Open-domain Question Answering

Presented by Kun Lu and Chris Sciavolino
03/10/2020
Open-domain Question Answering

DrQA Web UI: https://github.com/zaghaghi/drqa-webui
**A Brief History of Open-domain Question Answering**

- **Simmons et al. (1964)** did first exploration of answering questions from an expository text based on matching dependency parses of a question and answer.

- **Murax (Kupiec 1993)** aimed to answer questions over an online encyclopedia using IR and shallow linguistic processing.

- **The NIST TREC QA track** begun in 1999 first rigorously investigated answering fact questions over a large collection of documents.

- **IBM’s Jeopardy! System (DeepQA, 2011)** brought attention to a version of the problem; it used an ensemble of many methods.

- **DrQA (Chen et al. 2017)** uses IR followed by neural reading comprehension to bring deep learning to Open-domain QA.
IBM's Watson and Jeopardy! Challenge

Sample questions:

Q: Even a broken one of these on your wall is right twice a day

A: clock. Watson got it correctly.

Q: Its largest airport is named for a World War II Hero; its second largest for a World War II Battle

A: Chicago. Watson didn't get it correctly.

IBM Watson defeated two of Jeopardy's greatest champions in 2011
vs Reading Comprehension
vs Reading Comprehension

1. Much Harder!
vs Reading Comprehension

1. Much Harder!

Combining challenges of both large-scale open-domain QA and of machine comprehension
vs Reading Comprehension

2. Very General!
2. Very General!

the question can be any open-domain questions (instead of questions posed after reading the passage) and this meets people’s real information seeking
Overview

• (Chen et al, ACL’ 2017) Reading Wikipedia to Answer Open-Domain Questions
• (Lee et al, ACL’ 2019) Latent Retrieval for Weakly Supervised Open Domain Question Answering
Overview

• (Chen et al, ACL ’2017) Reading Wikipedia to Answer Open-Domain Questions

• (Lee et al, ACL’ 2019) Latent Retrieval for Weakly Supervised Open Domain Question Answering
Reading Wikipedia to Answer Open-Domain Questions

Danqi Chen*
Computer Science
Stanford University
Stanford, CA 94305, USA
danqi@cs.stanford.edu

Adam Fisch, Jason Weston & Antoine Bordes
Facebook AI Research
770 Broadway
New York, NY 10003, USA
{afisch,jase,abordes}@fb.com
Agenda

1. Introduction of DrQA
2. Document Retriever
3. Document Reader
4. Data
5. Results
Agenda

1. Introduction of DrQA
2. Document Retriever
3. Document Reader
4. Data
5. Results
Traditional QA System

Turn-of-the Millennium Full NLP QA:
[architecture of LCC (Harabagiu/Moldovan) QA system, circa 2003]
Complex systems but they did work fairly well on “factoid” questions
System: DrQA

• Part 1. Document Retriever
  ✓ Finding relevant articles

• Part 2. Document Reader
  ✓ Extracting answers
Contributions

• DrQA was trying to *reduce this complex problem into a simple two-stage retriever and reader problem*, by combining the challenges from IR and reading comprehension (and this was just a few months after SQuAD came out)

• DrQA: *can be applied to any large collection of documents* (e.g. the whole Web) but we chose to use the English Wikipedia as the knowledge source, which consists of 5M articles.
Hello! Please ask a question.

What is question answering?

a computer science discipline within the fields of information retrieval and natural language processing

Who was the winning pitcher in the 1956 World Series?

Don Larsen

What is the answer to life, the universe, and everything?

42
Agenda

1. Introduction of DrQA
2. Document Retriever
3. Document Reader
4. Data
5. Results
Methods: two steps

1. TF-IDF bag-of-words vectors
2. Efficient bigram hashing (Weinberger et al., 2009)
Methods: two steps

• 1. TF-IDF bag-of-words vectors
• 2. Efficient bigram hashing (Weinberger et al., 2009)
TF-IDF bag-of-words vectors

• TF-IDF vectors:

\[ w_{i,j} = tf_{i,j} \times \log \left( \frac{N}{df_i} \right) \]

- \( tf_{i,j} \) = number of occurrences of \( i \) in \( j \)
- \( df_i \) = number of documents containing \( i \)
- \( N \) = total number of documents

• Improve unigram by considering local word order using n-gram
• Compare articles and questions to retrieve
TF-IDF bag-of-words vectors

• TF-IDF vectors:

\[ w_{i,j} = tf_{i,j} \times \log \left( \frac{N}{df_i} \right) \]

- \( tf_{i,j} \): number of occurrences of \( i \) in \( j \)
- \( df_i \): number of documents containing \( i \)
- \( N \): total number of documents

• Improve unigram by considering local word order using n-gram
• Compare articles and questions to retrieve

High dimensional issue?
Methods: two steps

1. TF-IDF bag-of-words vectors
2. Efficient bigram hashing (Weinberger et al., 2009)
Efficient bigram hashing (Weinberger et al., 2009)

• Map the bigrams to $2^{24}$ bins with an unsigned murmur3 hash
  • Preserving speed and memory efficiency (Weinberger et al., 2009)

• **Murmur3**: Map a word or string to a 32-bit or 128bit value
  • Online: [http://murmurhash.shorelabs.com/](http://murmurhash.shorelabs.com/)
Feature Hashing

<table>
<thead>
<tr>
<th>Sentence</th>
<th>Murmurhash3</th>
<th>Divide by</th>
<th>Reminder</th>
</tr>
</thead>
<tbody>
<tr>
<td>john</td>
<td>3487894951</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>likes</td>
<td>1103617568</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>movies</td>
<td>3188341541</td>
<td>8</td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Index</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>likes</td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>movies</td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>john</td>
</tr>
</tbody>
</table>
Feature Hashing

What if we have hash collisions?
Feature Hashing (Weinberger et al., 2009)

• Mathematical formula:

\[ \phi_i^{(h,\xi)}(x) = \sum_{j:h(j)=i} \xi(j)x_j \]

and \( \langle x, x' \rangle_\phi := \left\langle \phi^{(h,\xi)}(x), \phi^{(h,\xi)}(x') \right\rangle \).

• They proved exponential tail Bounds
Agenda

1. Introduction of DrQA
2. Document Retriever
3. Document Reader
4. Data
5. Results
Document Reader

Three steps:

1. Paragraph encoding
2. Question encoding
3. Prediction

similar to AttentiveReader (Hermann et al, 2015; Chen et al, 2016)
Document Reader

Three steps:
1. Paragraph encoding
2. Question encoding
3. Prediction
Who did Genghis Khan unite before he began conquering the rest of Eurasia?

Bidirectional LSTMs

\[ P_s(i) = \text{softmax}_i(qW_s\tilde{p}_i) \quad \text{predict start token} \]

\[ P_e(i) = \text{softmax}_i(qW_e\tilde{p}_i) \quad \text{predict end token} \]
Paragraph encoding

1. Represent tokens $p_i$ in a paragraph as a sequence of feature vectors $\tilde{p}_i \in \mathbb{R}^d$
   - Word embedding
   - Exact match
   - Token features
   - Aligned question embedding

2. Pass features $\tilde{p}_i$ as the input to a RNN (multi-layer Bidirectional LSTM):

$$\{p_1, \ldots, p_m\} = \text{RNN}(\{\tilde{p}_1, \ldots, \tilde{p}_m\})$$
Word Embeddings

- $f_{emb}(p_i) = E(p_i)$
- 300-dimensional Glove word embeddings
- Keep most of the pre-trained word embeddings fixed and only fine tune the 1000 most frequent question key words: what, how, which... (crucial for QA system)
Exact match

- \( f_{\text{exact match}} (p_i) = \mathbb{1} (p_i \in q) \)
- Binary features indicating whether \( p_i \) can be exactly matched to one question word in \( q \), either in original, lowercase, or lemma form
Token features:

- $f_{token}(p_i) = (\text{POS}(p_i), \text{NER}(p_i), \text{TF}(p_i))$
- Part of speech (POS)
- Entity recognition (NER)
- Normalized term frequency (TF)
Aligned Question Embeddings

• $f_{align}(p_i) = \Sigma_j a_{i,j} \mathbb{E} (q_j)$

• Where

• $a_{i,j} = \frac{\exp(\alpha(\mathbb{E}(p_i) \cdot \alpha(\mathbb{E}(q_j))))}{\mathbb{E}_j, \exp(\alpha(\mathbb{E}(p_i) \cdot \alpha(\mathbb{E}(q_{j'}))))}$

• $a_{i,j}$ captures the similarity between $p_i$ and $q_j$, and $\alpha(\cdot)$ is a single layer with ReLu nonlinearity
Document Reader

Three steps:

1. Paragraph encoding
2. Question encoding
3. Prediction
Who did Genghis Khan unite before he began conquering the rest of Eurasia?

\[ Q \]

\[ q \]

**Bidirectional LSTMs**

\[ P \]

\[ \tilde{p}_i \]

\[ P_s(i) = \text{softmax}_i(qW_s\tilde{p}_i) \quad P_e(i) = \text{softmax}_i(qW_e\tilde{p}_i) \]

\[ \longrightarrow \text{predict start token} \quad \longrightarrow \text{predict end token} \]
Question Encoding

1. Apply another RNN to top of word embeddings of $q_i$ and get $q_j$

2. Combining the resulting units into one single vector

$$q = \Sigma_j b_j q_j,$$

Here $b_j = \frac{\exp(w \cdot q_j)}{\Sigma_{j'} \exp(w \cdot q_{j'})}$, and $w$ is a weight vector to learn.
Document Reader

Three steps:
1. Paragraph encoding
2. Question encoding
3. Prediction
Who did Genghis Khan unite before he began conquering the rest of Eurasia?

\[ P_s(i) = \text{softmax}_i(qW_s\tilde{p}_i) \quad \rightarrow \quad \text{predict start token} \]

\[ P_e(i) = \text{softmax}_i(qW_e\tilde{p}_i) \quad \rightarrow \quad \text{predict end token} \]
Prediction

• Goal: predict the span of tokens that is most likely the correct answer
• Method: train two classifier independently for predicting two ends of span

\[
\max_{i,j} P_{\text{start}}(i) \times P_{\text{end}}(j)
\]

such that \( i \leq j \leq i + 15 \), where \( P_{\text{start}}(i) \) and \( P_{\text{end}}(j) \) is probability of each token being start and end:

\[
P_{\text{start}}(i) \propto \exp(p_i W_s q)
\]
\[
P_{\text{end}}(i) \propto \exp(p_i W_e q)
\]
Agenda

1. Introduction of DrQA
2. Document Retriever
3. Document Reader
4. Data
5. Results
Three types of data:

- Wikipedia: knowledge source for finding answers
- SQuAD: main source to train the Document Reader
- Three more QA datasets (CuratedTREC, WebQuestions, WikiMovies): In addition to SQuAD, are used to test the DrQA
Curated TREC

# 2.3
Q: What are titles of the group’s releases?
Chocolate Starfish and Hot Dog Flavored Water
Significant Others
Three Dollar Bill, Y'all
Nookie
Break Stuff

# 4.4
Q: What movies did James Dean appear in?
East of Eden
Fixed Bayonet
Giant
Rebel Without a Cause

# 6.3
Q: Famous people who have been Rhodes scholars.
Maine Congressman Tom Allen
Australian Labor leader Kim Beazley
Alan Bersin
Newark Councilman, Cory Booker
Pres. candidate Bill Bradley
Wesley Clark
President Clinton
Peter Dawkins, Heisman Trophy Winner
Author and conceptual thinker Edward DiBono
Berkeley Law Prof., Judge William Fletcher
Environmentalist William Cronon
VA Sec. of Education Barbara Harmon
Kris Kristofferson
Alain Leroy Locke, 1st black Rhodes Scholar
Terrence Malick, Producer
Author Willie Morris
Mathew Polley, editor of "I Can't Believe It's Not the NYT"
Kurt Schork, correspondent killed
George Stephanopolis
Strobe Talbott
Louisiana Legislator David Vitter
Supreme Court Justice Byron White
Author John Edgar Wideman
Author Naomi Wolf

https://trec.nist.gov/data/qa.html
WebQuestions

1. Example 1:
   • Utterance: what is the name of justin bieber brother?
   • TargetValue: Jazmyn Bieber, Jaxon Bieber

2. Example 2:
   • Utterance: what character did natalie portman play in star wars?
   • TargetValue: Padm Amidala

https://nlp.stanford.edu/software/sempre/
WikiMovies

Doc: Wikipedia Article for Blade Runner (partially shown)

Blade Runner is a 1982 American neo-noir dystopian science fiction film directed by Ridley Scott and starring Harrison Ford, Rutger Hauer, Sean Young, and Edward James Olmos. The screenplay, written by Hampton Fancher and David Peoples, is a modified film adaptation of the 1968 novel “Do Androids Dream of Electric Sheep?” by Philip K. Dick. The film depicts a dystopian Los Angeles in November 2019 in which genetically engineered replicants, which are visually indistinguishable from adult humans, are manufactured by the powerful Tyrell Corporation as well as by other “mega-corporations” around the world. Their use on Earth is banned and replicants are exclusively used for dangerous, menial, or leisure work on off-world colonies. Replicants who defy the ban and return to Earth are hunted down and “retired” by special police operatives known as “Blade Runners”.

KB entries for Blade Runner (subset)
Blade Runner directed_by Ridley Scott
Blade Runner written_by Philip K. Dick, Hampton Fancher
Blade Runner starred_actors Harrison Ford, Sean Young, …
Blade Runner release_year 1982
Blade Runner has_tags dystopian, noir, police, androids, …

IE entries for Blade Runner (subset)
Blade Runner, Ridley Scott directed dystopian, science fiction, film
Hampton Fancher written Blade Runner
Blade Runner starred Harrison Ford, Rutger Hauer, Sean Young…
Blade Runner labelled 1982 neo noir
special police, Blade retired Blade Runner
Blade Runner, special police known Blade

Questions for Blade Runner (subset)
Ridley Scott directed which films?
What year was the movie Blade Runner released?
Who is the writer of the film Blade Runner?
Which films can be described by dystopian?
Which movies was Philip K. Dick the writer of?
Can you describe movie Blade Runner in a few words?

Table 1: WIKIMOVIES: Questions, Doc, KB and IE sources.

https://research.fb.com/downloads/babi/
Table 2: Number of questions for each dataset used in this paper. DS: distantly supervised training data. *: These training sets are not used as is because no paragraph is associated with each question. †: Corresponds to SQuAD development set.
Distantly Supervised Data

(Q, A) \rightarrow (P, Q, A) if P is retrieved and A can be found in P

Q: What part of the atom did Chadwick discover?  
A: neutron

Atom

From Wikipedia, the free encyclopedia

The atomic mass of these isotopes varied by integer amounts, called the whole number rule.\textsuperscript{[23]} The explanation for these different isotopes awaited the discovery of the neutron, an uncharged particle with a mass similar to the proton, by the physicist James Chadwick in 1932. Isotopes were then explained as elements with the same number of protons, but different numbers of neutrons within the nucleus.
## Example training data

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Example</th>
<th>Article / Paragraph</th>
</tr>
</thead>
</table>
| SQuAD         | **Q:** How many provinces did the Ottoman empire contain in the 17th century?  
A: 32                                          | **Article:** Ottoman Empire  
**Paragraph:** ... At the beginning of the 17th century the empire contained 32 provinces and numerous vassal states. Some of these were later absorbed into the Ottoman Empire, while others were granted various types of autonomy during the course of centuries. |
| CuratedTREC    | **Q:** What U.S. state’s motto is “Live free or Die”?  
A: New Hampshire       | **Article:** Live Free or Die  
**Paragraph:** "Live Free or Die" is the official motto of the U.S. state of New Hampshire, adopted by the state in 1945. 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. |
| WebQuestions   | **Q:** What part of the atom did Chadwick discover?  
A: neutron                  | **Article:** Atom  
**Paragraph:** ... The atomic mass of these isotopes varied by integer amounts, called the whole number rule. The explanation for these different isotopes awaited the discovery of the neutron, an uncharged particle with a mass similar to the proton, by the physicist James Chadwick in 1932. |
| WikiMovies     | **Q:** Who wrote the film Gigli?  
A: Martin Brest            | **Article:** Gigli  
**Paragraph:** Gigli is a 2003 American romantic comedy film written and directed by Martin Brest and starring Ben Affleck, Jennifer Lopez, Justin Bartha, Al Pacino, Christopher Walken, and Lainie Kazan. |
Agenda

1. Introduction of DrQA
2. Document Retriever
3. Document Reader
4. Data
5. Results
Three Parts of Evaluation

1. Document Retriever Evaluation
2. Document Reader Evaluation
3. Full Wikipedia Question Answering
Three Parts of Evaluation

1. Document Retriever Evaluation
2. Document Reader Evaluation
3. Full Wikipedia Question Answering
## Document retrieval results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Wiki Search</th>
<th>Doc. Retriever</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>plain</td>
</tr>
<tr>
<td>SQuAD</td>
<td>62.7</td>
<td>76.1</td>
</tr>
<tr>
<td>CuratedTREC</td>
<td>81.0</td>
<td>85.2</td>
</tr>
<tr>
<td>WebQuestions</td>
<td>73.7</td>
<td>75.5</td>
</tr>
<tr>
<td>WikiMovies</td>
<td>61.7</td>
<td>54.4</td>
</tr>
</tbody>
</table>

### Table 3: Document retrieval results.

% of questions for which the answer segment appears in one of the top 5 pages returned by the method.
Three Parts of Evaluation

1. Document Retriever Evaluation
2. Document Reader Evaluation
3. Full Wikipedia Question Answering
On SQuAD

<table>
<thead>
<tr>
<th>Method</th>
<th>Dev EM</th>
<th>Dev F1</th>
<th>Test EM</th>
<th>Test F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic Coattention Networks (Xiong et al., 2016)</td>
<td>65.4</td>
<td>75.6</td>
<td>66.2</td>
<td>75.9</td>
</tr>
<tr>
<td>Multi-Perspective Matching (Wang et al., 2016)†</td>
<td>66.1</td>
<td>75.8</td>
<td>65.5</td>
<td>75.1</td>
</tr>
<tr>
<td>BiDAF (Seo et al., 2016)</td>
<td>67.7</td>
<td>77.3</td>
<td>68.0</td>
<td>77.3</td>
</tr>
<tr>
<td>R-net†</td>
<td>n/a</td>
<td>n/a</td>
<td>71.3</td>
<td>79.7</td>
</tr>
<tr>
<td>DrQA (Our model, Document Reader Only)</td>
<td>69.5</td>
<td>78.8</td>
<td>70.0</td>
<td>79.0</td>
</tr>
</tbody>
</table>

Table 4: Evaluation results on the SQuAD dataset (single model only). †: Test results reflect the SQuAD leaderboard (https://stanford-qa.com) as of Feb 6, 2017.

Surpass all the published results and can match the top performance on the SQuAD leaderboard at the time of writing.
Three Parts of Evaluation

1. Document Retriever Evaluation
2. Document Reader Evaluation
3. Full Wikipedia Question Answering
Full Wikipedia Question Answer

Three versions of DrQA:

• **SQuAD**: A single Document Reader model is trained on the SQuAD training set only and used on all evaluation sets.

• **Fine-tune (DS)**: A Document Reader model is pre-trained on SQuAD and then fine-tuned for each dataset independently using its distant supervision (DS) training set.

• **Multitask (DS)**: A single Document Reader model is jointly trained on the SQuAD training set and all the DS sources.
### Full Wikipedia Results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>YodaQA</th>
<th>SQuAD</th>
<th>+Fine-tune (DS)</th>
<th>+Multitask (DS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQuAD (All Wikipedia)</td>
<td>n/a</td>
<td>27.1</td>
<td>28.4</td>
<td>29.8</td>
</tr>
<tr>
<td>CuratedTREC</td>
<td>31.3</td>
<td>19.7</td>
<td>25.7</td>
<td>25.4</td>
</tr>
<tr>
<td>WebQuestions</td>
<td>39.8</td>
<td>11.8</td>
<td>19.5</td>
<td>20.7</td>
</tr>
<tr>
<td>WikiMovies</td>
<td>n/a</td>
<td>24.5</td>
<td>34.3</td>
<td>36.5</td>
</tr>
</tbody>
</table>

Table 6: Full Wikipedia results. Top-1 exact-match accuracy (in %, using SQuAD eval script). +Fine-tune (DS): Document Reader models trained on SQuAD and fine-tuned on each DS training set independently. +Multitask (DS): Document Reader single model trained on SQuAD and all the distant supervision (DS) training sets jointly. YodaQA results are extracted from [https://github.com/brimson/yodaqa/wiki/Benchmarks](https://github.com/brimson/yodaqa/wiki/Benchmark) and use additional resources such as Freebase and DBpedia, see Section 2.

Multitask performs the best, reasonable performance across four datasets.

Seems to be better in these two tasks, anything wrong?
What is YodaQA?

YodaQA is an open source system modeled after IBM's DeepQA (Watson) system, which is a hybrid system which answers questions based on different types of data, including unstructured text, websites, databases etc.
Nothing wrong!

• It is not a direct comparison between YodaQA and DrQA as YodaQA relies on additional resources such as Freebase, while DrQA is more challenging by using single source

• WebQuestions is a dataset which is designed to answer questions over Freebase
Main Take-Aways

• DrQA was the first attempt to scale up reading comprehension to open-domain question answering, by combining IR techniques and neural reading comprehension models.

• Although we achieved good accuracy on SQuAD in 2017 (EM = 70.. vs state-of-the-art EM = 90 in 2020), the final QA accuracy still remains low: 20.7 - 36.5.

• Distant supervision + multi-task learning helps!
Latent Retrieval for Weakly Supervised Open Domain Question Answering

Kenton Lee
Ming-Wei Chang
Kristina Toutanova
Google Research
Seattle, WA

{kentonl,mingweichang,kristout}@google.com
ORQA

1. Introduction & Motivation

2. Model

3. Evaluation

4. Analysis
ORQA

1. Introduction & Motivation

2. Model

3. Evaluation

4. Analysis
Limitations of Current Models

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

DrQA Model

What if we learned the information retrieval section?
Problem Space

Typically in open-domain question answering

Input: question string $q$
Output: answer string $a$

Where do you get the evidence to go from input to output?
- In reading comprehension, it’s given to you
- Here, it can come from anywhere

*It's a modeling choice here!*
Discussion Question

Discuss the differences between unsupervised QA, strongly supervised QA, weakly supervised QA settings in open-domain question answering.
Discussion Question

Discuss the differences between unsupervised QA, strongly supervised QA, weakly supervised QA settings in open-domain question answering.

**Unsupervised**: No training data or question-answer pairs  
**Strongly supervised**: Assumes reading comprehension dataset with gold evidence and question-answer pairs  
**Weakly supervised**: Only have access to question-answer pairs without any gold evidence

<table>
<thead>
<tr>
<th>Task</th>
<th>Training</th>
<th>Evaluation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Evidence</td>
<td>Answer</td>
<td>Evidence</td>
</tr>
<tr>
<td>Reading Comprehension</td>
<td>given</td>
<td>span</td>
<td>given</td>
</tr>
<tr>
<td>Open-domain QA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unsupervised QA</td>
<td>none</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>Strongly Supervised QA</td>
<td>given</td>
<td>span</td>
<td>heuristic</td>
</tr>
<tr>
<td>Weakly Supervised QA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Closed Retrieval QA</td>
<td>heuristic</td>
<td>string</td>
<td>heuristic</td>
</tr>
<tr>
<td>Open Retrieval QA</td>
<td>learned</td>
<td>string</td>
<td>learned</td>
</tr>
</tbody>
</table>
1. Introduction & Motivation

2. Model

3. Evaluation

4. Analysis
The Model!

This is from 2019, it’s time for BERT models!
The Model!

Information Retrieval (IR)

$S_{\text{retr}}(0, q)$ → $\text{BERT}_B(0)$

[CLS]...The term ‘ZIP’ is an acronym for Zone Improvement Plan... [SEP]

$S_{\text{retr}}(1, q)$ → $\text{BERT}_B(1)$

[CLS]...group of zebras are referred to as a herd or dazzle... [SEP]

$S_{\text{retr}}(2, q)$ → $\text{BERT}_B(2)$

[CLS]...ZIPs for other operating systems may be preceded by... [SEP]

$S_{\text{retr}}(..., q)$ → $\text{BERT}_B(...)$

Reader (QA)

$\text{BERT}_R(q, 0)$

Top K

$S_{\text{read}}(0, “The term”, q)$

[CLS] What does the zip in zip code stand for? [SEP]...The term ‘ZIP’ is an acronym for Zone Improvement Plan... [SEP]

$S_{\text{read}}(0, “Zone Improvement Plan”, q)$

$S_{\text{read}}(0, ..., q)$

$\text{BERT}_R(q, 2)$

Top K

$S_{\text{read}}(2, “ZIPs”, q)$

[CLS] What does the zip in zip code stand for? [SEP]...ZIPs for other operating systems may be preceded by... [SEP]

$S_{\text{read}}(2, “operating systems”, q)$

$S_{\text{read}}(2, ..., q)$
The Model!

Information Retrieval (IR)

$S_{\text{retire}}(0, q) \rightarrow \text{BERT}_R(0)$

[CLS] The term ‘ZIP’ is an acronym for Zone Improvement Plan... [SEP]

$S_{\text{retire}}(1, q) \rightarrow \text{BERT}_R(1)$

[CLS] ...group of zebras are referred to as a herd or dazzle... [SEP]

$S_{\text{retire}}(2, q) \rightarrow \text{BERT}_R(2)$

[CLS] ...ZIPs for other operating systems may be preceded by... [SEP]

$S_{\text{retire}}(..., q) \rightarrow \text{BERT}_R(...)$

Reader (QA)

$\text{S}_{\text{read}}(0, \text{“The term”, } q)$

[CLS] What does the zip in zip code stand for? [SEP] The term ‘ZIP’ is an acronym for Zone Improvement Plan... [SEP]

$\text{S}_{\text{read}}(0, \text{“Zone Improvement Plan”, } q)$

$\text{S}_{\text{read}}(0, \ldots, q)$

$\text{S}_{\text{read}}(2, \text{“ZIPs”, } q)$

[CLS] What does the zip in zip code stand for? [SEP] ZIPs for other operating systems may be preceded by... [SEP]

$\text{S}_{\text{read}}(2, \text{“operating systems”, } q)$

$\text{S}_{\text{read}}(2, \ldots, q)$
Question $q$

What does the zip in zip code stand for?

Information Retrieval (IR)
What does the zip in zip code stand for?

In the United States, zip codes are used to classify mail and are composed of five digits. The first four digits represent the postal zone or sectional center facility, which is a grouping of mail processing facilities. The fifth digit is the delivery point bar code, which represents a specific delivery point, such as a home, business, or post office box. Together, these digits allow for efficient mail sorting and delivery.
Information Retrieval (IR)

2. Run the question and each evidence block through BERT encoders

All of Wikipedia

Question $q$
What does the zip in zip code stand for?

Evidence Block 1

Evidence Block 2

Evidence Block 3

Evidence Block 4

Evidence Block 5

Evidence Block 6

Bert $Q$

Bert $B$
What does the zip in zip code stand for?

Each evidence block $b$

By type and use:

• Unique assigned to a site.
• Post Office Box only.
• Standard at other ZIP Codes.
• ZIP Codes are used for governmental agencies, diplomatic offices, and linking to the national address designating high volume of mail that they need their own ZIP Codes. Government examples include SSAS at the Central Intelligence Agency in Washington, D.C., BON for the United States Census Bureau (USBC) in Washington, D.C., and the Department of Economic Adjustment (DEA) implemented postal for the United States Postal Service. (USPS) implemented postal address for the United States Postal Service.

3. Pass outputs through a linear layer
Information Retrieval (IR)

Question $q$
What does the zip in zip code stand for?

Each evidence block $b$

4. Score each evidence block as the inner product between $h_q$ and $h_b$

All of Wikipedia

Evidence Block 1

Evidence Block 2

Evidence Block 3

Evidence Block 4

Evidence Block 5

Evidence Block 6
Information Retrieval (IR)

Question $q$

What does the zip in zip code stand for?

Each evidence block $b$

$S_{retr}(b, q) = h_q^T h_b$

5. Pick and output the top $k$ scoring evidence blocks.
The Model!

Information Retrieval (IR)

[BERT_Q(q)] What does the zip in zip code stand for? [SEP]

[S_retr(0, q)]

[CLS] The term ‘ZIP’ is an acronym for Zone Improvement Plan... [SEP]

[BERT_B(0)]

[S_retr(1, q)]

[CLS] group of zebras are referred to as a herd or dazzle... [SEP]

[BERT_B(1)]

[S_retr(2, q)]

[CLS] ZIPs for other operating systems may be preceded by... [SEP]

[BERT_B(2)]

[S_retr(..., q)]

[BERT_B(...)]

Reader (QA)

[S_read(0, “The term”, q)]

[CLS] What does the zip in zip code stand for? [SEP] The term ‘ZIP’ is an acronym for Zone Improvement Plan... [SEP]

[S_read(0, ..., q)]

[S_read(2, “ZIPs”, q)]

[CLS] What does the zip in zip code stand for? [SEP] ZIPs for other operating systems may be preceded by... [SEP]

[S_read(2, “operating systems”, q)]

[S_read(2, ..., q)]
Reader (QA)

Top-$k$ Evidence Blocks
Concat. with Question

1. Pass each (question, block) pair from the retriever through the $BERT_R$ model to find the start token hidden state

Very similar to the QA model in the BERT paper!
Reader (QA)

Top-\(k\) Evidence Blocks
Concat. with Question

2. Pass each (question, block) pair from the retriever through the \(BERT_R\) model to find the end token hidden state.
Reader (QA)

Top-\(k\) Evidence Blocks
Concat. with Question

1. Question \(q\)
2. Top Evidence Block 1
3. Top Evidence Block 2

...  

3. Concatenate the \(h_{\text{start}}\) and \(h_{\text{end}}\) hidden states into a single vector

\[ [h_{\text{start}}; h_{\text{end}}] \]
4. Calculate the $S_{\text{read}}$ score for a given span $s$ in block $b$ for question $q$.

Output the best scoring span

$S_{\text{read}}(b, s, q)$
Inference & Learning Challenges

1.) The search space is huge! There’s around 13 million evidence blocks to choose from. (English Wikipedia ~5 million articles)

2.) How to pick relevant evidence blocks is latent (not explicitly learned) since we’re learning on the right answer.

<table>
<thead>
<tr>
<th>Example</th>
<th>Supportive Evidence</th>
<th>Spurious Ambiguity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q: Who is credited with developing the XY coordinate plane? A: René Descartes</td>
<td>...invention of Cartesian coordinates by René Descartes revolutionized...</td>
<td>...René Descartes was born in La Haye en Touraine, France...</td>
</tr>
<tr>
<td>Q: How many districts are in the state of Alabama? A: seven</td>
<td>...Alabama is currently divided into seven congressional districts, each represented by ...</td>
<td>...Alabama is one of seven states that levy a tax on food at the same rate as other goods...</td>
</tr>
</tbody>
</table>

“Spuriously ambiguous derivations”
Inference & Learning Challenges

1.) The search space is **huge**! There’s around 13 million evidence blocks to choose from. (English Wikipedia ~5 million articles)

2.) How to pick relevant evidence blocks is *latent* (not explicitly learned) since we’re learning on the right answer.

<table>
<thead>
<tr>
<th>Example</th>
<th>Supportive Evidence</th>
<th>Spurious Ambiguity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q: Who is credited with developing the XY coordinate plane? A: René Descartes</td>
<td>...invention of Cartesian coordinates by <strong>René Descartes</strong> revolutionized...</td>
<td>...<strong>René Descartes</strong> was born in La Haye en Touraine, France...</td>
</tr>
<tr>
<td>Q: How many districts are in the state of Alabama? A: seven</td>
<td>...Alabama is currently divided into <strong>seven</strong> congressional districts, each represented by ...</td>
<td>...Alabama is one of <strong>seven</strong> states that levy a tax on food at the same rate as other goods...</td>
</tr>
</tbody>
</table>

*Solved using Inverse Cloze Task (ICT) Pre-training!*

“Spuriously ambiguous derivations”
**Cloze Task**

Predict the masked-out *sentence based on its context*

**Given**

[CLS]
...Zebras have four gaits: walk, trot, canter, and gallop.

______________________________

______________________________

_____ When chased, a zebra will zig-zag from side to side...

[SEP]

**Choices**

[CLS] They are generally slower than horses, but their great stamina helps them outrun predators [SEP]

[CLS] Gagarin was further selected for an elite training group known as the Sochi Six [SEP]

...
Inverse Cloze Task

Predict the masked-out context based on its sentence

**Given**

[CLS]
______________________________
___________
They are generally slower than horses, but their great stamina helps them outrun predators.
______________________________
_____________________
[SEP]

**Choices**

[CLS] …Zebras have four gaits: walk, trot, canter and gallop. When chased, a zebra will zig-zag from size to side…[SEP]

[CLS]…Gagarin was further selected for an elite training group known as the Sochi Six…[SEP]

…

Question: “pseudo-query”

Answer: “psuedo evidence text”
How often do we mask the pseudo-query?

What if we masked the pseudo-query 100% of the time?
- Trouble learning basic **word overlap** between evidence and query

What if we masked the pseudo-query 0% of the time?
- Task becomes **trivial** and doesn't learn much! Find the evidence with the query in it

**In response**, ORQA removes the pseudo-question sentence 90% of the time.
- 90% of the time, focus on abstract representations
- 10% of the time, focus on word matching

“They are generally slower than horses, but their great stamina helps them outrun predators.”
Retrieval Pre-Training & Inference

• Pre-trains the IR sub-model on the Inverse Cloze Task (ICT) with sentences
• Masks the sentences 90% of the time
• Freeze the \( \text{BERT}_B(b) \) model afterwards
• Pre-compute the hidden representations for all of the evidence blocks \((h_b)\) into a giant index
• Beam-search on \( k \) top blocks
• Solves large search space problem!
Retrieval Pre-Training & Inference

- Pre-trains the IR sub-model on the Inverse Cloze Task (ICT) with sentences
- Masks the sentences 90% of the time
- Freeze the $BERT_B(b)$ model afterwards
- Pre-compute the hidden representations for all of the evidence blocks ($h_b$) into a giant index
- Beam-search on $k$ top blocks

This is really difficult too!
- Uses Locality Sensitive Hashing (LSH) to quickly find maximum inner products!
- Really important to make finding the best evidence blocks efficient!
Fine-Tuning & Learning

Probability distribution of any span $s$ in any top-$k$ block $b$ given question $q$:

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

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

Softmax over every span in the top-k blocks

Given a gold answer $a$, we find all spans that exactly match $a$ and optimize their marginal log-likelihood:

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

$k \sim 5$

Exact match!
**Fine-Tuning & Learning**

**Early Learning** consider a larger set of $c$ evidence blocks and update the retrieval score:

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

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

Provides additional training to the $BERT_Q(q)$ encoder!

$c \sim 5,000$
Fine-Tuning & Learning

**Final Loss** is the combination of both $L_{\text{full}}$ and $L_{\text{early}}$

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

If the answer isn’t found in the top-$k$ blocks, then discard the example.

Because ICT pre-training is such an effective strategy, only $< 10\%$ of examples are discarded!
Model Overview

**Trained Information Retrieval** picks the top-\(k\) scoring evidence blocks from all Wikipedia documents by taking the inner product between the question and block encodings.

**Trained Reader** uses beam-search to find the answer *span* within the top-\(k\) evidence blocks.

**Inverse Cloze Task Pre-training** initializes the block encoder weights to a sufficient starting point.

**Pre-computed Block Encoding Index** computes all the encodings for each evidence block after ICT pre-training.

**Fine-Tuning** on each task helps train the reader and the question encoder.

**Early Updates** help train the question encoder by calculating loss on the top-\(c\) evidence blocks.
ORQA

1. Introduction & Motivation
2. Model
3. Evaluation
4. Analysis
Datasets

Natural Questions (Kwiatkowski et al., 2019) from Google Search API, discard evidence and long answers

WebQuestions (Berant et al., 2013) from Google Suggest, only keep string representations

CuratedTrec (Baudis and Sedivy, 2015) QA data from TREC

TriviaQA (Joshi et al., 2017) is a trivia QA collection from the web, discard evidence

SQuAD (Rajpurkar et al., 2016) is a well-known QA dataset, discard given evidence

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Train</th>
<th>Dev</th>
<th>Test</th>
<th>Example Question</th>
<th>Example Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Natural Questions</td>
<td>79168</td>
<td>8757</td>
<td>3610</td>
<td>What does the zip in zip code stand for?</td>
<td>Zone Improvement Plan</td>
</tr>
<tr>
<td>WebQuestions</td>
<td>3417</td>
<td>361</td>
<td>2032</td>
<td>What airport is closer to downtown Houston?</td>
<td>William P. Hobby Airport</td>
</tr>
<tr>
<td>CuratedTrec</td>
<td>1353</td>
<td>133</td>
<td>694</td>
<td>What metal has the highest melting point?</td>
<td>Tungsten</td>
</tr>
<tr>
<td>TriviaQA</td>
<td>78785</td>
<td>8837</td>
<td>11313</td>
<td>What did L. Fran Baum, author of The Wonderful Wizard of Oz, call his home in Hollywood?</td>
<td>Ozcot</td>
</tr>
<tr>
<td>SQuAD</td>
<td>78713</td>
<td>8886</td>
<td>10570</td>
<td>Other than the Automobile Club of Southern California, what other AAA Auto Club chose to simplify the divide?</td>
<td>California State Automobile Association</td>
</tr>
</tbody>
</table>
## Datasets Biases

Natural Questions, WebQuestions, and CuratedTrec all have tool-assisted answers - bias towards the tool TriviaQA and SQuAD question writers are aware of the answers.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Question writer knows answer</th>
<th>Question writer knows evidence</th>
<th>Tool-assisted answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Natural Questions</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>WebQuestions</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CuratedTrec</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>TriviaQA</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>SQuAD</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Datasets Biases

SQuAD Paragraph
The largest living species is the emperor penguin (*Aptenodytes forsteri*): on average, adults are about 1.1 m (3 ft 7 in) tall and weigh 35 kg (77 lb). The smallest penguin species is the little blue penguin (*Eudyptula minor*), also known as the fairy penguin, which stands around 40 cm (16 in) tall and weighs 1 kg (2.2 lb). Among extant penguins, larger penguins inhabit colder regions, while smaller penguins are generally found in temperate or even tropical climates.

Question + Answer?
The largest living species is the emperor penguin (*Aptenodytes forsteri*): on average, adults are about 1.1 m (3 ft 7 in) tall and weigh 35 kg (77 lb). The **smallest penguin species is the** little blue penguin (*Eudyptula minor*), also known as the fairy penguin, which stands around 40 cm (16 in) tall and weighs 1 kg (2.2 lb). Among extant penguins, larger penguins inhabit colder regions, while smaller penguins are generally found in temperate or even tropical climates.

**Question + Answer**

Q: What **is the smallest penguin species**?

A: **Highlighted**
Baseline Models

**BM25+BERT** is like the 2019 version of DrQA
- BM25 is an updated version of TF-IDF
- BERT is an updated version of DocumentReader

**NNLM** is a context-*independent* embedding from feed-forward language models

**ELMo** is a context-*dependent* bidirectional LSTM language model

All models use the **same BERT-based reader as ORQA**

**NNLM and ELMo** both use the same scoring heuristic as ORQA for retrieval
## Results

<table>
<thead>
<tr>
<th>Model</th>
<th>BM25 +BERT</th>
<th>NNLG +BERT</th>
<th>ELMo +BERT</th>
<th>ORQA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Natural Questions Dev</td>
<td>24.8</td>
<td>3.2</td>
<td>3.6</td>
<td>31.3</td>
</tr>
<tr>
<td>WebQuestions</td>
<td>20.8</td>
<td>9.1</td>
<td>17.7</td>
<td>38.5</td>
</tr>
<tr>
<td>CuratedTrec</td>
<td>27.1</td>
<td>6.0</td>
<td>8.3</td>
<td>36.8</td>
</tr>
<tr>
<td>TriviaQA</td>
<td><strong>47.2</strong></td>
<td>7.3</td>
<td>6.0</td>
<td>45.1</td>
</tr>
<tr>
<td>SQuAD</td>
<td><strong>28.1</strong></td>
<td>2.8</td>
<td>1.9</td>
<td>26.5</td>
</tr>
<tr>
<td>Test Dev</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Natural Questions</td>
<td>26.5</td>
<td>4.0</td>
<td>4.7</td>
<td>33.3</td>
</tr>
<tr>
<td>WebQuestions</td>
<td>17.7</td>
<td>7.3</td>
<td>15.6</td>
<td>36.4</td>
</tr>
<tr>
<td>CuratedTrec</td>
<td>21.3</td>
<td>4.5</td>
<td>6.8</td>
<td>30.1</td>
</tr>
<tr>
<td>TriviaQA</td>
<td><strong>47.1</strong></td>
<td>7.1</td>
<td>5.7</td>
<td>45.0</td>
</tr>
<tr>
<td>SQuAD</td>
<td><strong>33.2</strong></td>
<td>3.2</td>
<td>2.3</td>
<td>20.2</td>
</tr>
</tbody>
</table>

Performing well on datasets with low question-evidence overlap!

Performed at-par with BM25 on SQuAD and TriviaQA

Generally poor baselines
Results Takeaways

- **Successful End-to-End Training**… when there isn’t “bias” in the dataset
- Previous neural retrieval methods (NNLM, ELMo-based) were very bad, but ORQA does a lot better
- ICT pre-trained retriever outperforms BM25 by 6 - 19 points depending on the dataset
- 128-dimensional vector may be too small to represent every word in the evidence
- SQuAD’s 100k questions are derived from only 536 documents! Good retrievals are highly correlated between examples
Strongly Supervised Model Comparison

DrQA is the state of the art unsupervised open-domain question-answering method.

BERT_Serini is another BERT-based model using BM25 that splits on paragraphs instead of blocks (i.e. more evidence blocks).

BM25+BERT is the best performing model from the results.

Evaluate on just the SQuAD testing dataset.
### Results

Excellent on an evidence retrieved efficiency level!

<table>
<thead>
<tr>
<th>Model</th>
<th>Evidence Retrieved</th>
<th>SQuAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>DrQA</td>
<td>5 documents</td>
<td>27.1</td>
</tr>
<tr>
<td>DrQA (DS)</td>
<td>5 documents</td>
<td>28.4</td>
</tr>
<tr>
<td>DrQA (DS + MTL)</td>
<td>5 documents</td>
<td>29.8</td>
</tr>
<tr>
<td>BERTSERINI</td>
<td>5 documents</td>
<td>19.1</td>
</tr>
<tr>
<td>BERTSERINI</td>
<td>29 paragraphs</td>
<td>36.6</td>
</tr>
<tr>
<td>BERTSERINI</td>
<td>100 paragraphs</td>
<td>38.6</td>
</tr>
<tr>
<td>BM25 + BERT (gold deriv.)</td>
<td>5 blocks</td>
<td>34.7</td>
</tr>
</tbody>
</table>

**Best performing combo is comparable to SOTA**

BERT + BM25 + paragraphs instead of blocks
What if we vary the ICT masking rate?

0% masking rate means only looking for word similarity!

100% masking rate means not looking for word similarity and only using semantic meaning!
### Error Comparisons

**ORQA > BM25+BERT** for separating semantically distinct text with high lexical overlap

<table>
<thead>
<tr>
<th>Example</th>
<th>ORQA</th>
<th>BM25 + BERT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q: what is the new orleans saints symbol called</td>
<td>...The team’s primary colors are old gold and black; their logo is a simplified <strong>fleur-de-lis</strong>. They played their home games in Tulane Stadium through the 1974 NFL season.</td>
<td>...the <strong>SkyDome</strong> was owned by Sportsco at the time... the sale of the New Orleans Saints with team owner Tom Benson... the Saints became a symbol for that community...</td>
</tr>
<tr>
<td>A: fleur-de-lis</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q: how many senators per state in the us</td>
<td>...powers of the Senate are established in Article One of the U.S. Constitution. Each U.S. state is represented by <strong>two</strong> senators...</td>
<td>...The Georgia Constitution mandates a maximum of <strong>56</strong> senators, elected from single-member districts...</td>
</tr>
<tr>
<td>A: two</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q: when was germany given a permanent seat on the council of the league of nations</td>
<td>...Under the Weimar Republic, Germany (in fact the “Deutsches Reich” or German Empire) was admitted to the League of Nations through a resolution passed on September 8 <strong>1926</strong>. An additional 15 countries joined later...</td>
<td>...the accession of the German Democratic Republic to the Federal Republic of Germany, it was effective on <strong>3 October 1990</strong>...Germany has been elected as a non-permanent member of the United Nations Security Council...</td>
</tr>
<tr>
<td>A: 1926</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q: when was diary of a wimpy kid double down published</td>
<td>...“Diary of a Wimpy Kid” first appeared on FunBrain in 2004, where it was read 20 million times. The abridged hardcover adaptation was released on <strong>April 1, 2007</strong>...</td>
<td>Diary of a Wimpy Kid: Double Down is the eleventh book in the ”Diary of a Wimpy Kid” series by Jeff Kinney... The book was published on <strong>November 1, 2016</strong>...</td>
</tr>
<tr>
<td>A: November 1, 2016</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**BM25+BERT > ORQA** for very specific representations better represented by sparse vectors
Conclusion

Significant Contributions
1.) First retriever-reader trained jointly end-to-end using only question-answer pairs
2.) Made possible because of the novel pre-training task: Inverse Cloze Task
3.) Learned retrieval proved to be successful when the question writers don’t know the answer (“true” information seeking)

Potential Additions
1.) Only uses 128 dimension vectors, what happens when we increase this?
2.) Can we quantify the bias in TriviaQA and SQuAD?
Discussion Question

(Lee et al, 2019) made a distinction between different types of QA datasets and demonstrated that in some cases, a traditional unsupervised retrieval method (e.g., BM25, TF-IDF) works better while in some other cases, it is more effective to "learn" a retriever.

Can you state the argument and do you agree with it?
Discussion Question

(Lee et al, 2019) made a distinction between different types of QA datasets and demonstrated that in some cases, a traditional unsupervised retrieval method (e.g., BM25, TF-IDF) works better while in some other cases, it is more effective to "learn" a retriever.

Can you state the argument and do you agree with it?

Learned retrievers are better for “true” information-seeking tasks and succeed when question writers don’t know the answer ahead of time.

Do you agree?
Appendix

Hyperparameters and Specifics
What if we vary the ICT masking rate?

In all uses of BERT, they used an uncased base model
- 12 transformer layers
- 768 hidden size
- Default optimizer

\( h_q \) and \( h_b \) have 128 dimensions - small because they wanted it to run on a single machine

ICT Pre-training
- Learning rate of \( 10^{-4} \)
- Batch size 4096

Fine-tuning
- Learning rate of \( 10^{-5} \)
- Batch size of 1 on a single machine with 12GB GPU

Answer spans limited to 10 tokens

2 epochs of fine-tuning on large datasets with 20 for smaller ones