sewon Min, Danqi Chen, Hannaneh Hajishirzi, Luke Zettlemoyer. A Discrete Hard EM Approach for Weakly Supervised Question Answering. EMNLP 2019.
11. Haitian Sun, Tania Bedrax-Weiss, William W. Cohen. PullNet: Open Domain Question Answering with Iterative Retrieval on Knowledge Bases and Text. EMNLP 2019.
12. Sewon Min, Danqi Chen, Luke Zettlemoyer, Hannaneh Hajishirzi. Knowledge Guided Text Retrieval and Reading for Open Domain Question Answering. arXiv 2019.
13. Jinhyuk Lee, Minjoon Seo, Hannaneh Hajishirzi, Jaewoo Kang. Contextualized Sparse Representations for Real-Time Open-Domain Question Answering. ACL 2020.
14. Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, Caiming Xiong. Learning to Retrieve Reasoning Paths over Wikipedia Graph for Question Answering. ICLR 2020.
15. Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, Ming-Wei Chang. REALM: Retrieval-Augmented Language Model Pre-Training. ICML 2020.

References

16. Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, Wen-tau Yih. Dense Passage Retrieval for Open-Domain Question Answering. arXiv 2020.
17. Adam Roberts, Colin Raffel, Noam Shazeer. How Much Knowledge Can You Pack Into the Parameters of a Language Model?. arXiv 2020.
18. Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, Douwe Kiela. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. arXiv 2020.
19. Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, Dario Amodei. Language Models are Few-Shot Learners. arXiv 2020.
20. R. F. Simmons. Answering english questions by computer: a survey. Communications of the ACM, 8(1):53-70, 1965.
21. Green, Wolf, Chomsky & Laughery, BASEBALL: An automatic question answerer. Computers and Thought, 1963.
22. Krisch. Computer Interpretation of English Text and Picture Patterns. IEEE Transactions on Electronic Computers, 1964.
23. Bobrow. Natural Language Input for a Computer Problem Solving System. 1964. Phd Thesis.
24. E. Brill, S. Dumais, and M. Banko. An analysis of the AskMSR question-answering system. EMNLP 2002.
25. A. M. Gliozzo, A. Kalyanpur, and J. Fan. Natural language processing in Watson. NAACL-HLT 2012 Tutorial.
26. S. Harabagiu and D. Moldovan. Open-domain textual question answering. NAACL-HLT 2001 Tutorial.
27. M. Richardson, C. J. Burges, and E. Renshaw. MCTest: A challenge dataset for the open-domain machine comprehension of text. EMNLP 2013.
28. E. M. Voorhees and D. M. Tice. Building a question answering test collection. SIGIR 2000.
29. X. Li and D. Roth. Learning Question Classifiers. COLING 2002.
30. W. Yih, M. Chang, X. He, and J. Gao. Semantic parsing via staged query graph generation: Question answering with knowledge base. ACL 2015.