Certainly! Here's a SPARQL query that you can use to retrieve the list of countries in the European Union (EU) from DBPedia:

```
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
PREFIX dbp: <http://dbpedia.org/property/>

SELECT ?country ?countryName WHERE {
  ?country dbp:name ?countryName .
  ?country a dbpedia-owl:Country .
}
ORDER BY ?countryName
```
is dbp:membership of

- dbr:Paris_Club
- dbr:Global_Health_Security_Initiative
- dbr:European_Economic_Area
- dbr:European_Go_Federation

dct:subject

- dbc:Italy
- dbc:G20_nations
- dbc:Southwestern_European_countries
- dbc:Italian-speaking_countries_and_territories
- dbc:Transcontinental_countries
- dbc:1861_establishments_in_Europe
- dbc:States_and_territories_established_in_1861
- dbc:Member_states_of_the_United_Nations
- dbc:Countries_in_Europe
- dbc:Member_states_of_NATO
- dbc:Member_states_of_the_Council_of_Europe
- dbc:Member_states_of_the_Union_for_the_Mediterranean
- dbc:Republics
- dbc:Southern_European_countries
- dbc:Member_states_of_the_European_Union
- dbc:G7_nations
Default Data Set Name (Graph IRI)
http://dbpedia.org

Query Text

```sparql
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX dbc: <http://dbpedia.org/resource/Category>

SELECT ?country ?countryName WHERE {
  ?country dbp:commonName ?countryName .
  ?country a dbo:Country .
} ORDER BY ?countryName
```

Results Format
HTML

Execute Query  Reset
<table>
<thead>
<tr>
<th>country</th>
<th>countryName</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://dbpedia.org/resource/Austria">http://dbpedia.org/resource/Austria</a></td>
<td>&quot;Austria&quot;@en</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Belgium">http://dbpedia.org/resource/Belgium</a></td>
<td>&quot;Belgium&quot;@en</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Bulgaria">http://dbpedia.org/resource/Bulgaria</a></td>
<td>&quot;Bulgaria&quot;@en</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Croatia">http://dbpedia.org/resource/Croatia</a></td>
<td>&quot;Croatia&quot;@en</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Cyprus">http://dbpedia.org/resource/Cyprus</a></td>
<td>&quot;Cyprus&quot;@en</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Estonia">http://dbpedia.org/resource/Estonia</a></td>
<td>&quot;Estonia&quot;@en</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Finland">http://dbpedia.org/resource/Finland</a></td>
<td>&quot;Finland&quot;@en</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/France">http://dbpedia.org/resource/France</a></td>
<td>&quot;France&quot;@en</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Germany">http://dbpedia.org/resource/Germany</a></td>
<td>&quot;Germany&quot;@en</td>
</tr>
</tbody>
</table>
Asking Questions in Natural Language

Pasquale Minervini
p.minervini@ed.ac.uk
Main References


2. Recent Advances in Text-to-SQL: A Survey of What We Have and What We Expect (COLING 2022) — https://aclanthology.org/2022.coling-1.190.pdf
Did Socrates use a laptop?
1. When did Aristotle live?
2. When was the laptop invented?
3. Is #2 before #1?

"The first laptop was... in 1980."

No

Was Aristotle alive when the laptop was invented?

Did Aristotle use a laptop?

---

Geva et al. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies
1. When did Aristotle live?
2. When was the laptop invented?
3. Is #2 before #1?

"The first laptop was... in 1980."

No

Was Aristotle alive when the laptop was invented?

Did Aristotle use a laptop?

Geva et al. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies
Did Socrates use a laptop?

Q1 Did Aristotle use a laptop?  
Q2 Was Aristotle alive when the laptop was invented?  

D 1. When did Aristotle live?  
2. When was the laptop invented?  
3. Is #2 before #1?

Geva et al. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies
1. When did Aristotle live?
2. When was the laptop invented?
3. Is #2 before #1?

“The first laptop was... in 1980.”

“Aristotle (384 –322 BC) was a philosopher...”

“...and was an ancient Greek philosopher.”

Geva et al. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies
1. When did Aristotle live?
2. When was the laptop invented?
3. Is #2 before #1?

“Aristotle (384 –322 BC) was a philosopher…”
“The first laptop was... in 1980.”
Did Socrates use a laptop?

No, Socrates did not use a laptop. Socrates lived in ancient Greece from 469/470 BCE to 399 BCE, long before the invention of laptops. Laptops were not developed until the 1970s, more than 2000 years after Socrates' death. Instead of using a laptop, Socrates engaged in philosophical conversations with his students and peers, often in public places like the agora, the marketplace of ancient Athens. His ideas were later recorded in the writings of his students, particularly Plato, who documented many of Socrates' dialogues in his works.
What happened between Ukraine and Russia on February 2022?

I apologize, but as of my last knowledge update in September 2021, I do not have information on events that occurred in February 2022 or beyond. It's important to stay updated with reliable news sources or consult recent information to learn about the specific events that took place between Ukraine and Russia during that time.
What happened to Mahsa Amini?

As of my last knowledge update in September 2021, I do not have information on Mahsa Amini. It is possible that this person became known or was involved in an event that occurred after my knowledge cutoff. To obtain the most accurate and up-to-date information on Mahsa Amini, I recommend referring to reliable news sources or conducting an online search for recent developments.
When did OpenAI launch GPT-4?

As of my last knowledge update in September 2021, OpenAI had not released GPT-4. At that time, the most recent version available was GPT-3, which was released in June 2020. However, please note that developments may have occurred since then, and it is recommended to refer to the latest news and updates from OpenAI or reliable sources to obtain information on any potential release of GPT-4 or subsequent versions.
Large Language Models

Extremely impressive:

• Store large quantities of knowledge in their parameters
• Impressive results on a variety of tasks
• Widely applicable

However ..

• Difficult to update and control knowledge
• Black-boxes — lack of provenance and interpretability
• Very large and expensive
Large Language Models

Extremely impressive:

• Store large quantities of knowledge in their parameters ✅
Large Language Models

Extremely impressive:

- Store large quantities of knowledge in their parameters ✔️
- Impressive results on a variety of tasks ✔️
Large Language Models

Extremely impressive:

• Store large quantities of knowledge in their parameters ✅
• Impressive results on a variety of tasks ✅
• Widely applicable ✅

However ..
Large Language Models

Extremely impressive:

• Store large quantities of knowledge in their parameters ✓
• Impressive results on a variety of tasks ✓
• Widely applicable ✓

However ..

• Difficult to update and control knowledge ✗
Large Language Models

Extremely impressive:

- Store large quantities of knowledge in their parameters ✅
- Impressive results on a variety of tasks ✅
- Widely applicable ✅

However ..

- Difficult to update and control knowledge ❌
- Black-boxes — lack of provenance and interpretability ❌
Large Language Models

Extremely impressive:

• Store large quantities of knowledge in their parameters ✓
• Impressive results on a variety of tasks ✓
• Widely applicable ✓

However ..

• Difficult to update and control knowledge ×
• Black-boxes — lack of provenance and interpretability ×
• Very large and expensive ×
Integrating Factual Knowledge in Neural Models

(via retrieval-augmentation and differentiable external memories)
### Knowledge-Intensive NLP Tasks

<table>
<thead>
<tr>
<th>question (string)</th>
<th>answer (sequence)</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;where did they film hot tub time machine&quot;</td>
<td>[ &quot;Fernie Alpine Resort&quot; ]</td>
</tr>
<tr>
<td>&quot;who has the right of way in international waters&quot;</td>
<td>[ &quot;Neither vessel&quot; ]</td>
</tr>
<tr>
<td>&quot;who does annie work for attack on titan&quot;</td>
<td>[ &quot;Marley&quot; ]</td>
</tr>
<tr>
<td>&quot;when was the immigration reform and control act passed&quot;</td>
<td>[ &quot;November 6, 1986&quot; ]</td>
</tr>
<tr>
<td>&quot;when was puerto rico added to the usa&quot;</td>
<td>[ &quot;1950&quot; ]</td>
</tr>
<tr>
<td>&quot;who has been chosen for best supporting actress in 64 national filmfare award&quot;</td>
<td>[ &quot;Zaira Wasim&quot; ]</td>
</tr>
<tr>
<td>&quot;which side of the white house is the front&quot;</td>
<td>[ &quot;North&quot; ]</td>
</tr>
<tr>
<td>&quot;names of the metropolitan municipalities in south africa&quot;</td>
<td>[ &quot;Mangaung Metropolitan Municipality&quot;, &quot;Nelson Mandela Bay Metropolitan Municipality&quot;, &quot;eThekwini...&quot; ]</td>
</tr>
</tbody>
</table>
# Knowledge-Intensive NLP Tasks

<table>
<thead>
<tr>
<th>input (string)</th>
<th>output (list)</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Nikolaj Coster-Waldau worked with the Fox Broadcasting Company.&quot;</td>
<td>[ { &quot;answer&quot;: &quot;SUPPORTS&quot;, &quot;provenance&quot;: [ ] } ]</td>
</tr>
<tr>
<td>&quot;Roman Atwood is a content creator.&quot;</td>
<td>[ { &quot;answer&quot;: &quot;SUPPORTS&quot;, &quot;provenance&quot;: [ ] } ]</td>
</tr>
<tr>
<td>&quot;History of art includes architecture, dance, sculpture, music, painting, poetry literature,...&quot;</td>
<td>[ { &quot;answer&quot;: &quot;SUPPORTS&quot;, &quot;provenance&quot;: [ ] } ]</td>
</tr>
<tr>
<td>&quot;Adrienne Bailon is an accountant.&quot;</td>
<td>[ { &quot;answer&quot;: &quot;REFUTES&quot;, &quot;provenance&quot;: [ ] } ]</td>
</tr>
<tr>
<td>&quot;Homeland is an American television spy thriller based on the Israeli television series Prisoners...&quot;</td>
<td>[ { &quot;answer&quot;: &quot;SUPPORTS&quot;, &quot;provenance&quot;: [ ] } ]</td>
</tr>
</tbody>
</table>
The KILT Benchmark

- Open-Domain Question Answering (Natural Questions, TriviaQA, HotPotQA, ELI5)
- Fact-Checking (FEVER)
- Slot Filling (T-REx, zsRE)
- Dialogue (Wizard of Wikipedia)
- Entity Linking (AIDA, WNED-WIKI, WNED-CWEB)

Petroni et al. KILT: A Benchmark for Knowledge Intensive Language Tasks
Solving Knowledge-Intensive NLP Tasks

Parametric Models

T5, BART, LLaMA, BLOOM..
Solving Knowledge-Intensive NLP Tasks

Parametric Models

T5, BART, LLaMA, BLOOM...

Q: Last time LA Dodgers won the World Series?
Solving Knowledge-Intensive NLP Tasks

Parametric Models

T5, BART, LLaMA, BLOOM...

Q: last time LA Dodgers won the World Series?

Accurate: ✗  Interpretable: ✗  Fast: ~
Solving Knowledge-Intensive NLP Tasks

Parametric Models

- Training data
- Seq2Seq
- T5, BART, LLaMA, BLOOM...
- Q: last time did Dodgers win the world series?
- Accurate: ✗
- Interpretable: ✗
- Fast: ~

Retrieval-Augmented Models

- Wikipedia
- Retriever
- Reader
- Q: last time did Dodgers win the world series?
- 1988
Solving Knowledge-Intensive NLP Tasks

**Parametric Models**

- **Training data** → Seq2Seq
- Q: last time LA Dodgers won the world series? → Seq2Seq → 1988

Accurate: ✗  Interpretable: ✗  Fast: ~

**T5, BART, LLaMA, BLOOM..**

**Retrieval-Augmented Models**

- Q: last time LA Dodgers won the world series? → Retriever → Reader → 1988

Accurate: ✓  Interpretable: ~  Fast: ✗
Solving Knowledge-Intensive NLP Tasks

Retrieval-Augmented Models

Q: last time la dodgers won the world series?

Accurate: ✅ Interpretable: ~ Fast: ❌

Retrieval-Augmented Models in a nutshell:
Solving Knowledge-Intensive NLP Tasks

Retrieval-Augmented Models in a nutshell:

- Document representations are stored in a large (1M-1B entries) dense vector index
Solving Knowledge-Intensive NLP Tasks

Retrieval-Augmented Models in a nutshell:

- Document representations are stored in a large (1M-1B entries) dense vector index
- A **retriever** retrieves the most relevant document are retrieved via **similarity search**
Solving Knowledge-Intensive NLP Tasks

Retrieval-Augmented Models in a nutshell:

- Document representations are stored in a large (1M-1B entries) dense vector index.
- A retriever retrieves the most relevant document via similarity search.
- A reader produces the output conditioned on the input and the retrieved documents.
ATLAS — A Retrieval-Augmented Model

Masked Language Modelling:
Bermuda Triangle is in the <MASK> of the Atlantic Ocean.

Fact checking:
Bermuda Triangle is in the western part of the Himalayas.

Question answering:
Where is the Bermuda Triangle?

The Bermuda Triangle is an urban legend focused on a loosely-defined region in the western part of the North Atlantic Ocean.

Western part of the North Atlantic Ocean

False

Pretraining

Few-shot
## ATLAS — Open-Domain QA

<table>
<thead>
<tr>
<th>Model</th>
<th>NQ</th>
<th>TriviaQA filtered</th>
<th>TriviaQA unfiltered</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>64-shot</td>
<td>Full</td>
<td>64-shot</td>
</tr>
<tr>
<td>GPT-3 (Brown et al., 2020)</td>
<td>29.9</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Gopher (Rae et al., 2021)</td>
<td>28.2</td>
<td>-</td>
<td>57.2</td>
</tr>
<tr>
<td>Chinchilla (Hoffmann et al., 2022)</td>
<td>35.5</td>
<td>-</td>
<td>64.6</td>
</tr>
<tr>
<td>PaLM (Chowdhery et al., 2022)</td>
<td>39.6</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>RETRO (Borgeaud et al., 2021)</td>
<td>-</td>
<td>45.5</td>
<td>-</td>
</tr>
<tr>
<td>FiD (Izacard &amp; Grave, 2020)</td>
<td>-</td>
<td>51.4</td>
<td>-</td>
</tr>
<tr>
<td>FiD-KD (Izacard &amp; Grave, 2021)</td>
<td>-</td>
<td>54.7</td>
<td>-</td>
</tr>
<tr>
<td>R2-D2 (Fajcik et al., 2021)</td>
<td>-</td>
<td>55.9</td>
<td>-</td>
</tr>
</tbody>
</table>
### ATLAS — Open-Domain QA

<table>
<thead>
<tr>
<th>Model</th>
<th>NQ</th>
<th>TriviaQA filtered</th>
<th>TriviaQA unfiltered</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>64-shot</td>
<td>Full</td>
<td>64-shot</td>
</tr>
<tr>
<td>GPT-3 (Brown et al., 2020)</td>
<td>29.9</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Gopher (Rae et al., 2021)</td>
<td>28.2</td>
<td>-</td>
<td>57.2</td>
</tr>
<tr>
<td>Chinchilla (Hoffmann et al., 2022)</td>
<td>35.5</td>
<td>-</td>
<td>64.6</td>
</tr>
<tr>
<td>PaLM (Chowdhery et al., 2022)</td>
<td>39.6</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>RETRO (Borgeaud et al., 2021)</td>
<td>-</td>
<td>45.5</td>
<td>-</td>
</tr>
<tr>
<td>FiD (Izacard &amp; Grave, 2020)</td>
<td>-</td>
<td>51.4</td>
<td>-</td>
</tr>
<tr>
<td>FiD-KD (Izacard &amp; Grave, 2021)</td>
<td>-</td>
<td>54.7</td>
<td>-</td>
</tr>
<tr>
<td>R2-D2 (Fajcik et al., 2021)</td>
<td>-</td>
<td>55.9</td>
<td>-</td>
</tr>
</tbody>
</table>
### ATLAS — Open-Domain QA

<table>
<thead>
<tr>
<th>Model</th>
<th>NQ 64-shot</th>
<th>NQ Full</th>
<th>TriviaQA filtered 64-shot</th>
<th>TriviaQA filtered Full</th>
<th>TriviaQA unfiltered 64-shot</th>
<th>TriviaQA unfiltered Full</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPT-3 (Brown et al., 2020)</td>
<td>29.9</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>71.2</td>
<td>-</td>
</tr>
<tr>
<td>Gopher (Rae et al., 2021)</td>
<td>28.2</td>
<td>-</td>
<td>57.2</td>
<td>-</td>
<td>61.3</td>
<td>-</td>
</tr>
<tr>
<td>Chinchilla (Hoffmann et al., 2022)</td>
<td>35.5</td>
<td>-</td>
<td>64.6</td>
<td>-</td>
<td>72.3</td>
<td>-</td>
</tr>
<tr>
<td>PaLM (Chowdhery et al., 2022)</td>
<td>39.6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>81.4</td>
<td>-</td>
</tr>
<tr>
<td>RETRO (Borgeaud et al., 2021)</td>
<td>-</td>
<td>45.5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>FiD (Izacard &amp; Grave, 2020)</td>
<td>-</td>
<td>51.4</td>
<td>-</td>
<td>67.6</td>
<td>-</td>
<td>80.1</td>
</tr>
<tr>
<td>FiD-KD (Izacard &amp; Grave, 2021)</td>
<td>-</td>
<td>54.7</td>
<td>-</td>
<td>73.3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>R2-D2 (Faicik et al., 2021)</td>
<td>-</td>
<td>55.9</td>
<td>-</td>
<td>69.9</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>ATLAS</strong></td>
<td><strong>42.4</strong></td>
<td><strong>60.4</strong></td>
<td><strong>74.5</strong></td>
<td><strong>79.8</strong></td>
<td><strong>84.7</strong></td>
<td><strong>89.4</strong></td>
</tr>
</tbody>
</table>
Generalisation

<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 Only</td>
<td>Total</td>
</tr>
</tbody>
</table>

- The question appears in the training set
- The answer appears in the training set
# Generalisation

<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>Closed book T5-11B+SSM</td>
<td>36.6</td>
<td>77.2</td>
<td>22.2</td>
</tr>
<tr>
<td>Closed book BART</td>
<td>26.5</td>
<td>67.6</td>
<td>10.2</td>
</tr>
</tbody>
</table>

- The question appears in the training set
- The answer appears in the training set
## Generalisation

<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>
</tbody>
</table>
# Generalisation

<table>
<thead>
<tr>
<th>Model</th>
<th>Natural Questions</th>
<th>TriviaQA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total</td>
<td>Overlap</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Liu et al. Challenges in Generalisation in Open-Domain Question Answering
## Generalisation

<table>
<thead>
<tr>
<th>Model</th>
<th>Natural Questions</th>
<th>TriviaQA</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total</td>
<td>Overlap</td>
<td>Comp-gen</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>Overlap</td>
<td></td>
</tr>
<tr>
<td>T5-11B+SSM</td>
<td>36.59</td>
<td>81.48</td>
<td>17.47</td>
</tr>
<tr>
<td>BART</td>
<td>26.54</td>
<td>76.34</td>
<td>5.88</td>
</tr>
</tbody>
</table>

Liu et al. Challenges in Generalisation in Open-Domain Question Answering
# Generalisation

<table>
<thead>
<tr>
<th>Model</th>
<th>Natural Questions</th>
<th>TriviaQA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total</td>
<td>Overlap</td>
</tr>
<tr>
<td></td>
<td>Comp-gen</td>
<td>Novel-entity</td>
</tr>
<tr>
<td>Non-parametric</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RAG</td>
<td>44.49</td>
<td>75.75</td>
</tr>
<tr>
<td>FiD</td>
<td><strong>53.13</strong></td>
<td><strong>78.85</strong></td>
</tr>
<tr>
<td>DPR</td>
<td>41.27</td>
<td>71.33</td>
</tr>
<tr>
<td>RePAQ</td>
<td>47.26</td>
<td>78.61</td>
</tr>
<tr>
<td>Parametric</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T5-11B+SSM</td>
<td>36.59</td>
<td><strong>81.48</strong></td>
</tr>
<tr>
<td>BART</td>
<td>26.54</td>
<td>76.34</td>
</tr>
</tbody>
</table>

Liu et al. Challenges in Generalisation in Open-Domain Question Answering
### Updateability

<table>
<thead>
<tr>
<th>query (string)</th>
<th>answer (list)</th>
<th>date (string)</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Valentino Rossi plays for <em>X</em>.&quot;</td>
<td>[ { &quot;wikidata_id&quot;: &quot;Q1085474&quot;, &quot;name&quot;: [ &quot;Monster Energy...&quot; } ]</td>
<td>&quot;2010&quot;</td>
</tr>
<tr>
<td>&quot;Valentino Rossi plays for <em>X</em>.&quot;</td>
<td>[ { &quot;wikidata_id&quot;: &quot;Q271812&quot;, &quot;name&quot;: [ &quot;Ducati&quot;, &quot;Ducati...&quot; } ]</td>
<td>&quot;2011&quot;</td>
</tr>
<tr>
<td>&quot;Valentino Rossi plays for <em>X</em>.&quot;</td>
<td>[ { &quot;wikidata_id&quot;: &quot;Q271812&quot;, &quot;name&quot;: [ &quot;Ducati&quot;, &quot;Ducati...&quot; } ]</td>
<td>&quot;2012&quot;</td>
</tr>
<tr>
<td>&quot;Valentino Rossi plays for <em>X</em>.&quot;</td>
<td>[ { &quot;wikidata_id&quot;: &quot;Q1085474&quot;, &quot;name&quot;: [ &quot;Monster Energy...&quot; } ]</td>
<td>&quot;2013&quot;</td>
</tr>
<tr>
<td>&quot;Valentino Rossi plays for <em>X</em>.&quot;</td>
<td>[ { &quot;wikidata_id&quot;: &quot;Q1085474&quot;, &quot;name&quot;: [ &quot;Monster Energy...&quot; } ]</td>
<td>&quot;2014&quot;</td>
</tr>
</tbody>
</table>

Izacard et al. Few-shot Learning with Retrieval-Augmented Language Models
## Updateability

<table>
<thead>
<tr>
<th>Train Set</th>
<th>Test-time Index</th>
<th>2017 Test Set Acc.</th>
<th>2020 Test Set Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Closed-book</td>
<td>ATLAS</td>
</tr>
<tr>
<td>2017 answers</td>
<td>2017</td>
<td>T5-XXL</td>
<td>Retrieval-augmented</td>
</tr>
<tr>
<td></td>
<td>2020</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2020 answers</td>
<td>2017</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2020</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Updateability

<table>
<thead>
<tr>
<th>Train Set</th>
<th>Test-time Index</th>
<th>2017 Test Set Acc.</th>
<th>2020 Test Set Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Closed-book</td>
<td>ATLAS</td>
</tr>
<tr>
<td>2017 answers</td>
<td>2017</td>
<td>12.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2020</td>
<td>12.1</td>
<td>Retrieval-augmented</td>
</tr>
<tr>
<td>2020 answers</td>
<td>2017</td>
<td>4.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2020</td>
<td>4.8</td>
<td></td>
</tr>
</tbody>
</table>
### Updateability

<table>
<thead>
<tr>
<th>Train Set</th>
<th>Test-time Index</th>
<th>2017 Test Set Acc.</th>
<th>2020 Test Set Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Closed-book</td>
<td>ATLAS</td>
</tr>
<tr>
<td>2017 answers</td>
<td>2017</td>
<td>12.1</td>
<td>57.7</td>
</tr>
<tr>
<td></td>
<td>2020</td>
<td>12.1</td>
<td>10.2</td>
</tr>
<tr>
<td>2020 answers</td>
<td>2017</td>
<td>4.8</td>
<td>50.1</td>
</tr>
<tr>
<td></td>
<td>2020</td>
<td>4.8</td>
<td>3.5</td>
</tr>
</tbody>
</table>
Retrieval-Augmented Models

• Control and updateability ✅

• Some degree of interpretability and provenance ✅

• More factual generations; fewer hallucinations ✅

• Towards decoupling factual knowledge from parameters ✅

• Need to generate and index billions of document representations ❌

• The reader needs to read hundreds of documents ❌
Retrieval-Augmented Models

• Control and updateability ✅

• Some degree of interpretability and provenance ✅

• Need to generate and index billions of document representations ❌

• The reader needs to read hundreds of documents ❌
Retrieval-Augmented Models

• Control and updateability ✅
• Some degree of interpretability and provenance ✅
• More factual generations; fewer hallucinations ✅
Retrieval-Augmented Models

• Control and updateability ✅
• Some degree of interpretability and provenance ✅
• More factual generations; fewer hallucinations ✅
• Towards decoupling factual knowledge from parameters ✅
Retrieval-Augmented Models

- Control and updateability ✓
- Some degree of interpretability and provenance ✓
- More factual generations; fewer hallucinations ✓
- Towards decoupling factual knowledge from parameters ✓
- Need to generate and index billions of document representations ✗
Retrieval-Augmented Models

• Control and updateability ✔
• Some degree of interpretability and provenance ✔
• More factual generations; fewer hallucinations ✔
• Towards decoupling factual knowledge from parameters ✔
• Need to generate and index billions of document representations ❌
• The reader needs to read hundreds of documents ❌
Retrieval-Augmented Models

• Control and updateability ✔
• Some degree of interpretability and provenance ✔
• More factual generations; fewer hallucinations ✔
• Towards decoupling factual knowledge from parameters ✔
• Need to generate and index billions of document representations ✗
• The reader needs to read hundreds of documents ✗
PAQ: A Condensed Wikipedia

We generated 65 million synthetic Question-Answer pairs..

Generation

1. Passage selector
2. Answer Extractor
3. Question Generator
4. Global Filtering

PAQ

65M Probably-Asked Questions

Lewis et al. PAQ: 65 Million Probably-Asked Questions and What You Can Do With Them
and designed RePAQ, a retrieval-augmented QA model that uses the generated question-answer pairs as knowledge source.
PAQ: A Condensed Wikipedia

and designed RePAQ, a retrieval-augmented QA model that uses the generated question-answer pairs as knowledge source.

<table>
<thead>
<tr>
<th>Model</th>
<th>Retriever</th>
<th>Reranker</th>
<th>Exact Match</th>
<th>Q/sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>FiD-large</td>
<td>-</td>
<td>-</td>
<td>51.4</td>
<td>0.5</td>
</tr>
<tr>
<td>FiD-base</td>
<td>-</td>
<td>-</td>
<td>48.2</td>
<td>2</td>
</tr>
</tbody>
</table>

Lewis et al. PAQ: 65 Million Probably-Asked Questions and What You Can Do With Them
Lewis et al. PAQ: 65 Million Probably-Asked Questions and What You Can Do With Them

and designed RePAQ, a retrieval-augmented QA model that uses the generated question-answer pairs as knowledge source.

<table>
<thead>
<tr>
<th>Model</th>
<th>Retriever</th>
<th>Reranker</th>
<th>Exact Match</th>
<th>Q/sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>FiD-large</td>
<td>-</td>
<td>-</td>
<td>51.4</td>
<td>0.5</td>
</tr>
<tr>
<td>FiD-base</td>
<td>-</td>
<td>-</td>
<td>48.2</td>
<td>2</td>
</tr>
<tr>
<td>RePAQ</td>
<td>base</td>
<td>-</td>
<td>40.9</td>
<td>1400</td>
</tr>
<tr>
<td>RePAQ</td>
<td>large</td>
<td>-</td>
<td>41.2</td>
<td>1100</td>
</tr>
<tr>
<td>RePAQ</td>
<td>xlarge</td>
<td>-</td>
<td>41.5</td>
<td>800</td>
</tr>
<tr>
<td>RePAQ</td>
<td>base</td>
<td>base</td>
<td>45.7</td>
<td>55</td>
</tr>
<tr>
<td>RePAQ</td>
<td>large</td>
<td>xlarge</td>
<td>46.2</td>
<td>10</td>
</tr>
<tr>
<td>RePAQ</td>
<td>xlarge</td>
<td>xxlarge</td>
<td>47.6</td>
<td>6</td>
</tr>
</tbody>
</table>
# Efficient QA Challenge at NeurIPS 2020

## Systems Under 500Mb

<table>
<thead>
<tr>
<th>Rank</th>
<th>Model</th>
<th>Participant</th>
<th>Affiliation</th>
<th>Attempt Date</th>
<th>Accuracy</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ucInlp-fair-efficientqa-lg</td>
<td>ucInlp-efficientqa</td>
<td>University College London and Facebook AI Research</td>
<td>14/11/2020</td>
<td>33.44</td>
<td>336.23 MB</td>
</tr>
</tbody>
</table>

## Smallest Systems Achieving 25% Accuracy

<table>
<thead>
<tr>
<th>Rank</th>
<th>Model</th>
<th>Participant</th>
<th>Affiliation</th>
<th>Attempt Date</th>
<th>Accuracy</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ucInlp-fair-efficientqa</td>
<td>ucInlp-efficientqa</td>
<td>University College London and Facebook AI Research</td>
<td>07/11/2020</td>
<td>26.78</td>
<td>29.43 MB</td>
</tr>
</tbody>
</table>
EMATs combine the strengths of Parametric and Retrieval-Augmented models:

- Knowledge encoded as a differentiable key-value memory
- Memory retrieval via fast MIPS search

Efficient Memory-Augmented Transformers

Generated QA pairs (for example) can be integrated in Transformer-based models:

Q: last time LA dodgers won the world series?

1988
Efficient Memory-Augmented Transformers

Generated QA pairs (for example) can be integrated in Transformer-based models:

Wu et al. An Efficient Memory-Augmented Transformer for Knowledge-Intensive NLP Tasks
EMATs combine the strengths of Parametric and Retrieval-Augmented models:

Generated QA pairs (for example) can be integrated in Transformer-based models:

Wu et al. An Efficient Memory-Augmented Transformer for Knowledge-Intensive NLP Tasks
EMATs combine the strengths of Parametric and Retrieval-Augmented models:

- Knowledge encoded as a **differentiable key-value memory**

Wu et al. An Efficient Memory-Augmented Transformer for Knowledge-Intensive NLP Tasks
EMATs combine the strengths of Parametric and Retrieval-Augmented models:

- Knowledge encoded as a differentiable key-value memory
- Memory retrieval via fast MIPS search

Wu et al. An Efficient Memory-Augmented Transformer for Knowledge-Intensive NLP Tasks
Efficient Memory-Augmented Transformers
EMAT in Knowledge-Intensive Tasks

Open-Domain Question Answering

- T5-base
- BART-large
- RePAQ-base
- RePAQ rerank
- QAMAT
- DensePhrases
- RAG
- FiD-base
- DPR
- FiD-large
- EMAT-SKSV
- EMAT-FRSV

EMAT in Knowledge-Intensive Tasks

- Open-Domain Question Answering
- Open-Domain Dialogue
- Long-Form Question Answering
EMAT in Knowledge-Intensive Tasks

Open-Domain Question Answering

- T5-base
- RePAQ_base
- EMAT_FSV
- EMAT_SKSV

Open-Domain Dialogue

- T5-base
- EMAT_SKSV

Graphs showing performance metrics such as EM score and ROUGE-L for various models.
Combination of discrete optimisation algorithms (e.g., an ILP solver, Dijkstra's shortest path algorithm) and arbitrary neural modules

Combination of complex discrete probability distributions (e.g., discrete attention distributions over the input, graph-structured representations) with arbitrary neural modules

EMAT in Knowledge-Intensive Tasks

Open-Domain Question Answering

- T5-base
- T5-3B
- DPR
- FID-base
- FID-large
- QAMAT
- RePAQ rerank
- RePAQ-base
- EMAT-skSV
- EMAT-frSV
- BART-large

Open-Domain Dialogue

- T5-base
- T5-3B
- DPR
- FID-base
- FID-large
- QAMAT
- RePAQ rerank
- RePAQ-base
- EMAT-skSV
- EMAT-frSV
- BART-large

Long-Form Question Answering

- T5-base
- T5-3B
- DPR
- FID-base
- FID-large
- QAMAT
- RePAQ rerank
- RePAQ-base
- EMAT-skSV
- EMAT-frSV
- BART-large

EMAT in Knowledge-Intensive Tasks
Text-to-SQL/SPARQL
Text-to-SQL Systems

The framework for text-to-SQL systems: given a database schema and user utterance, the system outputs a corresponding SQL query to query the database system or the results.

SELECT T1.CITY_NAME FROM CITY AS T1 WHERE T1.POPULATION > 150000 AND T1.STATE_NAME = "Kansas" ;
What is Spider?

Spider is a large-scale complex and cross-domain semantic parsing and text-to-SQL dataset annotated by 11 Yale students. The goal of the Spider challenge is to develop natural language interfaces to cross-domain databases. It consists of 10,181 questions and 5,893 unique complex SQL queries on 200 databases with multiple tables covering 136 different domains. In Spider 1.0, different complex SQL queries and databases appear in train and test sets. To do well on it, systems must generalize well to not only new SQL queries but also new database schemas.

Why we call it “Spider”? It is because our dataset is complex and cross-domain like a spider crawling across multiple complex (with many foreign keys) nests (databases).

Leaderboard - Execution with Values

Our current models do not predict any value in SQL conditions so that we do not provide execution accuracies. However, we encourage you to provide it in the future submissions. For value prediction, your model should be able to 1) copy from the question inputs, 2) retrieve from the database content (database content is available), or 3) generate numbers (e.g. 3 in “LIMIT 3”). Notice: Test results after May 02, 2020 are reported on the new release (collected some annotation errors).

<table>
<thead>
<tr>
<th>Rank</th>
<th>Model</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MiniSeek</td>
<td>91.2</td>
</tr>
<tr>
<td></td>
<td>Anonymous</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Code and paper coming soon</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>DAIL-SQL + GPT-4 + Self-Consistency Alibaba Group (Gao and Wang et al., 2023) code</td>
<td>86.6</td>
</tr>
<tr>
<td>2</td>
<td>DAIL-SQL + GPT-4 Alibaba Group (Gao and Wang et al., 2023) code</td>
<td>86.2</td>
</tr>
<tr>
<td>3</td>
<td>DAIL-SQL + GPT-4 Alibaba Group (Gao and Wang et al., 2023) code</td>
<td>85.8</td>
</tr>
<tr>
<td>4</td>
<td>DAIL-SQL + GPT-4 Alibaba Group (Gao and Wang et al., 2023) code</td>
<td>85.6</td>
</tr>
</tbody>
</table>
The left half shows retrieval repository construction in three steps. The top three sentences are three specific instances each. The Green dashed box presents the training set. The right half is a dynamic revision chain with SQL queries generated by LLM iterations as nodes (green boxes). The output of steps 2 and 4 are collectively referred to as fine-grained feedback.
Language as Programs

Remind me to buy milk after my last meeting on Monday.

semantic parsing

\[
\text{Add}(\text{Buy}(\text{Milk}), \text{argmax}(\text{Meetings} \cap \text{HasDate}(2016-07-18), \text{EndTime}))
\]

execute

[reminder added]
Building a Semantic Parser Overnight

Yushi Wang*  
Stanford University  
yushiw@cs.stanford.edu

Jonathan Berant*  
Stanford University  
joberant@stanford.edu

Percy Liang  
Stanford University  
pliang@cs.stanford.edu

(1) by builder (~30 minutes)

Seed lexicon

article → TYPENP[article]
pub date → RELNP[publicationDate]
cites → VP/NP[cites]
...

(2) via domain-general grammar

Logical forms and canonical utterances

article that has the largest publication date
argmax(type.article,publicationDate)
person that is author of the most number of article
argmax(type.person,R(λx.count(type.article P author.x)))
...

(3) via crowdsourcing (~5 hours)

Paraphrases

what is the newest published article?
who has published the most articles?
...

(4) by training a paraphrasing model

Semantic parser