Natural Language Processing with Deep Learning
CS224N/Ling284

Diyi Yang
Lecture 12: Question Answering
(slides based on John Hewitt, Danqi Chen, Nelson Liu)
Overview

1. What is question answering? (10 mins)

2. Reading comprehension (30 mins)
   - How to answer questions over a single passage of text

3. Open-domain (textual) question answering (20 mins)
   - How to answer questions over a large collection of documents

4. Retrieval-augmented generation for question answering (10 mins)

• Next lecture: Nicholas Carlini on Privacy and Security in LLMs
• Assignment 5 due on Feb 18th
1. What is question answering?

The goal of question answering is to build systems that **automatically** answer questions posed by humans in a **natural language**.

The earliest QA systems dated back to 1960s! (Simmons et al., 1964)
1. What is question answering?

- Lots of immediate applications: search engines, dialogue systems
- QA is an important testbed for evaluating how well compute systems understand human language
- “Since question can be devised to query any aspect of text comprehension, the ability to answer questions is the strongest possible demonstration of understanding”
1. What is question answering?

- People ask a lot of questions to Digital Personal Assistants
Question answering taxonomy

• Factoid questions vs. Non-factoid questions

• **Answers**
  • A short span of text
  • A paragraph
  • Yes/No
  • A database entry
  • A list

• **Context**
  • A passage, a document, a large collection of documents
  • Knowledge base
  • Semi-structured tables
  • Images
  • The web
2011: IBM Watson beat Jeopardy champions

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

(1) Question processing, (2) Candidate answer generation, (3) Candidate answer scoring, and (4) Confidence merging and ranking.
Almost all the state-of-the-art question answering systems are built on top of end-to-end training and pre-trained language models (e.g., BERT)!
How did some suspect that Polo learned about China instead of by actually visiting it?

**Answer:** through contact with Persian traders
Textual Question Answering

James the Turtle was always getting in trouble. Sometimes he’d reach into the freezer and empty out all the food. Other times he’d sled on the deck and get a splinter. His aunt Jane tried as hard as she could to keep him out of trouble, but he was sneaky and got into lots of trouble behind her back.

One day, James thought he would go into town and see what kind of trouble he could get into. He went to the grocery store and pulled all the pudding off the shelves and ate two jars. Then he walked to the fast food restaurant and ordered 15 bags of fries. He didn’t pay, and instead headed home.

His aunt was waiting for him in his room. She told James that she loved him, but he would have to start acting like a well-behaved turtle.

After about a month, and after getting into lots of trouble, James finally made up his mind to be a better turtle.

1) What is the name of the trouble making turtle?
   A) Fries
   B) Pudding
   C) James
   D) Jane

2) What did James pull off of the shelves in the grocery store?
   A) pudding
   B) fries
   C) food
   D) splinters

(Richardson et al, 2013): MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text
The Virginia governor’s race, billed as the marquee battle of an otherwise anticlimactic 2013 election cycle, is shaping up to be a foregone conclusion. Democrat Terry McAuliffe, the longtime political fixer and moneyman, hasn’t trailed in a poll since May. Barring a political miracle, Republican Ken Cuccinelli will be delivering a concession speech on Tuesday evening in Richmond. In recent ...

Q: What are the candidates running for?  
A: Governor

Q: Where?  
A: Virginia

Q: Who is the democratic candidate?  
A: Terry McAuliffe

Q: Who is his opponent?  
A: Ken Cuccinelli

Q: What party does he belong to?  
A: Republican

Q: Which of them is winning?  

(Retty et al, 2019): CoQA: A Conversational Question Answering Challenge
Long-form Question Answering

How do jellyfish function without a brain and a nervous system?

Abstractive

Jellyfish don’t have brains. Jellyfish have a nervous system. They know how to swim, they don’t even know what to do with their bodies. They have a “nervous system”, which is basically a bunch of nerve cells that connect to the body. The nervous system is made up of neurons, neurons, nerves, etc. Each neuron doesn’t have a specific set of “neurons”, it just has a different set of nerves. It has a set of sensory neurons, which are all connected to each other. It just knows that it does react to the stimulus, and it knows that the stimulus is there. They have a bunch of nervous systems that act like a filter to get information back.

Extractive

But none of these hypotheses make much sense for a jellyfish because they don’t have brains at all. They just have a nerve net—a loose ring of neurons that runs around the rim of their pulsating bells. They have an unusual nervous system, 451 because jellyfish are not bilaterally symmetric. One side and a right side nervous system, and they coordinate...

https://ai.facebook.com/blog/longform-qa/
(Fan et al, 2019): ELI5: Long Form Question Answering

13
Open-domain Question Answering

(Chen et al, 2017): Reading Wikipedia to Answer Open-Domain Questions
Knowledge Base Question Answering

(Berant et al, 2013): Semantic Parsing on Freebase from Question-Answer Pairs
Table-based Question Answering


(x = Greece held its last Summer Olympics in which year?)

y = 2004
Visual Question Answering

How many slices of pizza are there? Is this a vegetarian pizza?

ChatGPT
There are 8 slices of pizza in the image. As for whether it’s a vegetarian pizza, it appears to have toppings that could be vegetables, such as pieces that look like diced tomatoes and possibly green peppers. However, without a clearer image or a list of ingredients, it’s impossible to be certain if there are no meat products on the pizza. Vegetarian pizzas typically do not include meat, but they can include a variety of vegetable toppings like onions, peppers, mushrooms, olives, and more. If there is any meat or meat-based toppings visible, then it would not be a vegetarian pizza.
Why do we care about this problem?

• Reading comprehension is an important testbed for evaluating how well computer systems understand human language
  • Wendy Lehnert 1977: “Since questions can be devised to query any aspect of text comprehension, the ability to answer questions is the strongest possible demonstration of understanding.”
• Many other NLP tasks can be reduced to a reading comprehension problem:

**Information extraction**
(Barack Obama, educated_at, ?)

Question: Where did Barack Obama graduate from?
Passage: Obama was born in Honolulu, Hawaii. After graduating from Columbia University in 1983, he worked as a community organizer in Chicago.

(Levy et al., 2017)

**Semantic role labeling**

```
UCD finished the 2006 championship as Dublin champions, by beating St Vincents in the final.
```

Who finished something? - UCD
What did someone finish? - the 2006 championship
What did someone finish something as? - Dublin champions
How did someone finish something? - by beating St Vincents in the final

Who beat someone? - UCD
When did someone beat someone? - in the final
Who did someone beat? - St Vincents

(He et al., 2015)
Stanford Question Answering Dataset (SQuAD)

- 100k annotated (passage, question, answer) triples
  - Large-scale supervised datasets are also a key ingredient for training effective neural models for reading comprehension!

- Passages are selected from English Wikipedia, usually 100~150 words.
  - Questions are crowd-sourced.
  - Each answer is a short segment of text (or span) in the passage.
  - This is a limitation— not all the questions can be answered in this way!
Stanford Question Answering Dataset (SQuAD)

• SQuAD was for years the most popular reading comprehension dataset; it is “almost solved” today (though the underlying task is not,) and the state-of-the-art exceeds the estimated human performance.

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

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

- **Evaluation**: exact match (0 or 1) and F1 (partial credit).
- For development and testing sets, 3 gold answers are collected, because there could be multiple plausible answers.
- We compare the predicted answer to each gold answer *(a, an, the, punctuations are removed)* and take max scores. Finally, we take the average of all the examples for both exact match and F1.

Q: Rather than taxation, what are private schools largely funded by?

A: {tuition, charging their students tuition, tuition}
Other question answering datasets

- **TriviaQA**: Questions and answers by trivia enthusiasts. Independently collected web paragraphs that contain the answer and seem to discuss the question, but no human verification that the paragraph supports the answer to the question.

- **Natural Questions**: Question drawn from frequently asked Google search questions. Answers from Wikipedia paragraphs. Answer can be a substring, yes, no, or **NOT_PRESENT**. Verified by human annotation.

- **HotpotQA**. Constructed questions to be answered from the whole of Wikipedia which involve getting information from two pages to answer a multistep query:
  - Q: Which novel by the author of “Armada” will be adapted as a feature film by Steven Spielberg?  
    A: *Ready Player One*
Conventional feature-based methods for reading comprehension

- Generate a list of candidate answers \((a_1, a_2, \ldots, a_M)\)

- Define a feature vector \(\phi(p, q, a_i) \in R^d:\)
  - Word/bigram features
  - Parse tree matches
  - Dependency labels, length, part-of-speech tags

- Apply a multi-class logistic regression model

(Rajpurkar et al, 2016): SQuAD: 100,000+ Questions for Machine Comprehension of Text
Neural models for reading comprehension

• How can we build a neural model to solve SQuAD?

• Problem formulation
  • Input: $C = (c_1, c_2, \ldots, c_N), Q = (q_1, q_2, \ldots, q_M), c_i, q_i \in V$
  • Output: $1 \leq \text{start} \leq \text{end} \leq N$
  • $N \sim 100, M \sim 15$

• Stanford Attentive Reader [Chen, Bolton, & Manning 2016] [Chen, Fisch, Weston & Bordes 2017]
  • Demonstrated a minimal, highly successful architecture for reading comprehension and question answering
  • Became known as the Stanford Attentive Reader
Stanford Attentive Reader

Input

Passage (P)

Question (Q)

Output

Answer (A)

Which team won Super Bowl 50?
Who did Genghis Khan unite before he began conquering the rest of Eurasia?
Stanford Attentive Reader

Who did Genghis Khan unite before he began conquering the rest of Eurasia?

Bidirectional LSTMs

Attention

$$\alpha_i = \text{softmax}_i (q^T W^a_i)$$

→ predict start token

Attention

$$\alpha'_i = \text{softmax}_i(q^T W'_{s\tilde{p}_i})$$

→ predict end token

He came to power by uniting many of the nomadic tribes of Northeast Asia. After founding the Mongol Empire and being proclaimed “Genghis Khan”, he started the Mongol invasions that resulted in the conquest of most of Eurasia. These included raids or invasions of the Qara Khitai, Caucasus, Khwarezmid Empire, Western Xia and Jin dynasties. The early campaigns were often accompanied by unspeakable massacres of the civilian populations – especially in the Khwarezmian and Xia controlled lands. By the end of his life, the Mongol Empire occupied a substantial portion of Central Asia and China.
### SQuAD 1.1 Results (single model, c. Feb 2017)

<table>
<thead>
<tr>
<th>Model Description</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic regression</td>
<td>51.0</td>
</tr>
<tr>
<td>Fine-Grained Gating (Carnegie Mellon U)</td>
<td>73.3</td>
</tr>
<tr>
<td>Match-LSTM (Singapore Management U)</td>
<td>73.7</td>
</tr>
<tr>
<td>DCN (Salesforce)</td>
<td>75.9</td>
</tr>
<tr>
<td>BiDAF (UW &amp; Allen Institute)</td>
<td>77.3</td>
</tr>
<tr>
<td>Multi-Perspective Matching (IBM)</td>
<td>78.7</td>
</tr>
<tr>
<td>ReasoNet (MSR Redmond)</td>
<td>79.4</td>
</tr>
<tr>
<td>DrQA (Chen et al. 2017)</td>
<td>79.4</td>
</tr>
<tr>
<td>r-net (MSR Asia) [Wang et al., ACL 2017]</td>
<td>79.7</td>
</tr>
<tr>
<td>Google Brain / CMU (Feb 2018)</td>
<td>88.0</td>
</tr>
<tr>
<td>Human performance</td>
<td>91.2</td>
</tr>
<tr>
<td>Pretrained + Finetuned Models circa 2021</td>
<td>&gt;93.0</td>
</tr>
</tbody>
</table>
BiDAF: the Bidirectional Attention Flow model

- Encode the question using word/char embeddings; pass on an biLSTM encoder
- Encode the passage similarly
- Passage-to-question and question-to-passage attention
- Modeling layer: another BiLSTM layer
- Output layer: two classifiers for predicting start and end points
- The entire model can be trained in an end-to-end way

(Seo et al, 2017): Bidirectional Attention Flow for Machine Comprehension
BiDAF: the Bidirectional Attention Flow model

(Seo et al, 2017): Bidirectional Attention Flow for Machine Comprehension
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. As this was the 50th Super Bowl, the league emphasized the "golden anniversary" with various gold-themed initiatives, as well as temporarily suspending the tradition of naming each Super Bowl game with Roman numerals (under which the game would have been known as "Super Bowl L"), so that the logo could prominently feature the Arabic numerals 50.

Where did Super Bowl 50 take place?

at, the, at, Stadium, Levi, in, Santa, Ana
[ ]
Super, Super, Super, Super, Super Bowl, Bowl, Bowl, Bowl, Bowl 50

initiatives
# BiDAF: results on SQuAD

<table>
<thead>
<tr>
<th></th>
<th>Single Model</th>
<th>Ensemble</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EM</td>
<td>F1</td>
</tr>
<tr>
<td>Logistic Regression Baseline&lt;sup&gt;a&lt;/sup&gt;</td>
<td>40.4</td>
<td>51.0</td>
</tr>
<tr>
<td>Dynamic Chunk Reader&lt;sup&gt;b&lt;/sup&gt;</td>
<td>62.5</td>
<td>71.0</td>
</tr>
<tr>
<td>Fine-Grained Gating&lt;sup&gt;c&lt;/sup&gt;</td>
<td>62.5</td>
<td>73.3</td>
</tr>
<tr>
<td>Match-LSTM&lt;sup&gt;d&lt;/sup&gt;</td>
<td>64.7</td>
<td>73.7</td>
</tr>
<tr>
<td>Multi-Perspective Matching&lt;sup&gt;e&lt;/sup&gt;</td>
<td>65.5</td>
<td>75.1</td>
</tr>
<tr>
<td>Dynamic Coattention Networks&lt;sup&gt;f&lt;/sup&gt;</td>
<td>66.2</td>
<td>75.9</td>
</tr>
<tr>
<td>R-Net&lt;sup&gt;g&lt;/sup&gt;</td>
<td>68.4</td>
<td>77.5</td>
</tr>
<tr>
<td>BiDAF (Ours)</td>
<td>68.0</td>
<td>77.3</td>
</tr>
</tbody>
</table>

(a) Results on the SQuAD test set

<table>
<thead>
<tr>
<th></th>
<th>EM</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>No char embedding</td>
<td>65.0</td>
<td>75.4</td>
</tr>
<tr>
<td>No word embedding</td>
<td>55.5</td>
<td>66.8</td>
</tr>
<tr>
<td>No C2Q attention</td>
<td>57.2</td>
<td>67.7</td>
</tr>
<tr>
<td>No Q2C attention</td>
<td>63.6</td>
<td>73.7</td>
</tr>
<tr>
<td>Dynamic attention</td>
<td>63.5</td>
<td>73.6</td>
</tr>
<tr>
<td>BiDAF (single)</td>
<td>67.7</td>
<td>77.3</td>
</tr>
<tr>
<td>BiDAF (ensemble)</td>
<td>72.6</td>
<td>80.7</td>
</tr>
</tbody>
</table>

(b) Ablations on the SQuAD dev set
BERT for reading comprehension

- BERT is pre-trained on two training objectives:
  - Masked language model (MLM)
  - Next sentence prediction (NSP)

- BERT_{base} has 12 layers and 110M parameters, BERT_{large} has 24 layers and 330M parameters
BERT for reading comprehension

\[ \mathcal{L} = - \log p_{\text{start}}(s^*) - \log p_{\text{end}}(e^*) \]

\[ p_{\text{start}}(i) = \text{softmax}_i(w_{\text{start}}^T \mathbf{h}_i) \]

\[ p_{\text{end}}(i) = \text{softmax}_i(w_{\text{end}}^T \mathbf{h}_i) \]

where \( \mathbf{h}_i \) is the hidden vector of \( c_i \), returned by BERT
BERT for reading comprehension

\[ \mathcal{L} = -\log p_{\text{start}}(s^*) - \log p_{\text{end}}(e^*) \]

- All the BERT parameters (e.g., 110M) as well as the newly introduced parameters (e.g., 768 x 2 = 1536) are optimized together for \( h_{\text{start}}, h_{\text{end}} \) (e.g., 768x2=1536) are optimized together for \( \mathcal{L} \)

- It works amazing well. Stronger pre-trained language models can lead to even better performance and SQuAD becomes a standard dataset for testing pre-trained models.

<table>
<thead>
<tr>
<th>Model</th>
<th>F1</th>
<th>EM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human performance</td>
<td>91.2*</td>
<td>82.3*</td>
</tr>
<tr>
<td>BiDAF</td>
<td>77.3</td>
<td>67.7</td>
</tr>
<tr>
<td>BERT-base</td>
<td>88.5</td>
<td>80.8</td>
</tr>
<tr>
<td>BERT-large</td>
<td>90.9</td>
<td>84.1</td>
</tr>
<tr>
<td>XLNet</td>
<td>94.5</td>
<td>89.0</td>
</tr>
<tr>
<td>RoBERTa</td>
<td>94.6</td>
<td>88.9</td>
</tr>
<tr>
<td>ALBERT</td>
<td>94.8</td>
<td>89.3</td>
</tr>
</tbody>
</table>

(dev set, except for human performance)
Comparisons between BiDAF and BERT models

• BERT model has many many more parameters (110M or 330M) BiDAF has ~2.5M parameters.
• BiDAF is built on top of several bidirectional LSTMs while BERT is built on top of Transformers (no recurrence architecture and easier to parallelize).
• BERT is pre-trained while BiDAF is only built on top of GloVe (and all the remaining parameters need to be learned from the supervision datasets).

Pre-training is clearly a game changer but it is expensive..
Is reading comprehension solved?
Is reading comprehension solved?

• We have already surpassed human performance on SQuAD. Does it mean that reading comprehension is already solved?

• The current systems still perform poorly on adversarial examples or examples from out-of-domain distributions (Jia and Liang, 2017)

• Systems trained on one dataset can’t generalize to other datasets (Sen and Saffari, 2020)
Is reading comprehension solved?

- BERT model trained on SQuAD

| MFT: change in one person only | 41.5 | C: Both Luke and Abigail were writers, but there was a change in Abigail, who is now a model. Q: Who is a model? A: Abigail \( ^\hat{\circ} \): Abigail were writers, but there was a change in Abigail |
| MFT: Understanding before/after, last/first | 82.9 | C: Logan became a farmer before Danielle did. Q: Who became a farmer last? A: Danielle \( ^\hat{\circ} \): Logan |
| MFT: Context has negation | 67.5 | C: Aaron is not a writer. Rebecca is. Q: Who is a writer? A: Rebecca \( ^\hat{\circ} \): Aaron |
| MFT: \( Q \) has negation, \( C \) does not | 100.0 | C: Aaron is an editor. Mark is an actor. Q: Who is not an actor? A: Aaron \( ^\hat{\circ} \): Mark |
| MFT: Simple coreference, he/she. | 100.0 | C: Melissa and Antonio are friends. He is a journalist, and she is an adviser. Q: Who is a journalist? A: Antonio \( ^\hat{\circ} \): Melissa |
| MFT: Simple coreference, his/her. | 100.0 | C: Victoria and Alex are friends. Her mom is an agent Q: Whose mom is an agent? A: Victoria \( ^\hat{\circ} \): Alex |
| MFT: former/latter | 100.0 | C: Kimberly and Jennifer are friends. The former is a teacher Q: Who is a teacher? A: Kimberly \( ^\hat{\circ} \): Jennifer |
| SRL: subject/object distinction | 60.8 | C: Richard bothers Elizabeth. Q: Who is bothered? A: Elizabeth \( ^\hat{\circ} \): Richard |
| SRL: subj/obj distinction with 3 agents | 95.7 | C: Jose hates Lisa. Kevin is hated by Lisa. Q: Who hates Kevin? A: Lisa \( ^\hat{\circ} \): Jose |

(Ribeiro et al., 2020): Beyond Accuracy: Behavioral Testing of NLP Models with CheckList
3. Open-domain question answering

- Different from reading comprehension, we don’t assume a given passage.
- Instead, we only have access to a large collection of documents (e.g., Wikipedia). We don’t know where the answer is located, and the goal is to return the answer for any open-domain questions.

- Much more challenging and a more practical problem!
Retriever-Reader framework

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

https://github.com/facebookresearch/DrQA

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

- Input: a large collection of documents $\mathcal{D} = D_1, D_2, \ldots, D_N$ and $Q$
- Output: an answer string $A$

- Retriever: $f(\mathcal{D}, Q) \rightarrow P_1, \ldots, P_K$
  - $K$ is pre-defined (e.g., 100)
- Reader: $g(Q, \{P_1, \ldots, P_K\}) \rightarrow A$
  - A reading comprehension problem!

In DrQA,

- Retriever = A standard TF-IDF information-retrieval sparse model (a fixed module)
- Reader = a neural reading comprehension model that we just learned
  - Trained on SQuAD and other distantly-supervised QA datasets

Distantly-supervised examples: $(Q, A) \rightarrow (P, Q, A)$

Chen et al., 2017. Reading Wikipedia to Answer Open-domain Questions
We can train the retriever! Joint training of retriever and reader

- Each text passage can be encoded as a vector using BERT and the retriever score can be measured as the dot product between the question representation and passage representation.
- However, it is not easy to model as there are a huge number of passages (e.g., 21M in English Wikipedia)

Lee et al., 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering
We can train the retriever! REALM: retrieval-augmented LM

REALM augments language model pre-training with a neural knowledge retriever that retrieves knowledge from a textual knowledge corpus (e.g., all of Wikipedia).

Signal from the language modeling objective backpropagates all the way through the retriever, which must consider millions of documents in $Z$—a significant computational challenge.

(Guu et al., 2020)
We can train the retriever! REALM: retrieval-augmented LM

(Understanding and retrieving knowledge from pre-trained corpora)
We can train the retriever! REALM: retrieval-augmented LM

\[ p(y | z, x) = \prod_{j=1}^{J_x} p(y_j | z, x) \]

\[ p(y | x) = \sum_{z \in Z} p(y | z, x) p(z | x) \]

\[ p(z | x) = \frac{\exp f(x, z)}{\sum_{z'} \exp f(x, z')} \]

\[ f(x, z) = \text{Embed}_{\text{input}}(x)^\top \text{Embed}_{\text{doc}}(z) \]

(Guu et al., 2020)
We can train the retriever! REALM: retrieval-augmented LM

Unlabeled text, from pre-training corpus ($\mathcal{X}$)

The [MASK] at the top of the pyramid ($x$)

Textual knowledge corpus ($\mathcal{Z}$)

Retrieved document

The pyramidion on top allows for less material higher up the pyramid. ($z$)

Query and document

[CLS] The [MASK] at the top of the pyramid
[SEP] The pyramidion on top allows for less material higher up the pyramid. ($x, z$)

Knowledge-Augmented Encoder $\sim p_\phi(y|x, z)$

End-to-end backpropagation

$\text{Answer} \quad \text{[MASK]} = \text{pyramidion} (y)$

(Guu et al., 2020)

Knowledge augmented encoder

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

- Key computational challenge is that the marginal probability involves a summation over all docs.
- Approximate this by summing over top k docs with highest prob under $p(z \mid x)$ --- via Maximum inner product search (MIPS)
We can train the retriever! REALM: retrieval-augmented LM

<table>
<thead>
<tr>
<th>Name</th>
<th>Architectures</th>
<th>Pre-training</th>
<th>NQ (79k/4k)</th>
<th>WQ (3k/2k)</th>
<th>CT (1k/1k)</th>
<th># params</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT-Baseline (Lee et al., 2019)</td>
<td>Sparse Retr.+Transformer</td>
<td>BERT</td>
<td>26.5</td>
<td>17.7</td>
<td>21.3</td>
<td>110m</td>
</tr>
<tr>
<td>T5 (base) (Roberts et al., 2020)</td>
<td>Transformer Seq2Seq</td>
<td>T5 (Multitask)</td>
<td>27.0</td>
<td>29.1</td>
<td>-</td>
<td>223m</td>
</tr>
<tr>
<td>T5 (large) (Roberts et al., 2020)</td>
<td>Transformer Seq2Seq</td>
<td>T5 (Multitask)</td>
<td>29.8</td>
<td>32.2</td>
<td>-</td>
<td>738m</td>
</tr>
<tr>
<td>T5 (11b) (Roberts et al., 2020)</td>
<td>Transformer Seq2Seq</td>
<td>T5 (Multitask)</td>
<td>34.5</td>
<td>37.4</td>
<td>-</td>
<td>11318m</td>
</tr>
<tr>
<td>DrQA (Chen et al., 2017)</td>
<td>Sparse Retr.+DocReader</td>
<td>N/A</td>
<td>-</td>
<td>20.7</td>
<td>25.7</td>
<td>34m</td>
</tr>
<tr>
<td>HardEM (Min et al., 2019a)</td>
<td>Sparse Retr.+Transformer</td>
<td>BERT</td>
<td>28.1</td>
<td>-</td>
<td>-</td>
<td>110m</td>
</tr>
<tr>
<td>GraphRetriever (Min et al., 2019b)</td>
<td>GraphRetriever+Transformer</td>
<td>BERT</td>
<td>31.8</td>
<td>31.6</td>
<td>-</td>
<td>110m</td>
</tr>
<tr>
<td>PathRetriever (Asai et al., 2019)</td>
<td>PathRetriever+Transformer</td>
<td>MLM</td>
<td>32.6</td>
<td>-</td>
<td>-</td>
<td>110m</td>
</tr>
<tr>
<td>ORQA (Lee et al., 2019)</td>
<td>Dense Retr.+Transformer</td>
<td>ICT+BERT</td>
<td>33.3</td>
<td>36.4</td>
<td>30.1</td>
<td>330m</td>
</tr>
<tr>
<td>ORQA (more fine-tune epochs)</td>
<td>Dense Retr.+Transformer</td>
<td>ICT+BERT</td>
<td>34.8</td>
<td>35.4</td>
<td>28.7</td>
<td>330m</td>
</tr>
<tr>
<td>Ours ($\mathcal{X} = \text{Wikipedia, } \mathcal{Z} = \text{Wikipedia}$)</td>
<td>Dense Retr.+Transformer</td>
<td>REALM</td>
<td>39.2</td>
<td>40.2</td>
<td>46.8</td>
<td>330m</td>
</tr>
<tr>
<td>Ours ($\mathcal{X} = \text{CC-News, } \mathcal{Z} = \text{Wikipedia}$)</td>
<td>Dense Retr.+Transformer</td>
<td>REALM</td>
<td>40.4</td>
<td>40.7</td>
<td>42.9</td>
<td>330m</td>
</tr>
</tbody>
</table>

Test results on Open-QA benchmarks: NaturalQuestions, WebQuestions, CuratedTrec dataset

(Guu et al., 2020)
We can train the retriever

- Dense passage retrieval (DPR)
- We can also just train the retriever using question-answer pairs!

- Trainable retriever (using BERT) largely outperforms traditional IR retrieval models

Karpukhin et al., 2020. Dense Passage Retrieval for Open-Domain Question Answering
Deep retrieval + generative models

- Recent work shows that it is beneficial to generate answers instead of to extract answers.

<table>
<thead>
<tr>
<th>Model</th>
<th>NaturalQuestions</th>
<th>TriviaQA</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORQA (Lee et al., 2019)</td>
<td>31.3</td>
<td>45.1</td>
</tr>
<tr>
<td>REALM (Guu et al., 2020)</td>
<td>38.2</td>
<td>-</td>
</tr>
<tr>
<td>DPR (Karpukhin et al., 2020)</td>
<td>41.5</td>
<td>57.9</td>
</tr>
<tr>
<td>SpanSeqGen (Min et al., 2020)</td>
<td>42.5</td>
<td>-</td>
</tr>
<tr>
<td>RAG (Lewis et al., 2020)</td>
<td>44.5</td>
<td>56.1</td>
</tr>
<tr>
<td>T5 (Roberts et al., 2020)</td>
<td>36.6</td>
<td>60.5</td>
</tr>
<tr>
<td>GPT-3 few shot (Brown et al., 2020)</td>
<td>29.9</td>
<td>71.2</td>
</tr>
<tr>
<td>Fusion-in-Decoder (base)</td>
<td>48.2</td>
<td>65.0</td>
</tr>
<tr>
<td>Fusion-in-Decoder (large)</td>
<td><strong>51.4</strong></td>
<td><strong>67.6</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>80.1</strong></td>
</tr>
</tbody>
</table>

Izacard and Grave 2020. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering
LLMs can do open-domain QA well

- ... without an explicit retriever stage

Roberts et al., 2020. How Much Knowledge Can You Pack Into the Parameters of a Language Model?
Maybe the reader model is not necessary too!

- It is possible to encode all the phrases (60 billion phrases in Wikipedia) using dense vectors and only do nearest neighbor search without a BERT model at inference time!

Seo et al., 2019. Real-Time Open-Domain Question Answering with Dense-Sparse Phrase Indexing
Lee et al., 2020. Learning Dense Representations of Phrases at Scale
LLM based QA (with web search!)

Where does Christopher D. Manning teach?

Christopher D. Manning is a professor of computer science and linguistics at Stanford University [1][2]. He is also the Director of the Stanford Artificial Intelligence Laboratory [1], and has been teaching at Stanford since 1994.

1. Christopher Manning's Profile | Stanford Profiles
   https://profiles.stanford.edu/chris-manning

2. Introduction to Information Retrieval: Manning, Christopher D ...
   https://www.amazon.com/Introduction-Information-Retrieval-Christopher-Manning/dp/0521865719
Problem with LLM based QA

- Seems totally reasonable!

- But (1) it’s not his most cited paper, and (2) it doesn’t have that many citations. Yikes! Also the reference to a web page doesn’t help.
LLMs cannot memorize everything

- LLMs store an impressive amount of information in their parameters
  - But LMs can’t memorize everything
  - The world changes over time
  - You might want it to use private documents that aren’t on the web

- Also, black-box LLMs are opaque
  - Given a query, it produces an answer, but it’s difficult to verify if the answer is correct
Using retrieval to overcome LLMs’ shortcomings

• Instead of asking the LLM to memorize everything, can we provide the LLM with relevant and useful content just-in-time?
• **Retrieval** is a common mechanism for identifying such relevant information.
  • **Dynamic**: it’s easy to update / add documents to your retrieval system
  • **Interpretable**: LM can generate pointers to retrieved documents that support human verification of its generations (citations)
4: Retrieval Augmented Generation (RAG)

- Retrieval Augmented Generation (RAG) is very powerful!
4: Retrieval Augmented Generation (RAG)

- State of the art for open-domain QA
  - Combine strengths of Open (non-parametric) and Closed-book (parametric)

- More specific, diverse, and factual generation than seq2seq (e.g., BART)

<table>
<thead>
<tr>
<th>Model</th>
<th>NQ</th>
<th>TQA</th>
<th>WQ</th>
<th>CT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Closed T5-11B</td>
<td>34.5</td>
<td>-</td>
<td>37.4</td>
<td>-</td>
</tr>
<tr>
<td>Book T5-11B+SSM</td>
<td>36.6</td>
<td>/50.1</td>
<td>44.7</td>
<td>-</td>
</tr>
<tr>
<td>Open REALM</td>
<td>40.4</td>
<td>-</td>
<td>40.7</td>
<td>46.8</td>
</tr>
<tr>
<td>Book DPR</td>
<td>41.5</td>
<td>57.9</td>
<td>41.1</td>
<td>50.6</td>
</tr>
<tr>
<td>RAG-Token</td>
<td>44.1</td>
<td>55.2/66.1</td>
<td>45.5</td>
<td>50.0</td>
</tr>
<tr>
<td>RAG-Seq.</td>
<td><strong>44.5</strong></td>
<td>56.8/68.0</td>
<td>45.2</td>
<td><strong>52.2</strong></td>
</tr>
</tbody>
</table>

[Piktus+ 2021]
4: Retrieval Augmented Generation (RAG)

How well does RAG work, anyway? Is it more factual?
Problem #1: how many documents can use?

- The retriever is key: if we have to use only 1 document, then we have to get that right.
- Why not get lots of documents and pass it to the LM?
LLM’s can’t pay attention to the entire context

- The long-context problem bites you – LLMs do not pay attention to its context well!
- Setup: 1 relevant document, all others irrelevant

Best Closed-Book performance: GPT-3.5-Turbo, ~56%
Best Oracle (only feed in relevant doc) performance: GPT-3.5-Turbo, ~88.5%

[Liu+ 2023]
In practice: LLMs can’t use many documents

- Retriever performance (yellow) rises slowly to 90% recall
- RAG performance (other lines) saturate very quickly – after 10-20 documents.
Problem #2: useful and usable citations

- A unique benefit of RAG: citing your sources
How accurate are these citations?

- These citations themselves are generated by LLMs
- So the citations could also be hallucinated. How often does that happen?
Existing systems have high fluency, but low correctness for citations

- Outputs are easy to read / appear useful to rater (1-5 scale)

<table>
<thead>
<tr>
<th>Perceived Utility (↑)</th>
<th>Fluency (↑)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Over All Queries</td>
<td>Average Over All Queries</td>
</tr>
<tr>
<td>Bing Chat</td>
<td>4.34</td>
</tr>
<tr>
<td>NeevaAI</td>
<td>4.48</td>
</tr>
<tr>
<td>perplexity.ai</td>
<td>4.56</td>
</tr>
<tr>
<td>YouChat</td>
<td>4.62</td>
</tr>
<tr>
<td>Average</td>
<td>4.50</td>
</tr>
</tbody>
</table>

[Liu+ 2023]
Existing systems have high fluency, but low correctness for citations

- But precision and recall are both low ...

<table>
<thead>
<tr>
<th>Citation Precision (%; ↑)</th>
<th>Citation Recall (%; ↑)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Average Over All Queries</strong></td>
<td><strong>Average Over All Queries</strong></td>
</tr>
<tr>
<td>Bing Chat 89.5</td>
<td>Bing Chat 58.7</td>
</tr>
<tr>
<td>NeevaAI 72.0</td>
<td>NeevaAI 67.6</td>
</tr>
<tr>
<td>perplexity.ai 72.7</td>
<td>perplexity.ai 68.7</td>
</tr>
<tr>
<td>YouChat 63.6</td>
<td>YouChat 11.1</td>
</tr>
<tr>
<td><strong>Average</strong> 74.5</td>
<td><strong>Average</strong> 51.5</td>
</tr>
</tbody>
</table>
Overview

1. What is question answering?

2. Reading comprehension
   ✓ How to answer questions over a single passage of text

3. Open-domain (textual) question answering
   ✓ How to answer questions over a large collection of documents

4. Retrieval-augmented generation for question answering