# 第5章 基于知识库的问答系统

通过上一章节的学习，我们基于常用问答数据构建了一个简单的问答系统，在实际生成实践中，这种基于信息检索以及数据查询的方式有很大的不足。首先在构建整理问答数据的过程就需要耗费庞大的人力物力，其次这种方式仅仅适用于问题规模小且集中，存在少量语义异构信息的场景，而且对于精确率和召回率的要求较高。

对于用户特殊且复杂的信息需求，如何从当前多样化且非结构化的信息中获取用户所需要的答案，同时可以对用户的问题进行自动化的语义理解？本章我们将介绍基于知识库的问答系统，这种问答系统不仅可以实现对复杂问题的语义理解，而且可以对不同知识库的知识进行融合，并针对复杂问题进行相应的知识推理。

本章主要涉及到的知识点有：

* 基于知识库的问答系统介绍。
* 知识图谱介绍。
* Neo4j数据库安装配置。
* 导入数据到图数据库。
* Neo4j数据的简单操作增删改查。

注意：本章代码地址：xxx

## 5.1 基于知识库的问答系统

本节首先介绍什么是基于知识库的问答系统以及主流的基于知识库的问答系统的实现方法。

### 5.1.1 基于知识库的问答系统简介

基于知识库的问答系统（Knowledge Based Question Answering，KBQA，下文统称为KBQA）是目前应用最广泛的问答系统之一，适用于人们生活的方方面面，再医疗、金融、保险、零售等行业建立相应专业知识的问答系统，可以给用户提供更好的服务。

知识库（Knowledge Base，KB）是用于知识管理的一种特殊的数据库，用于相关领域知识的采集、整理和提取。知识库中的知识来源于各个领域的专家，是求解问题所需领域知识的集合，包括一些基本事实、规则和其他相关信息。知识库的表示形式通常是一个对象模型（object model），通常称为本体，包含一些类、子类和实体。不同于传统的数据库，知识库中存放的知识蕴含特殊的知识表示，其结构比传统数据库更加复杂，可以存放更多复杂语义表示的数据。

KBQA是依托于知识库中的专业知识而建立的问答系统，也是当前最主流的问答系统。常见的知识库如Freebase、DBpedia等，知识库一般采用RDF格式对其中的知识进行表示。此外还有一些非结构化信息的知识库如维基百科、百度百科等。

KBQA一般包含问句理解模块、答案信息抽取模块、答案排序和生成模块等核心模块，其基本架构图如图5.1所示。



图5.1 KBQA基本架构图

KBQA系统中的问句理解模块主要完成提取问题中的实体信息的任务，答案信息抽取模块通过在知识库中查询以该实体节点为中心的知识库子图，并依据某些规则或模板从提取到的子图中抽取相应的信息，得到表征问题和候选答案特征的特征向量，最后将候选答案的特征向量作为分类模型的输入，通过模型输出的分类结果来筛选答案。KBQA系统的各模块间的关系如图5.2所示，其中主要包括问句分析（Question Analysis）、短语映射（Phrase Mapping）、语义消歧（Semantic Disambiguation）和构建查询（Query Construction）。



图5.2 KBQA各模块间关系

通过信息抽取的方法从知识库中提取有价值的信息，并构建知识图谱。对于用户输入的问题通过问句理解抽取其问题中的实体信息，再结合知识推理等方法为用户提供更深层次的语义理解的答案。

### 5.1.2 主流的KBQA实现方法

注意：该部分的数据爬取并非本章节重点，爬取的数据集读者在开源项目中直接获取

最终获取到一个包含四个字段的数据集。数据字段说明及部分数据如表4.1所示。

表4.1 保险问答数据源数据格式字段介绍

|  |  |  |
| --- | --- | --- |
| 字段 | 说明 | 备注 |
| Title | 问题的标题 |  |
| Question | 问题内容 | 可为空 |
| Reply | 回复内容 |  |
| Is\_best | 是否为页面上显示的最佳回答 | 1表示改回答为最佳回答，反之为0 |

## 5.2 基于知识库的问答系统

### 5.1.1 知识图谱

想象一下，当我们在网上购物需要支付的时候，发现无法支付或者支付失败，这个时候我们会先去找一个小助手帮忙解答。如图4.1为支付宝的客服助手展示页面，图4.2为客服助手的问题搜索页面，图4.3为客服助手的问题搜索结果展示页面。



图4.1 支付宝客服助手



图4.2 支付宝客服助手搜索页面



图4.3 支付宝客服助手搜索问题结果

通过在输入框描述我们遇到的问题，系统会自动给出答案，帮助我们解决问题。

这样一个系统即为智能客服，广泛应用于电商、政务等服务行业，当我们遇到问题时，智能客服可以高效便捷地帮助我们。

智能客服的实现是基于一个庞大的常见问题答案库，也就是FQA。通过构建问答库，然后将用户输入的问题与问答库中的问题一一比较，选取相同及相似问题作为候选问题，并将候选问题的答案进行筛选后返回给用户。

### 5.1.2 知识图谱的应用场景

### 5.1.3 常用的图数据库及优缺点

本项目中我们采用从百度知道人工爬取的保险相关问答数据，该数据集中共包含8000余条保险相关问题及答案，经过数据清理，去除了原数据集中的问题id、url 、qid、reply\_t、user字段，对question、reply做了脱敏处理。

注意：该部分的数据爬取并非本章节重点，爬取的数据集读者在开源项目中直接获取

最终获取到一个包含四个字段的数据集。数据字段说明及部分数据如表4.1所示。

表4.1 保险问答数据源数据格式字段介绍

|  |  |  |
| --- | --- | --- |
| 字段 | 说明 | 备注 |
| Title | 问题的标题 |  |
| Question | 问题内容 | 可为空 |
| Reply | 回复内容 |  |
| Is\_best | 是否为页面上显示的最佳回答 | 1表示改回答为最佳回答，反之为0 |

部分数据如下：

title,question,reply,is\_best

最近在安邦长青树中看到什么豁免，这个是什么意思？,,您好，这个是重疾险中给予投保者的一项权利，安\*长青树保障责任规定，投保者可以享受多次赔付，豁免等权益。也就是说不同轻症累计5次赔付，理赔1次轻症豁免后期所交保费，人性化的设计，无需加保费。,1

和老婆利用假期去澳\*探亲，但是第一次去不大熟悉，有没有相关保险呢？,,您好，HUTS保险中的乐游全球（探亲版）-慧择旅游保险澳新计划是澳\*新西兰探亲专属保障，承保年龄可达90周岁，含有50万高额医疗保障，完全满足境外医疗保障需求，需要注意的是这款产品仅承保出行目的为境外探亲的人群，理赔时需提供相关签证或亲属关系证明等,0

HUTS中有没有适合帆船比赛的保险，我男朋友这周就要开始了,,您好，水上运动比赛，尤其是带有奖金的比赛一般承保的公司比较少。不过，HUTS保险中的众行天下-水上运动保险赛事版B就是适合帆船等水上比赛的产品，含户外溺水保障，是水上运动专属定制的保障，意外住院有津贴，保障期限灵活可选，还可以投保有奖金的赛事，您可以根据情况看看。,1

计划端午节和男朋友自驾去九\*山，买保险三天要多少钱？,,"您好，端午出行的人比较多，而且自驾存在一定风险，所以有保险意识还是很好的。考虑到价格以及保障内容等相关因素，您可以看看HUTS保险中的畅玩神州-慧择旅游保险计划三，适合驾驶私家车走南闯北国内旅游，自驾意外累计赔付,承保的范围也较为广泛，适合带家人出游，保障全面，三天仅需75元，性价比还是蛮高的。",1

获取到数据后，我们需要对数据进行进一步处理，为了提高我们智能客服的问题回复质量，我们仅将数据集中回答为最佳回答的问答数据抽取出来。具体实现代码如下：

**代码4.1 数据预处理代码**

#!/usr/bin/env python

# \_\*\_ coding:utf-8 \_\*\_

import csv

def read\_data():

'''

读取保险数据源文件数据，将其作为数组返回

:return:

'''

# 源数据文件地址

csv\_dir = './baoxianzhidao\_filter.csv'

# 存储读取到的保险问题描述

insurance\_ques = []

# 存储读取到的保险问题对应答案

insurance\_ans = []

# 读取源数据

with open(csv\_dir, 'r', newline='', encoding='utf-8') as csvfile:

csv\_reader = csv.reader(csvfile)

for row in csv\_reader:

# 选取源数据中答案为最优答案的问答数据

if len(row) == 4 and row[3] == '1':

# 若question字段非空，则将问题title字段与question字段拼接为问题描述

# 若question字段为空，则将问题title字段作为问题描述

if row[1]:

insurance\_ques.append(row[0] + row[1])

else:

insurance\_ques.append(row[0])

# 选取replay字段作为答案描述

insurance\_ans.append(row[2])

return insurance\_ques, insurance\_ans

def save\_data(insurance\_ques, insurance\_ans):

'''

存储保险问答数据

:param insurance\_ques:

:param insurance\_ans:

:return:

'''

# 遍历存储问答数据为csv格式

for idx in range(len(insurance\_ans)):

with open('insurance\_data.csv', 'a', newline='', encoding='utf-8') as csvfile:

spamwriter = csv.writer(csvfile)

spamwriter.writerow([insurance\_ques[idx], insurance\_ans[idx]])

if \_\_name\_\_ == '\_\_main\_\_':

insurance\_ques, insurance\_ans = read\_data()

save\_data(insurance\_ques, insurance\_ans)

最终获取到保险问答数据格式如表4.2所示：

表4.2 保险问答数据格式字段介绍

|  |  |  |
| --- | --- | --- |
| 字段 | 说明 | 备注 |
| insurance\_ques, | 保险数据问题 |  |
| insurance\_ans | 保险数据问题相应答案 |  |

insurance\_ques, insurance\_ans

最近在安邦长青树中看到什么豁免，这个是什么意思？,您好，这个是重疾险中给予投保者的一项权利，安\*长青树保障责任规定，投保者可以享受多次赔付，豁免等权益。也就是说不同轻症累计5次赔付，理赔1次轻症豁免后期所交保费，人性化的设计，无需加保费。  
HUTS中有没有适合帆船比赛的保险，我男朋友这周就要开始了,您好，水上运动比赛，尤其是带有奖金的比赛一般承保的公司比较少。不过，HUTS保险中的众行天下-水上运动保险赛事版B就是适合帆船等水上比赛的产品，含户外溺水保障，是水上运动专属定制的保障，意外住院有津贴，保障期限灵活可选，还可以投保有奖金的赛事，您可以根据情况看看。  
计划端午节和男朋友自驾去九\*山，买保险三天要多少钱？,"您好，端午出行的人比较多，而且自驾存在一定风险，所以有保险意识还是很好的。考虑到价格以及保障内容等相关因素，您可以看看HUTS保险中的畅玩神州-慧择旅游保险计划三，适合驾驶私家车走南闯北国内旅游，自驾意外累计赔付,承保的范围也较为广泛，适合带家人出游，保障全面，三天仅需75元，性价比还是蛮高的。"  
端午我们准备要举行赛龙舟，说是要份保险，什么好,您好，赛龙舟是一项比较传统的活动，很有意义。不过由于是在水上活动，建议安全保障还要做足，HUTS保险中有针对水上运动风险特别定制的众行天下-水上运动保险，可以针对这种赛事进行保障，含有意外住院有津贴以及一系列保障，性价比较高，关键是费用也比较实惠。  
老婆买了安\*长\*树，她在网上投保的，以后缴费怎么办,您好，这点是不用担心的。投保后保险公司会在约定的保险费交纳日从消费者购买时填写的银行账号中划扣当期应交的安\*长青树重疾险的保险费，所以您老婆是不用亲自去保险公司缴费的。

## 5.2 Neo4j数据库的安装配置

Neo4j 是目前最流行的图形数据库，支持完整的事务，在属性图中，图是由顶点（Vertex），边（Edge）和属性（Property）组成的，顶点和边都可以设置属性，顶点也称作节点，边也称作关系，每个节点和关系都可以由一个或多个属性。Neo4j创建的图是用顶点和边构建一个有向图，通过查询语言cypher来遍历获取图上的节点信息。

关系型数据库只对单个Join操作进行优化查询，而多重Join操作查询的性能显著下降。图形数据库适合查询关系数据，由于图形遍历的局部性，不管图形中由多少节点和关系，根据遍历规则，Neo4j只访问与遍历相关的节点，不受到总数据集大小的影响，从而保持期待的性能；相应地，遍历的节点越多，遍历速度越慢，但是变慢是线性的，这使得图形数据库不适合做海量数据统计分析。对于存在大量丰富关系的数据，遍历的性能不受图形数据量大小的影响，这使得Neo4j成为解决图形问题的理想数据库。

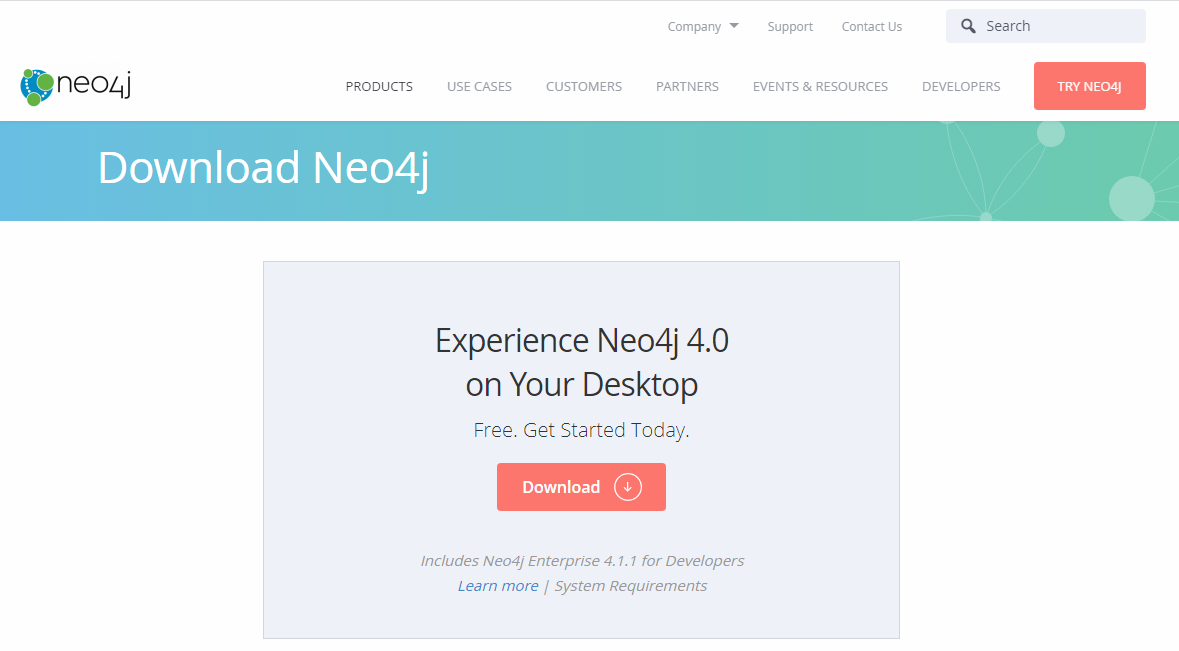
下面介绍Neo4j的安装配置：

### 5.2.1 下载安装Neo4j

Neo4j是基于Java的图形数据库，运行Neo4j需要启动JVM进程，因此需要预先安装JDK，且版本需要在1.8以上。

JDK的安装这里暂不介绍

点击下载页面



激活密钥

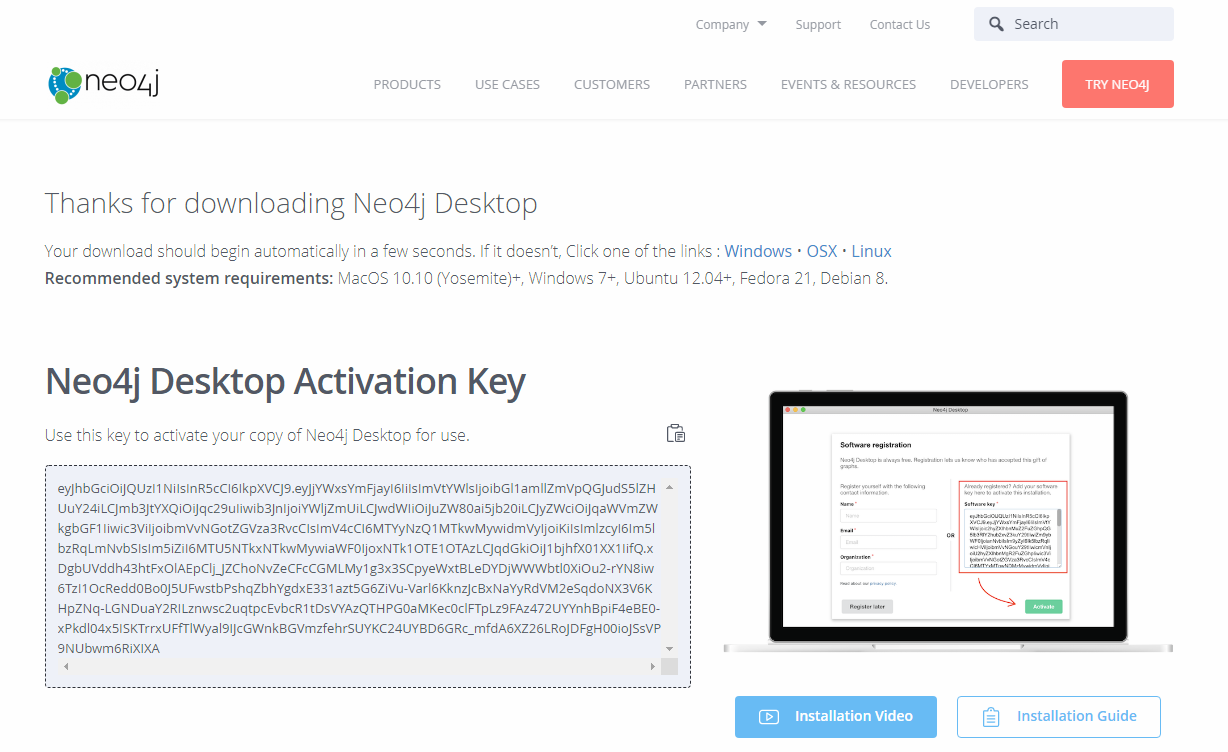
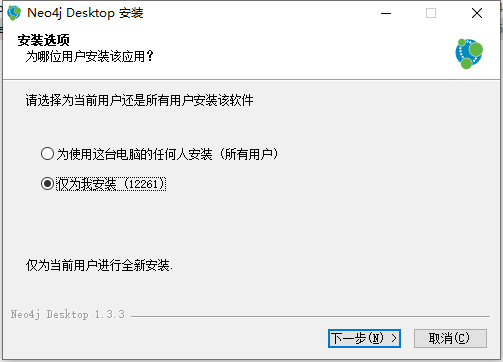
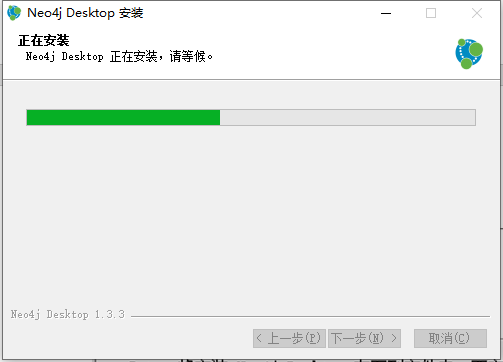
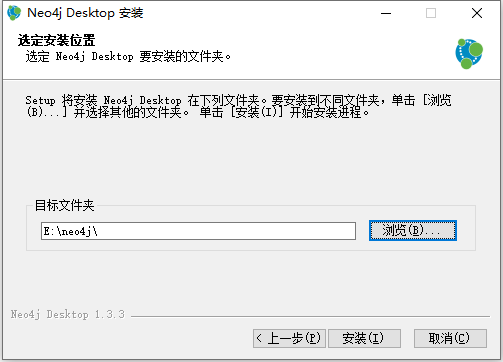


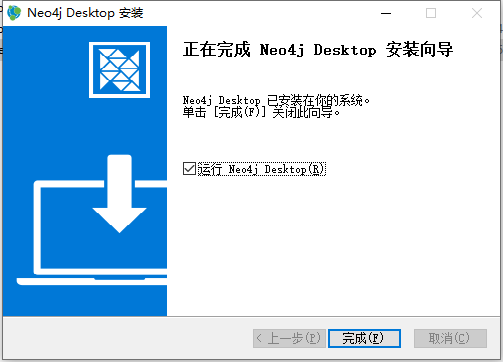
图4.4 智能客服系统架构

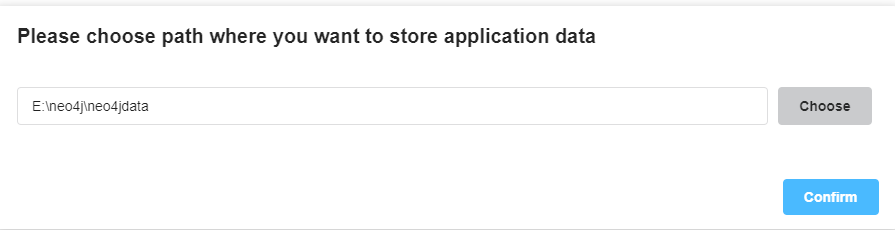
从官网下载最新版Neo4j社区版并解压，这里安装的是neo4j-community-3.5.5-windows

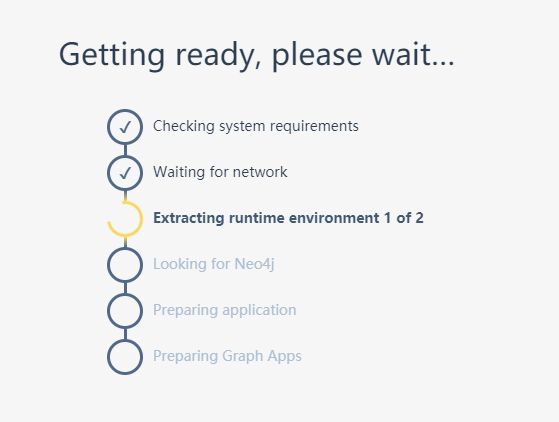
<https://neo4j.com/download/>





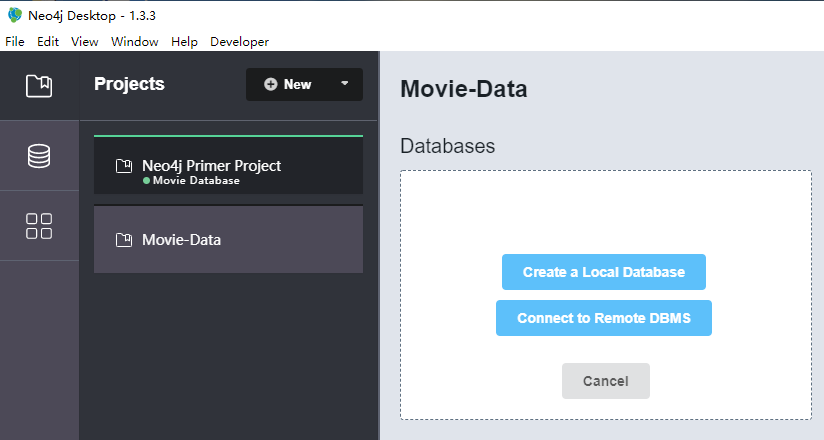




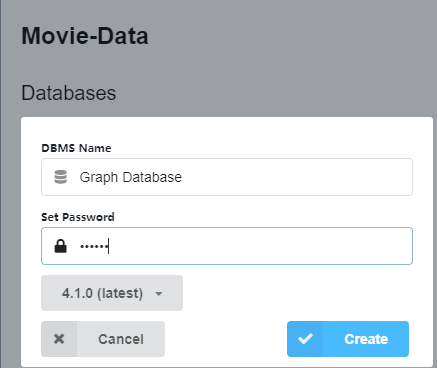


### 5.2.2 运行Neo4j

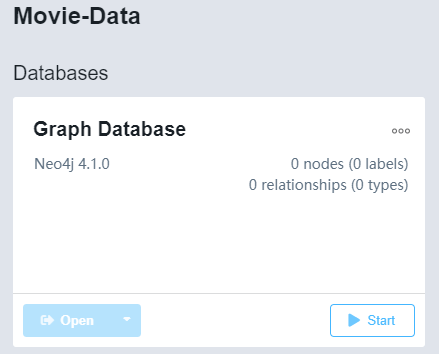
创建Project Movie-Data

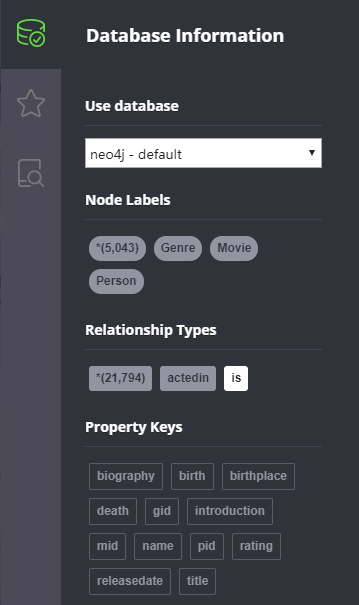


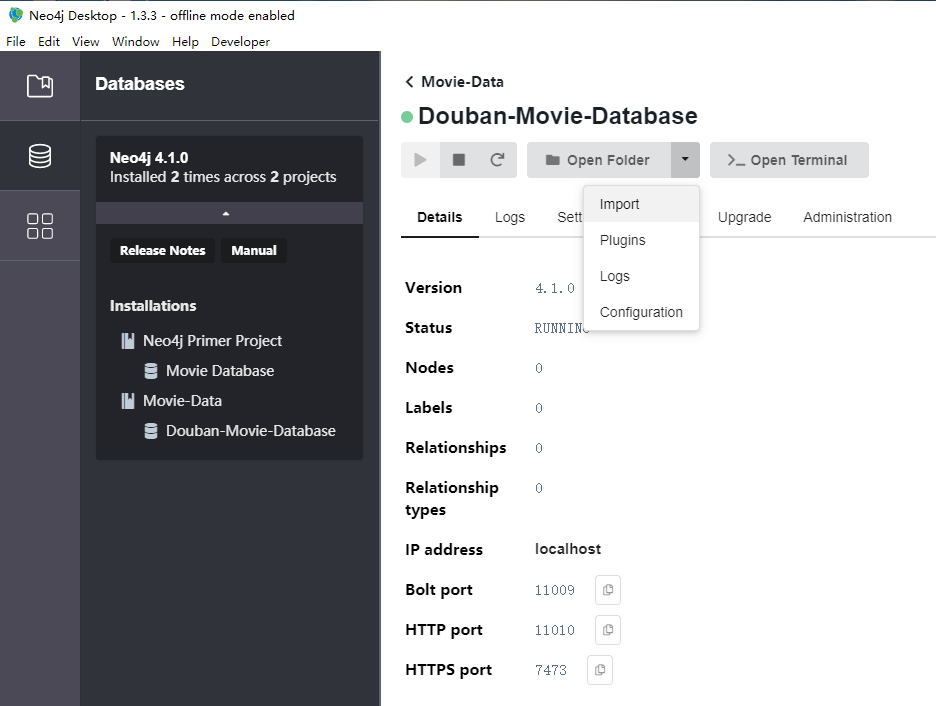
创建本地数据库 设置密码

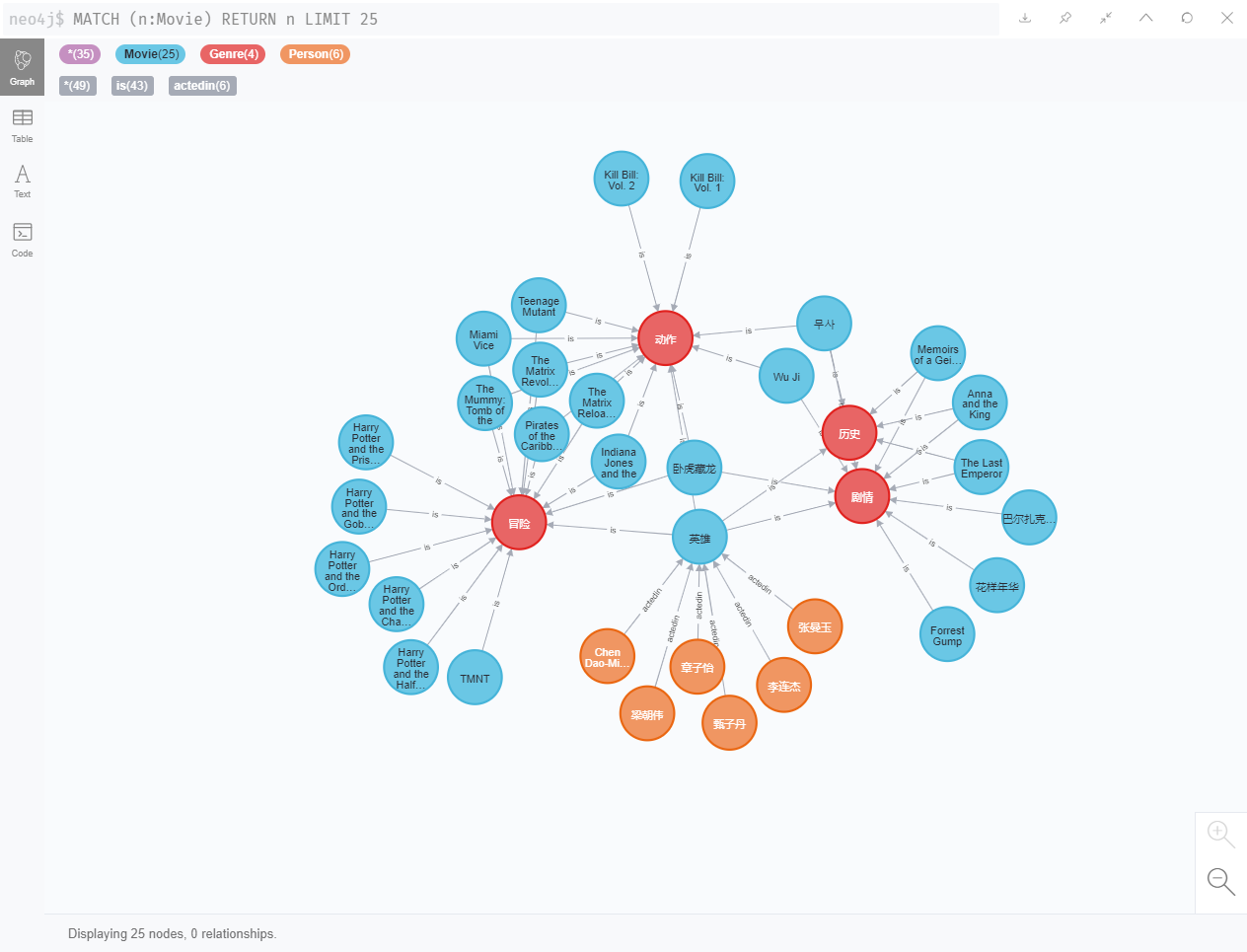


启动数据库









### 4.2.4 封装API并部署在服务器

为了便于访问我们的智能客服服务，我们将我们的代码封装成一个API，并部署在服务器上，是的通过http请求即可访问获取。这里采用aiohttp来实现服务部署。相关知识可以访问：<https://docs.aiohttp.org/en/stable/>

智能客服服务部署在我们本地，接口设置为9010，故访问地址为：<http://127.0.0.1:9010>

通过GET请求进行访问，参数为question

项目整体代码在如下地址提供：xxx

### 4.2.5 智能客服效果展示

服务部署以后，我们通过在浏览器输入想查询的问题，系统会给出相应回复。问题数据通过在访问地址中拼接?question=XXX的方式传递。

如：

案例1：

<http://127.0.0.1:9010/?question=>天热了，很想去潜水，除了装备、教练以及费用外还要什么保险？

{

"系统回复": "",

"相似问题": "",

"推荐答案": "您好！除了您所说的准备、教练等专业人士陪同之外，购买一份合适的户外运动保险很有必要，毕竟潜水存在一定风险。目前国内关于潜水运动的保险并不多，一般都要求是下潜不超过18m。具体您可以了解下HUTS保险中的适合国外潜水等海岛相关水上运动的产品众行天下-水上运动保险，是海岛潜水专属的保障。主要是因为它同时包括潜水意外医疗补偿以及紧急医疗运送，是高风险水上运动的理想选择。"

}

案例2：

<http://127.0.0.1:9010/?question=>本周末公司组织一次大型户外拓展活动，100人什么HUTS保险合适？

{

"系统回复": "小助手没有这个问题的答案呢，给您推荐以下相似问题及答案以供参考哦~\n",

"相似问题": "相似问题：本周末公司会组织200人的户外拓展活动，什么保险能保障员工安全？\n",

"推荐答案": "推荐答案：您好，关于户外拓展活动，存在一定的风险，并且贵公司有200人参加，所以建议可以考虑下HUTS保险中的众行天下-拓展训练保险，是专门针对户外运动这块的保险。最高可赔付10万元，关键是价格比较实惠，适合集体投保。"

}

案例3：

<http://127.0.0.1:9010/?question=>太阳系有几个行星呢

{

"系统回复": "啊哦，小助手还没有掌握这方面的知识呢，我会将您的问题记录下来，并尽快找到专业的答案。",

"相似问题": "",

"推荐答案": ""

}

## 5.3 导入数据到Neo4j数据库

将已经整理好的电影数据导入Neo4j数据库。

**代码5.9 导入CSV文件**

//导入节点 电影类型 == 注意类型转换

LOAD CSV WITH HEADERS FROM "file:///genre.csv" AS line

MERGE (p:Genre{gid:toInteger(line.gid),name:line.gname})

//导入节点 演员信息

LOAD CSV WITH HEADERS FROM 'file:///person.csv' AS line

MERGE (p:Person { pid:toInteger(line.pid),birth:line.birth,

death:line.death,name:line.name,

biography:line.biography,

birthplace:line.birthplace})

// 导入节点 电影信息

LOAD CSV WITH HEADERS FROM "file:///movie.csv" AS line

MERGE (p:Movie{mid:toInteger(line.mid),title:line.title,introduction:line.introduction,

rating:toFloat(line.rating),releasedate:line.releasedate})

// 导入关系 actedin 电影是谁参演的 1对多

LOAD CSV WITH HEADERS FROM "file:///person\_to\_movie.csv" AS line

match (from:Person{pid:toInteger(line.pid)}),(to:Movie{mid:toInteger(line.mid)})

merge (from)-[r:actedin{pid:toInteger(line.pid),mid:toInteger(line.mid)}]->(to)

//导入关系 电影是什么类型 == 1对多

LOAD CSV WITH HEADERS FROM "file:///movie\_to\_genre.csv" AS line

match (from:Movie{mid:toInteger(line.mid)}),(to:Genre{gid:toInteger(line.gid)})

merge (from)-[r:is{mid:toInteger(line.mid),gid:toInteger(line.gid)}]->(to)

## 5.4 Neo4j数据的简单操作增删改查