Language Models and Knowledge

Mengzhou Xia & Jane Pan

October 10th, 2022
Outline

1. What is a knowledge base?

2. Can language models be used as knowledge bases? (Petroni et al., 2019)

3. Can a closed-book QA LM perform as well as other open-book methods? (Roberts et al., 2020)

4. How to update facts? (Dai et al., 2021, Mitchell et al., 2022)
Introduction
Motivation

- The corpora used to pretrain language models are **huge aggregations** of information and data from the internet.
Motivation

- The corpora used to pretrain language models are huge aggregations of information and data from the internet
- Consider The Pile (Gao et al., 2020): **800GB total**
Motivation

- The corpora used to pretrain language models are huge aggregations of information and data from the internet
- Consider **The Pile** (Gao et al., 2020): **800GB total**
Motivation

- The corpora used to pretrain language models are huge aggregations of information and data from the internet
- Consider **The Pile** (Gao et al., 2020): **800GB total**

Graphic from Gao et al., 2020

This little purple box is the entirety of Wikipedia :)
Motivation

- The corpora used to pretrain language models are huge aggregations of information and data from the internet
- Consider The Pile (Gao et al., 2020): 800GB total
Pretraining and knowledge

- Pre-training allows language models to learn robust task-agnostic features, which is critical for high performance on downstream tasks.
Pretraining and knowledge

- Pre-training allows language models to learn robust **task-agnostic features**, which is critical for high performance on downstream tasks.

- As **language models and their pre-training corpora scale**, the amount of information encoded in the pretrained language models also increases:
  - Few-shot learning
  - In-context learning
Pretraining and knowledge

- Pre-training allows language models to learn robust task-agnostic features, which is critical for high performance on downstream tasks.

- As language models and their pre-training corpora scale, the amount of information encoded in the pretrained language models also increases:
  - Few-shot learning
  - In-context learning

Can we directly retrieve the knowledge learned in pre-training from a language model?

Key Question
Today, we take LLMs’ ability to “store” knowledge for granted.
Today, we take LLMs’ ability to “store” knowledge for granted

GPT-3 Zero-shot Knowledge Retrieval

- This was not so obvious to NLP researchers *three years ago*!
- Instead, **traditional knowledge bases** were often used
What is a knowledge base?
What is a knowledge base?

Graphic from Megan Leszczynski (Stanford)
What is a knowledge base?

A knowledge base is a collection of nodes, which are entities or abstract classes, interconnected by relationships such as date of birth, place of birth, and spouse. In the context of the diagram, Franklin D. Roosevelt and Eleanor Roosevelt are depicted as nodes connected by these relationships. The knowledge base includes structured information about entities like dates and places, and relationships between them.
What is a knowledge base?

- **knowledge base**
- **edges**
- relations between entities

**Graphic from Megan Leszczynski (Stanford)**
What is a knowledge base?

```
SELECT date of birth
WHERE person = "Franklin D. Roosevelt"
```
How were knowledge bases formed?
How were knowledge bases formed?

Knowledge Extraction Pipeline

unstructured text

knowledge base
How were knowledge bases formed?

[Diagram showing the Knowledge Extraction Pipeline: unstructured text to knowledge base]
Downsides of using knowledge bases
Downsides of using knowledge bases

Unstructured text

“Born in St. Louis, Missouri, to a prominent Boston Brahmin family…”

Requires supervised data to train the pipeline and/or fill the knowledge base.
Downsides of using knowledge bases

Requires **supervised data** to train the pipeline and/or fill the knowledge base

“Born in St. Louis, Missouri, to a prominent Boston Brahmin family…”
Downsides of using knowledge bases

Requires **supervised data** to train the pipeline and/or fill the knowledge base

"Born in St. Louis, Missouri, to a prominent Boston Brahmin family..."
Downsides of using knowledge bases

 Requires **supervised data** to train the pipeline and/or fill the knowledge base.
Downsides of using knowledge bases

Populating the knowledge base often involves complicated, multi-step NLP pipelines.
Downsides of using knowledge bases

Knowledge Extraction Pipeline

(T.S. Eliot, BORN-IN, Boston)

incorrect extraction

Prone to error propagation (from human annotations or knowledge extraction)
Downsides of using knowledge bases

“Born in St. Louis, Missouri, to a prominent Boston Brahmin family...”

Incorrect extraction

Prone to error propagation (from human annotations or knowledge extraction)
Downsides of using knowledge bases

Reliant on **fixed schemas** to store or query data

(T.S. Eliot, BORN IN, X)

slightly incorrect query

Graphic from Petroni et al., 2019
Downsides of using knowledge bases

Reliant on fixed schemas to store or query data

slightly incorrect query

(T.S. Eliot, BORN IN, X)

ERROR
“BORN IN” is not a legal relation.

(T.S. Eliot, BORN IN, X) → Boston → T.S. Eliot

Graphic from Petroni et al., 2019
Traditional knowledge bases are **inflexible** and require **significant manual effort**.

Are there better alternatives?
Language Models as Knowledge Bases? (Petroni et al., 2019)
Language models as knowledge bases?

Why language models?

- Pretrained on a huge corpus of data
- Doesn’t require annotations/supervision
- More flexible with natural language queries
- Can be used off-the-shelf
Language models as knowledge bases?

Why language models?

- Pretrained on a huge corpus of data
- Doesn’t require annotations/supervision
- More flexible with natural language queries
- Can be used off-the-shelf

But first, we need to see if language models really do store knowledge.

**Question:** How do we check this?
Language models as knowledge bases?

Why language models?
- Pretrained on a huge corpus of data
- Doesn’t require annotations/supervision
- More flexible with natural language queries
- Can be used off-the-shelf

But first, we need to see if language models really do store knowledge.

**Question:** How do we check this?

**Answer:**
Language models as knowledge bases?

Why language models?
- Pretrained on a huge corpus of data
- Doesn’t require annotations/supervision
- More flexible with natural language queries
- Can be used off-the-shelf

But first, we need to see if language models really do store knowledge.

Question: How do we check this?

Answer:
Goal: evaluate factual + commonsense knowledge in language models
LAMA Probe

- **Goal**: evaluate **factual + commonsense knowledge** in language models

- Collect set of **knowledge sources** (i.e. set of facts) and test to see how well the model’s knowledge captures these facts
LAMA Probe

- **Goal:** evaluate factual + commonsense knowledge in language models

- Collect set of **knowledge sources** (i.e. set of facts) and test to see how well the model’s knowledge captures these facts

- *How do we know how “knowledgeable” a LM is about a particular fact?*
Goal: evaluate factual + commonsense knowledge in language models

Collect set of knowledge sources (i.e. set of facts) and test to see how well the model’s knowledge captures these facts

How do we know how “knowledgeable” a LM is about a particular fact?

Given a cloze statement that queries the model for a missing token, knowledgeable LMs rank ground truth tokens high and other tokens lower.

Language Model Analysis
Evaluation of LM via LAMA

Given a cloze statement that queries the model for a missing token, knowledgeable LMs rank ground truth tokens high and other tokens lower.
Evaluation of LM via LAMA

Given a cloze statement that queries the model for a missing token, knowledgeable LMs rank ground truth tokens high and other tokens lower

“The color of the sky is [MASK].”

*according to the LM
Evaluation of LM via LAMA

Given a cloze statement that queries the model for a missing token, knowledgeable LMs rank ground truth tokens high and other tokens lower.

“The color of the sky is [MASK].”

Language Model

\[\text{candidate vocab} \]

*according to the LM
Evaluation of LM via LAMA

Given a cloze statement that queries the model for a missing token, knowledgeable LMs rank ground truth tokens high and other tokens lower

“The color of the sky is [MASK].”

Language Model

| Bob: blue | 0.003% |
| Bob: red  | 49%    |
| Bob: grass| 12%    |
| Bob: grey | 0.01%  |
| pear:     | 0.09%  |

*according to the LM
Evaluation of LM via LAMA

Given a cloze statement that queries the model for a missing token, knowledgeable LMs rank ground truth tokens high and other tokens lower

“The color of the sky is [MASK].”

Language Model

Bob: 0.003%
blue: 49%
red: 12%
grass: 0.01%
grey: 3%
. . .
pear: 0.09%

blue
red
grey
. .
grass
pear
Bob

probability scores
rankings

most likely*
least likely*

*according to the LM
Evaluation of LM via LAMA

Given a cloze statement that queries the model for a missing token, knowledgeable LMs rank ground truth tokens high and other tokens lower.

"The color of the sky is [MASK]."

<table>
<thead>
<tr>
<th>Probability Scores</th>
<th>Rankings</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bob:</strong> 0.003%</td>
<td><strong>blue:</strong></td>
</tr>
<tr>
<td><strong>blue:</strong> 49%</td>
<td>red: 12%</td>
</tr>
<tr>
<td>red: 12%</td>
<td>grey: 0.01%</td>
</tr>
<tr>
<td>grass: 0.01%</td>
<td>pear: 3%</td>
</tr>
<tr>
<td>grey: 3%</td>
<td>. . .</td>
</tr>
<tr>
<td>pear: 0.09%</td>
<td>. . .</td>
</tr>
</tbody>
</table>

**P@k: precision at k**

"Does ground truth exist in the top k ranks?"
Evaluation of LM via LAMA

Given a cloze statement that queries the model for a missing token, knowledgeable LMs rank ground truth tokens high and other tokens lower

“The color of the sky is [MASK].”
Evaluation of LM via LAMA

Given a cloze statement that queries the model for a missing token, knowledgeable LMs rank ground truth tokens high and other tokens lower.

“The color of the sky is [MASK].”

<table>
<thead>
<tr>
<th>Probability Scores</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob: 0.1%</td>
</tr>
<tr>
<td><strong>blue</strong>: 15%</td>
</tr>
<tr>
<td>red: 18%</td>
</tr>
<tr>
<td>grass: 0.04%</td>
</tr>
<tr>
<td>grey: 30%</td>
</tr>
<tr>
<td>pear: 0.003%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rankings</th>
</tr>
</thead>
<tbody>
<tr>
<td>grey</td>
</tr>
<tr>
<td>red</td>
</tr>
<tr>
<td><strong>blue</strong></td>
</tr>
<tr>
<td>Bob</td>
</tr>
<tr>
<td>grass</td>
</tr>
<tr>
<td>pear</td>
</tr>
</tbody>
</table>
Evaluation of LM via LAMA

Given a cloze statement that queries the model for a missing token, knowledgeable LMs rank ground truth tokens high and other tokens lower.

“The color of the sky is [MASK].”

Language Model #2

- Bob: 0.1%
- blue: 15%
- red: 18%
- grass: 0.04%
- grey: 30%
- pear: 0.003%

P@1: precision at rank 1

<table>
<thead>
<tr>
<th>grey</th>
<th>red</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>.</td>
<td>.</td>
<td>Bob</td>
</tr>
<tr>
<td>.</td>
<td>grass</td>
<td></td>
</tr>
<tr>
<td>pear</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Evaluation of LM via LAMA

Given a cloze statement that queries the model for a missing token, knowledgeable LMs rank ground truth tokens high and other tokens lower

“*The color of the sky is [MASK].”*

**Language Model #2**

<table>
<thead>
<tr>
<th>Probability Scores</th>
<th>Rankings</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bob:</strong> 0.1%</td>
<td>grey</td>
</tr>
<tr>
<td><strong>blue:</strong> 15%</td>
<td>red</td>
</tr>
<tr>
<td><strong>red:</strong> 18%</td>
<td>.</td>
</tr>
<tr>
<td><strong>grass:</strong> 0.04%</td>
<td>.</td>
</tr>
<tr>
<td><strong>grey:</strong> 30%</td>
<td>.</td>
</tr>
<tr>
<td>pear: 0.003%</td>
<td>.</td>
</tr>
</tbody>
</table>

**P@1:** precision at rank 1

No ground truth. **P@1 = 0**
Evaluation of LM via LAMA

Given a cloze statement that queries the model for a missing token, knowledgeable LMs rank ground truth tokens high and other tokens lower.

“The color of the sky is [MASK].”

Language Model #2

<table>
<thead>
<tr>
<th>Probability Scores</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob: 0.1%</td>
</tr>
<tr>
<td>blue: 15%</td>
</tr>
<tr>
<td>red: 18%</td>
</tr>
<tr>
<td>grass: 0.04%</td>
</tr>
<tr>
<td>grey: 30%</td>
</tr>
<tr>
<td>.</td>
</tr>
<tr>
<td>.</td>
</tr>
<tr>
<td>pear: 0.003%</td>
</tr>
</tbody>
</table>

“grey red blue”

rankings

P@3: precision at rank 3

Contains ground truth! P@3 = 1
Architecture of the LAMA probe
Architecture of the LAMA probe

Step 1: Compile knowledge sources

Knowledge Sources
Architecture of the LAMA probe

Step 2: Formulate facts into triplets or question-answer pairs

Knowledge Sources

Fact #1
Fact #2
Fact #3

Facts

Fact #1
Fact #2
Fact #3

(subject, relation, object)

(question, answer)
Architecture of the LAMA probe

Step 2: Formulate facts into triplets or question-answer pairs

Knowledge Sources

FACT #1
FACT #2
FACT #3

Facts

FACT #1
FACT #2
FACT #3

(Mozart, Born-in, Austria)

(Who wrote Hamlet, Shakespeare)
Architecture of the LAMA probe

Step 3: Create cloze statements, either manually or via templates

Knowledge Sources

<table>
<thead>
<tr>
<th>Fact #1</th>
<th>Fact #2</th>
<th>Fact #3</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Fact #1</th>
<th>Fact #2</th>
<th>Fact #3</th>
</tr>
</thead>
</table>

(subject, relation, object)

(question, answer)

Cloze Statement Maker
Architecture of the LAMA probe

Step 3: Create cloze statements, either manually or via templates

Knowledge Sources  
Fact #1  
Fact #2  
Fact #3  

Facts  
(Mozart, BORN-IN, Austria)  

Cloze Statement Maker  
(Template)  

Cloze Statements
Architecture of the LAMA probe

Step 3: Create cloze statements, either manually or via templates

Knowledge Sources → Facts → Cloze Statement Maker (Template) → Cloze Statements

FACT #1
FACT #2
FACT #3

(Mozart, BORN-IN, Austria)

Cloze Statement Maker (Template)

“[ SUBJECT ] was born in [MASK]”

manually crafted template for “BORN-IN” relation
Architecture of the LAMA probe

Step 3: Create cloze statements, either manually or via templates

Knowledge Sources

Facts

- Mozart
- BORN-IN
- Austria

Cloze Statement Maker (Template)

"[SUBJECT] was born in [MASK]"

manually crafted template for “BORN-IN” relation
Architecture of the LAMA probe

Step 3: Create cloze statements, either manually or via templates

Knowledge Sources

Facts

Cloze Statement Maker (Template)

“Mozart was born in [MASK]”

manually crafted template for “BORN-IN” relation
Architecture of the LAMA probe

Step 3: Create cloze statements, either manually or via templates

Knowledge Sources

Facts

Cloze Statements

Fact #1
Fact #2
Fact #3

Fact #1
Fact #2
Fact #3

(Cloze Statement (Manual))

(Who wrote Hamlet, Shakespeare)
Architecture of the LAMA probe

Step 3: Create cloze statements, either manually or via templates

Cloze Statement (Manual)

(Who wrote Hamlet, Shakespeare)
Architecture of the LAMA probe

Step 3: Create cloze statements, either manually or via templates

Knowledge Sources

Facts

Cloze Statements

Fact #1
Fact #2
Fact #3

Who wrote Hamlet, Shakespeare

Cloze Statement (Manual)

“The writer of Hamlet is [MASK].”

human-generated statement
LAMA’s Knowledge Sources: Google-RE

- Manually extracted facts from Wikipedia
- Only consider 3 kinds of relations: place of birth, date of birth, place of death
LAMA’s Knowledge Sources: Google-RE

- Manually extracted facts from Wikipedia
- Only consider 3 kinds of relations: place of birth, date of birth, place of death

Original Fact: (T.S. Eliot, birth-place, St. Louis)
LAMA’s Knowledge Sources: Google-RE

- Manually extracted facts from Wikipedia
- Only consider 3 kinds of relations: place of birth, date of birth, place of death

Original Fact: (T.S. Eliot, birth-place, St. Louis)

Question: “T.S. Eliot was born in [MASK]”

Answer: St. Louis
LAMA’s Knowledge Sources: T-REx

- Automatically extracted facts from Wikipedia (may have some errors)
- For multiple right answers: throw away all but one
LAMA's Knowledge Sources: **T-REx**

- Automatically extracted facts from Wikipedia (may have some errors)
- For multiple right answers: throw away all but one
LAMA’s Knowledge Sources: **T-REx**

- Automatically extracted facts from Wikipedia (may have some errors)
- **For multiple right answers:** throw away all but one

Original Fact: (Francesco Conti, born-in, [Florence, Italy])
LAMA’s Knowledge Sources: **T-REx**

- Automatically extracted facts from Wikipedia (may have some errors)
- For multiple right answers: throw away all but one

Original Fact: (Francesco Conti, born-in, [Florence, It, v])

Question: “Francesco Conti was born in [MASK]”

Answer: Florence
LAMA’s Knowledge Sources: **ConceptNet**

- For each ConceptNet triple, find the relevant *Open Mind Common Sense (OMCS)* sentences and mask the object

**ConceptNet Triple**

(ravens, CapableOf, fly)
LAMA’s Knowledge Sources: ConceptNet

- For each ConceptNet triple, find the relevant Open Mind Common Sense (OMCS) sentences and mask the object

**ConceptNet Triple**: (ravens, CapableOf, fly)

**OMCS Sentence**: “Ravens can fly.”
LAMA's Knowledge Sources: **ConceptNet**

- For each ConceptNet triple, find the relevant *Open Mind Common Sense (OMCS)* sentences and mask the object

**ConceptNet Triple**
(ravens, CapableOf, fly)

**Question**
“Ravens can [MASK]”

**Answer**
fly
LAMA’s Knowledge Sources: **SQuAD**

- **Question-answer dataset:** pick only context-insensitive questions with single-token answers
- Originally created via Wikipedia
LAMA’s Knowledge Sources: **SQuAD**

- **Question-answer dataset:** pick only context-insensitive questions with single-token answers
- Originally created via Wikipedia

SQuAD Question-Answer Pair

(“Who developed the theory of relativity?”, Einstein)
LAMA’s Knowledge Sources: **SQuAD**

- **Question-answer dataset:** pick only context-insensitive questions with single-token answers
- Originally created via Wikipedia

**SQuAD Question-Answer Pair**

- **Question:** “The theory of relativity was developed by **[MASK]**”
- **Answer:** Einstein
# Dataset Statistics

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Facts</th>
<th># of Relations</th>
<th># Tokens in Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google-RE</td>
<td>5.5k</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>T-REx</td>
<td>34k</td>
<td>41</td>
<td>1</td>
</tr>
<tr>
<td>ConceptNet</td>
<td>11.4k</td>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td>SQuAD</td>
<td>300</td>
<td>-</td>
<td>1</td>
</tr>
</tbody>
</table>
### Dataset Statistics

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Facts</th>
<th># of Relations</th>
<th># Tokens in Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google-RE</td>
<td>5.5k</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>T-REx</td>
<td>34k</td>
<td>41</td>
<td>1</td>
</tr>
<tr>
<td>ConceptNet</td>
<td>11.4k</td>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td>SQuAD</td>
<td>300</td>
<td>-</td>
<td>1</td>
</tr>
</tbody>
</table>

**Note:** all ground truth answers are single-token!
Baselines
Baselines

- **Freq**: ranks candidates by frequency of appearance as objects for a subject-relation pair
  - Analogous to majority classifier
Baselines

- **Freq**: ranks candidates by frequency of appearance as objects for a subject-relation pair
  - Analogous to majority classifier

- **Pretrained models**
  - RE (Sorokin and Gurevych, 2017): extracts relation triples from sentence
Baseline

- **Freq**: ranks candidates by frequency of appearance as objects for a subject-relation pair
  - Analogous to majority classifier

- **Pretrained models**
  - **RE** (Sorokin and Gurevych, 2017): extracts relation triples from sentence
    - **RE**\(_n\): uses exact string matching for entity linking
      - **RE**\(_n\) has to find the subject/object entities itself
Baselines

● **Freq**: ranks candidates by frequency of appearance as objects for a subject-relation pair
  ○ Analogous to majority classifier

● **Pretrained models**
  ○ **RE** (Sorokin and Gurevych, 2017): extracts relation triples from sentence
    ■ **RE**<sub>n</sub>: uses exact string matching for entity linking
      ● **RE**<sub>n</sub> has to find the subject/object entities itself
    ■ **RE**<sub>o</sub>: uses oracle for entity linking
      ● As long as **RE**<sub>o</sub> gets the right relation type, it gets the answer for free
Baselines

- **Freq**: ranks candidates by frequency of appearance as objects for a subject-relation pair
  - Analogous to majority classifier

- **Pretrained models**
  - **RE** (Sorokin and Gurevych, 2017): extracts relation triples from sentence
    - **RE**\textsubscript{n}: uses exact string matching for entity linking
    - **RE**\textsubscript{o}: uses oracle for entity linking
  - **DRQA** (Chen et al., 2017): uses TF/IDF to retrieve relevant arguments from a set of documents, then extracts answers from the best $k$ articles
# Pre-trained language models

<table>
<thead>
<tr>
<th>Model</th>
<th>Base Model</th>
<th>Training Corpus</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>fairseq-fconv  (Fs)</td>
<td>ConvNet</td>
<td>WikiText-103 corpus</td>
<td>324M</td>
</tr>
<tr>
<td>Transformer-XL large (Txl)</td>
<td>Transformer</td>
<td>WikiText-103 corpus</td>
<td>257M</td>
</tr>
<tr>
<td>ELMo</td>
<td>ELMo (Eb)</td>
<td>BiLSTM</td>
<td>Google Billion Word</td>
</tr>
<tr>
<td></td>
<td>ELMo 5.5B (E5B)</td>
<td></td>
<td>Wikipedia + WMT 2008-2012</td>
</tr>
<tr>
<td>BERT</td>
<td>BERT-base (Bb)</td>
<td>Transformer</td>
<td>Wikipedia (en) &amp; BookCorpus</td>
</tr>
<tr>
<td></td>
<td>BERT-large (Bl)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Results:** both BERT models outperform other models on Google-RE

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Relation</th>
<th>Baselines</th>
<th></th>
<th>KB</th>
<th></th>
<th>LM</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Freq</td>
<td>DrQA</td>
<td>RE$_n$</td>
<td>RE$_o$</td>
<td>Fs</td>
<td>Txl</td>
</tr>
<tr>
<td>Google-RE</td>
<td>birth-place</td>
<td>4.6</td>
<td>-</td>
<td>3.5</td>
<td>13.8</td>
<td>4.4</td>
<td>2.7</td>
</tr>
<tr>
<td></td>
<td>birth-date</td>
<td>1.9</td>
<td>-</td>
<td>0.0</td>
<td><strong>1.9</strong></td>
<td>0.3</td>
<td>1.1</td>
</tr>
<tr>
<td></td>
<td>death-place</td>
<td>6.8</td>
<td>-</td>
<td>0.1</td>
<td>7.2</td>
<td>3.0</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>4.4</td>
<td>-</td>
<td>1.2</td>
<td>7.6</td>
<td>2.6</td>
<td>1.6</td>
</tr>
<tr>
<td>T-REx</td>
<td>1-1</td>
<td>1.78</td>
<td>-</td>
<td>0.6</td>
<td>10.0</td>
<td>17.0</td>
<td>36.5</td>
</tr>
<tr>
<td></td>
<td>N-1</td>
<td>23.85</td>
<td>-</td>
<td>5.4</td>
<td><strong>33.8</strong></td>
<td>6.1</td>
<td>18.0</td>
</tr>
<tr>
<td></td>
<td>N-M</td>
<td>21.95</td>
<td>-</td>
<td>7.7</td>
<td><strong>36.7</strong></td>
<td>12.0</td>
<td>16.5</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>22.03</td>
<td>-</td>
<td>6.1</td>
<td><strong>33.8</strong></td>
<td>8.9</td>
<td>18.3</td>
</tr>
<tr>
<td>ConceptNet</td>
<td>Total</td>
<td>4.8</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>3.6</td>
<td>5.7</td>
</tr>
<tr>
<td>SQuAD</td>
<td>Total</td>
<td>-</td>
<td><strong>37.5</strong></td>
<td>-</td>
<td>-</td>
<td>3.6</td>
<td>3.9</td>
</tr>
</tbody>
</table>

**P@1:** precision at rank 1

Table from Petroni et al., 2019
**Results:** BERT models does better on T-REx when there’s only one correct answer...

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Relation</th>
<th>Baselines</th>
<th>KB</th>
<th>LM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Freq</td>
<td>DrQA</td>
<td>RE&lt;sub&gt;n&lt;/sub&gt;</td>
</tr>
<tr>
<td>Google-RE</td>
<td>birth-place</td>
<td>4.6</td>
<td>-</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td>birth-date</td>
<td>1.9</td>
<td>-</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>death-place</td>
<td>6.8</td>
<td>-</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>4.4</td>
<td>-</td>
<td>1.2</td>
</tr>
<tr>
<td>T-REx</td>
<td>N-1</td>
<td>23.85</td>
<td>-</td>
<td>5.4</td>
</tr>
<tr>
<td></td>
<td>N-M</td>
<td>21.95</td>
<td>-</td>
<td>7.7</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>22.03</td>
<td>-</td>
<td>6.1</td>
</tr>
<tr>
<td></td>
<td>ConceptNet</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>4.8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>SQuAD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

**P@1:** precision at rank 1
Results: ...but when there are multiple answers, $\text{RE}_o$ is best

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Relation</th>
<th>Baselines</th>
<th></th>
<th>KB</th>
<th></th>
<th>LM</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Freq</td>
<td>DrQA</td>
<td>$\text{RE}_m$</td>
<td>$\text{RE}_o$</td>
<td>Fs</td>
<td>Txl</td>
<td>Eb</td>
<td>E5B</td>
<td>Bb</td>
</tr>
<tr>
<td>Google-RE</td>
<td>birth-place</td>
<td>4.6</td>
<td>-</td>
<td>3.5</td>
<td>13.8</td>
<td>4.4</td>
<td>2.7</td>
<td>5.5</td>
<td>7.5</td>
<td>14.9</td>
</tr>
<tr>
<td></td>
<td>birth-date</td>
<td>1.9</td>
<td>-</td>
<td>0.0</td>
<td>1.9</td>
<td>0.3</td>
<td>1.1</td>
<td>0.1</td>
<td>0.1</td>
<td>1.5</td>
</tr>
<tr>
<td></td>
<td>death-place</td>
<td>6.8</td>
<td>-</td>
<td>0.1</td>
<td>7.2</td>
<td>3.0</td>
<td>0.9</td>
<td>0.3</td>
<td>1.3</td>
<td>13.1</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>4.4</td>
<td>-</td>
<td>1.2</td>
<td>7.6</td>
<td>2.6</td>
<td>1.6</td>
<td>2.0</td>
<td>3.0</td>
<td>9.8</td>
</tr>
<tr>
<td>T-REx</td>
<td>1-1</td>
<td>1.78</td>
<td>-</td>
<td>0.6</td>
<td>10.0</td>
<td>17.0</td>
<td>36.5</td>
<td>10.1</td>
<td>13.1</td>
<td>68.0</td>
</tr>
<tr>
<td></td>
<td>$N$-1</td>
<td>23.85</td>
<td>-</td>
<td>5.4</td>
<td>33.8</td>
<td>6.1</td>
<td>18.0</td>
<td>3.6</td>
<td>6.5</td>
<td>32.4</td>
</tr>
<tr>
<td></td>
<td>$N$-$M$</td>
<td>21.95</td>
<td>-</td>
<td>7.7</td>
<td>36.7</td>
<td>12.0</td>
<td>16.5</td>
<td>5.7</td>
<td>7.4</td>
<td>24.7</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>22.03</td>
<td>-</td>
<td>6.1</td>
<td>33.8</td>
<td>8.9</td>
<td>18.3</td>
<td>4.7</td>
<td>7.1</td>
<td>31.1</td>
</tr>
<tr>
<td>ConceptNet</td>
<td>Total</td>
<td>4.8</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>3.6</td>
<td>5.7</td>
<td>6.1</td>
<td>6.2</td>
<td>15.6</td>
</tr>
<tr>
<td>SQuAD</td>
<td>Total</td>
<td>-</td>
<td>37.5</td>
<td>-</td>
<td>-</td>
<td>3.6</td>
<td>3.9</td>
<td>1.6</td>
<td>4.3</td>
<td>14.1</td>
</tr>
</tbody>
</table>

$P@1$: precision at rank 1
**Results:** BERT models outperform other LMs on T-REx

Performance on T-REx

BERT models perform the best by a large margin
**Results:** BERT models outperform other LMs on T-REx

Next best is Transformer-XL
**Results**: BERT models outperform other LMs on T-REx

Performance on T-REx

The worst performers are ELMo and fairseq-conv

Figure from Petroni et al., 2019
**Results:** BERT models are less sensitive to query variations than other LMs
Results: BERT models are less sensitive to query variations than other LMs

Figure from Petroni et al., 2019
**Results:** What factors correlate with better performance for BERT on T-REx?

![Pearson Correlation Coefficients for P@1 (BERT-large, T-REx)]

<table>
<thead>
<tr>
<th>Factor</th>
<th>Correlation Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td># of subject mentions in training corpus</td>
<td>-0.05</td>
</tr>
<tr>
<td># of object mentions in training corpus</td>
<td>0.2</td>
</tr>
<tr>
<td>logprob of first prediction</td>
<td>0.42</td>
</tr>
<tr>
<td>cosine similarity between subject + object vectors</td>
<td>0.31</td>
</tr>
<tr>
<td># tokens in subject (standard tokenizer)</td>
<td>0.12</td>
</tr>
<tr>
<td># tokens in subject (BERT Wordpiece tokenizer)</td>
<td>0.035</td>
</tr>
</tbody>
</table>

Figure from Petroni et al., 2019
Conclusion

- BERT-large recalls knowledge better than its competitors, and competitively with non-neural/supervised alternatives

- BERT-large is competitive with a RE knowledge base that was trained on the “best possible” data and used the entity-linking oracle

- Dealing with variance in performance in response to different natural language templates is a challenge
Question 1

Describe what the LAMA Probe is in (Petroni et al., 2019) - How do they probe different knowledge sources (Wikidata triples, ConceptNet, QA pairs)?

- A collection of knowledge sources either for relation extraction or QA
- Convert facts to cloze statements (either manually or using templates)
- Ask LM to rank candidate vocabulary and see if ground truth is in top $k$ rank

Can you think of any drawbacks of the probes?

- Answers must be single-token
- Relies on manual templates
- Questions are constrained to very specific and simple types of questions
Is there a better alternative to manual templates?
Is there a better alternative to manual templates?

- **Prompt mining** ([Jiang et al., 2020](#)): automated prompt extraction via dependency parsing, simple heuristics, or automated paraphrasing.
Is there a better alternative to manual templates?

- **Prompt mining (Jiang et al., 2020):** automated prompt extraction via dependency parsing, simple heuristics, or automated paraphrasing

- **AutoPrompt (Shin et al., 2020):** prompt is constructed by adding tokens found via gradient-guided search to a simple prompt

```
Original Input \( x_{\text{inp}} \)
a real joy.

Trigger Tokens \( x_{\text{trig}} \)
atmosphere, alot, dialogue, Clone...

Template \( \lambda (x_{\text{inp}}, x_{\text{trig}}) \)
{sentence}[T][T][T][T][T][P].

AutoPrompt \( x_{\text{prompt}} \)
a real joy. atmosphere alot dialogue Clone totally [MASK].
```
Is there a better alternative to manual templates?

- **Prompt mining** ([Jiang et al., 2020](#)): automated prompt extraction via dependency parsing, simple heuristics, or automated paraphrasing

- **AutoPrompt** ([Shin et al., 2020](#)): prompt is constructed by adding tokens found via gradient-guided search to a simple prompt

- **OptiPrompt** ([Zhong et al., 2021](#)): directly optimize prompt in embedding space, rather than in discrete space
  - Similar to prompt tuning
How Much Knowledge Can You Pack Into the Parameters of a Language Model? (Roberts et al., 2020)
Motivation

- Petroni et al., 2019 measures knowledge in a model with its **pre-training objective** with a **synthetic task**
- This work measures **transfer learning performance** on knowledge on **question answering tasks with a closed-book approach**
To solve open-domain QA: Two approaches

- Open-book QA: questions and external resources are given

- Closed-book QA: questions
Experimental Setup - Datasets

- **Natural Questions:**
  - Web queries each with an oracle Wikipedia article
  - Rich annotation, different answer types (yes/no, unanswerable, multiple answers, short answer and long answer)
  - E.g., *Who are the members of the Beatles?*

- **WebQuestions**
  - Web queries
  - E.g., *Which college did Obama go to?*

- **TriviaQA**
  - Questions from quiz league websites, each with web pages that might contain answers
  - E.g., *Who won the Nobel Peace Prize in 2009?*
Experimental Setup - Models

- **Pre-training resources**
  - T5 v1.0: trained with the unsupervised “span corruption” task on C4 as well as *supervised translation, summarization, classification, and reading comprehension tasks*
  - T5 v1.1: trained only with the C4

- **Model size**
  - Base (220 million parameters)
  - Large (770 million)
  - 3B (3 billion)
  - 11B (11 billion)

- **Additional pre-training**
  - Salient Span Masking ([Guu et al. 2020](#)), mask salient spans (named entities & dates)
  - Continue pre-training the T5 for 100k steps

*Henri Hutin* invented Brie cheese while living in North of Meuse, France
## Results

<table>
<thead>
<tr>
<th>SOTA Retrieval-based Models (can access external documents)</th>
<th>NQ</th>
<th>WQ</th>
<th>TQA dev</th>
<th>TQA test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chen et al. (2017)</td>
<td>–</td>
<td>20.7</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Lee et al. (2019)</td>
<td>33.3</td>
<td>36.4</td>
<td>47.1</td>
<td>–</td>
</tr>
<tr>
<td>Min et al. (2019a)</td>
<td>28.1</td>
<td>–</td>
<td>50.9</td>
<td>–</td>
</tr>
<tr>
<td>Min et al. (2019b)</td>
<td>31.8</td>
<td>31.6</td>
<td>55.4</td>
<td>–</td>
</tr>
<tr>
<td>Asai et al. (2019)</td>
<td>32.6</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Ling et al. (2020)</td>
<td>–</td>
<td>–</td>
<td>35.7</td>
<td>–</td>
</tr>
<tr>
<td>Guu et al. (2020)</td>
<td>40.4</td>
<td>40.7</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Févry et al. (2020)</td>
<td>–</td>
<td>–</td>
<td>43.2</td>
<td>53.4</td>
</tr>
<tr>
<td>Karpukhin et al. (2020)</td>
<td><strong>41.5</strong></td>
<td><strong>42.4</strong></td>
<td><strong>57.9</strong></td>
<td>–</td>
</tr>
</tbody>
</table>

Metric: Exact Match
### Results: SSM clearly leads to improved performance

Metric: Exact Match

<table>
<thead>
<tr>
<th>Model/Config</th>
<th>NQ</th>
<th>WQ</th>
<th>TQA dev</th>
<th>TQA test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chen et al. (2017)</td>
<td>–</td>
<td>20.7</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Lee et al. (2019)</td>
<td>33.3</td>
<td>36.4</td>
<td>47.1</td>
<td>–</td>
</tr>
<tr>
<td>Min et al. (2019a)</td>
<td>28.1</td>
<td>–</td>
<td>50.9</td>
<td>–</td>
</tr>
<tr>
<td>Min et al. (2019b)</td>
<td>31.8</td>
<td>31.6</td>
<td>55.4</td>
<td>–</td>
</tr>
<tr>
<td>Asai et al. (2019)</td>
<td>32.6</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Ling et al. (2020)</td>
<td>–</td>
<td>–</td>
<td>35.7</td>
<td>–</td>
</tr>
<tr>
<td>Guu et al. (2020)</td>
<td>40.4</td>
<td>40.7</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Févry et al. (2020)</td>
<td>–</td>
<td>–</td>
<td>43.2</td>
<td>53.4</td>
</tr>
<tr>
<td>Karpukhin et al. (2020)</td>
<td><strong>41.5</strong></td>
<td><strong>42.4</strong></td>
<td><strong>57.9</strong></td>
<td>–</td>
</tr>
</tbody>
</table>

**Non-SSM Retrieval-based Models**

<table>
<thead>
<tr>
<th>Model/Config</th>
<th>NQ</th>
<th>WQ</th>
<th>TQA dev</th>
<th>TQA test</th>
</tr>
</thead>
<tbody>
<tr>
<td>T5-Base</td>
<td>25.9</td>
<td>27.9</td>
<td>23.8</td>
<td>29.1</td>
</tr>
<tr>
<td>T5-Large</td>
<td>28.5</td>
<td>30.6</td>
<td>28.7</td>
<td>35.9</td>
</tr>
<tr>
<td>T5-3B</td>
<td>30.4</td>
<td>33.6</td>
<td>35.1</td>
<td>43.4</td>
</tr>
<tr>
<td>T5-11B</td>
<td>32.6</td>
<td>37.2</td>
<td>42.3</td>
<td>50.1</td>
</tr>
<tr>
<td>T5-11B + SSM</td>
<td>34.8</td>
<td>40.8</td>
<td>51.0</td>
<td>60.5</td>
</tr>
</tbody>
</table>

**SSM Closed-Book QA models with fine-tuning (relies only on internal parameters)**

<table>
<thead>
<tr>
<th>Model/Config</th>
<th>NQ</th>
<th>WQ</th>
<th>TQA dev</th>
<th>TQA test</th>
</tr>
</thead>
<tbody>
<tr>
<td>T5.1.1-Base</td>
<td>25.7</td>
<td>28.2</td>
<td>24.2</td>
<td>30.6</td>
</tr>
<tr>
<td>T5.1.1-Large</td>
<td>27.3</td>
<td>29.5</td>
<td>28.5</td>
<td>37.2</td>
</tr>
<tr>
<td>T5.1.1-XL</td>
<td>29.5</td>
<td>32.4</td>
<td>36.0</td>
<td>45.1</td>
</tr>
<tr>
<td>T5.1.1-XXL</td>
<td>32.8</td>
<td>35.6</td>
<td>42.9</td>
<td>52.5</td>
</tr>
<tr>
<td>T5.1.1-XXL + SSM</td>
<td>35.2</td>
<td><strong>42.8</strong></td>
<td>51.9</td>
<td><strong>61.6</strong></td>
</tr>
</tbody>
</table>

**Closed-Book QA model without fine-tuning**

<table>
<thead>
<tr>
<th>Model/Config</th>
<th>NQ</th>
<th>WQ</th>
<th>TQA dev</th>
<th>TQA test</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPT-3 few-shot</td>
<td>29.9</td>
<td>41.5</td>
<td><strong>71.2</strong></td>
<td>–</td>
</tr>
<tr>
<td>SOTA</td>
<td>51.4</td>
<td>–</td>
<td>80.1</td>
<td>–</td>
</tr>
</tbody>
</table>

**SOTA Retrieval-based Models (can access external documents)**
# Results: Scale correlates with performance

Metric: Exact Match

<table>
<thead>
<tr>
<th></th>
<th>NQ</th>
<th>WQ</th>
<th>TQA dev</th>
<th>TQA test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chen et al. (2017)</td>
<td>–</td>
<td>20.7</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Lee et al. (2019)</td>
<td>33.3</td>
<td>36.4</td>
<td>47.1</td>
<td>–</td>
</tr>
<tr>
<td>Min et al. (2019a)</td>
<td>28.1</td>
<td>–</td>
<td>50.9</td>
<td>–</td>
</tr>
<tr>
<td>Min et al. (2019b)</td>
<td>31.8</td>
<td>31.6</td>
<td>55.4</td>
<td>–</td>
</tr>
<tr>
<td>Asai et al. (2019)</td>
<td>32.6</td>
<td>–</td>
<td>35.7</td>
<td>–</td>
</tr>
<tr>
<td>Ling et al. (2020)</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Guu et al. (2020)</td>
<td>40.4</td>
<td>40.7</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Févry et al. (2020)</td>
<td>–</td>
<td>–</td>
<td>43.2</td>
<td>53.4</td>
</tr>
<tr>
<td>Karpukhin et al. (2020)</td>
<td><strong>41.5</strong></td>
<td><strong>42.4</strong></td>
<td><strong>57.9</strong></td>
<td>–</td>
</tr>
</tbody>
</table>

**Increasing size**

- T5-Base
- T5-Large
- T5-3B
- T5-11B
- T5-11B + SSM

**Increasing performance**

- T5.1.1-Base
- T5.1.1-Large
- T5.1.1-XL
- T5.1.1-XXL
- T5.1.1-XXL + SSM

**Table Values:**
- NQ: Normal Question
- WQ: Weak Question
- TQA dev: Test/dev
- TQA test: Test/test

**Table Note:**
- Bold values indicate the highest performance for each model configuration.
Additional Evaluation on NQ

- Previous results adopt evaluation used in previous work
  - Long answers and unanswerable questions are not considered
  - Only output single answer
  - Only trained with the first answer if a question has multiple answers
  - Answers with longer than 5 tokens are excluded
  - Answers are normalized (lowercased, strip of articles, punctuation etc.)

- **Leaderboard evaluation**
  - Long answers and unanswerable questions are not considered
  - Models are trained to predict all ground-truth answers
  - Only considered correct if it predicts *all answers* correctly

T5-11B + SSM achieves a recall of **36.2** on the validation set, which lags behind the state-of-the-art score of **51.9** from Pan et al. (2019) at the time.
Human Evaluation + Qualitative Error Analysis

- Exact Match is a very harsh metric → potentially lots of **false negatives**
- Use human evaluation to see what percent of predicted negatives area are actually true negatives

**38% of T5’s “incorrect” predictions are actually correct!**

<table>
<thead>
<tr>
<th>Category</th>
<th>Percentage</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>True Negative</td>
<td>62.0%</td>
<td>what does the ghost of christmas present sprinkle from his torch</td>
</tr>
<tr>
<td></td>
<td></td>
<td>who plays red on orange is new black</td>
</tr>
<tr>
<td></td>
<td></td>
<td>where does the us launch space shuttles from who is the secretary of</td>
</tr>
<tr>
<td></td>
<td></td>
<td>state for northern ireland</td>
</tr>
<tr>
<td></td>
<td></td>
<td>little warmth, warmth</td>
</tr>
<tr>
<td></td>
<td></td>
<td>kate mulgrew</td>
</tr>
<tr>
<td></td>
<td></td>
<td>florida</td>
</tr>
<tr>
<td></td>
<td></td>
<td>karen bradley</td>
</tr>
<tr>
<td>Phrasing Mismatch</td>
<td>13.3%</td>
<td>confetti</td>
</tr>
<tr>
<td>Incomplete Annotation</td>
<td>13.3%</td>
<td>katherine kiernan</td>
</tr>
<tr>
<td></td>
<td></td>
<td>maria mulgrew</td>
</tr>
<tr>
<td></td>
<td></td>
<td>kennedy lc39b</td>
</tr>
<tr>
<td>Unanswerable</td>
<td>11.3%</td>
<td>james brokenshire</td>
</tr>
</tbody>
</table>
Conclusion

- Large language models pretrained on unstructured text perform competitively on open-domain QA, even compared to competitors with access to external knowledge.

- Scale is critical to performance – needed largest (11B) model to compete on par with SOTA.

- Using LMs as knowledge bases suffers from lack of interpretability, and LMs are prone to hallucinating “realistic” answers.
Question 2

Compared to (Petroni et al., 2019), can you state the key differences in (Roberts et al., 2021)?

- (Roberts et al., 2021) handles harder questions that may require multiple tokens. LAMA uses specific/easier types of questions with single-token answers.
- Since T5 can’t do zero-shot well, (Roberts et al., 2021) fine-tunes the model for QA tasks and compares against other retrieval-based fine-tuned models. LAMA does not fine-tune the models.

Do you think the accuracy of answering these open-domain questions reflects how much knowledge is already encoded in LLMs?

- To some extent. (Roberts et al., 2021) fine-tunes the model on the question-answer datasets, so it could be argued that it does not 100% accurately test how much knowledge is encoded in the pre-training stage.
## Comparison of the Two Works

<table>
<thead>
<tr>
<th></th>
<th>Petroni et al., 2019</th>
<th>Roberts et al., 2020</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Objective</strong></td>
<td>MLM</td>
<td>seq2seq</td>
</tr>
<tr>
<td><strong>Format</strong></td>
<td>filling in the blank</td>
<td>generation</td>
</tr>
<tr>
<td><strong>Finetune?</strong></td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td><strong>Answer length</strong></td>
<td>1</td>
<td>&gt; 1</td>
</tr>
</tbody>
</table>
How much does train-test overlap affect performance?

- Many of the knowledge sources we’ve discussed were extracted from Wikipedia.

- However, pre-training corpora for language models almost always contain data from Wikipedia...

- How much of the amazing knowledge retrieval is due to **train-test overlap** in the knowledge probing benchmarks?
Train-test overlap is responsible for LM’s ability to do knowledge retrieval! (Lewis et al., 2020)

<table>
<thead>
<tr>
<th>Model</th>
<th>Open Natural Questions</th>
<th>TriviaQA</th>
<th>WebQuestions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total</td>
<td>Question Overlap</td>
<td>Answer Overlap Only</td>
</tr>
<tr>
<td>Open book</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RAG</td>
<td>44.5</td>
<td>70.7</td>
<td>34.9</td>
</tr>
<tr>
<td>DPR</td>
<td>41.3</td>
<td>69.4</td>
<td>34.6</td>
</tr>
<tr>
<td>FID</td>
<td>51.4</td>
<td>71.3</td>
<td>48.3</td>
</tr>
<tr>
<td>Closed book</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T5-11B+SSM</td>
<td>36.6</td>
<td>77.2</td>
<td>22.2</td>
</tr>
<tr>
<td>BART</td>
<td>26.5</td>
<td>67.6</td>
<td>10.2</td>
</tr>
<tr>
<td>Nearest Dense</td>
<td>26.7</td>
<td>69.4</td>
<td>7.0</td>
</tr>
<tr>
<td>Nearest Neighbor TF-IDF</td>
<td>22.2</td>
<td>56.8</td>
<td>4.1</td>
</tr>
</tbody>
</table>

When there is question overlap, both open and closed-book LMs perform well.
Train-test overlap is responsible for LM’s ability to do knowledge retrieval! (Lewis et al., 2020)

<table>
<thead>
<tr>
<th>Model</th>
<th>Open Natural Questions</th>
<th>TriviaQA</th>
<th>WebQuestions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total</td>
<td>Question Overlap</td>
<td>Answer Overlap Only</td>
</tr>
<tr>
<td>Open book</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RAG</td>
<td>44.5</td>
<td>70.7</td>
<td>34.9</td>
</tr>
<tr>
<td>DPR</td>
<td>41.3</td>
<td>69.4</td>
<td>26.6</td>
</tr>
<tr>
<td>FID</td>
<td>51.4</td>
<td>71.3</td>
<td>34.5</td>
</tr>
<tr>
<td>Closed book</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T5-11B+SSM</td>
<td>36.6</td>
<td>77.2</td>
<td>22.2</td>
</tr>
<tr>
<td>BART</td>
<td>26.5</td>
<td>67.6</td>
<td>10.2</td>
</tr>
<tr>
<td>Nearest Neighbor</td>
<td>26.7</td>
<td>69.4</td>
<td>7.0</td>
</tr>
<tr>
<td>TF-IDF</td>
<td>22.2</td>
<td>56.8</td>
<td>4.1</td>
</tr>
</tbody>
</table>

But with no question or answer overlap, performance drops sharply!
How to update knowledge in pre-trained models?
Edit What, Exactly?
Defining the problem

Who is the prime minister of the UK?
Edit What, Exactly?
Defining the problem

Who is the PM of the UK?

Who is the prime minister of the UK?

Edit example  Edit scope  In-scope

Slides from link
Edit What, Exactly?
Defining the problem

- Why is the sky blue?
- What club does Messi play for?
- Who is the prime minister of the UK?
- What continent is Everest on?

Edit example
Edit scope
In-scope
Out-of-scope

Slides from link
Edit What, Exactly?
Defining the problem

Where is Boris Johnson the PM?

Where did Boris Johnson go to university?

Who is the PM of the UK?

Who is the prime minister of the UK?

Why is the sky blue?

What club does Messi play for?

What continent is Everest on?

Who is the UK deputy PM?

Edit example  Edit scope  In-scope  Out-of-scope  Hard in/out-of-scope

Slides from link
Knowledge Neurons in Pretrained Transformers (Dai et al. 2021)
Knowledge Neurons

- What is a knowledge neuron
  - **Activations** after the first feed-forward layer

- Assumption
  - Knowledge neuron are associated with factual knowledge

- Implications
  - If we can identifying these neurons, we can alter them to edit (update/erase) knowledge.
  - No additional training is involved.
Suppressing or Amplifying Knowledge Neurons

**Suppressing** the neurons **hurt** performance and **amplifying** neurons **increase** performance by up to 30% on average.
Case Study - Updating Facts

- Update neuron values by subtracting the word embedding of the previous answer and adding the updated answer

<table>
<thead>
<tr>
<th>Metric</th>
<th>Knowledge Neurons</th>
<th>Random Neurons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change rate↑</td>
<td>48.5%</td>
<td>4.7%</td>
</tr>
<tr>
<td>Success rate↑</td>
<td>34.4%</td>
<td>0.0%</td>
</tr>
</tbody>
</table>

- They achieved a change rate and success rate that is better than random neurons.
- But is this good enough?
Fast Model Editing at Scale
(Mitchell et al. 2022)
Editing a Pre-trained Model with MEND
Editing a Pre-trained Model with MEND
The MEND network produces gradient updates for the pretrained model.

It's not the gradient of all the weights, it's a transformation of the gradient!
Editing a Pre-trained Model with MEND
Editing a Pre-trained Model with MEND

$x_e = \text{"Who is the prime minister of the UK?"}$

$y_e = \text{"Boris Johnson"}$

$x'_e = \text{"Who is the UK PM?"}$

The knowledge is updated!
Editing a Pre-trained Model with MEND

- Involves training
  - correctly updates the fact and the related facts
  - maintain answers to the irrelevant facts
- MEND network learns how to edit for one single fact change
Results

- FT: fine-tuning with updated facts
- FT + KL: fine-tuning with updated facts and locality loss

<table>
<thead>
<tr>
<th>Editor</th>
<th>ES ↑</th>
<th>acc. DD ↓</th>
<th>ES ↑</th>
<th>acc. DD ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>FT</td>
<td>0.58</td>
<td>&lt; 0.001</td>
<td>0.87</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>FT+KL</td>
<td>0.55</td>
<td>&lt; 0.001</td>
<td>0.85</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>MEND</td>
<td>0.88</td>
<td>0.001</td>
<td>0.89</td>
<td>&lt; 0.001</td>
</tr>
</tbody>
</table>

MEND shows the best **Edit success rate (ES)** and least interference to overall model perplexity or accuracy, i.e., **ppl. DD, acc.DD.**
## Comparison of the Two Works

<table>
<thead>
<tr>
<th></th>
<th>Knowledge Neurons</th>
<th>MEND</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Method</strong></td>
<td>Attribution-based</td>
<td>Learning-based</td>
</tr>
<tr>
<td><strong>Training?</strong></td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Restricted by</strong></td>
<td>Attribution algorithm</td>
<td>Need a lot of edits data</td>
</tr>
</tbody>
</table>
Conclusion
The world knowledge is constantly changing; for instance, the president was Donald Trump in 2020 and now is Joe Biden in 2022. However, LLMs are always trained on a static corpus of a fixed period.

1) Do you have any ideas about how to update and edit LLMs with real-world knowledge?

2) Do you think it is possible to decouple world knowledge and other knowledge encoded in LLMs?