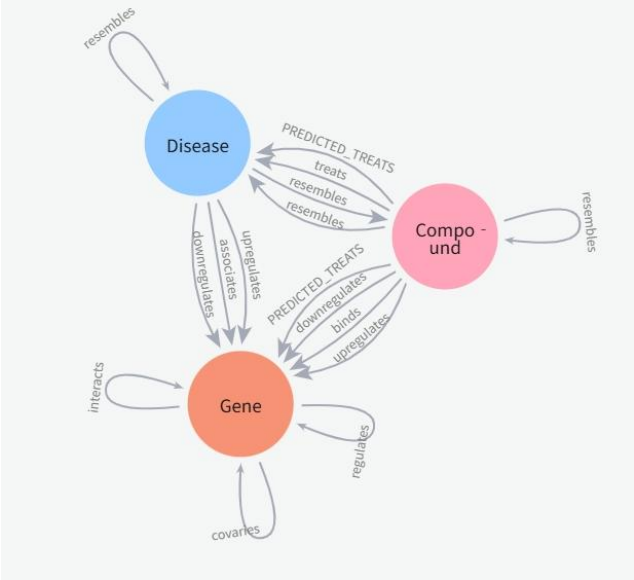
实验：使用 PyKEEN 和 Neo4j 补全知识图谱

1. 启动 neo4j

图谱 schema



2. 在 VS Code 中使用 Jupyter Notebook

记录 token

```
  Or copy and paste one of these URLs:
      http://localhost:8888/lab?token=3c1d5a8c5c624c1e828443e98bbcb2208f6f09331d022958
      http://127.0.0.1:8888/lab?token=3c1d5a8c5c624c1e828443e98bbcb2208f6f09331d022958
```

3. 执行 notebook 代码

【实验任务】完成 notebook 文件 pykeen-neo4j.ipynb 中的内容：

将 notebook 每个代码段执行成功的输出截图粘贴在下面

【代码段 1 执行截图】:

```python
# Define Neo4j connections
from neo4j import GraphDatabase
import pandas as pd

host = 'bolt://localhost:7687'
user = 'neo4j'
password = '12345678'
driver = GraphDatabase.driver(host,auth=(user, password))


def run_query(query, params={}):
    with driver.session() as session:
        result = session.run(query, params)
        return pd.DataFrame([r.values() for r in result], columns=result.keys())
```

✓ 0.0s

【代码段 2 执行截图】:

```
data = run_query("""
MATCH (s)-[r]->(t)
RETURN toString(id(s)) as source, toString(id(t)) AS target, type(r) as type
""")
```

✓ 17.7s Python

```
Received notification from DBMS server: {severity: WARNING} {code: Neo.ClientNotification.Statement.FeatureDeprecationWar
Received notification from DBMS server: {severity: WARNING} {code: Neo.ClientNotification.Statement.FeatureDeprecationWar
```

【代码段 3 执行截图】:

```
data.head()
```

✓ 0.0s

	source	target	type
0	18996	21900	interacts
1	514	5013	interacts
2	11912	20825	upregulates
3	19641	4335	interacts
4	5148	7779	binds

【代码段 4 执行截图】:

```
from pykeen.triples import TriplesFactory


tf = TriplesFactory.from_labeled_triples(
    data[["source", "type", "target"]].values,
    create_inverse_triples=False,
    entity_to_id=None,
    relation_to_id=None,
    compact_id=False,
    filter_out_candidate_inverse_relations=True,
    metadata=None,
)
```

✓ 1.2s Python

【代码段 5 执行截图】:

```
training, testing, validation = tf.split([.8, .1, .1])
```

✓ 0.1s Python

```
using automatically assigned random_state=38986874
```

【代码段 6 执行截图】:

```
from pykeen.pipeline import pipeline

result = pipeline(
    training=training,
    testing=testing,
    validation=validation,
    model='TransE',
    stopper='early',
    epochs=5,
    dimensions=10,
    random_seed=420

)
```

✓ 5m 24.2s Python

```
No cuda devices were available. The model runs on CPU
Training epochs on cpu: 100%| 5/5 [00:49<00:00,  9.94s/epoch, loss=0.218, prev_loss=0.249]
WARNING:pykeen.utils:Using automatic batch size on device.type='cpu' can cause unexplained out-of-memory crashes. Therefo
Evaluating on cpu:   0%|                                                    | 0.00/56.2k [00:00<?, ?triple/s]W
Evaluating on cpu: 100%| 56.2k/56.2k [04:27<00:00, 210triple/s]
INFO:pykeen.evaluation.evaluator:Evaluation took 269.69s seconds
```

【代码段 7 执行截图】:

```
result.save_to_directory("./model")
```

✓ 1.3s Python

```
INFO:pykeen.triples.triples_factory:Stored TriplesFactory(num_entities=19930, num_relations=10, create_inverse_triples=Fa
INFO:pykeen.pipeline.api:Saved to directory: C:\Users\34418\Desktop\KnoledgeEngineering\lab6-pykeen\lab6-pykeen\pykeen\mo
```

【代码段 8 执行截图】:

```
from pykeen import predict

compound_id = run_query("""
MATCH (s:Compound)
WHERE s.name = "L-Asparagine"
RETURN toString(id(s)) as id
""")['id'][0]

df = predict.predict_target(model=result.model, head=compound_id, relation="treats", triples_factory=result.
training).df
print(df.head(5))
```

✓ 0.1s Python

```
WARNING:neo4j.notifications:Received notification from DBMS server: {severity: WARNING} {code: Neo.ClientNotification.Sta
       tail_id     score tail_label
14044    14044 -6.925520         40
16815    16815 -7.171328       6832
12028    12028 -7.324925      22303
15525    15525 -7.361601       5517
7009      7009 -7.393231      17183
```

【代码段 9 执行截图】:

```
candidate_nodes = df.head(5)['tail_label'].to_list()

run_query("""
MATCH (n)
WHERE id(n) = toInteger($compound_id)
UNWIND $candidates as ca
MATCH (c)
WHERE id(c) = toInteger(ca)
MERGE (n)-[:PREDICTED_TREATS]->(c)
""", {'compound_id':compound_id, 'candidates': candidate_nodes})
```

✓ 0.6s Python

```
WARNING:neo4j.notifications:Received notification from DBMS server: {severity: WARNING} {code: Neo.ClientNotification.Sta
WARNING:neo4j.notifications:Received notification from DBMS server: {severity: WARNING} {code: Neo.ClientNotification.Sta
```

【代码段 10 执行截图】:

```
run_query("""
MATCH (c:Compound)-[:PREDICTED_TREATS]->(d:Disease)
RETURN c.name as compound, d.name as disease
""")
```

✓ 0.1s

	compound	disease
0	L-Asparagine	schizophrenia
1	L-Asparagine	bipolar disorder
2	L-Asparagine	type 2 diabetes mellitus
3	L-Asparagine	ulcerative colitis
4	L-Asparagine	skin cancer

【代码段 11 执行截图】:

```
run_query("""
MATCH (c:Compound {name: "L-Asparagine"}),(d:Disease {name:"colon cancer"})
WITH c,d
MATCH p=AllShortestPaths((c)-[r:binds|regulates|interacts|upregulates|downregulates|associates*1..4]-(d))
RETURN [n in nodes(p) | n.name] LIMIT 25
""")
```

✓ 0.2s

	[n in nodes(p) | n.name]
0	[L-Asparagine, ASRGL1, GDF15, colon cancer]
1	[L-Asparagine, SLC1A5, VEGFA, colon cancer]
2	[L-Asparagine, SLC38A3, VEGFA, colon cancer]
3	[L-Asparagine, NARS, TP53, colon cancer]
4	[L-Asparagine, ASNS, CCNB1, colon cancer]
5	[L-Asparagine, SLC1A5, FZD5, colon cancer]
6	[L-Asparagine, ASRGL1, AKR7A2, colon cancer]
7	[L-Asparagine, SLC38A3, MIF, colon cancer]
8	[L-Asparagine, SLC1A5, GNAI1, colon cancer]
9	[L-Asparagine, ASRGL1, PUF60, colon cancer]
10	[L-Asparagine, ASNS, CEBPB, colon cancer]
11	[L-Asparagine, ASNS, HSF1, colon cancer]
12	[L-Asparagine, SLC1A5, RUVBL1, colon cancer]
13	[L-Asparagine, ASRGL1, PSMG1, colon cancer]
14	[L-Asparagine, SLC1A5, G3BP1, colon cancer]
15	[L-Asparagine, ASNS, G3BP1, colon cancer]
16	[L-Asparagine, ASRGL1, CDK7, colon cancer]
17	[L-Asparagine, ASNS, BRAF, colon cancer]
18	[L-Asparagine, SLC38A3, MT1X, colon cancer]
19	[L-Asparagine, SLC1A5, OXCT1, colon cancer]
20	[L-Asparagine, SLC38A3, EZH2, colon cancer]

```
21        [L-Asparagine, ASRGL1, NME1, colon cancer]
22     [L-Asparagine, SLC38A3, RBM28, colon cancer]
23  [L-Asparagine, SLC38A3, PLXNA1, colon cancer]
24           [L-Asparagine, SLC1A5, APC, colon cancer]
```