COS 484/584

(Advanced) Natural Language Processing

L21: Question Answering

Spring 2021
Announcements

Guest lectures this Wednesday (April 21) and next Monday (April 26)

Mark Yatskar (UPenn)  
Fairness in NLP

Yonatan Belinkov (Technion)  
Interpretability in NLP
This lecture

1. What is question answering?
2. Reading comprehension
3. Open-domain question answering
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)
What information source does a system build on?
- A text passage, all Web documents, knowledge bases, tables, images..

Question type
- Factoid vs non-factoid, open-domain vs closed-domain, simple vs compositional, ..

Answer type
- A short segment of text, a paragraph, a list, yes/no, …
Lots of practical applications

Siberia

Lake Baikal, in Siberia, holds the distinction of being both the deepest lake in the world and the largest freshwater lake, holding more than 20% of the unfrozen fresh water on the surface of Earth.
Lots of practical applications

The best way to prevent illness is to avoid being exposed to this virus. Learn how COVID-19 spreads and practice these actions to help prevent the spread of this illness.

To help prevent the spread of COVID-19:
- Cover your mouth and nose with a mask when around people who don’t live with you. Masks work best when everyone wears one.
- Stay at least 6 feet (about 2 arm lengths) from others.
- Avoid crowds. The more people you are in contact with, the more likely you are to be exposed to COVID-19.
- Avoid unventilated indoor spaces. If indoors, bring in fresh air by opening windows and doors.
- Clean your hands often, either with soap and water for 20 seconds or a hand sanitizer that contains at least 60% alcohol.
- Get vaccinated against COVID-19 when it’s your turn.
- Avoid close contact with people who are sick.
- Cover your cough or sneeze with a tissue, then throw the tissue in the trash.
- Clean and disinfect frequently touched objects and surfaces daily.

Learn more on cdc.gov

For informational purposes only. Consult your local medical authority for advice.
Lots of practical applications

Smart Speaker Use Case Frequency January 2020

Listen to streaming music service: 88.7% daily, 73.6% weekly, 39.8%
Ask a question: 83.1% daily, 66.2% weekly, 29.4%
Check the weather: 77.0% daily, 58.8% weekly, 32.9%
Set a timer: 64.9% daily, 52.4% weekly, 20.3%
Set an alarm: 59.8% daily, 45.6% weekly, 26.3%
Listen to the radio: 59.8% daily, 42.6% weekly, 19.0%
Listen to News / Sports: 50.6% daily, 37.7% weekly, 18.9%
Use a favorite Alexa skill or Google Action: 47.9% monthly, 34.6% weekly, 16.4%
Play game or answer trivia: 46.1% monthly, 27.7% weekly, 9.0%

Listen to Podcast or other talk formats: 44.9% daily, 32.6% weekly, 11.4%
Control smart home devices: 43.4% daily, 31.9% weekly, 24.5%
Find a recipe or cooking instructions: 42.3% daily, 26.0% weekly, 5.4%
Call someone: 40.2% daily, 21.2% weekly, 8.5%
Search for product information: 38.2% daily, 27.9% weekly, 7.3%
Check traffic / directions: 35.1% daily, 23.7% weekly, 11.1%
Access my calendar: 32.1% daily, 19.9% weekly, 9.5%
Send a text message: 27.9% monthly, 11.9% weekly, 6.7%
Make a purchase: 25.2% monthly, 11.9% weekly, 4.9%

Source: Voicebot AI - 2020
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)!
Beyond textual QA problems

Today, we will mostly focus on how to answer questions based on unstructured text.

Knowledge based QA

Image credit: Percy Liang
Beyond textual QA problems

Today, we will mostly focus on how to answer questions based on unstructured text.

(Antol et al., 2015): Visual Question Answering
Tesla was the fourth of five children. He had an older brother named Dane and three sisters, Milka, Angelina and Marica. Dane was killed in a horse-riding accident when Nikola was five. In 1861, Tesla attended the "Lower" or "Primary" School in Smiljan where he studied German, arithmetic, and religion. In 1862, the Tesla family moved to Gospić, Austrian Empire, where Tesla's father worked as a pastor. Nikola completed "Lower" or "Primary" School, followed by the "Lower Real Gymnasium" or "Normal School."

Q: What language did Tesla study while in school?
A: German
Kannada language is the official language of Karnataka and spoken as a native language by about 66.54% of the people as of 2011. Other linguistic minorities in the state were Urdu (10.83%), Telugu language (5.84%), Tamil language (3.45%), Marathi language (3.38%), Hindi (3.3%), Tulu language (2.61%), Konkani language (1.29%), Malayalam (1.27%) and Kodava Takk (0.18%). In 2007 the state had a birth rate of 2.2%, a death rate of 0.7%, an infant mortality rate of 5.5% and a maternal mortality rate of 0.2%. The total fertility rate was 2.2.

Q: Which linguistic minority is larger, Hindi or Malayalam?
A: Hindi
Why do we care about this problem?

• Useful for many practical applications
• 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!

• SQuAD still remains the most popular reading comprehension dataset; it is “almost solved” today and the state-of-the-art exceeds the estimated human performance.

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

Estimated human performance: EM = 82.3, F1 = 91.2

Q: What did Tesla do in December 1878?

A: {left Graz, left Graz, left Graz and severed all relations with his family}

Prediction: {left Graz and severed}

Exact match: max{0, 0, 0} = 0

F1: max{0.67, 0.67, 0.61} = 0.67
Neural models for reading comprehension

How can we build a model to solve SQuAD?
(We are going to use passage, paragraph and context, as well as question and query interchangeably)

- 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: \( 1 \leq \text{start} \leq \text{end} \leq N \)

- A family of LSTM-based models with attention (2016-2018)
  
  Attentive Reader (Hermann et al., 2015), Stanford Attentive Reader (Chen et al., 2016), Match-LSTM (Wang et al., 2017), BiDFA (Seo et al., 2017), Dynamic coattention network (Xiong et al., 2017), DrQA (Chen et al., 2017), R-Net (Wang et al., 2017), ReasoNet (Shen et al., 2017).

- Fine-tuning BERT-like models for reading comprehension (2019+)

Answer is a span in the passage.
LSTM-based vs BERT models

Image credit: (Seo et al, 2017)

Image credit: J & M, edition 3
Recap: seq2seq model with attention

• Instead of source and target sentences, we also have two sequences: passage and question (lengths are imbalanced)

• We need to model which words in the passage are most relevant to the question (and which question words)

  Attention is the key ingredient here, similar to which words in the source sentence are most relevant to the current target word…

• We don’t need an autoregressive decoder to generate the target sentence word-by-word. Instead, we just need to train two classifiers to predict the start and end positions of the answer!
BiDAF: Bidirectional Attention Flow

(Seo et al., 2017): Bidirectional Attention Flow for Machine Comprehension
BiDAF: Encoding

- Use a concatenation of word embedding (GloVe) and character embedding (CNNs over character embeddings) for each word in context and query.

- Then, use two bidirectional LSTMs separately to produce contextual embeddings for both context and query.

\[
\begin{align*}
\overrightarrow{c}_i &= \text{LSTM}(\overrightarrow{c}_{i-1}, e(c_i)) \in \mathbb{R}^H \\
\overleftarrow{c}_i &= \text{LSTM}(\overleftarrow{c}_{i+1}, e(c_i)) \in \mathbb{R}^H \\
\overrightarrow{c}_i &= [\overrightarrow{c}_i; \overleftarrow{c}_i] \in \mathbb{R}^{2H} \\
\overrightarrow{q}_i &= \text{LSTM}(\overrightarrow{q}_{i-1}, e(q_i)) \in \mathbb{R}^H \\
\overleftarrow{q}_i &= \text{LSTM}(\overleftarrow{q}_{i+1}, e(q_i)) \in \mathbb{R}^H \\
\overrightarrow{q}_i &= [\overrightarrow{q}_i; \overleftarrow{q}_i] \in \mathbb{R}^{2H}
\end{align*}
\]
BiDAF: Attention

- Context-to-query attention: For each context word, choose the most relevant words from the query words.

Q: *Who leads the United States?*

C: *Barak Obama is the president of the USA.*

For each context word, find the most relevant query word.

(Slides adapted from Minjoon Seo)
BiDAF: Attention

- Query-to-context attention: choose the context words that are most relevant to one of query words.

While Seattle’s weather is very nice in summer, its weather is very rainy in winter, making it one of the most gloomy cities in the U.S. LA is ...

Q: Which city is gloomy in winter?

(Slides adapted from Minjoon Seo)
BiDAF: Attention

- First, compute a similarity score for every pair of \((c_i, q_j)\):
  \[
  S_{i,j} = \mathbf{w}_{\text{sim}}^T [c_i; q_j; c_i \odot q_j] \in \mathbb{R} \quad \mathbf{w}_{\text{sim}} \in \mathbb{R}^{6H}
  \]

- Context-to-query attention (which question words are more relevant to \(c_j\)):
  \[
  \alpha_{i,j} = \text{softmax}_j (S_{i,j}) \in \mathbb{R} \quad \mathbf{a}_i = \sum_{j=1}^{M} \alpha_{i,j} q_j \in \mathbb{R}^{2H}
  \]

- Query-to-context attention (which context words are relevant to some question words):
  \[
  \beta_i = \text{softmax}_i (\max_{j=1}^{M} (S_{i,j})) \in \mathbb{R}^N \quad \mathbf{b} = \sum_{i=1}^{N} \beta_i c_i \in \mathbb{R}^{2H}
  \]

The final output is
\[
\mathbf{g}_i = [c_i; \mathbf{a}_i; c_i \odot \mathbf{a}_i; c_i \odot \mathbf{b}] \in \mathbb{R}^{8H}
\]
**BiDAF: Modeling and output layers**

**Modeling layer:** pass $g_i$ to another two layers of bi-directional LSTMs.
- Attention layer is modeling interactions *between query and context*
- Modeling layer is modeling interactions *within context words*

**Output layer:** two classifiers predicting the start and end positions

The final training loss is

\[ \mathcal{L} = - \log p_{\text{start}}(s^*) - \log p_{\text{end}}(e^*) \]
### BiDAF: Performance on SQuAD

This model achieved 77.3 F1 on SQuAD v1.1.

- Without context-to-query attention → 67.7 F1
- Without query-to-context attention → 73.7 F1
- Without character embeddings → 75.4 F1

<table>
<thead>
<tr>
<th>Single Model</th>
<th>Published EM / F1</th>
<th>LeaderBoard EM / F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>LR Baseline (Rajpurkar et al., 2016)</td>
<td>40.4 / 51.0</td>
<td>40.4 / 51.0</td>
</tr>
<tr>
<td>Dynamic Chunk Reader (Yu et al., 2016)</td>
<td>62.5 / 71.0</td>
<td>62.5 / 71.0</td>
</tr>
<tr>
<td>Match-LSTM with Ans-Ptr (Wang &amp; Jiang, 2016)</td>
<td>64.7 / 73.7</td>
<td>64.7 / 73.7</td>
</tr>
<tr>
<td>Multi-Perspective Matching (Wang et al., 2016)</td>
<td>65.5 / 75.1</td>
<td>70.4 / 78.8</td>
</tr>
<tr>
<td>Dynamic Coattention Networks (Xiong et al., 2016)</td>
<td>66.2 / 75.9</td>
<td>66.2 / 75.9</td>
</tr>
<tr>
<td>FastQA (Weissenborn et al., 2017)</td>
<td>68.4 / 77.1</td>
<td>68.4 / 77.1</td>
</tr>
<tr>
<td>BiDAF (Seo et al., 2016)</td>
<td>68.0 / 77.3</td>
<td>68.0 / 77.3</td>
</tr>
<tr>
<td>SEDT (Liu et al., 2017a)</td>
<td>68.1 / 77.5</td>
<td>68.5 / 78.0</td>
</tr>
<tr>
<td>RaSoR (Lee et al., 2016)</td>
<td>70.8 / 78.7</td>
<td>69.6 / 77.7</td>
</tr>
<tr>
<td>FastQAExt (Weissenborn et al., 2017)</td>
<td>70.8 / 78.9</td>
<td>70.8 / 78.9</td>
</tr>
<tr>
<td>ReasoNet (Shen et al., 2017b)</td>
<td>69.1 / 78.9</td>
<td>70.6 / 79.4</td>
</tr>
<tr>
<td>Document Reader (Chen et al., 2017)</td>
<td>70.0 / 79.0</td>
<td>70.7 / 79.4</td>
</tr>
<tr>
<td>Ruminating Reader (Gong &amp; Bowman, 2017)</td>
<td>70.6 / 79.5</td>
<td>70.6 / 79.5</td>
</tr>
<tr>
<td>jNet (Zhang et al., 2017)</td>
<td>70.6 / 79.8</td>
<td>70.6 / 79.8</td>
</tr>
<tr>
<td>Conductor-net</td>
<td>N/A</td>
<td>72.6 / 81.4</td>
</tr>
<tr>
<td>Interactive AoA Reader (Cui et al., 2017)</td>
<td>N/A</td>
<td>73.6 / 81.9</td>
</tr>
<tr>
<td>Reg-RaSoR</td>
<td>N/A</td>
<td>75.8 / 83.3</td>
</tr>
<tr>
<td>DCN+</td>
<td>N/A</td>
<td>74.9 / 82.8</td>
</tr>
<tr>
<td>AIR-FusionNet</td>
<td>N/A</td>
<td>76.0 / 83.9</td>
</tr>
<tr>
<td>R-Net (Wang et al., 2017)</td>
<td>72.3 / 80.7</td>
<td>76.5 / 84.3</td>
</tr>
<tr>
<td>BiDAF + Self Attention + ELMo</td>
<td>N/A</td>
<td>77.9 / 85.3</td>
</tr>
<tr>
<td>Reinforced Mnemonic Reader (Hu et al., 2017)</td>
<td>73.2 / 81.8</td>
<td>73.2 / 81.8</td>
</tr>
</tbody>
</table>

(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.

Attention visualization

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

initiatives
BERT for reading comprehension
**Question** = Segment A

**Passage** = Segment B

**Answer** = predicting two endpoints in segment B

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

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

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

where \( h_i \) is the hidden vector of \( c_i \), returned by BERT

**Reference Text:** BERT-large is really big... it has 24 layers and an embedding size of 1,024, for a total of 340M parameters! Altogether it is 1.34GB, so expect it to take a couple minutes to download to your Colab instance.

Image credit: https://mccormickml.com/
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 \( w_{\text{start}}, w_{\text{end}} \) (e.g., 768 \( \times 2 = 1536 \)) 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></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) and 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..
Comparisons between BiDAF and BERT models

Are they really fundamentally different? Probably not.

- BiDAF and other models aim to model the interactions between question and passage.
- BERT uses self-attention between the concatenation of question and passage = \( \text{attention}(P, P) + \text{attention}(P, Q) + \text{attention}(Q, P) + \text{attention}(Q, Q) \)
- (Clark and Gardner, 2018) shows that adding a self-attention layer for the passage attention\((P, P)\) to BiDAF also improves performance.
Is reading comprehension solved?

• We have already surpassed human performance on SQuAD. Does it mean that reading comprehension is already solved? **Of course not!**

• The current systems still perform poorly on adversarial examples or examples from out-of-domain distributions

---

Article: Super Bowl 50
Paragraph: “Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver’s Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV.”

Question: “What is the name of the quarterback who was 38 in Super Bowl XXXIII?”

Original Prediction: John Elway
Prediction under adversary: Jeff Dean

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>71.4</td>
<td>75.4</td>
<td>75.5</td>
<td>80.0</td>
</tr>
<tr>
<td>AddSent</td>
<td>27.3</td>
<td>29.4</td>
<td>34.3</td>
<td>34.2</td>
</tr>
<tr>
<td>AddOneSent</td>
<td>39.0</td>
<td>41.8</td>
<td>45.7</td>
<td>46.9</td>
</tr>
<tr>
<td>AddAny</td>
<td>7.6</td>
<td>11.7</td>
<td>4.8</td>
<td>2.7</td>
</tr>
<tr>
<td>AddCommon</td>
<td>38.9</td>
<td>51.0</td>
<td>41.7</td>
<td>52.6</td>
</tr>
</tbody>
</table>

(Jia and Liang, 2017): Adversarial Examples for Evaluating Reading Comprehension Systems
## Is reading comprehension solved?

Systems trained on one dataset can’t generalize to other datasets:

<table>
<thead>
<tr>
<th>Fine-tuned on</th>
<th>Evaluated on</th>
<th>SQuAD</th>
<th>TriviaQA</th>
<th>NQ</th>
<th>QuAC</th>
<th>NewsQA</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQuAD</td>
<td>75.6</td>
<td>46.7</td>
<td>48.7</td>
<td>20.2</td>
<td>41.1</td>
<td></td>
</tr>
<tr>
<td>TriviaQA</td>
<td>49.8</td>
<td>58.7</td>
<td>42.1</td>
<td>20.4</td>
<td>10.5</td>
<td></td>
</tr>
<tr>
<td>NQ</td>
<td>53.5</td>
<td>46.3</td>
<td>73.5</td>
<td>21.6</td>
<td>24.7</td>
<td></td>
</tr>
<tr>
<td>QuAC</td>
<td>39.4</td>
<td>33.1</td>
<td>33.8</td>
<td><strong>33.3</strong></td>
<td>13.8</td>
<td></td>
</tr>
<tr>
<td>NewsQA</td>
<td>52.1</td>
<td>38.4</td>
<td>41.7</td>
<td>20.4</td>
<td><strong>60.1</strong></td>
<td></td>
</tr>
</tbody>
</table>

(Sen and Saffari, 2020): What do Models Learn from Question Answering Datasets?
Is reading comprehension solved?

### BERT-large model trained on SQuAD

<table>
<thead>
<tr>
<th>Test TYPE and Description</th>
<th>Failure Rate (%)</th>
<th>Example Test cases (with expected behavior and prediction)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Vocab</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MFT: comparisons</td>
<td>20.0</td>
<td>C: Victoria is younger than Dylan.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Q: Who is less young? A: Dylan &gt; Victoria</td>
</tr>
<tr>
<td>MFT: intensifiers to superlative: most/least</td>
<td>91.3</td>
<td>C: Anna is worried about the project. Matthew is extremely worried about the project.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Q: Who is least worried about the project? A: Anna &lt; Matthew</td>
</tr>
<tr>
<td>MFT: match properties to categories</td>
<td>82.4</td>
<td>C: There is a tiny purple box in the room. Q: What size is the box? A: tiny &lt; purple</td>
</tr>
<tr>
<td>MFT: nationality vs job</td>
<td>49.4</td>
<td>C: Stephanie is an Indian accountant.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Q: What is Stephanie’s job? A: accountant &gt; Indian accountant</td>
</tr>
<tr>
<td>MFT: animal vs vehicles</td>
<td>26.2</td>
<td>C: Jonathan bought a truck. Isabella bought a hamster.</td>
</tr>
<tr>
<td>MFT: comparison to antonym</td>
<td>67.3</td>
<td>C: Jacob is shorter than Kimberly.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Q: Who is taller? A: Kimberly &gt; Jacob</td>
</tr>
<tr>
<td>MFT: more/less in context, more/less antonym in question</td>
<td>100.0</td>
<td>C: Jeremy is more optimistic than Taylor.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Q: Who is more pessimistic? A: Taylor &gt; Jeremy</td>
</tr>
<tr>
<td><strong>Robust</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INV: Swap adjacent characters in Q (typo)</td>
<td>11.6</td>
<td>C: ...Newcomen designs had a duty of about 7 million, but most were closer to 5 million....</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Q: What was the ideal duty of a Newcomen engine? A: INV &gt; 7 million &gt; 5 million</td>
</tr>
<tr>
<td>INV: add irrelevant sentence to C</td>
<td>9.8</td>
<td>(no example)</td>
</tr>
</tbody>
</table>

(Ribeiro et al., 2020): Beyond Accuracy: Behavioral Testing of NLP Models with CheckList
Is reading comprehension solved?

| MFT: | 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  
\[\text{Abigail were writers, but there was a change in Abigail}\]  
Q: Who became a farmer last? A: Danielle  
\[\text{Logan}\] |
| MFT: | 82.9 | C: Logan became a farmer before Danielle did.  
Q: Who became a farmer last? A: Danielle  
\[\text{Logan}\] |
| MFT: | 67.5 | C: Aaron is not a writer. Rebecca is.  
Q: Who is a writer? A: Rebecca  
\[\text{Aaron}\]  
Q: Who is an actor? A: Aaron  
\[\text{Mark}\] |
| MFT: | 100.0 | C: Aaron is an editor. Mark is an actor.  
Q: Who is a writer? A: Aaron  
\[\text{Mark}\]  
Q: Who is an actor? A: Aaron  
\[\text{Mark}\] |
| MFT: | 100.0 | C: Melissa and Antonio are friends. He is a journalist, and she is an adviser.  
Q: Who is a journalist? A: Antonio  
\[\text{Melissa}\]  
Q: Who is an agent? A: Victoria  
\[\text{Alex}\] |
| MFT: | 100.0 | C: Victoria and Alex are friends. Her mom is an agent  
Q: Whose mom is an agent? A: Victoria  
\[\text{Alex}\]  
Q: Who is a teacher? A: Kimberly  
\[\text{Jennifer}\] |
| MFT: | 100.0 | C: Kimberly and Jennifer are friends. The former is a teacher  
Q: Who is a teacher? A: Kimberly  
\[\text{Jennifer}\] |
| MFT: | 60.8 | C: Richard bothers Elizabeth.  
Q: Who is bothered? A: Elizabeth  
\[\text{Richard}\] |
| MFT: | 95.7 | C: Jose hates Lisa. Kevin is hated by Lisa.  
Q: Who hates Kevin? A: Lisa  
\[\text{Jose}\] |

(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 but a more practical problem!

In contrast to closed-domain systems that deal with questions under a specific domain (medicine, technical support)...

Retriever-reader framework

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

https://github.com/facebookresearch/DrQA
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 too

- **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 too

- 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
We can train the retriever too

Title: Harry Potter (film series)  
Who tells harry potter that he is a wizard in the harry potter series?  
Retrieval ranking: #90  
P(p|q)=0.85  P(a|p,q)=1.00  P(a,p|q)=0.84  

... and uncle. At the age of eleven, half-giant Rubeus Hagrid informs him that he is actually a wizard and that his parents were murdered by an evil wizard named Lord Voldemort. Voldemort also attempted to kill one-year-old Harry on the same night, but his killing curse mysteriously rebounded and reduced him to a weak and helpless form. Harry became extremely famous in the Wizarding World as a result. Harry begins his first year at Hogwarts School of Witchcraft and Wizardry and learns about magic. During the year, Harry and his friends Ron Weasley and Hermione Granger become entangled in the...

Title: Harry Potter (character)  
... Harry Potter (character) Harry James Potter is the titular protagonist of J. K. Rowling’s "Harry Potter" series. The majority of the books’ plot covers seven years in the life of the orphan Potter, who, on his eleventh birthday, learns he is a wizard. Thus, he attends Hogwarts School of Witchcraft and Wizardry to practice magic under the guidance of the kindly headmaster Albus Dumbledore and other school professors along with his best friends Ron Weasley and Hermione Granger. Harry also discovers that he is already famous throughout the novel’s magical community, and that his fate is tied with that of...

Dense retrieval + generative models

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

Fusion-in-decoder (FID) = DPR + T5

<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>-</td>
</tr>
<tr>
<td>GPT-3 few shot (Brown et al., 2020)</td>
<td>29.9</td>
<td>-</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>
</tbody>
</table>
Large language models 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!

Lee et al., 2020. Learning Dense Representations of Phrases at Scale