|  |  |
| --- | --- |
| **自然语言处理及应用**  **实验报告** | |
|  | |
|  |  |
| **名称** | 基于知识图谱的电影问答系统 |
| **姓名** | 黄隆宁 |
| **班级** | 硕1224班 |
| **学号** | 3121155049 |
| Email | hln18773372567@stu.xjtu.edu.cn |
| **日期** | 2020-11-11 |

# 实验目的

随着互联网数据的不断增多，数据实体之间的关系日渐复杂。我们在使用问答系统的时候，希望能以人类自然语言的形式提出问题，系统能正确的分析提问者的意图，并选择正确的数据以合适的表述形式回答。对于我们的电影知识问答系统，则希望系统能流畅回答例如电影名称、评分、类别、演员等信息。

为了实现更加自然简单的交互方式，如果问答系统采用传统的关系型数据库，数据之间的关系需要通过多表连接得到，这些连接操作运算量大，并且极大程度依赖人工设计的复杂规则。我们在本实验通过知识图谱的引入，使用图数据库代替关系数据库，实现有效的数据实体和数据关系的储存。

本实验将以豆瓣电影数据作为主要数据源，建立图数据库形式的知识图谱，并实现电影信息问答系统。

本实验的全部代码均托管于GitHub，本人在项目中负责绝大部分系统设计和代码实现，以及汇报展示。

<https://github.com/HLNN/NLP>

在Google Colab中运行问答系统Demo：

<https://colab.research.google.com/github/HLNN/NLP/blob/master/Final/movie_QA_with_KQ.ipynb>

# 实验环境

Ubuntu 18.04

Python3.8

调用build-in库：

time：用于算法运行时间统计

math：用于数学函数的计算

os：文件系统及数据文件的管理和访问

csv,xml：中间数据结构的解析和处理

pip依赖：

numpy：绝大部分的数据表示和数学计算

nltk：用于自然语言数据的预处理

jieba：用于自定义分词和词性分析

sklearn：用于数据集特征提取及分类

py2neo, neo4j：用于从Neo4j图数据库中读写数据

flask：实现基于web的问答接口调用

# 实验方法

**数据准备：**

我们的电影知识库来源于IMDB数据库，该知识库包括电影、演员电影类别等信息。我们获取到的数据一共包括500多位演员、4500多部电影信息、以及14000多条演员出演电影数据等。

**数据导入：**

下一步我们将获得的数据导入Neo4j图数据库中，数据使用Google表格发布为在线CSV，并通过我们的csv2neo4j.py，将读取的csv格式数据导入到Neo4j图数据库中。

我们的数据一共分为五个csv表：genre.csv用于记录电影实体类别；person.csv用于记录演员实体信息；movie.csv用于记录电影实体信息；preson\_to\_movie.csv用于记录演员参演电影的关系信息；movie\_to\_genre.csv用于记录电影所属类别的关系信息。

图片包含 游戏机

描述已自动生成

前三类实体信息将导入图数据库成为实体，后两类关系信息将导入图数据库成为关系。



**问题模板抽象：**

用户在向系统提问时，相同含义的问题，可能会有不同的表述方式。针对自然语言的表意多样性，问答系统需要对自然语言问题进行抽象，转换为问答系统能处理的形式。

对此我们将电影问答系统，可能会遇到的问题，整理为13个问题类别。针对每一个问题，使用不同的表述方法提出一些问题模板。



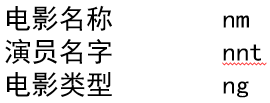
比如同样询问电影评分，我们设置了如下问题：



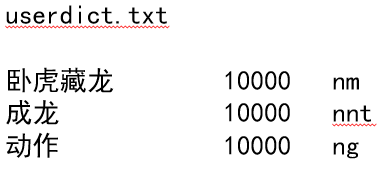
**实体抽象：**

除了问题模板抽象之外，我们还对实体类别进行了抽象。比如当用户询问电影《卧虎藏龙》的评分和电影《无间道》的评分时，用户希望知道的都是具体一部电影的评分。而不同电影的不同名字本身，可能会影响到对问题的正确分类。

因此我们为图数据库中的每一类实体，设置了自定义词性。电影的词性为nm，演员的词性为nnt，电影类别的词性为ng。



同时读取图数据库中的所有实体名称，将他们写入自定义词表中。这样jieba在分词和词性标注过程中，遇到这些实体名称时，会将其识别为我们自定义的词性。



将用户提问输入分类器之前，将所有的实体名称，都由其类别词性代替，这样不同的同类实体名称不会影响分类器精度。

**问题分类：**

经过问题模板抽象和实体抽象之后，问题被送入朴素贝叶斯多分类器进行分类。该分类器使用问题模板抽象部分提到的自定义问题模板进行训练。训练数据使用TF-IDF方法向量化。自定义的问题模板使用了不同的表述方式，因此分类器能稳定处理不同的提问方式。

因为朴素贝叶斯分类器和数据向量化，在之前的作业中都有涉及。因此该部分均使用sklearn中模块实现。

**微信公众号接口：**

问答系统通常时一问一答的对话形式，因此我们选择微信公众号作为系统载体。公众号的能提供沉浸式的一对一聊天形式的问答，同时依托微信的巨大用户量，相对网页和APP，能更为直接方便的触达用户。

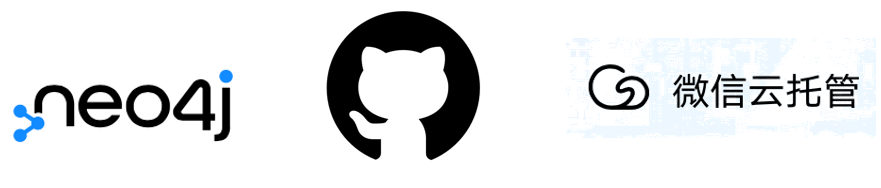
图形用户界面, 文本, 应用程序, 聊天或短信

描述已自动生成

问答系统使用Flask实现后台API，用户发给微信公众号的消息经微信发送到系统所在的服务器后，系统将回答返回给微信服务器，再又微信后台发送给用户。

**流水线部署：**

最后该问答系统实现了全基于云平台的自动化流水线部署。项目代码托管于GitHub，图数据库使用Neo4j的在线沙盒，问答系统运行于微信云托管。



代码更新推送到GitHub之后，会触发微信云托管，微信云托管后台自动拉取代码、构建镜像并发布。代码部署功能全部基于云服务且完全自动化。

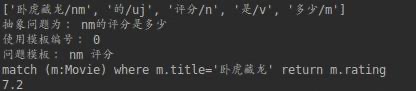
1. **实验结果**

问答系统部署于微信公众号之后，效果如下图：

图形用户界面, 文本, 应用程序, 聊天或短信

描述已自动生成

其中上方为用户发送的问题：“卧虎藏龙的评分时多少？”，下方为微信公众号回复的：“卧虎藏龙电影评分为7.2分！”。



在接受到用户的问题之后，会首先对问题进行分词，分词过程中，电影名称会正确被解析为nm。经常抽象的问题送往朴素贝叶斯分类器，得到应使用的问题模板编号为0号。下一步根据分类器得到的问题模板和分词器得到的电影名称，向图数据库查询评分数据。最后将图数据库返回的评分格式化为自然语言返回。

可见问答系统能准确的分析用户提问意图，正确查询图数据库数据，并使用使用自然语言回答。

因租用的云服务器已到期，微信公众号回答功能已下线。如需试用本问答系统，可在Google Colab中运行在线测试Demo。

<https://colab.research.google.com/github/HLNN/NLP/blob/master/Final/movie_QA_with_KQ.ipynb>

1. **遇到问题及解决思路**

本项目在微信公众号部署方面遇到过较多的问题。包括公众号后台于云托管服务器的之间，API接口的调用和鉴权，项目代码容器化处理等等。这些问题通过查阅官方问答和社区问答，都已逐一解决。

数据方面因不是我们自己爬取的数据，数据时间偏早，上映时间较近的电影并未收录其中。

数据中的英文单词之间会使用空格分隔，而问题预处理时，空格会被分词器分至不同单词，因此系统无法处理英文数据查找。为此我们去除了数据库中的所有空格，同时在输入问题分词前也去除全部空格。这样能初步解决包括英文的提问需求。