

# 问答精灵 - aowu

## 题目编号:S3

- 开发一个具有一定功能的小型问答系统，答案从给定的小型问答数据库中提取。

## 理论知识

教程资源-知识图谱导论-bilibili

### 什么是知识图谱？先摘一段百度百科：

知识图谱（Knowledge Graph），在图书情报界称为知识域可视化或知识领域映射地图，是显示知识发展进程与结构关系的一系列各种不同的图形，用可视化技术描述知识资源及其载体，挖掘、分析、构建、绘制和显示知识及它们之间的相互联系。知识图谱是通过将应用数学、图形学、信息可视化技术、信息科学等学科的理论与方法与计量学引文分析、共现分析等方法结合，并利用可视化的图谱形象地展示学科的核心结构、发展历史、前沿领域以及整体知识架构达到多学科融合目的的现代理论。它能为学科研究提供切实的、有价值的参考。

知识图谱

智能问答分类：按答案来源分

智能问答系统

知识图谱问答

表格问答

文本问答

社区问答

视觉问答



	TABLE	TABLE		ROW	ROW
1. Product width	0.00	-0.00	4.0	4.0	0.00
2. Product length	0.00	-0.00	0	0	0.00
3. Product floor length	0.00	-0.00	1.00	1.00	0.00
4. Color Width	0.00	-0.00	1.00	1.00	0.00
5. Machine Circumference	0.00	-0.00	10	10	0.00
6. Product Width	0.00	-0.00	1.0	1.0	0.00
7. Product width	0.00	-0.00	0	0	0.00
8. Product Circumference	0.00	-0.00	40	40	0.00
9. Weight Circumference	0.00	-0.00	40	40	0.00

Question: "Who's smiling like Nivona from the US in the late afternoon was one staying at the Hotel Tropic. By the time we get there and check in it will be rainy evening. Can anyone recommend a good restaurant nearby so that we don't have to venture too far on our first night after a long journey? The Hotel is at Calle Obispo y Cuba. Ciudad de La Habana. Any suggestions appreciated."

Answers:

1. "Los Mirlos" (790\_R\_7906)
2. "La Mirlo" (790\_R\_7906)

Who is wearing glasses?

man

woman





## 经验

- 文本回答 + 知识图谱问答
- 视觉问答 + 知识图谱问答

## 知识图谱问答评测数据集

- QALD
  - 多语种问答
  - 基于链接数据问答
- WebQuestions
  - 6000+个问答对
- SimpleQuestions
  - 简单 常用数据集
  - 10W+个自然问题
- MetaQA
  - 垂直电影领域知识图谱
  - 40w+问题

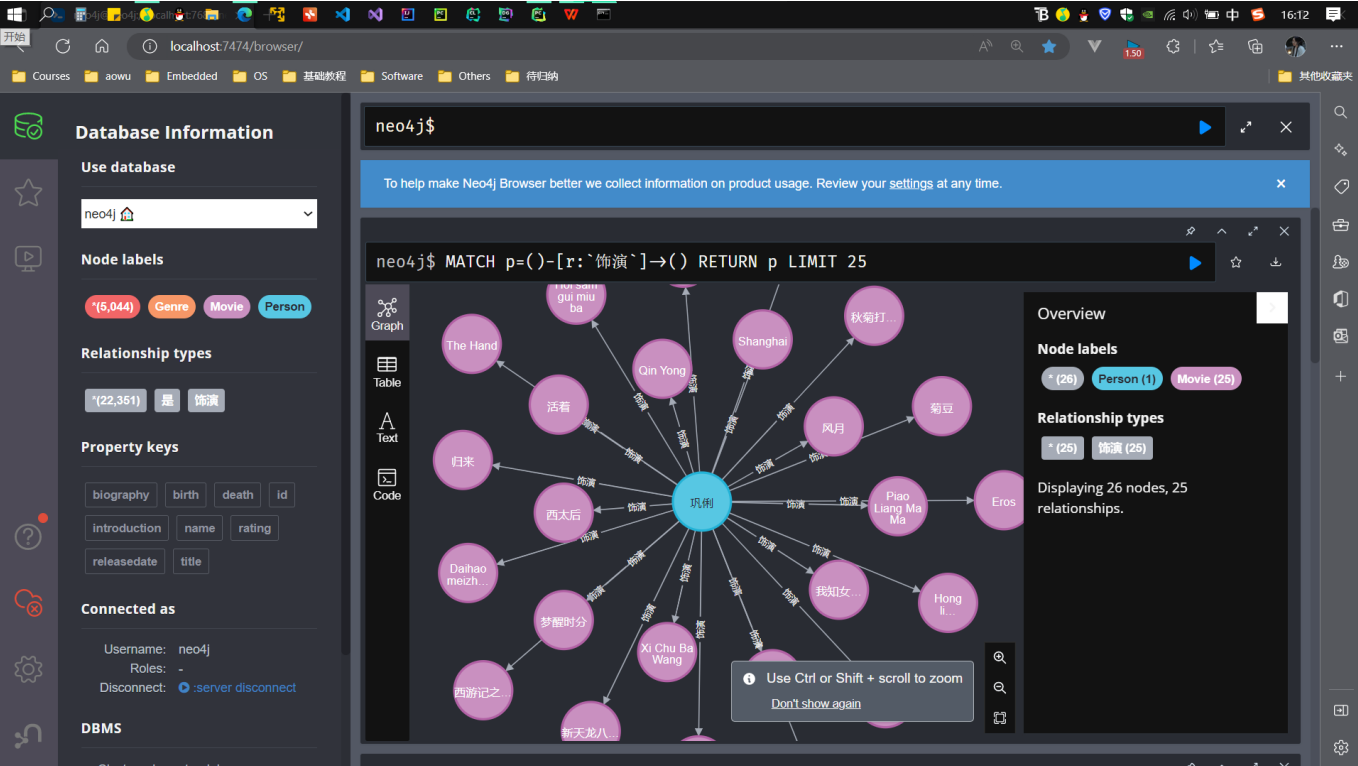
- Others

数据集	规模	数据形式	其他
ATIS	5410	问答对	
JOB5	640	问题+结构化查询	
GeoQuery	880	问答对+逻辑表达式	
QALD Series	150-300/年	问答对+结构化查询	
Free917	917	问答对+逻辑表达式	
WebQuestions	3782/2037	问答对	
WebQuestionsSP	3098/1639	问答对+结构化查询	
SimpleQuestions	108,442	问答对	
ComplexQuestions	2,100	问答对	
WikiMovies	100k	问答对	
MSParse	100k	问题+逻辑表达式	
GraphQuestions	5166	问题+结构化查询	
LcQuAD	5k	问题+结构化查询	
LcQuAD 2.0	30k	问题+结构化查询	

neo4j使用简介

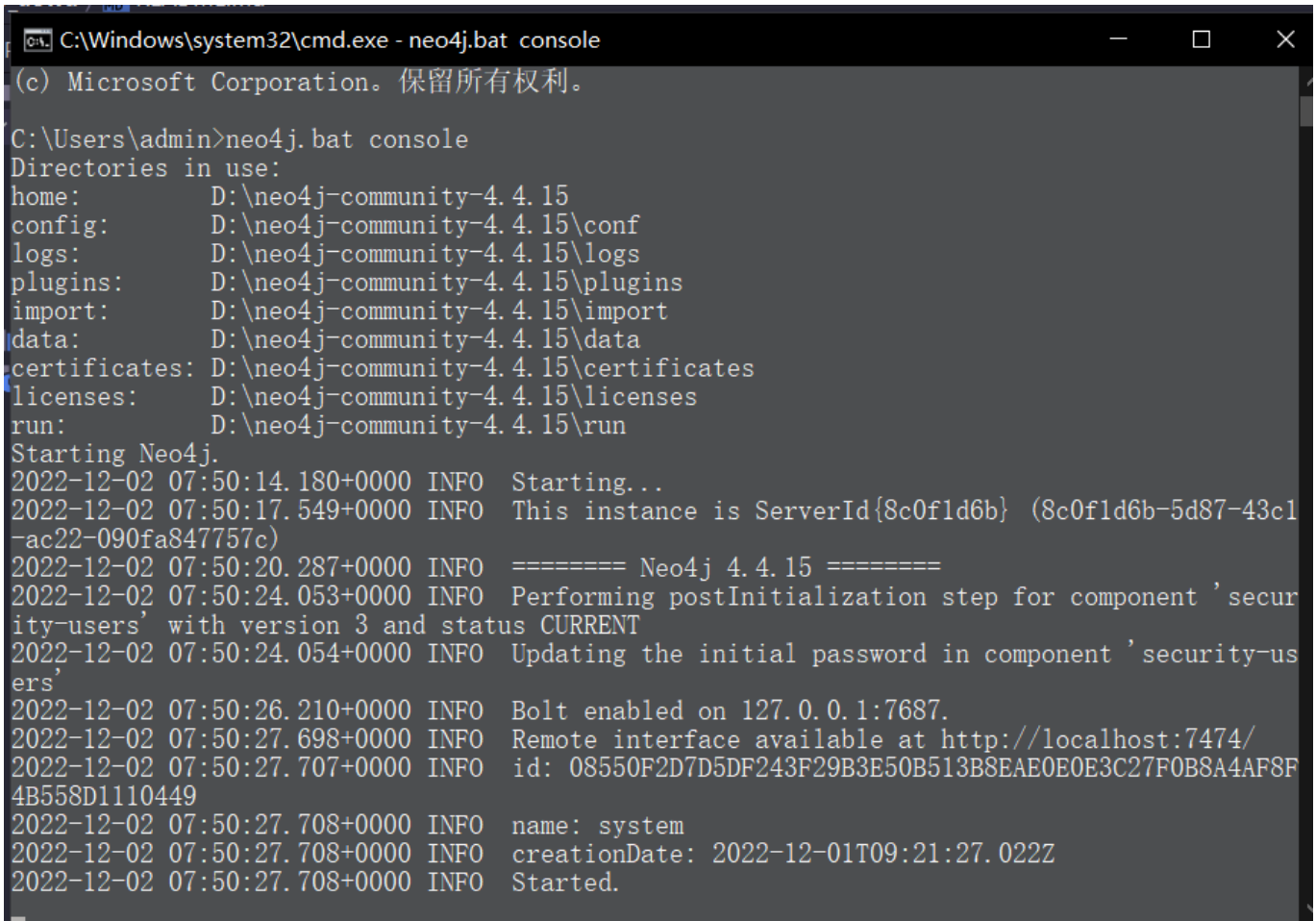
neo4j下载链接：[neo4j download](#)

操作界面展示



启动命令-CMD命令行

```
$ neo4j.bat console
```



```
C:\Windows\system32\cmd.exe - neo4j.bat console
(c) Microsoft Corporation。保留所有权利。

C:\Users\admin>neo4j.bat console
Directories in use:
home:          D:\neo4j-community-4.4.15
config:        D:\neo4j-community-4.4.15\conf
logs:          D:\neo4j-community-4.4.15\logs
plugins:       D:\neo4j-community-4.4.15\plugins
import:        D:\neo4j-community-4.4.15\import
data:          D:\neo4j-community-4.4.15\data
certificates:  D:\neo4j-community-4.4.15\certificates
licenses:      D:\neo4j-community-4.4.15\licenses
run:           D:\neo4j-community-4.4.15\run
Starting Neo4j.
2022-12-02 07:50:14.180+0000 INFO  Starting...
2022-12-02 07:50:17.549+0000 INFO  This instance is ServerId{8c0fld6b} (8c0fld6b-5d87-43c1-ac22-090fa847757c)
2022-12-02 07:50:20.287+0000 INFO  ===== Neo4j 4.4.15 =====
2022-12-02 07:50:24.053+0000 INFO  Performing postInitialization step for component 'security-users' with version 3 and status CURRENT
2022-12-02 07:50:24.054+0000 INFO  Updating the initial password in component 'security-users'
2022-12-02 07:50:26.210+0000 INFO  Bolt enabled on 127.0.0.1:7687.
2022-12-02 07:50:27.698+0000 INFO  Remote interface available at http://localhost:7474/
2022-12-02 07:50:27.707+0000 INFO  id: 08550F2D7D5DF243F29B3E50B513B8EAE0E0E3C27F0B8A4AF8F4B558D1110449
2022-12-02 07:50:27.708+0000 INFO  name: system
2022-12-02 07:50:27.708+0000 INFO  creationDate: 2022-12-01T09:21:27.022Z
2022-12-02 07:50:27.708+0000 INFO  Started.
```

## 打开操作界面

浏览器输入 `http://localhost:7474/`  
登录 - 初始用户: neo4j 初始密码: neo4j  
目前 - 用户名: neo4j 密码: 123456

## neo4j使用-参考博客链接

1. [Neo4j 安装、使用教程](#)
2. [neo4j新手教程](#)
3. [手把手教你快速入门知识图谱 - Neo4j教程](#)

## python操作简介

```
from py2neo import Graph, Node, Relationship, NodeMatcher
import py2neo

g = Graph("http://localhost:7474/", user='neo4j', password='123456')
# 结点
```

```

test_node_1 = Node("Person", name='P1')
test_node_2 = Node("Person", name='P2')
# 添加属性
test_node_1['age'] = 18
test_node_1['sex'] = '男'
test_node_2['age'] = 19
test_node_2['sex'] = '女'
# 创建结点
g.create(test_node_1)
g.create(test_node_2)
# 创建覆盖式结点
g.merge(test_node_1, "Person", "name")
g.merge(test_node_2, "Person", "name")
# 创建关系
friend = Relationship(test_node_1, 'friend', test_node_2)
g.merge(friend, "Person", "name")
# 查询结点
matcher = NodeMatcher(g)
print(matcher.match("Person", name="P1").first())

```

## 建立图谱

```

import csv
from py2neo import Graph, Node, Relationship, NodeMatcher
import py2neo

g = Graph("http://localhost:7474/", user='neo4j', password='123456')
with open('triples.csv', 'r', encoding='utf-8') as f:
    reader = csv.reader(f)
    for item in reader:
        if reader.line_num==1:
            continue
        print("当前行数", reader.line_num, "当前内容: ", item)
        start_node = Node("Person", name=item[0])
        end_node = Node("Person", name=item[1])
        relation = Relationship(start_node, item[3], end_node)
        g.merge(start_node, "Person", "name")
        g.merge(end_node, "Person", "name")
        g.merge(relation, "Person", "name")

```

## 数据查询

```
$ MATCH (p: Person {name: "贾宝玉"})-[k: 丫鬟]-(r) return p,k,r
```

## 查询所有有关系的节点

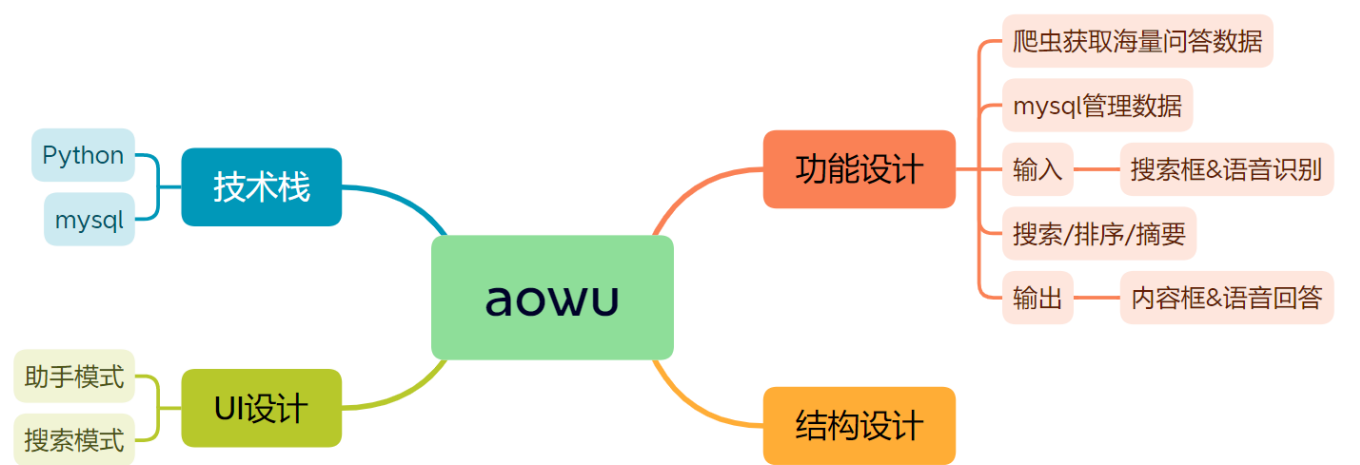
```
$ MATCH (a)--() RETURN a
```

删除数据库中以往的图

```
$ MATCH (n) DETACH DELETE n
```

功能设计

待完善



模块说明

主函数

```
$ Src/main.py
```

演示效果

开始界面



总结

参考链接

[个人Github](#)

