Natural Language Processing

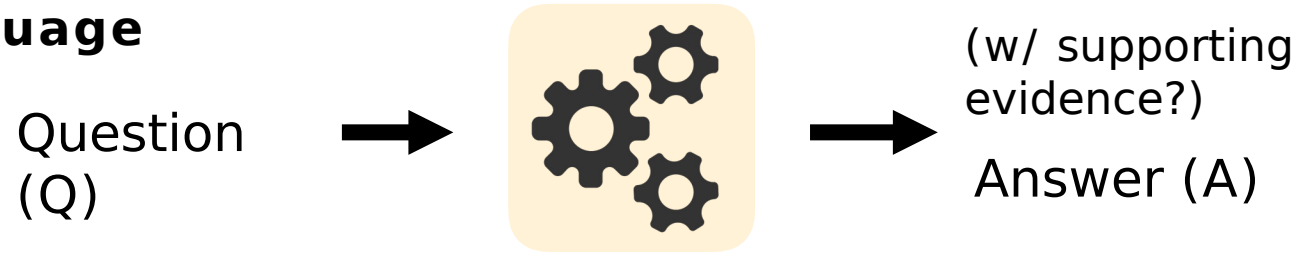
IR based Question Answering

Instructor: Yangqiu Song

Slides credits: Scott Yih, Danqi Chen, Akari Asai, Sewon Min, Zexuan Zhong

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*
  - *Q: Where does the energy in a nuclear explosion come from? A: high-speed nuclear reaction*
  - *Q: Where is Einstein's house? A: 112 Mercer St, Princeton, NJ*
  - *Q: How many papers were accepted by ACL*

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

Answer Questions using Structured Data

- General problem setting
  - Information source: A “database”
    - Collection of records
    - Tables
    - Large scale DB with complex schema
  - Input: a natural language question (instead of formal “query”)
  - Output: Answer

Early Work

- Small scale & domain-specific KBs
  - Simple schema
  - Small number of entities and relations
  - Limited set of sensible questions
- Approaches
  - Ad-hoc methods (e.g., manually crafting rules) can be quite effective
  - Semantic parsing (of questions)
- Issues
  - Not clear if the methods are scalable
  - Cannot support “open-domain” question answering

Modern Large-scale Knowledge Graphs

Google Knowledge Graph

Microsoft 悟 SATORI

NEL

Freebase

DBpedia

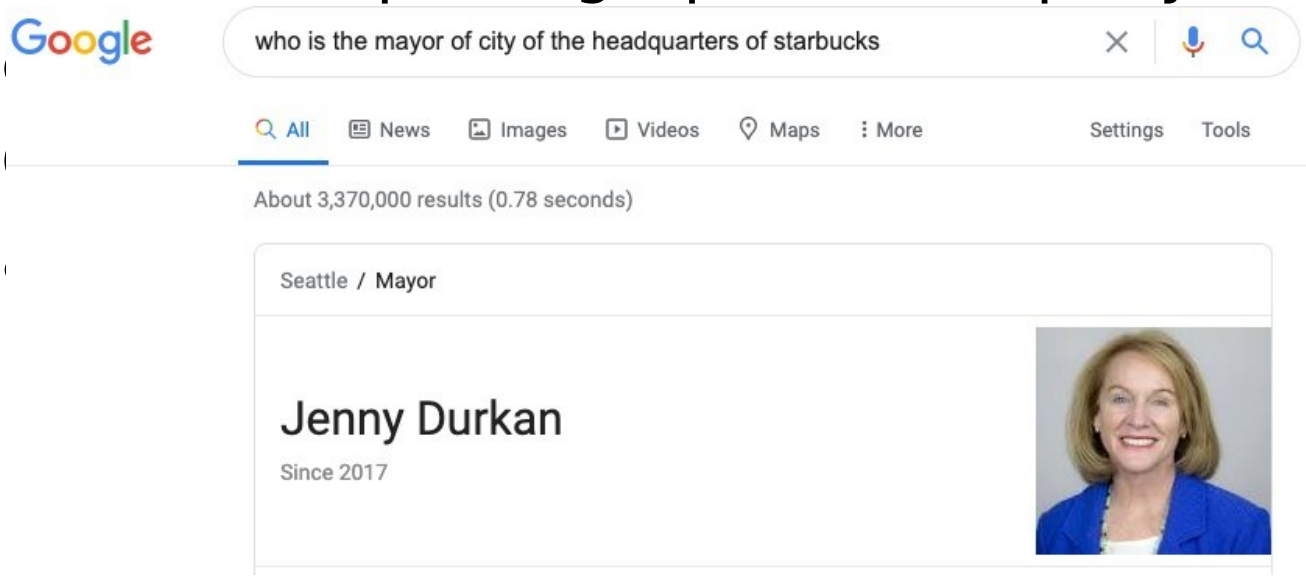
yago
select knowledge

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

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
  - ion



Key Challenges

- Language mismatch
  - Lots of ways to ask the same question
    - “Who played the role of Meg on Family Guy?”
    - “What is the name of the actress for Meg on Family Guy?”
    - “In the TV show Family Guy, who is the voice for Meg?”
  - Need to map questions to the predicates defined in KB
    - tv.tv_program.regular_cast – tv.regular_tv_appearance.actor
- Large search space
  - Some Freebase entities have > 160,000 immediate neighbors
- Compositionality
  - “What movies are directed by the person who won the most Academy and Golden Globe awards combined?”

Knowledge Graph is largely incomplete

Freebase

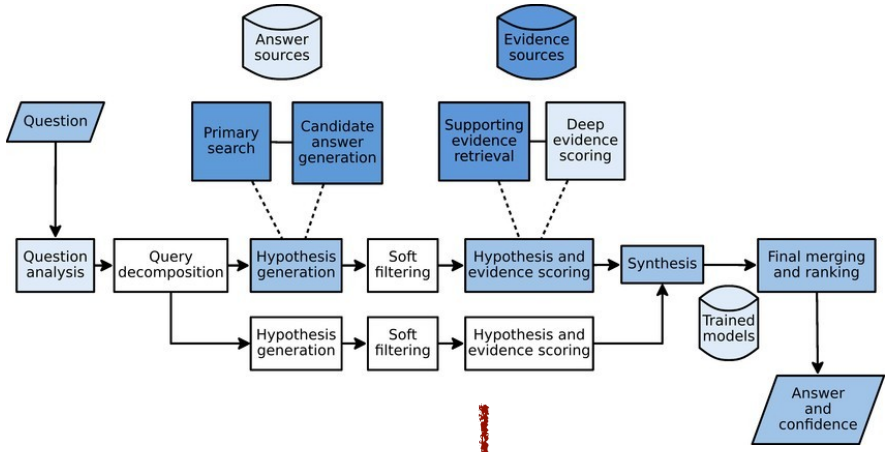
Relation	Percentage unknown	
	All 3M	*Top 100K*
PROFESSION	68%	24%
PLACE OF BIRTH	71%	13%
NATIONALITY	75%	21%
EDUCATION	91%	63%
SPOUSES	92%	68%
PARENTS	94%	77%
CHILDREN	94%	80%
SIBLINGS	96%	83%
ETHNICITY	99%	86%

ge Base Completion via Search-based Question Answering [Robert West, et al., W

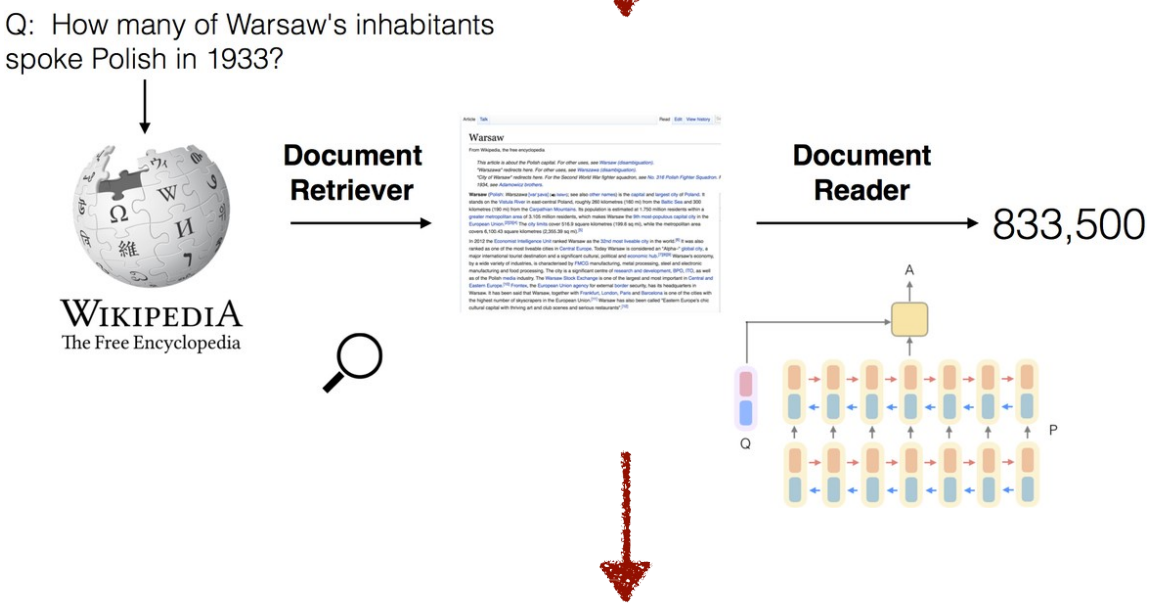
Search engines: from keyword matching to question answering with unstructured data

Paradigm Switching

Classical QA pipeline

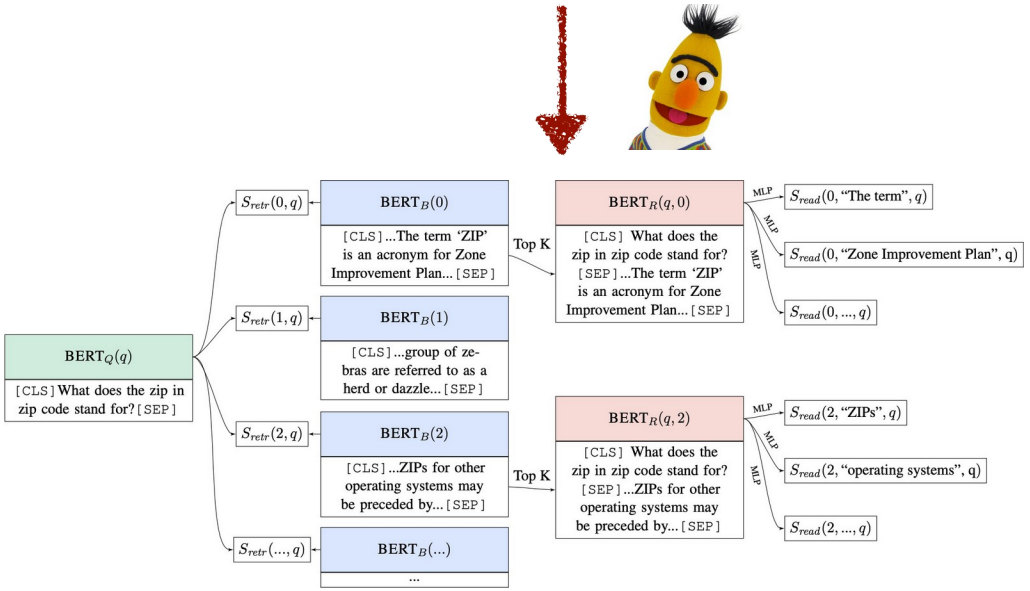


Two-stage Retriever-reader approaches

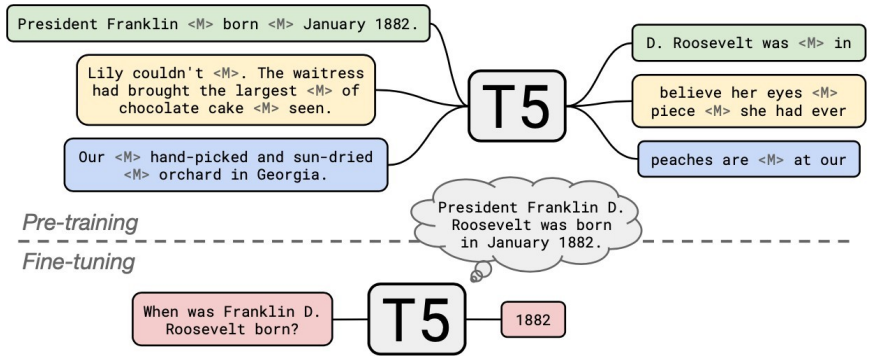


Paradigm Switching

End-to-end learning



Retrieval-free models



Outline

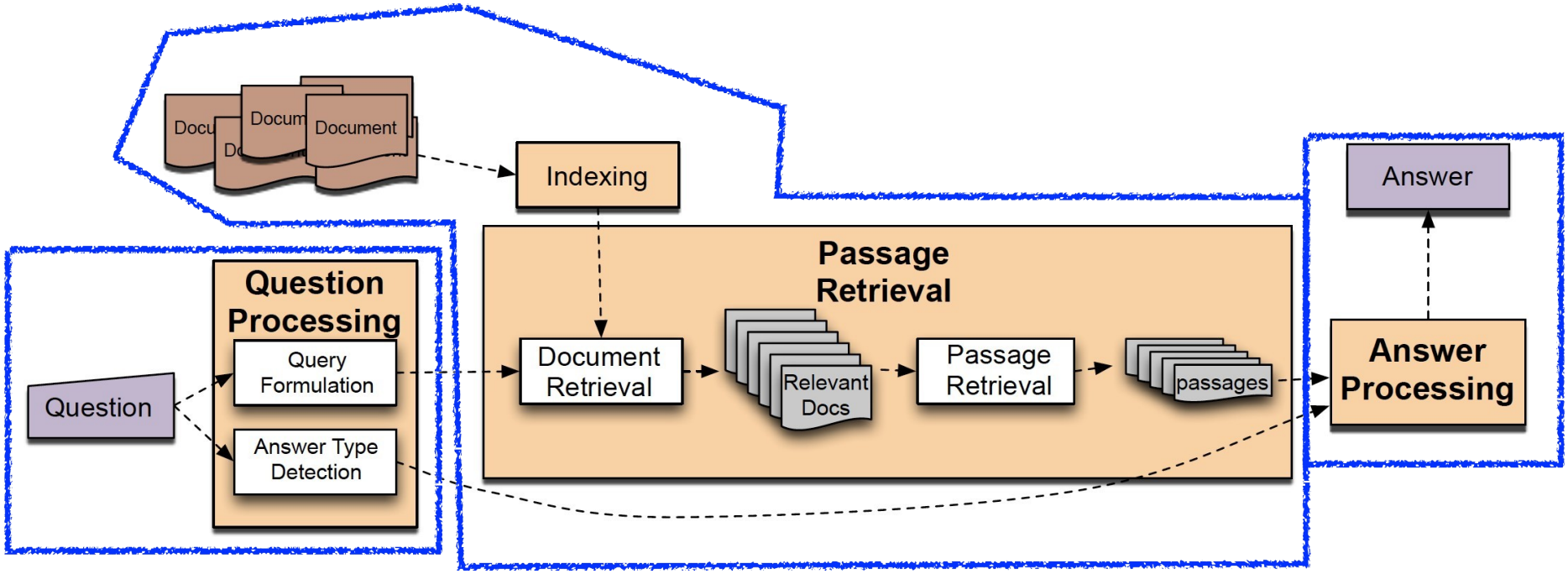
- History

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
  - Test set: 500 questions from Excite, Encarta, MSNSearch, AskJeeves
  - Human judges decide the correctness of the answers from QA systems

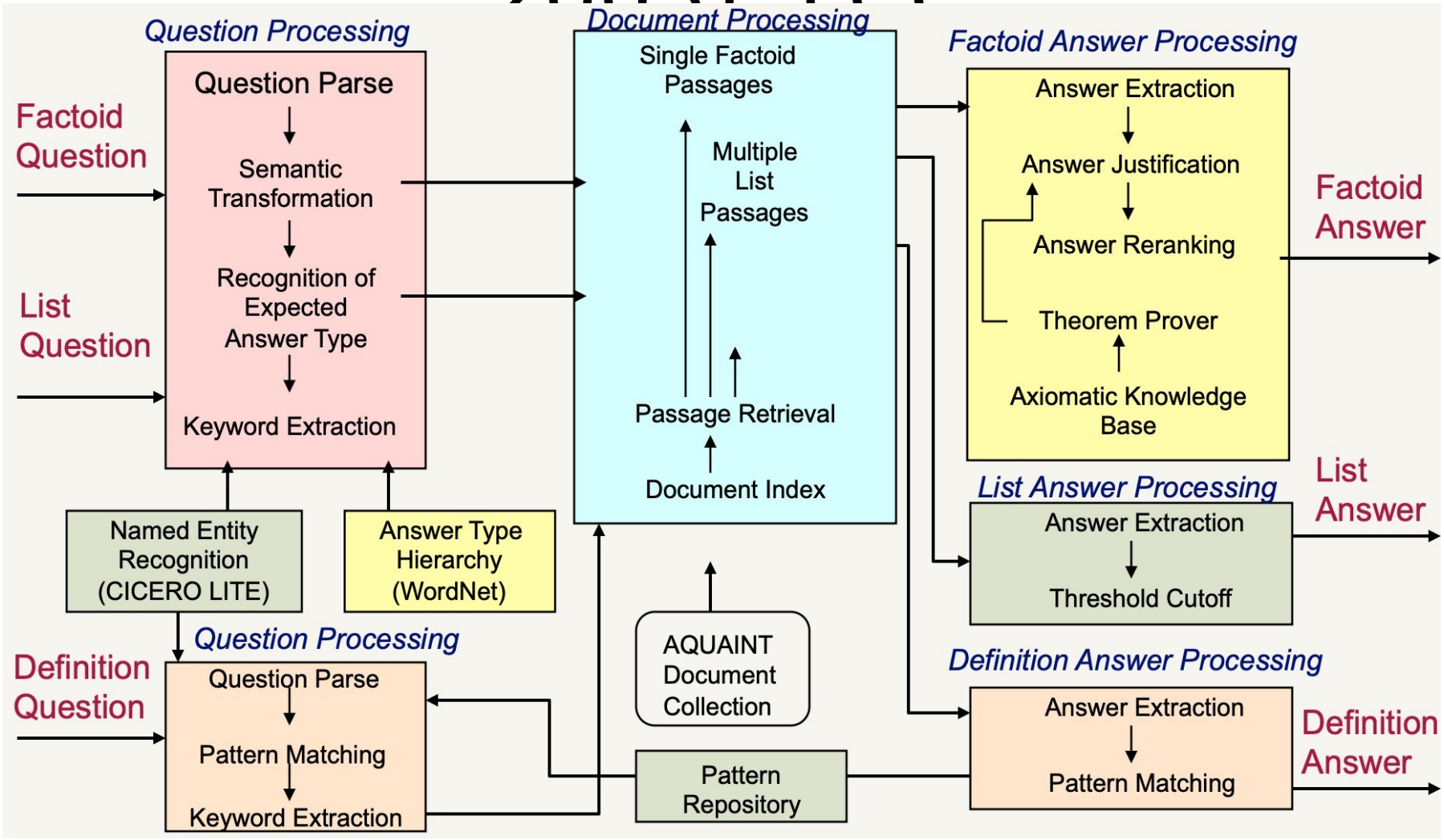
https://
trec.nist.gov/

Typical pipeline of TREC-QA systems



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

Top TREC-QA system (circa 2003): LCC



Factoid Questions: Sometimes the question asks for a body of information instead of a fact.

Harabagiu and Moldovan, 2003. Question Answering.

https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1086/handouts/cs224n-

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

Stanford Question Answering Dataset (SQuAD)

[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

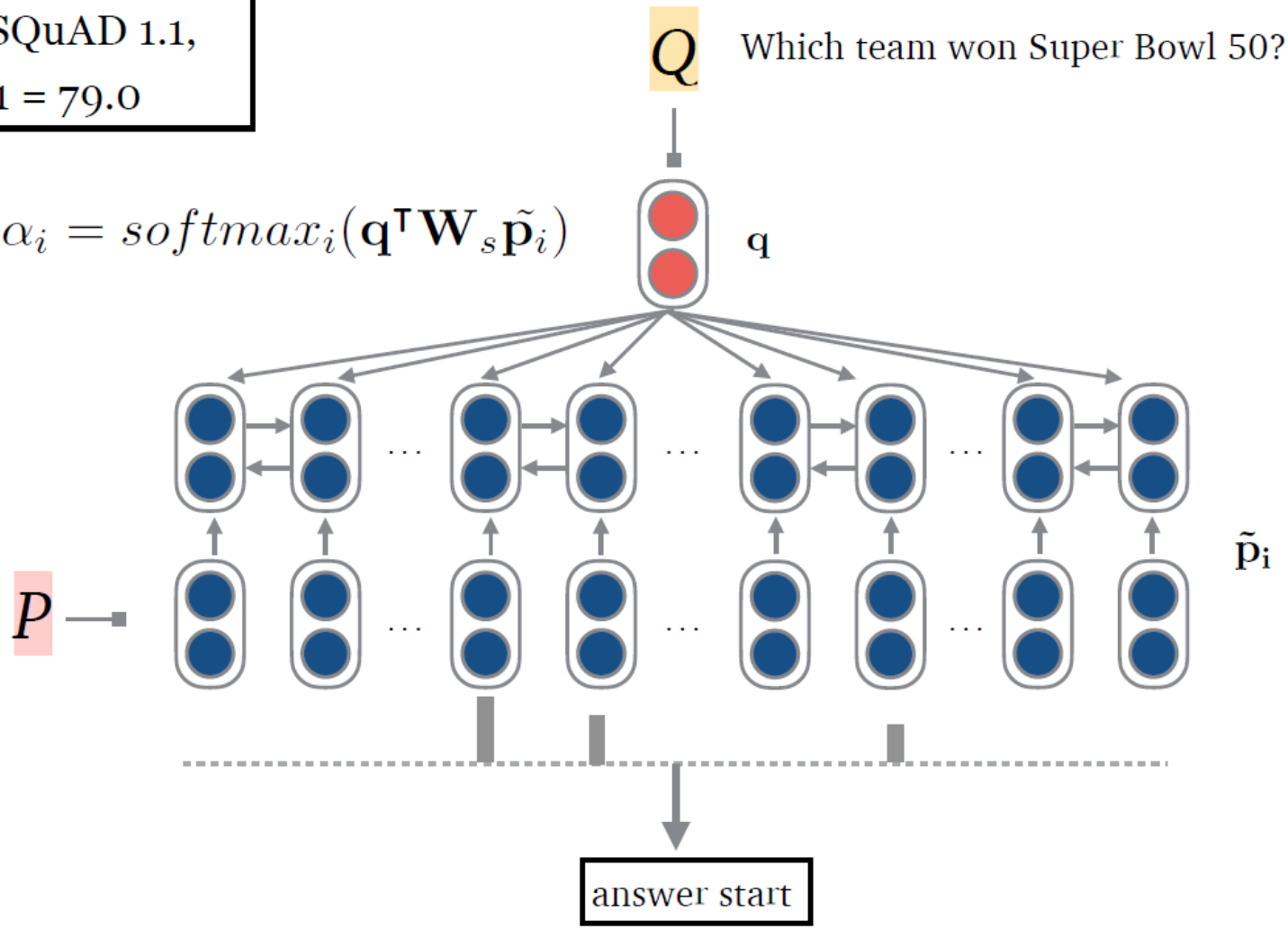
- (passage, question, answer) triples
- Passage is from Wikipedia, question is crowd-sourced
- Answer must be a span of text in the passage (aka. "extractive question answering")
- SQuAD 1.1: 100k answerable questions, SQuAD 2.0: another 50k unanswerable questions

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

Document Reader

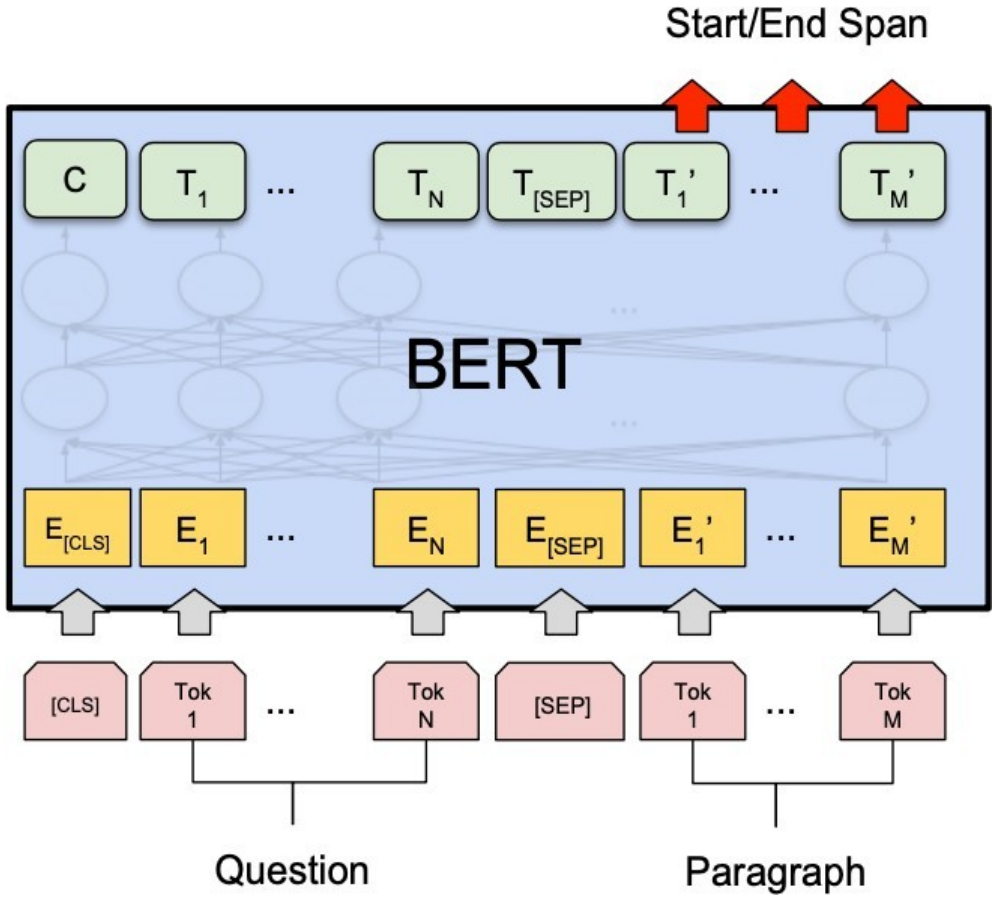
$$P_{start}(i) \propto \exp(\mathbf{p}_i \mathbf{W}_s \mathbf{q})$$
$$P_{end}(i) \propto \exp(\mathbf{p}_i \mathbf{W}_e \mathbf{q})$$
$$P_{start}(i) \times P_{end}(i') \text{ is maximized}$$

On SQuAD 1.1,
- F1 = 79.0



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,

- : F1 = 90.9
- RoBERTa: F1 = 94.6

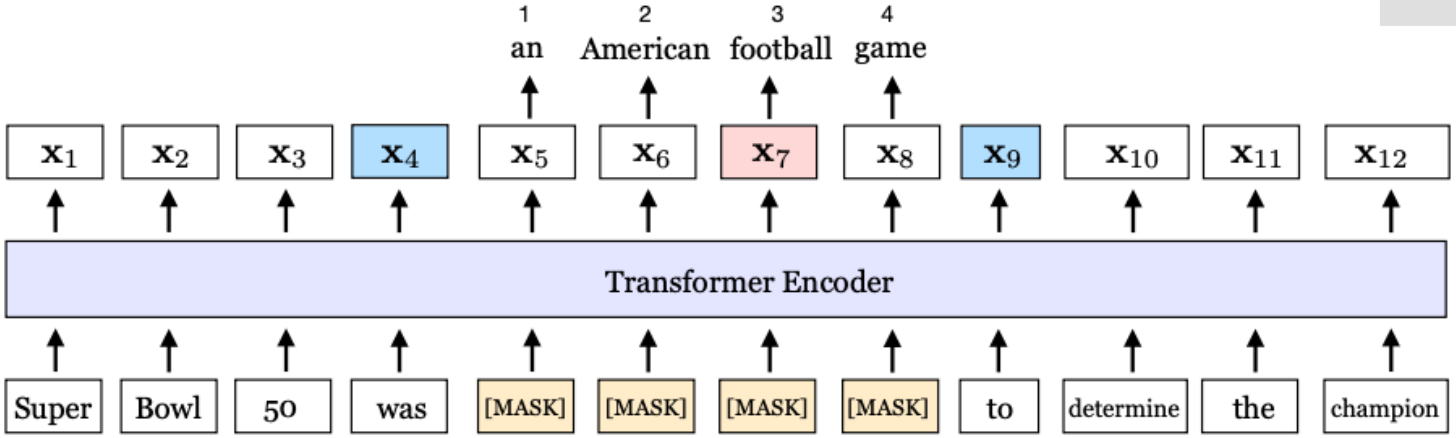
Devlin et al., 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

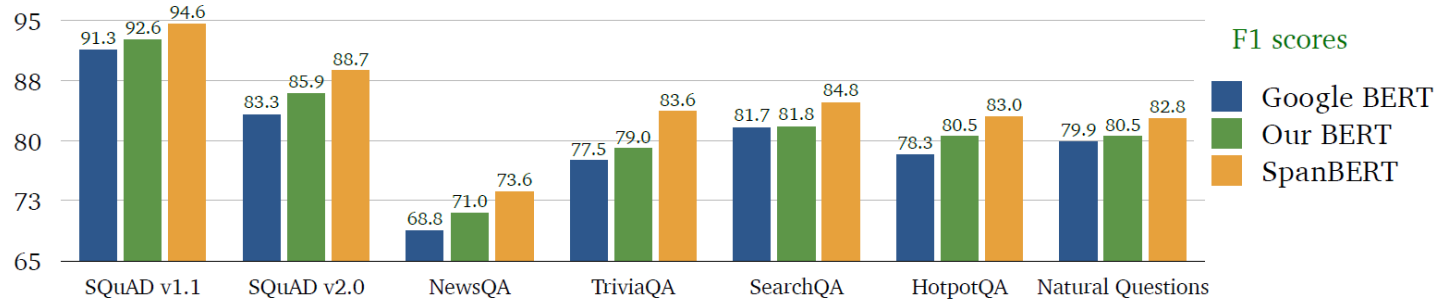
SpanBERT

An illustration of SpanBERT training. The span an American football game is masked. The SBO (span-boundary objective) uses the output representations of the boundary tokens

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

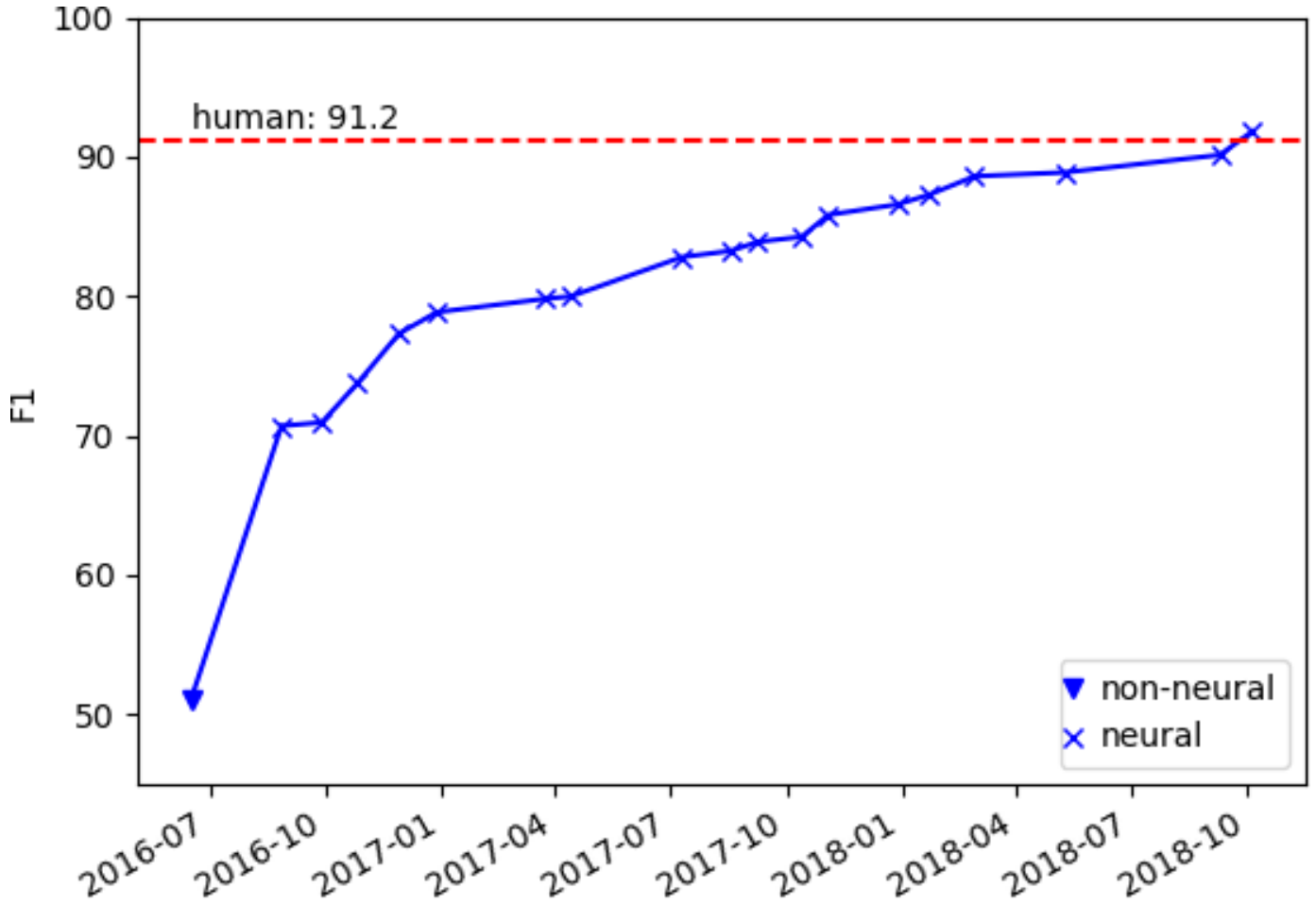
Position embeddin g





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

Stanford Question Answering Dataset (SQuAD)



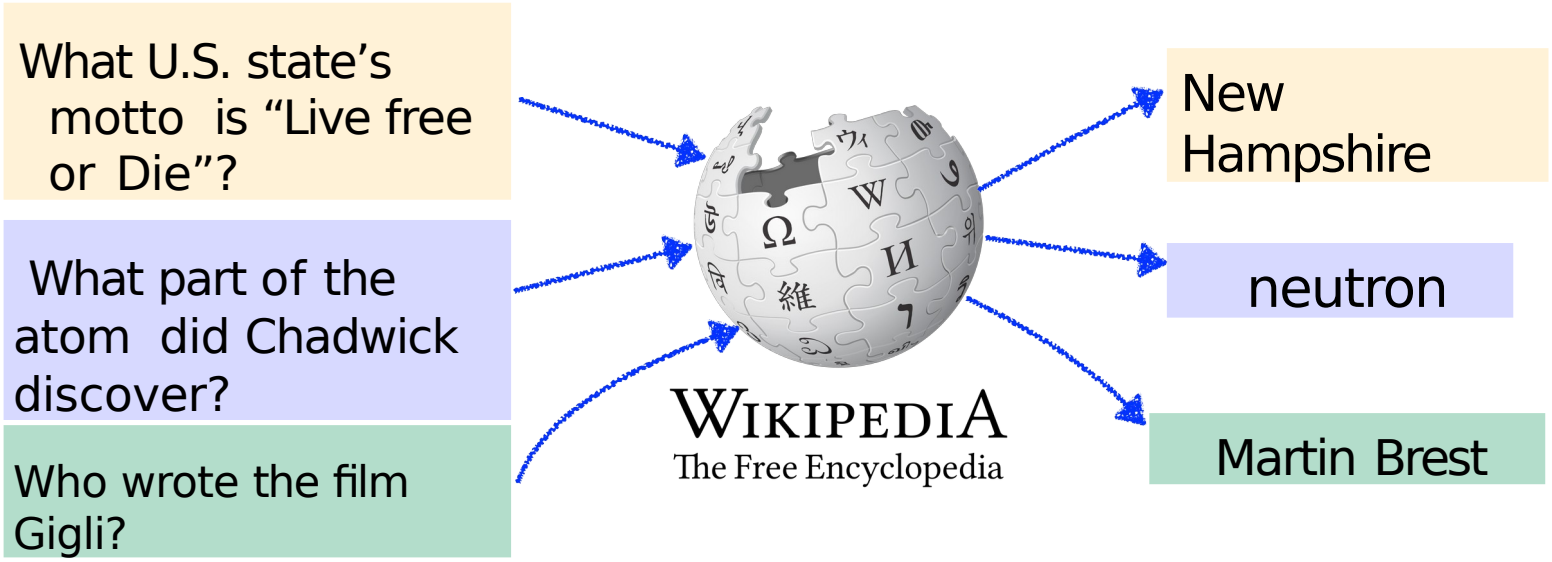
Rajpurkar et al., 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text

Outline

- Two-stage **retriever-reader** approaches

Problem setup

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



"Machine Reading at Scale"

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

Why is Wikipedia?

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

DrQA: a first neural open-domain QA system

[Chen et al., 2017]

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

Document Retriever

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:

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

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

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

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

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

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

Document Reader

paragraph, document,
arbitrary-length text blocks.

Cast as a *reading comprehension* problem:

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

A restricted setting is that A needs
to be a segment of text in P =>
"extractive question answering"

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

Passage

Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24–10 to earn their third Super Bowl title. The game was played on February 7, 2016, at Levi's Stadium in the San Francisco Bay Area at Santa Clara, California.

Question: Which NFL team won Super Bowl 50?
Answer: Denver Broncos

Question: What does AFC stand for?
Answer: American Football Conference

Question: What year was Super Bowl 50?
Answer: 2016

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

How to train the model?

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

$$D_{rc} = \{(P_i, Q_i, A_i)\}$$

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

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

$$D_{QA} = \{(Q_i, A_i)\}$$

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

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

if passage is retrieved and answer can be found in passage

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

How to train the model?

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

Answer: New Hampshire

Passage

Live Free or Die

From Wikipedia, the free encyclopedia

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

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

Putting together

Training time:

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

Inference time:

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

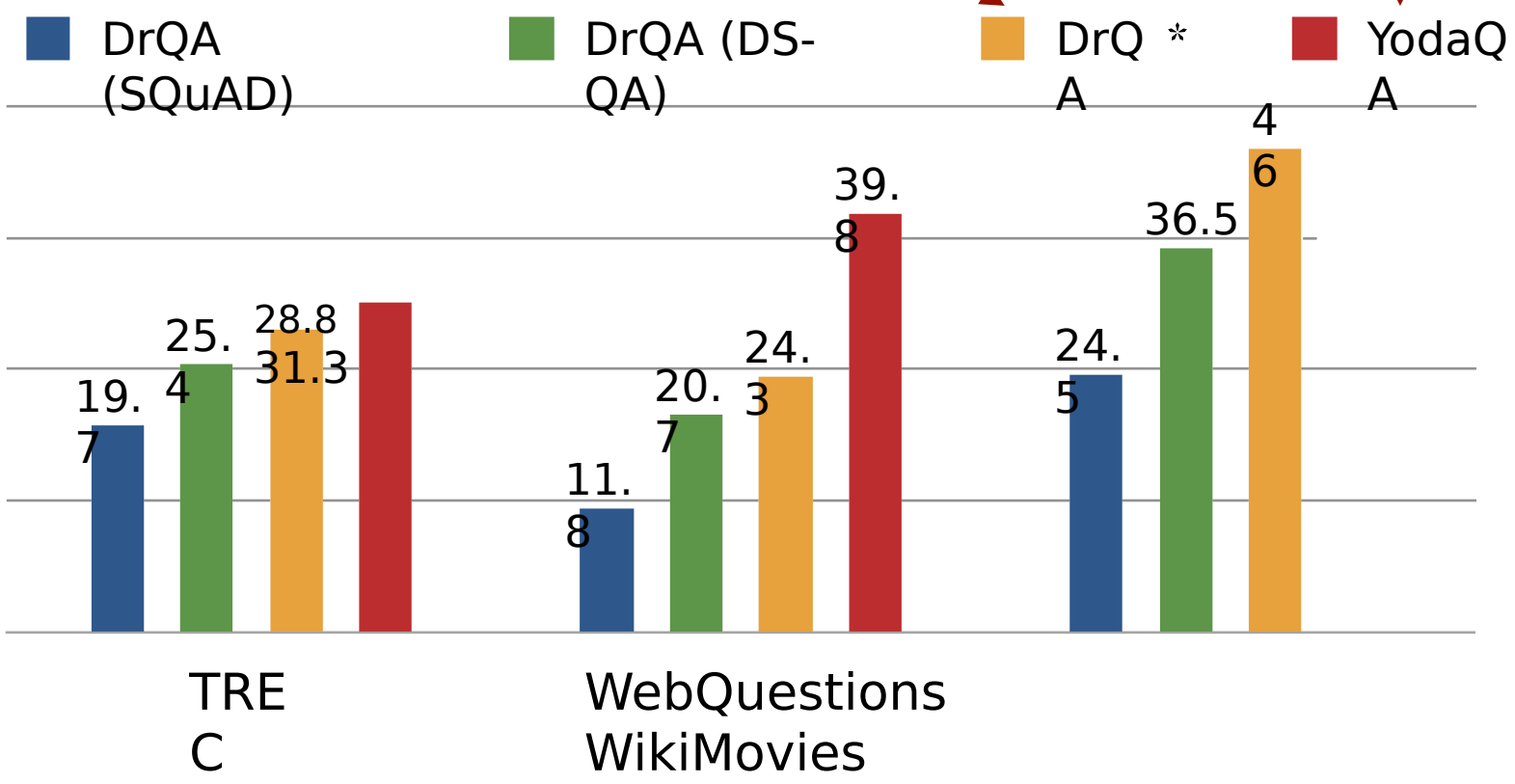
The system finally returns the answer with the highest (unnormalized) span score.

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

Experiments

See [Raison et al., 2018]
Trained reader on full documents

Hybrid system on KBs, text, ontologies etc
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 paragraph-level retriever work better?

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

  Does multi-passage training help?

- The importance of each passage has been omitted.

  Can we use passage retriever scores or even train a better ranker?

- The retriever is not trained!

  The focus of the next part.

Outline

- **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.
1934, see Adamowicz brothers.

Warsaw (Polish: Warszawa [vaˈʂava] (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.[3][5][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).[6]

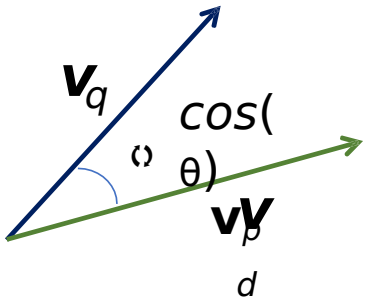
In 2012 the Economist Intelligence Unit ranked Warsaw as the 32nd most liveable city in the world.[8] It was also ranked as one of the most liveable cities in Central Europe. Today Warsaw is considered an "Alpha–" global city, a major international tourist destination and a significant cultural, political and economic hub.[7][9][10] 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

Can we also train the retriever component?

Can we use dense representations for

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] $\in \mathbb{R}^{d_2}$

They capture complementary information; Dense representations have never been

shown to outperform sparse representations in open-domain QA before 2019!

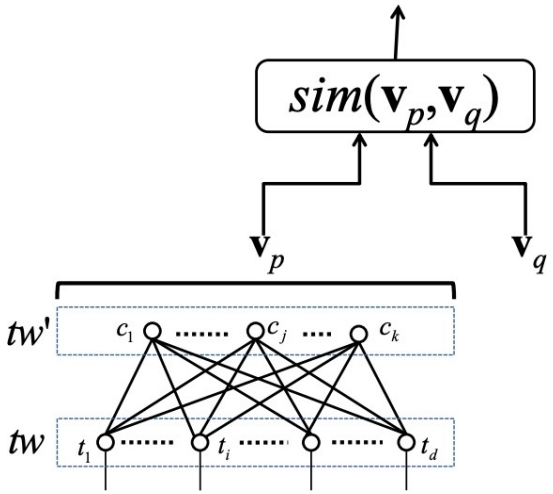
Why dense retrieval **now**?

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

CoNLL'2011 Best paper

Learning Discriminative Projections for Text Similarity Measures

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**?

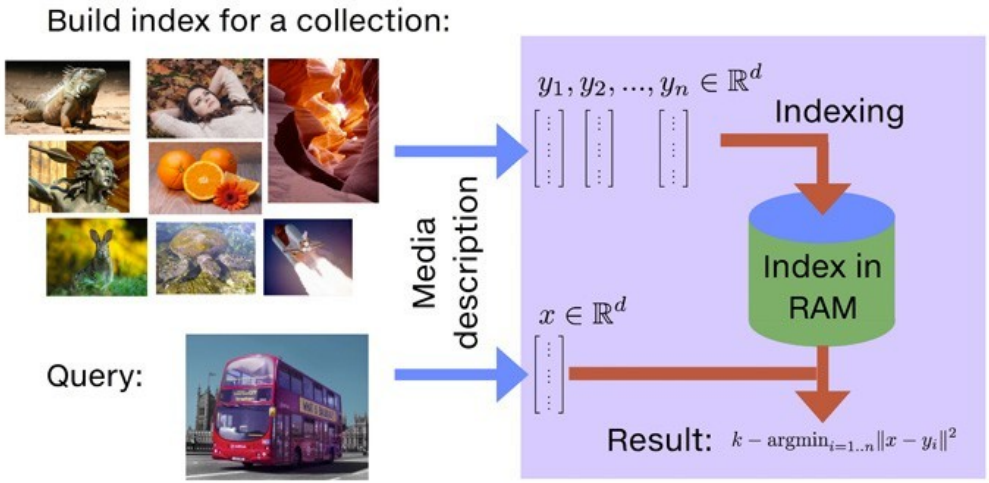
It is actually not easy to make these dense models “work”.

- 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: **O**pen-**R**etriever **Q**uestion **A**nswering

[Lee et al., 2019]

First model to learn retriever and reader jointly

Key contributions:

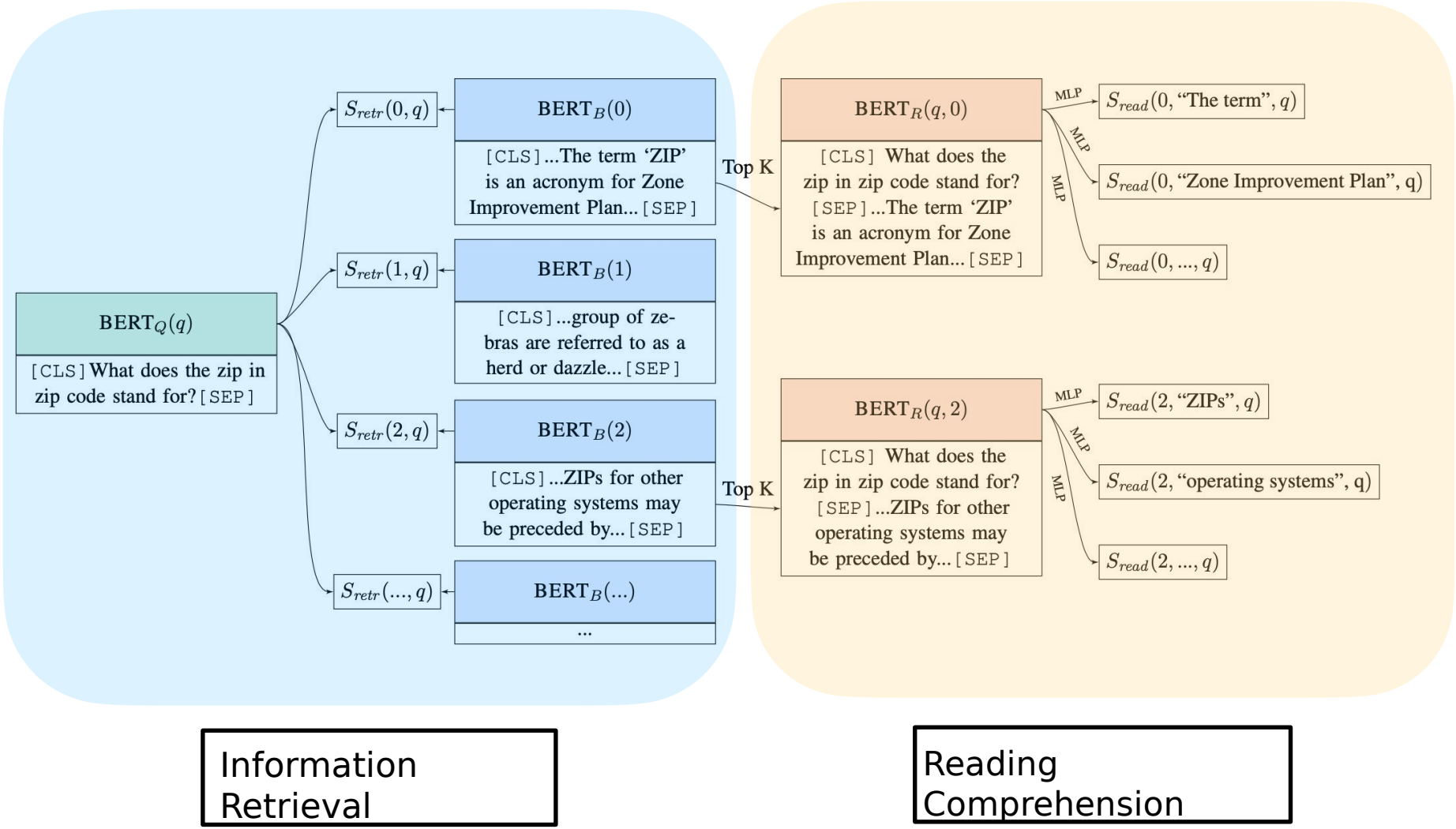
- Both retriever and reader are learnable with NNs (= BERT)
- Only learned from question-answer pairs; No reading comprehension datasets!

$$D_{\mathrm{QA}} = \{(Q_i, A_i)\}$$

- 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

$$S(b, s, q)$$

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

Fixed-length blocks as “passages”: 288 wordpieces => 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

Retriever score:

$S_{retr}(b, q)$

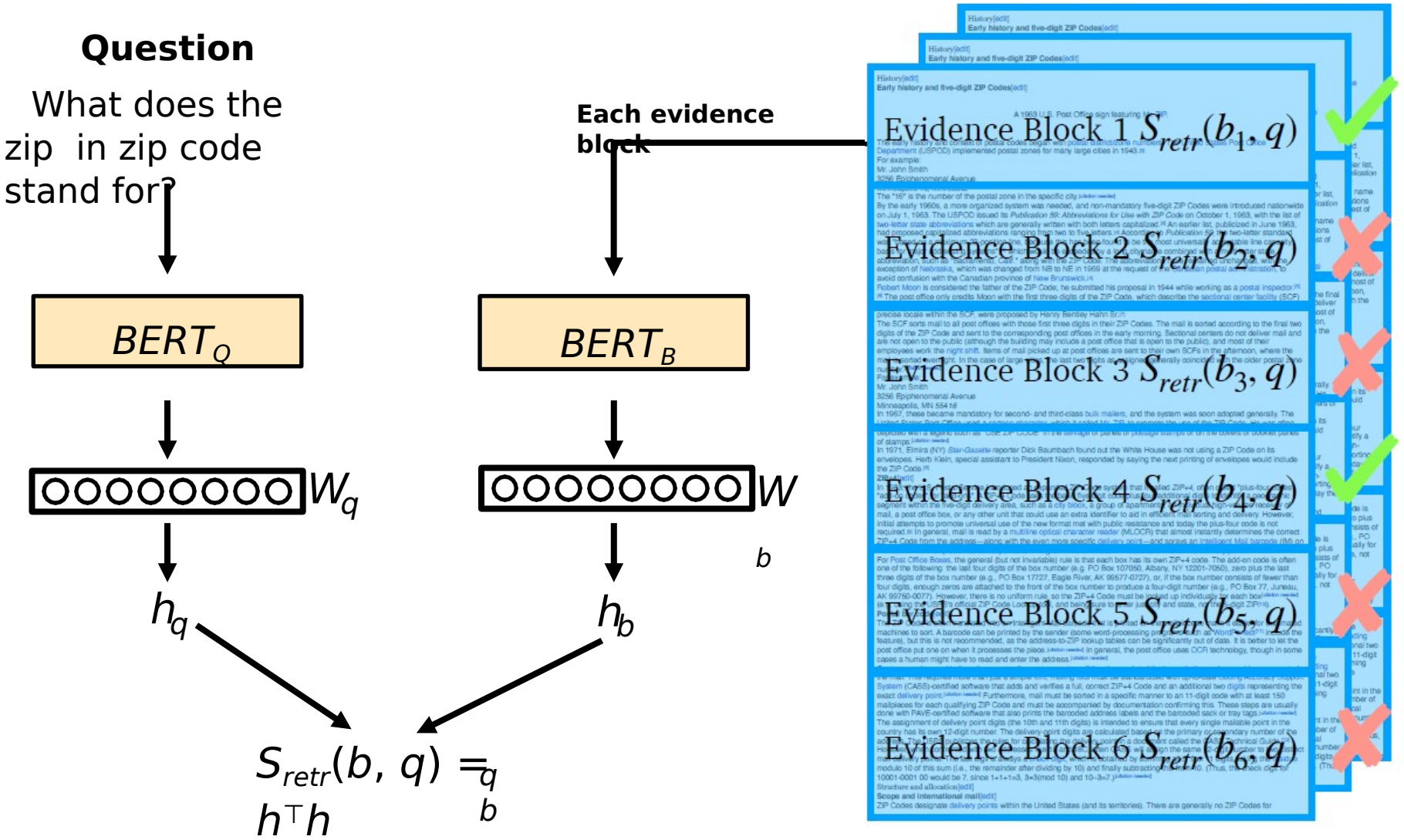
$$h_q = \mathbf{W_q}\mathbf{BERT}_Q(q)[\texttt{CLS}]$$
$$h_b = \mathbf{W_b}\mathbf{BERT}_B(b)[\texttt{CLS}]$$
$$S_{retr}(b, q) = h_q^\top h_b$$

All of Wikipedia: select top K

Question

What does the zip in zip code stand for?

Each evidence block

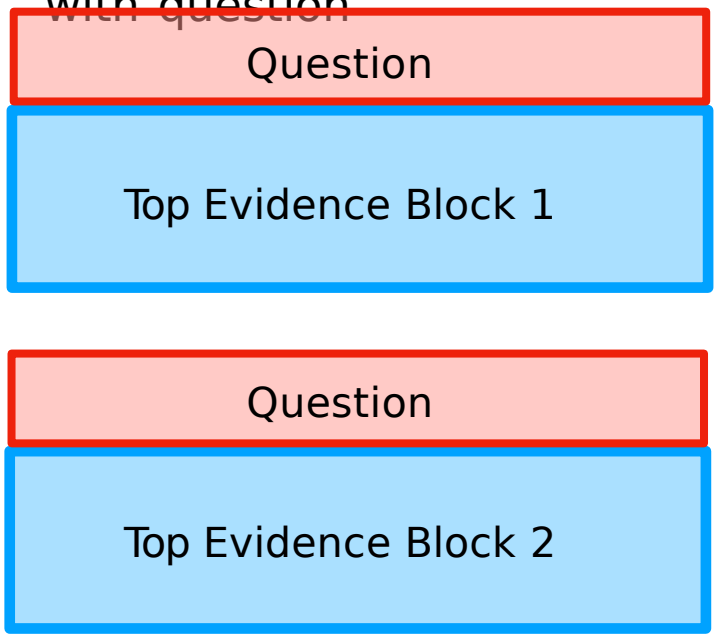


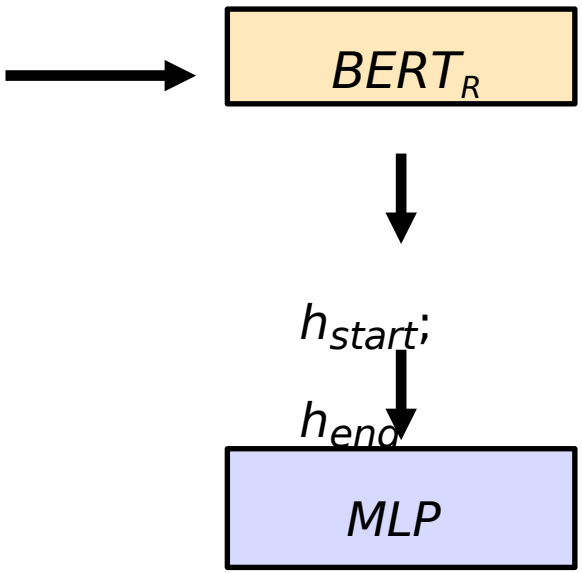
Reader score:

$S_{read}(b, s, q)$

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

Top-K evidence blocks concatenated with question





$S_{read}(b, s, q)$

Notations:

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

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

How is this model learned?

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

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

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

Loss function

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

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

How is this model learned?

Early learning: consider a larger set of evidence blocks

$c = 5000$

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

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

Final loss

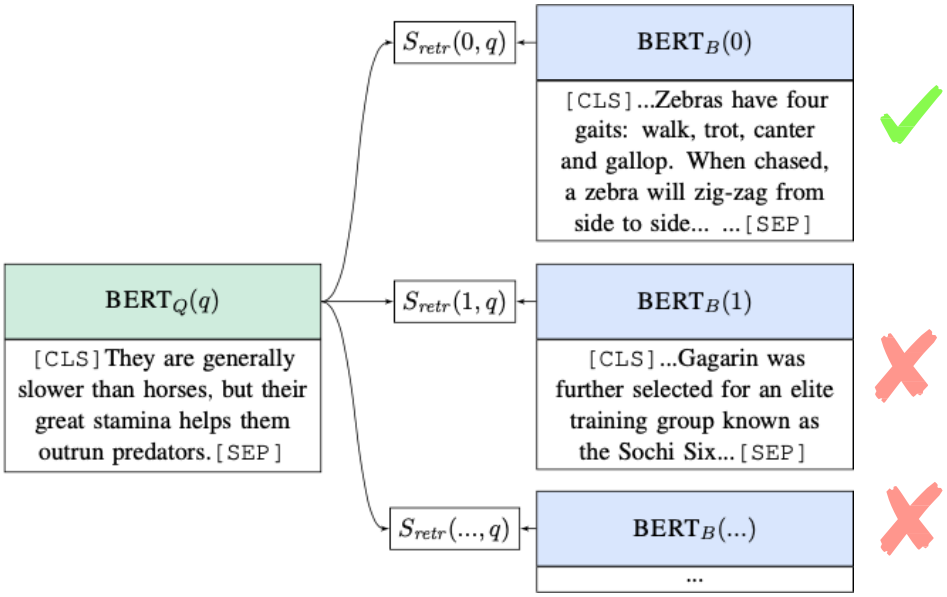
$$L(q,a) = L_{\text{early}}(q,a) + L_{\text{full}}(q,a)$$

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.

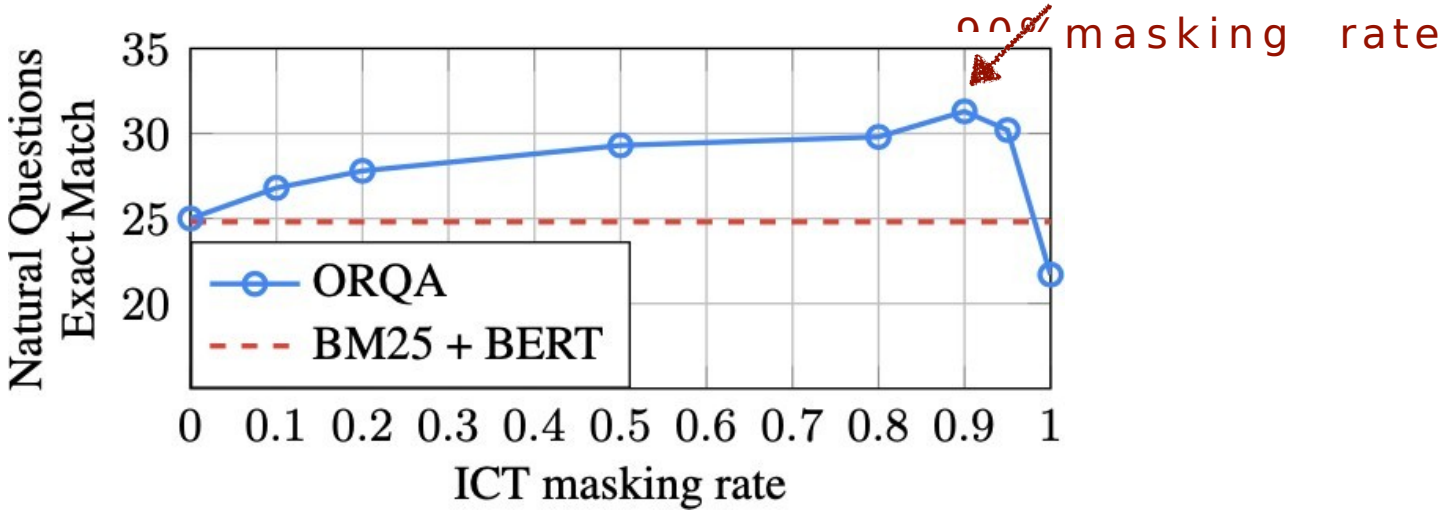
...Zebras have four gaits: walk, trot, canter and gallop. **They are generally slower than horses, but their great stamina helps them outrun predators.** When chased, a zebra will zig- zag from side to side...



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

Pre-training: Inverse Cloze Task (ICT)

Still encourages the model to learn word matching — only remove sentences from its context in 90% of the examples!



Important - 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

REALM: **Re**trieval-**a**ugmented **L**anguage **M**odel

[Guu et al., 2020]

Pre-training on both retriever and reader!

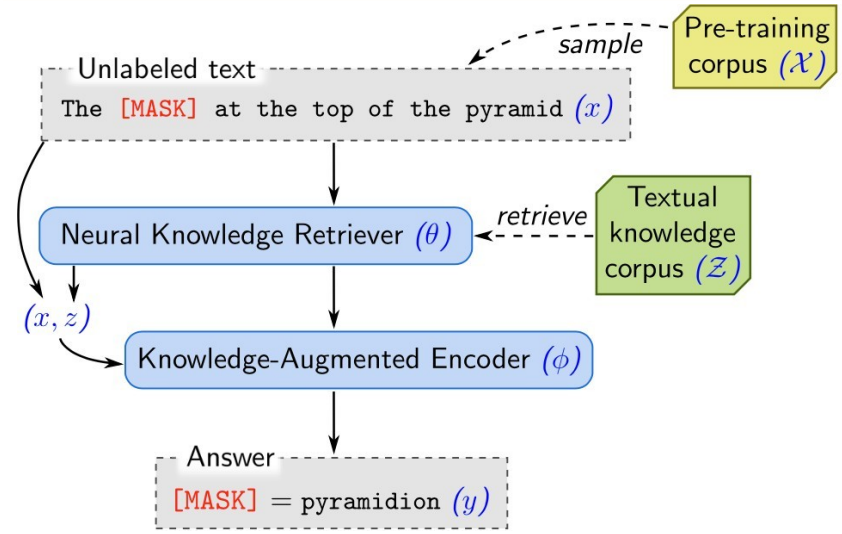
$$p(y \mid x) = \sum_{z \in \mathcal{Z}} p(y \mid z, x)\, p(z \mid x)$$

Z=Top(K) passages

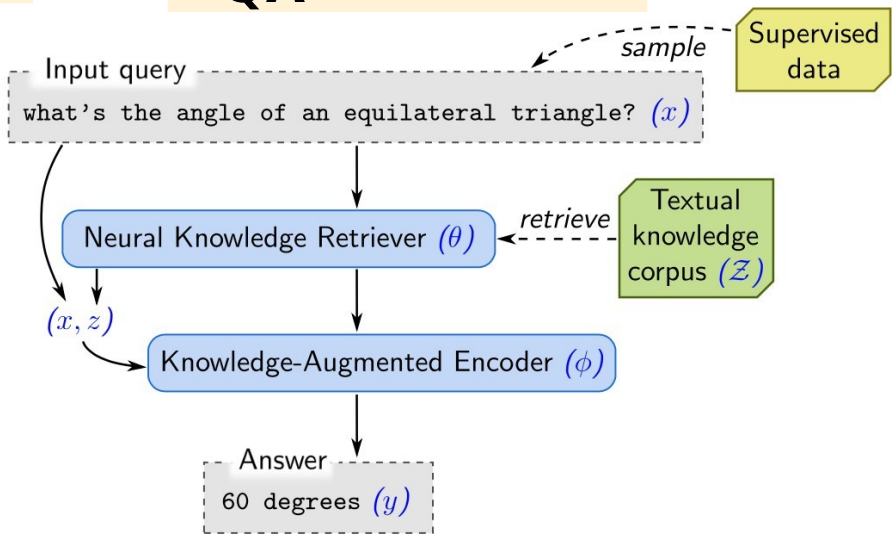
Compared to ORQA:

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

Pre-training: masked language model (MLM)

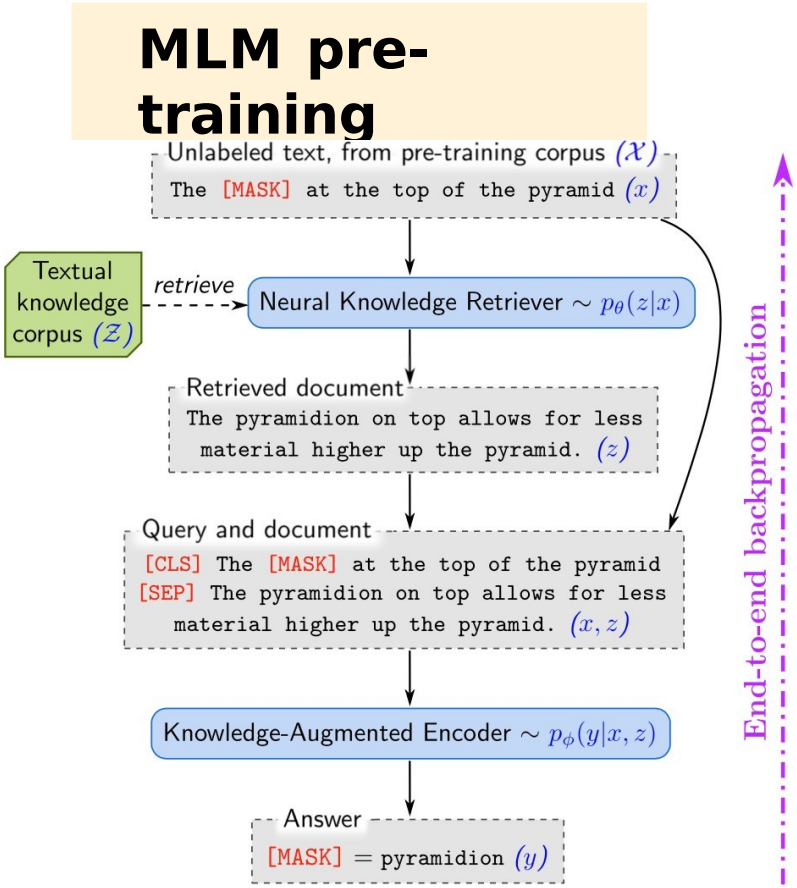


Fine-tuning: QA



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

REALM: MLM pre-training



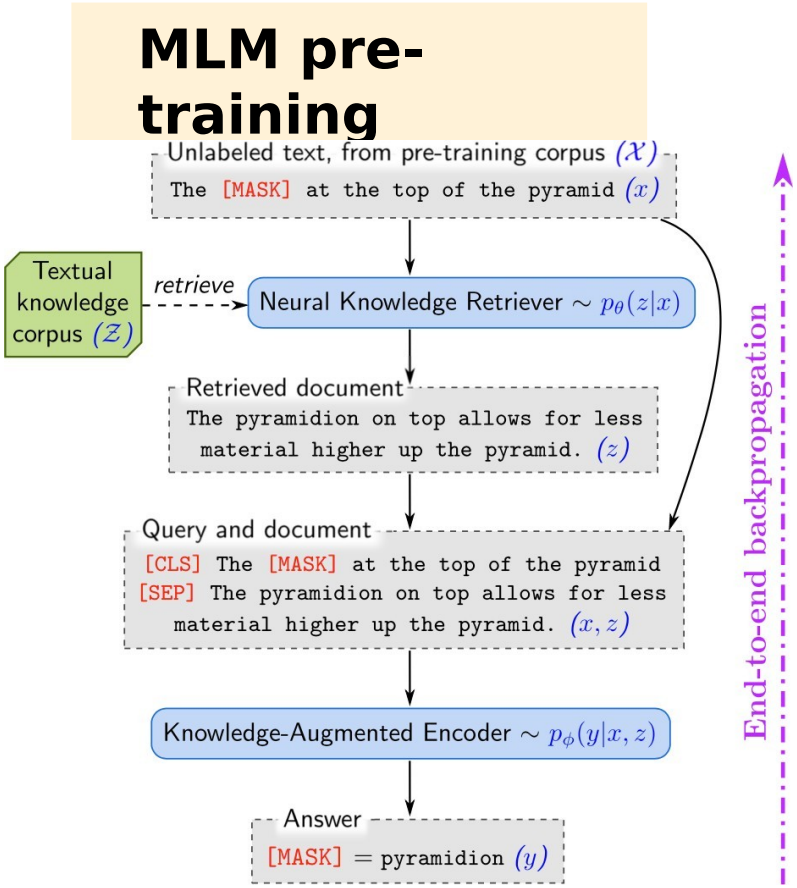
- Cold-start is hard => Use ICT as a first-stage pre-training
- Salient span masking: similar to span masking [Joshi et al., 2020] but only mask named entities and dates.

Evaluated on NQ

Random **uniform** masks: 32.3 Random **span** masks: 35.3 **Salient** span masks: 38.2

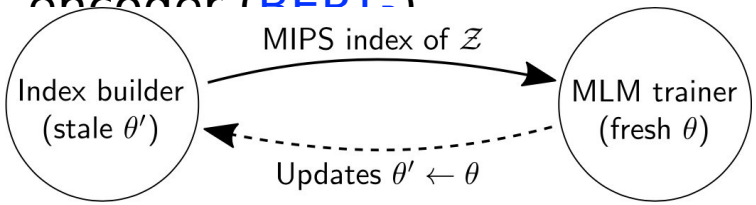
Guu et al., 2020. REALM: Retrieval-Augmented Language Model Pre-Training

REALM: MLM pre-training



- Can use corpora even larger than Wikipedia for pre-training
  - CC-News vs Wikipedia: 40.4 vs 39.2

- Can allow updating evidence encoder (REPT-)



MIPS: Maximum Inner Product Search

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?

Retrieval-free QA: Probing Language Models

- 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.

Language Models as Knowledge Bases

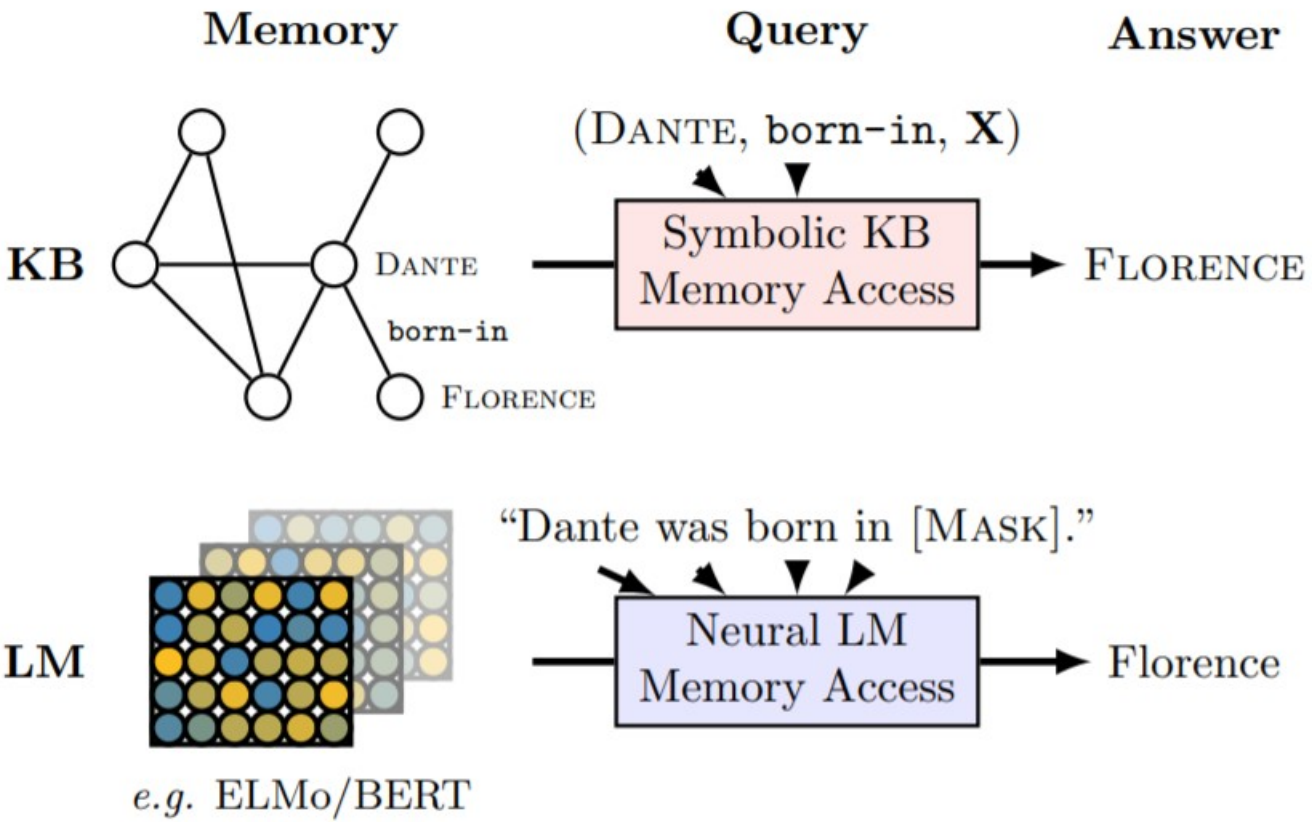


Figure 1: Querying knowledge bases (KB) and language models (LM) for factual knowledge.

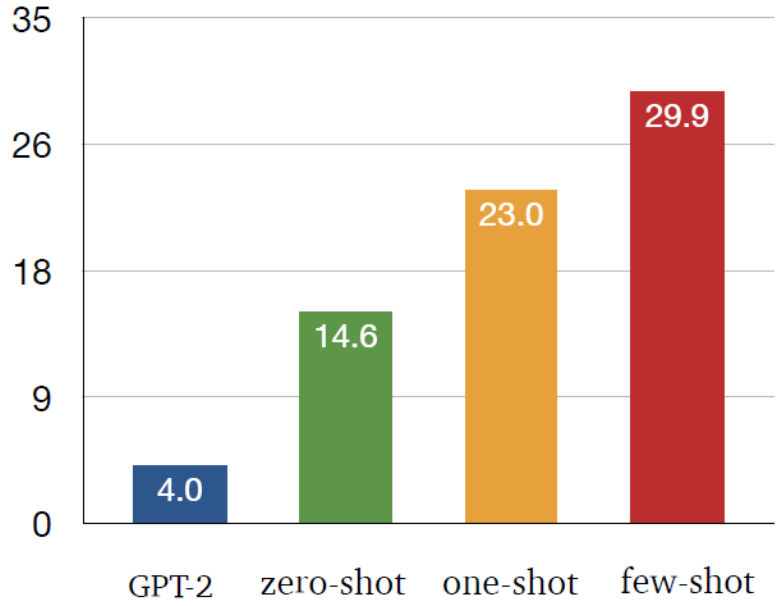
Petroni et al., 2019. Language Models as Knowledge

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



Few-shot learner

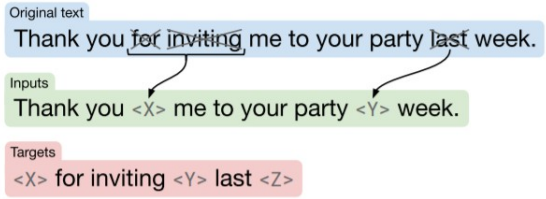
- Q_1, A_1, Q_2, A_2, .., Q_K, A_K, Q ?
- One-shot setting is a special case when only **one** example is given.

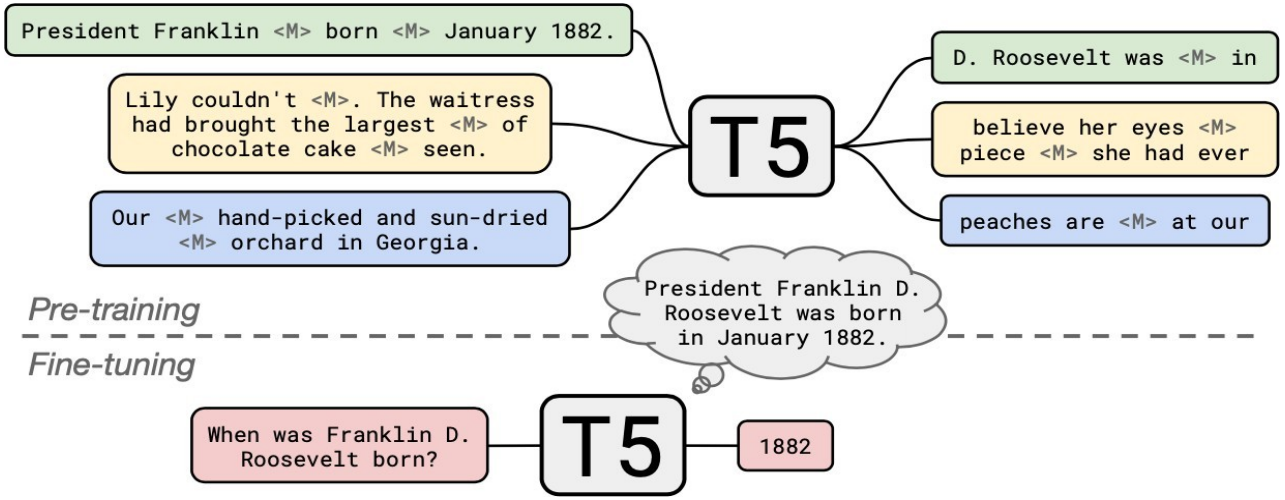
Brown et al., 2020. Language Models are Few-Shot Learners

T5: Fine-tuning leads to improved performance

[Roberts et al., 2020]

Text-**t**o-**T**ext **T**ransfer **T**ransformer [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
	T5-Large	29.8	32.2	35.9
770M	T5-3B	32.1	34.9	43.4
	T5-11B	34.5	37.4	50.1
3B	T5-11B + SSM	36.6	**44.7**	**60.5**
11B				

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?

Why retrieval-based LMs?

LLMs can't memorize all (long-tail) knowledge in their parameters

List 5 important papers authored by Geoffrey Hinton

ChatGPT

Geoffrey Hinton is a renowned computer scientist ... Here are five important papers authored by him:

1."**Learning Internal Representations by Error Propagation**" (with **D. E. Rumelhart** and **R. J. Williams**) - This paper, published in **1986**, .. ✓

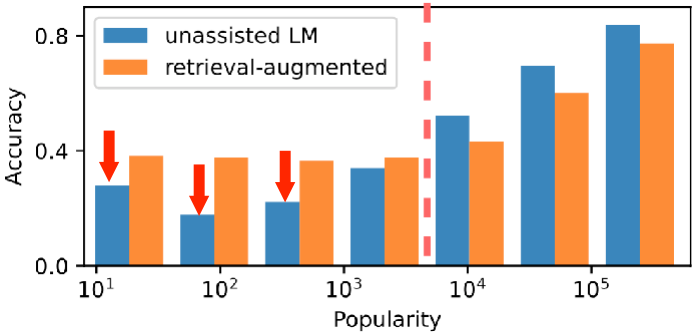
2."**Deep Boltzmann Machines**" (with **R. Salakhutdinov**) - Published in **2009**, .. ✓

...

4."**Deep Learning**" (with Y. Bengio and A. Courville) - Published as a book in **2016**,... ✗

5."**Attention Is All You Need**" (with V. Vaswani, N. Shazeer, et al.) - Published in **2017**, this paper introduced the Transformer model,... ✗

What is Kathy Saltzman's occupation?



(Mallen et al., 2023)

GPT-3 davinci-003: 20%-30% accuracy

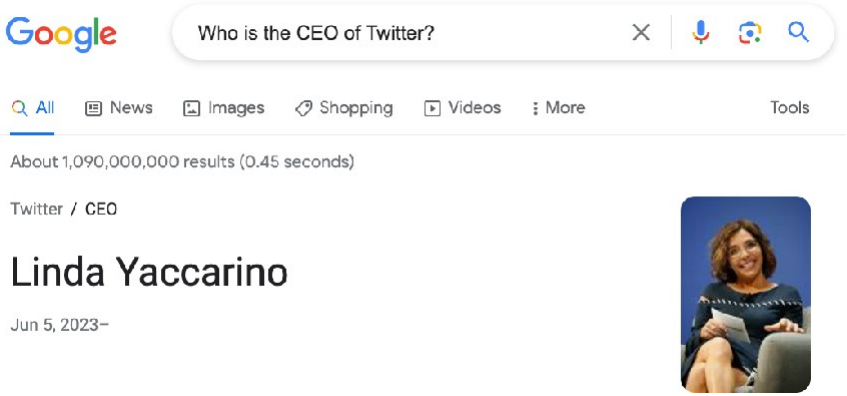
Why retrieval-based LMs?

LLMs' knowledge is easily outdated and hard to update

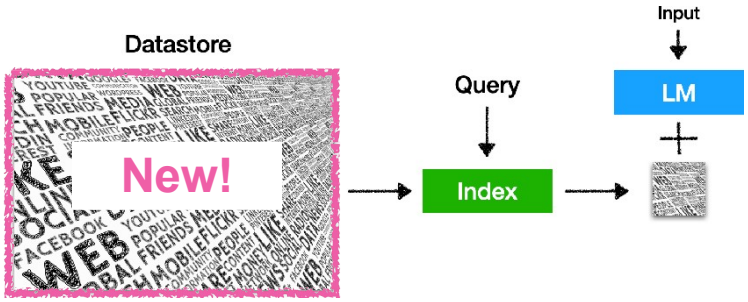
Who is the CEO of Twitter?

ChatGPT

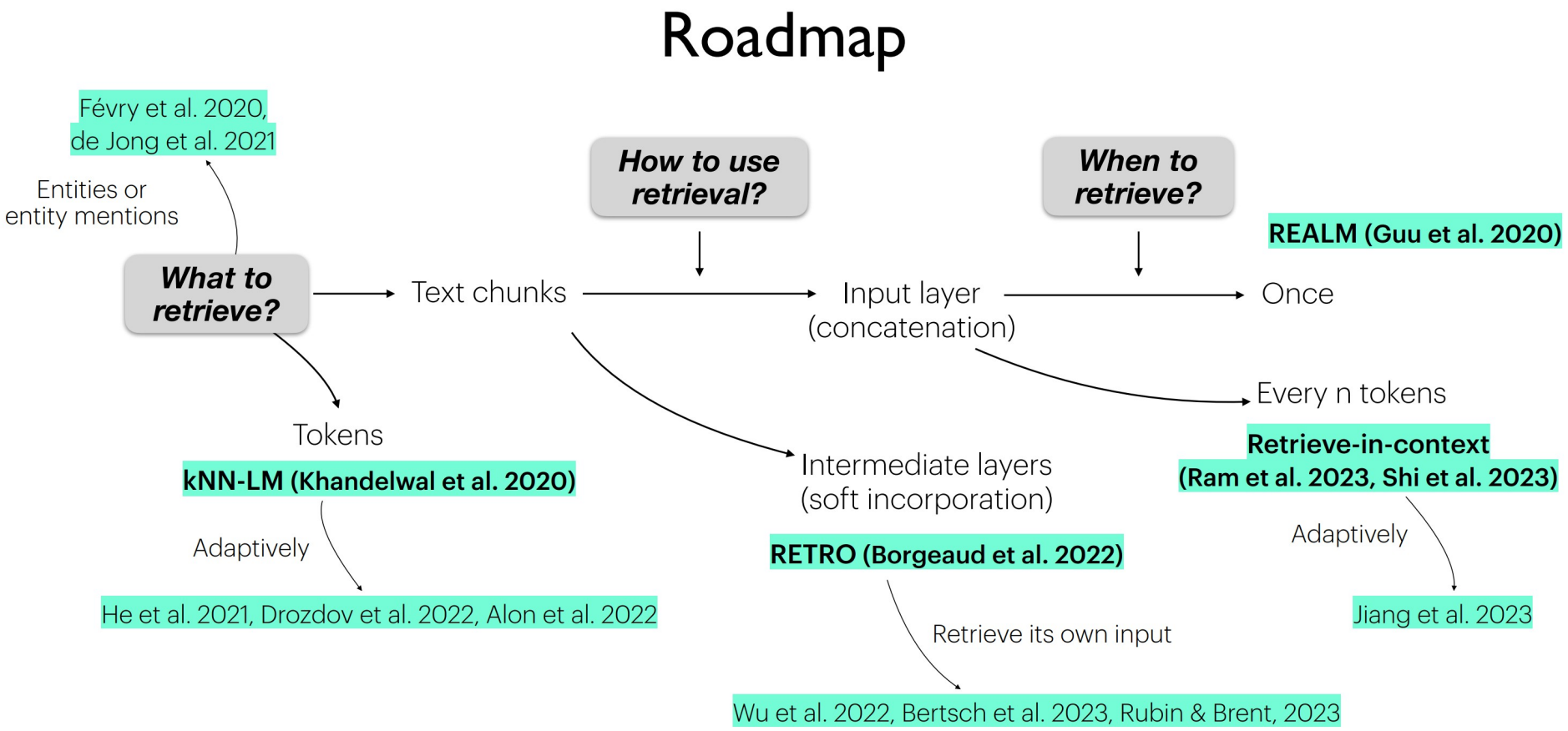
As of my **knowledge cutoff in September 2021**, the CEO of Twitter is **Jack Dorsey**….



- Existing **knowledge editing** methods are still NOT scalable (active research!)
- The datastore can be easily **updated** and **expanded** - even without retraining!



Retrieval Augmented Generation



REPLUG

- **Title:** REPLUG: Retrieval-Augmented Black Box Language Models
- **Authors:** Shi et al. (University of Washington, Stanford, KAIST, Meta AI)
- **Publication:** ArXiv 2023
- **Link:** https://arxiv.org/abs/2301.12652

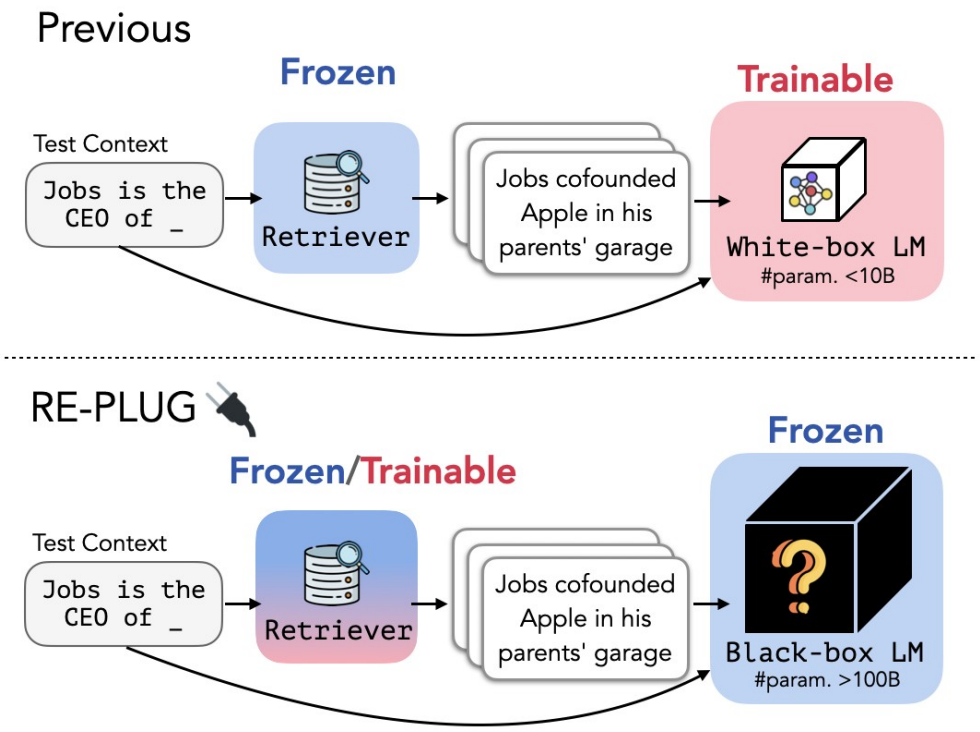
Motivation: RAG for Black-Box LMs

- **Issue:** Fine-tuning LMs with retrieval requires white-box access (i.e., access to the LM parameters)
  - To train the model
  - To index the datastore
- **In Practice:** Many LLMs are only accessible via API
  - Can't access model parameters!
  - Can't fine-tune!




Introducing REPLUG

- **Focus:** Retrieval-augmentation in the black-box setting (no access to model parameters)
  - LM is a black box
  - Retriever is optionally tunable
- **How does it work?**
  1. Get documents from retriev
  2. Prepend documents to LM i
  3. Feed input to LM



Implementation: Retrieval

- **Given:**
  - Corpus of documents
  - Input context
- **Output:**
  - The documents in that are most relevant to , denoted
- **How:**
  - Precompute embedding for each document (construct FAISS index)
  - Encode input to obtain embedding
  - Find top- most similar documents to using *cosine* similarity function

Hang On, Can We Fit Documents?

- **Context Window:** The input text for an LM
- **Context Window Size:** Number of tokens an LM can process as input
- **Problem:** Context window may not fit documents
  - The context window size varies by LM
  - The size of each document could vary
  - The number of retrieved documents could vary

Solution: Ensemble Inference Scheme

- **Idea:** Do separate LM predictions!
  1. Concatenate each with
  2. Ensemble the output probabilities from inference calls