Graph QA

Contents

This notebook goes over how to do question answering over a graph data structure.

Create the graph

In this section, we construct an example graph. At the moment, this works best for small pieces of text.

```
from langchain.indexes import GraphIndexCreator
from langchain.llms import OpenAI
from langchain.document_loaders import TextLoader
```

```
index_creator = GraphIndexCreator(llm=OpenAI(temperature=0))
```

```
with open("../../state_of_the_union.txt") as f:
    all_text = f.read()
```

We will use just a small snippet, because extracting the knowledge triplets is a bit intensive at the moment.

```
text = "\n".join(all_text.split("\n\n")[105:108])
```

```
text
```

Skip to main content

```
'It won’t look like much, but if you stop and look closely, you’ll see a “Field of
dreams,” the ground on which America’s future will be built. \nThis is where
Intel, the American company that helped build Silicon Valley, is going to build
its $20 billion semiconductor “mega site”. \nUp to eight state-of-the-art
factories in one place. 10,000 new good-paying jobs. '
```

```
graph = index_creator.from_text(text)
```

We can inspect the created graph.

```
graph.get_triples()
```

```
[('Intel', '$20 billion semiconductor "mega site"', 'is going to build'),
 ('Intel', 'state-of-the-art factories', 'is building'),
 ('Intel', '10,000 new good-paying jobs', 'is creating'),
 ('Intel', 'Silicon Valley', 'is helping build'),
 ('Field of dreams',
  "America's future will be built",
  'is the ground on which')]
```

Querying the graph

We can now use the graph QA chain to ask question of the graph

```
from langchain.chains import GraphQAChain
```

```
chain = GraphQAChain.from_llm(OpenAI(temperature=0), graph=graph, verbose=True)
```

```
chain.run("what is Intel going to build?")
```

```
> Entering new GraphQAChain chain...
Entities Extracted:
 Intel
Full Context:
Intel is going to build $20 billion semiconductor "mega site"
Intel is building state-of-the-art factories
```

Skip to main content

```
> Finished chain.
```

```
' Intel is going to build a $20 billion semiconductor "mega site" with state-of-
the-art factories, creating 10,000 new good-paying jobs and helping to build
Silicon Valley.'
```

Save the graph

We can also save and load the graph.

```
graph.write_to_gml("graph.gml")
```

```
from langchain.indexes.graph import NetworkxEntityGraph
```

```
loaded_graph = NetworkxEntityGraph.from_gml("graph.gml")
```

```
loaded_graph.get_triples()
```

```
[('Intel', '$20 billion semiconductor "mega site"', 'is going to build'),
 ('Intel', 'state-of-the-art factories', 'is building'),
 ('Intel', '10,000 new good-paying jobs', 'is creating'),
 ('Intel', 'Silicon Valley', 'is helping build'),
 ('Field of dreams',
  "America's future will be built",
  'is the ground on which')]
```