图标

描述已自动生成

**人工智能实验报告**

|  |  |  |  |
| --- | --- | --- | --- |
| **专业** | **：** | **计算机科学与技术** |  |
| **班级** | **：** | **CSM2403班** |  |
| **学号** | **：** | **M202474058** |  |
| **姓名** | **：** | **肖先荣** |  |
| **电话** | **：** | **19996815586** |  |
| **邮箱** | **：** | **3493602396@qq.com** |  |

**独创性声明**

本人郑重声明本报告内容，是由作者本人独立完成的。有关观点、方法、数据和文献等的引用已在文中指出。除文中已注明引用的内容外，本报告不包含任何其他个人或集体已经公开发表的作品成果，不存在剽窃、抄袭行为。

特此声明！

作者签名：肖先荣

日期：2024年12月25日

|  |  |
| --- | --- |
| 成绩 |  |
| 教师签名 |  |

目录

[1 概述 1](#_Toc185801606)

[2 实验一：实现豆瓣电影知识图谱可视化 2](#_Toc185801607)

[2.1 实验设计 2](#_Toc185801608)

[2.2 实验结果 7](#_Toc185801609)

[3 实验二：十五数码问题求解与路径规划 9](#_Toc185801610)

[3.1 实验设计 9](#_Toc185801611)

[3.2 实验结果 12](#_Toc185801612)

[4 实验四：基于情感分析的社交媒体数据分析 14](#_Toc185801613)

[4.1 实验设计 14](#_Toc185801614)

[4.2 实验结果分析 16](#_Toc185801615)

[5 总结和源码 18](#_Toc185801616)

# 概述

本组选取了三个实验，分别是：

1）实验一：实现豆瓣电影知识图谱可视化。

2）实验二：十五数码问题求解与路径规划。

3）实验四：基于情感分析的社交媒体数据分析。

本文将对每个实验进行实验设计、实验结果分析。

# 实验一：实现豆瓣电影知识图谱可视化

**任务描述：**利用Python编程语言和Neo4j图数据库技术，实现豆瓣电影数据的知识图谱可视化，呈现豆瓣电影的相关信息及其关联，如下图所示。

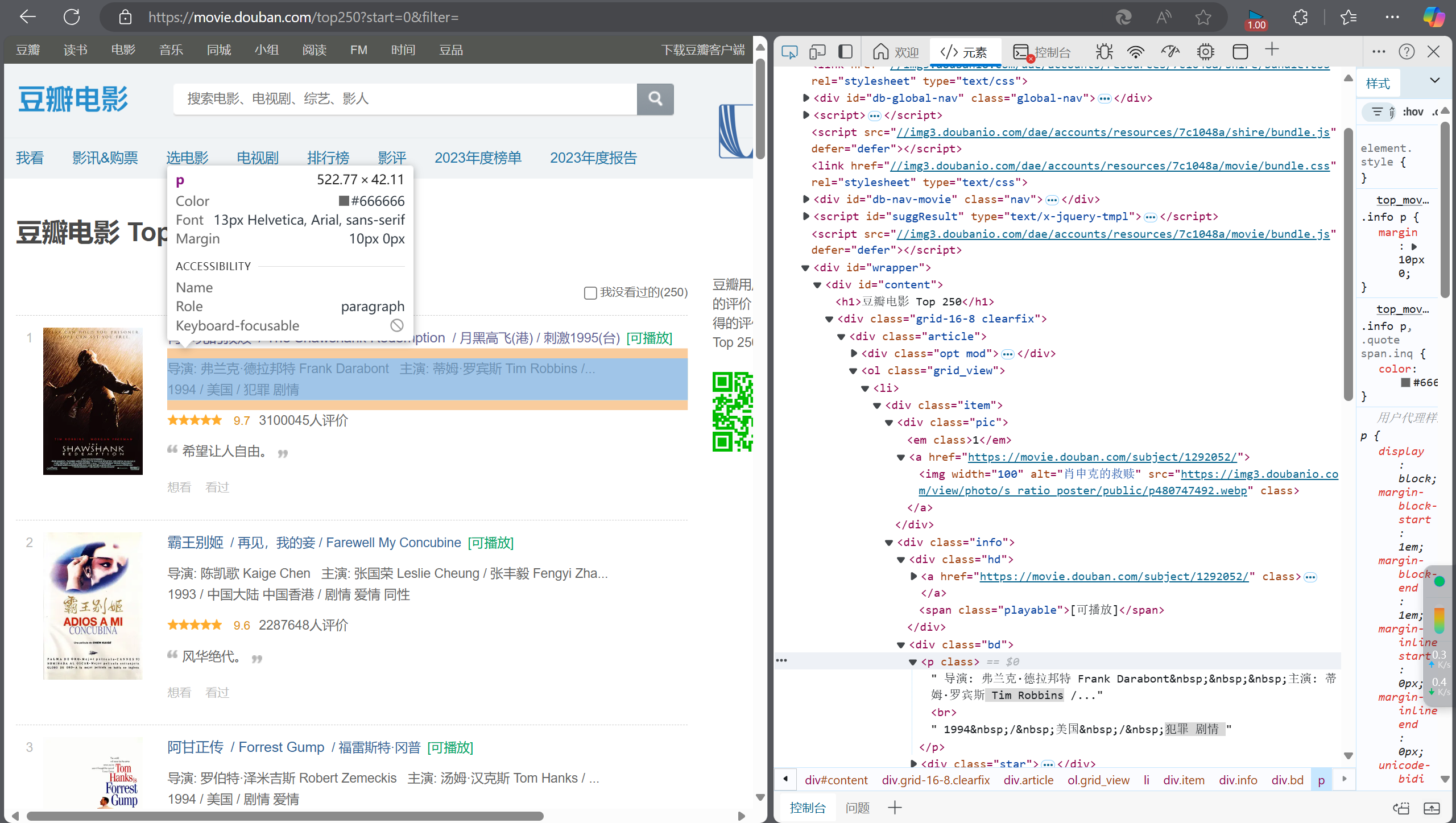
**实验要求：**

1. 对网页数据进行分析，确定节点和关系
2. 节点数据爬取（包括电影名称、导演、演员、类型）
3. 节点关系数据爬取（包括合作、属于、参演、指导）
4. 使用python+neo4j创建知识图谱

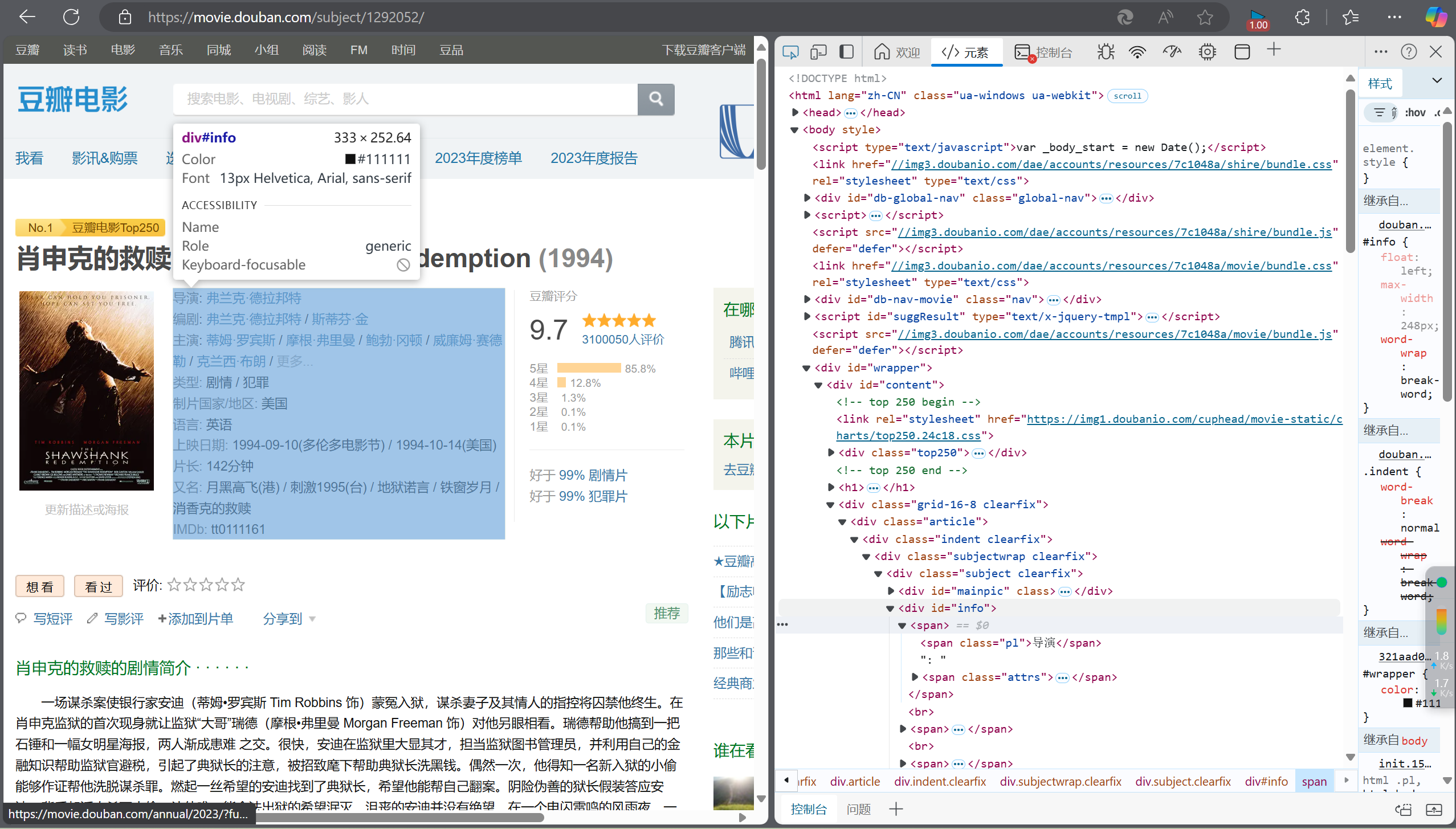
## 实验设计

**1、爬取网页数据**

以豆瓣top250电影为例，我们访问网站[豆瓣电影 Top 250](https://movie.douban.com/top250?start=0&filter=)分析网页源码，如下图所示。



每个list有25个item，每个item包含指定一部电影的网页。我们找到每个item的链接访问，分析网页源码如下图所示：



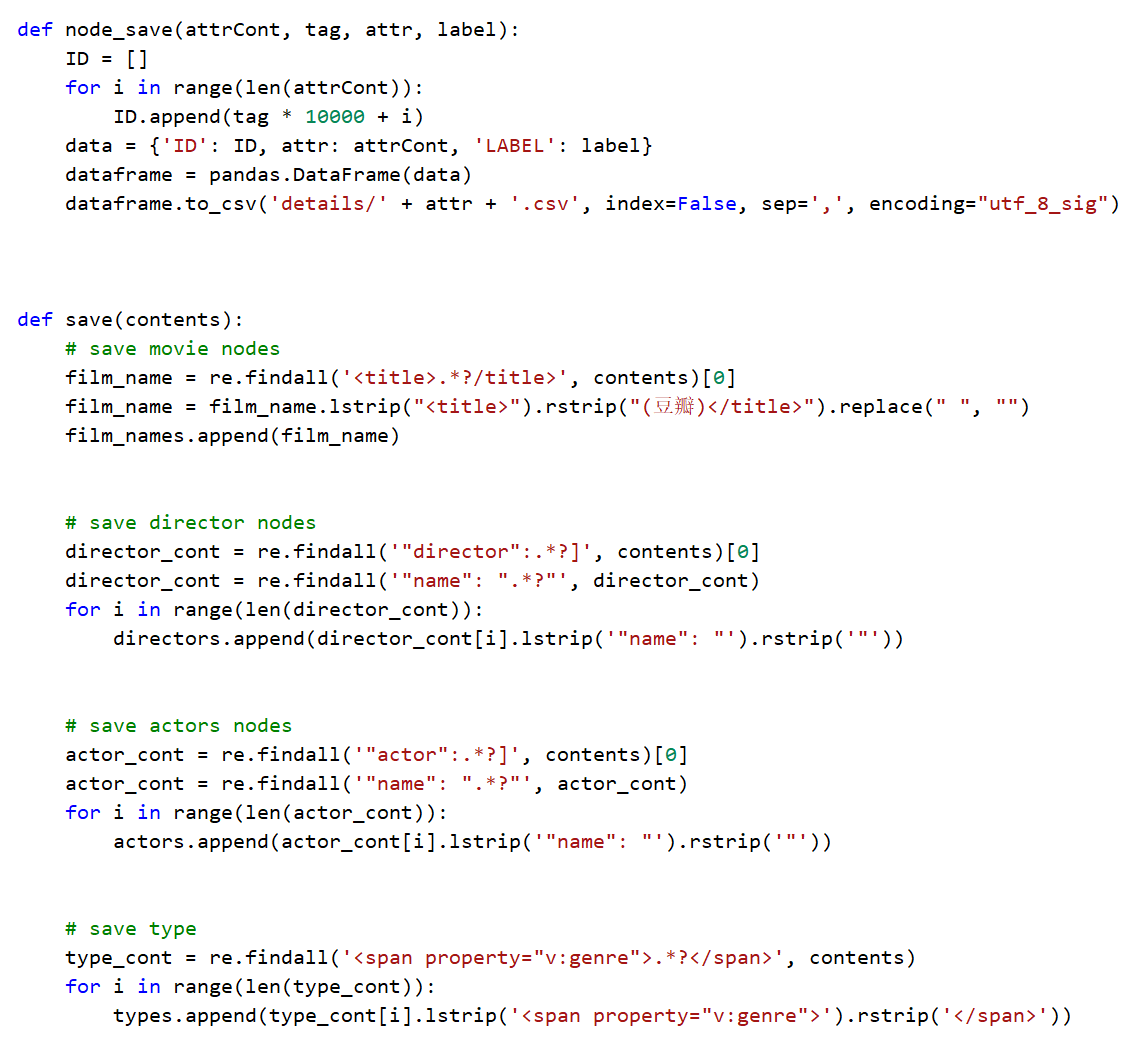
因此我们可以先爬取list然后找到每个item的链接，访问链接获得每个item的信息。在这里就是获取每个电影的四个结点，分别是电影名、导演、演员和类型。核心代码如下：



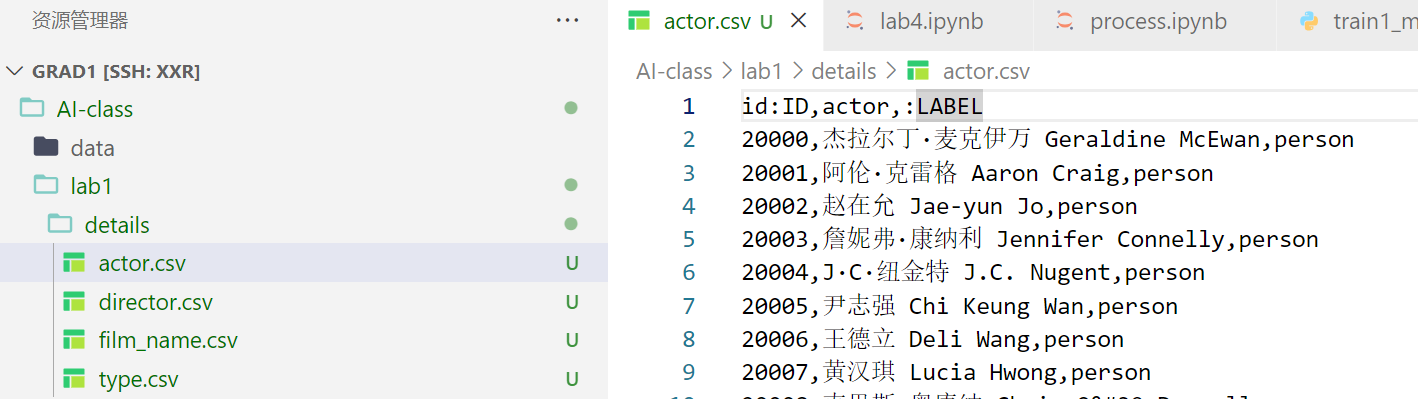
这里选取了前125个电影，经过上述代码我们下载了125个电影网页文本中。

**2、结点数据的爬取**

对这125个网页文本进行处理，提取出四个结点并进行去重，并最终保存到四个csv文件中。



获得了四个csv文件。



**3、结点关系的爬取**

我们分别考虑以下四种关系。

acted\_in关系：提取电影名称（fillm\_name）和演员列表（actor），将它们分别加入列表中。

directed关系：提取电影名称（fillm\_name）和导演列表（director），将它们分别加入列表中。

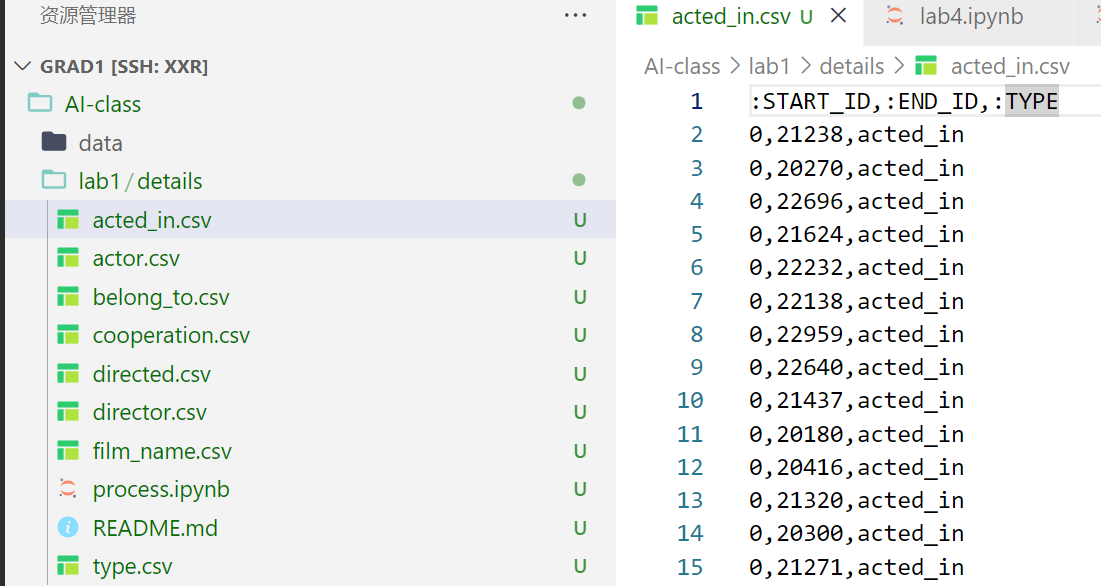
belong\_to关系：提取电影名称（fillm\_name）和类型列表（type）。

cooperation关系：提取演员列表（actor）和导演列表（director）。

核心代码如下：

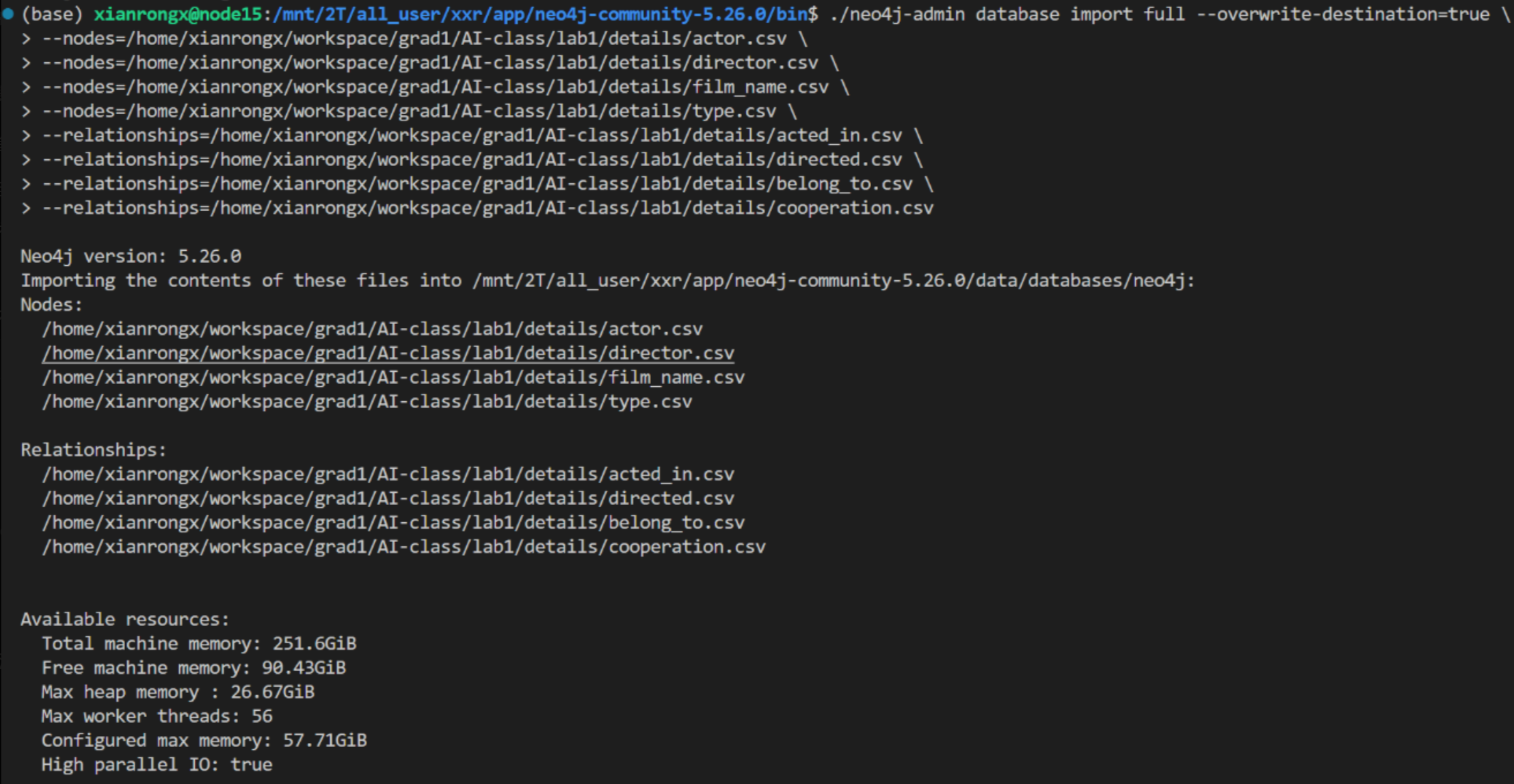


获得四个csv文件：

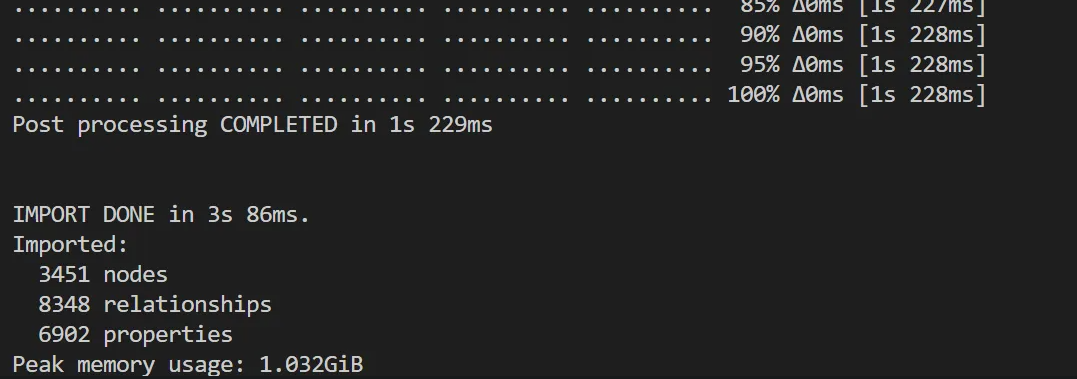


**4、创建知识图谱**

利用python+neo4j创建知识图谱。导入数据：

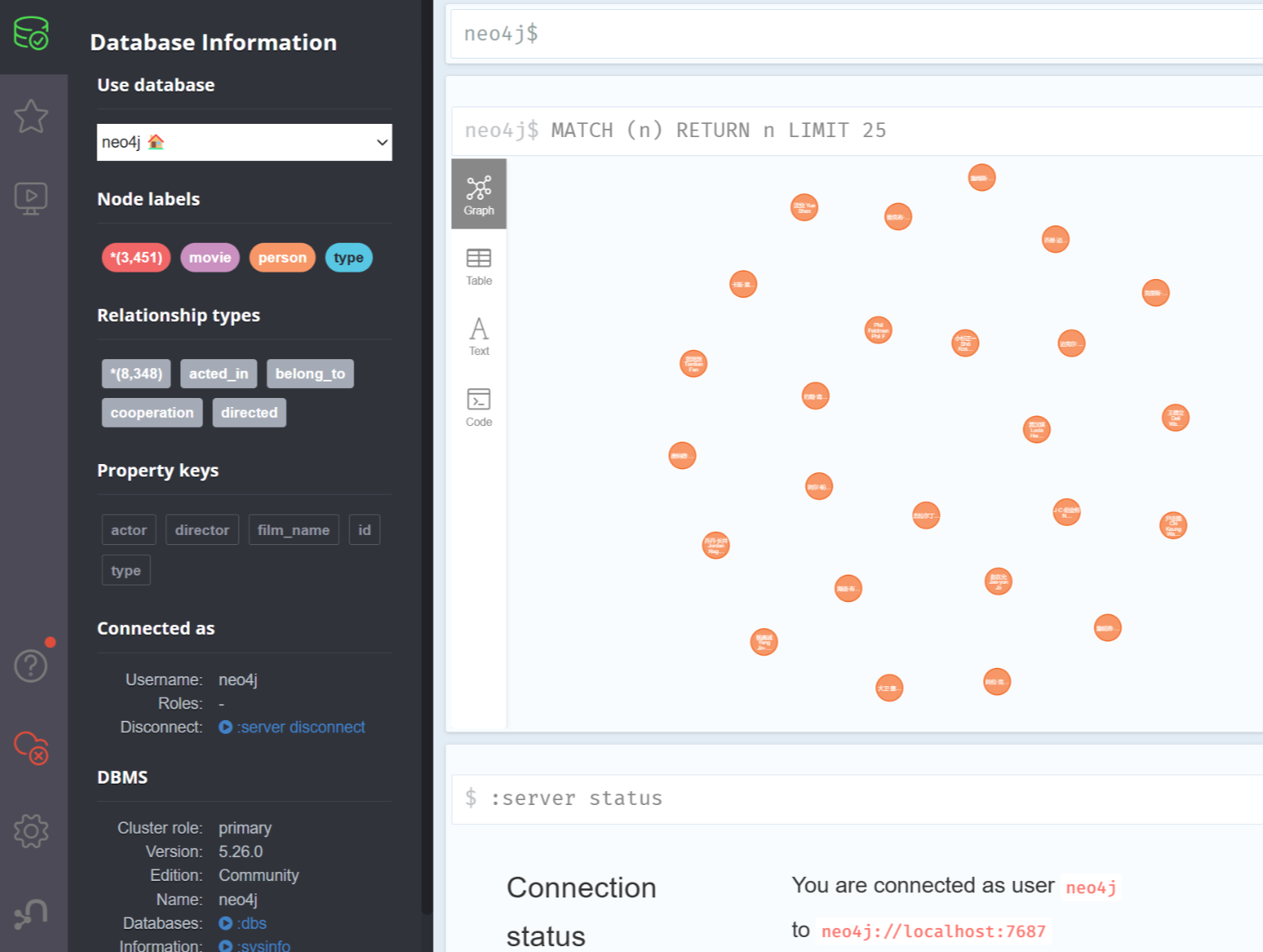


导入成功，显示3451个结点，8348个关系，6902个属性。

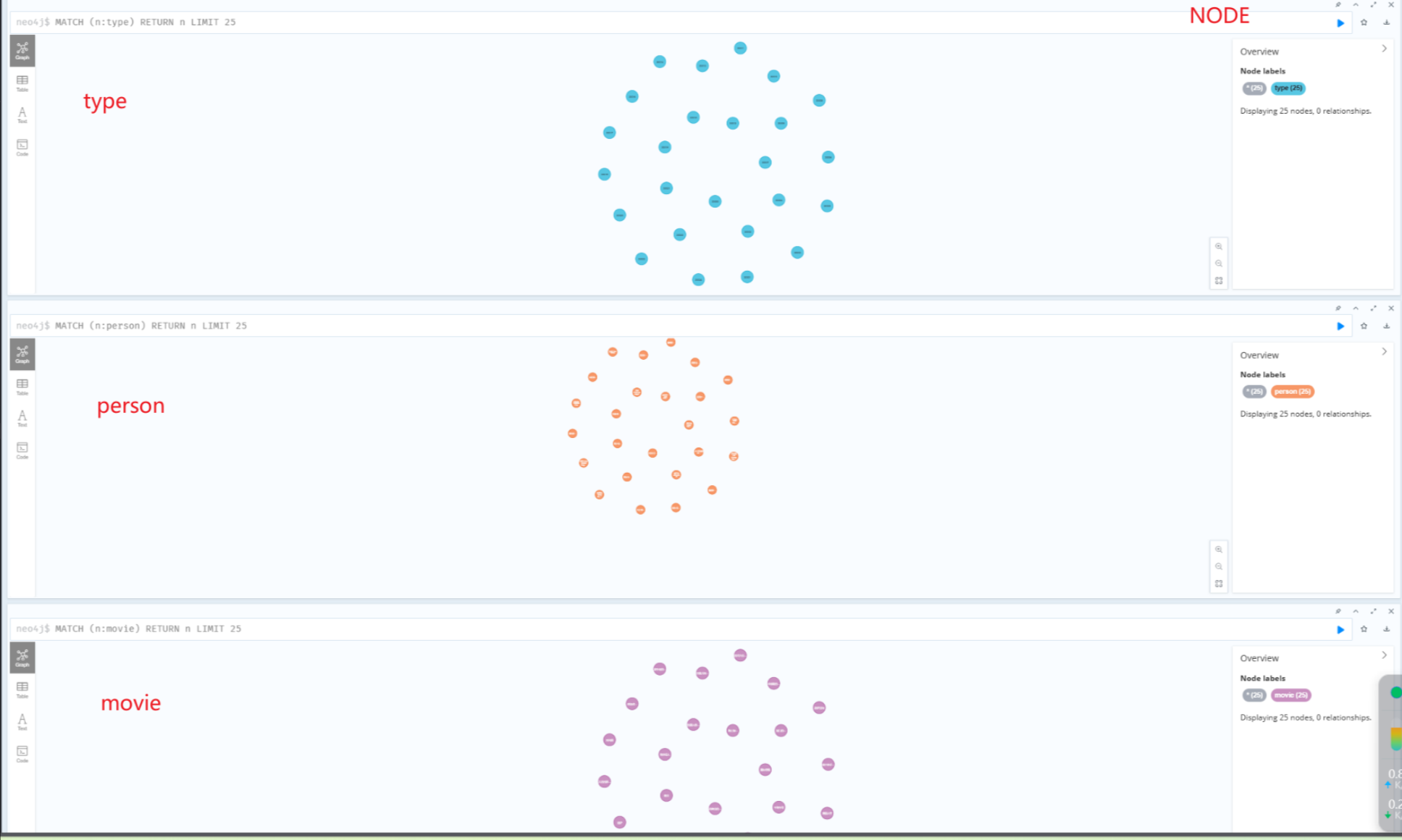


## 实验结果

导入数据成功后，启动neo4j，并在本地端口查看建立的知识图谱。



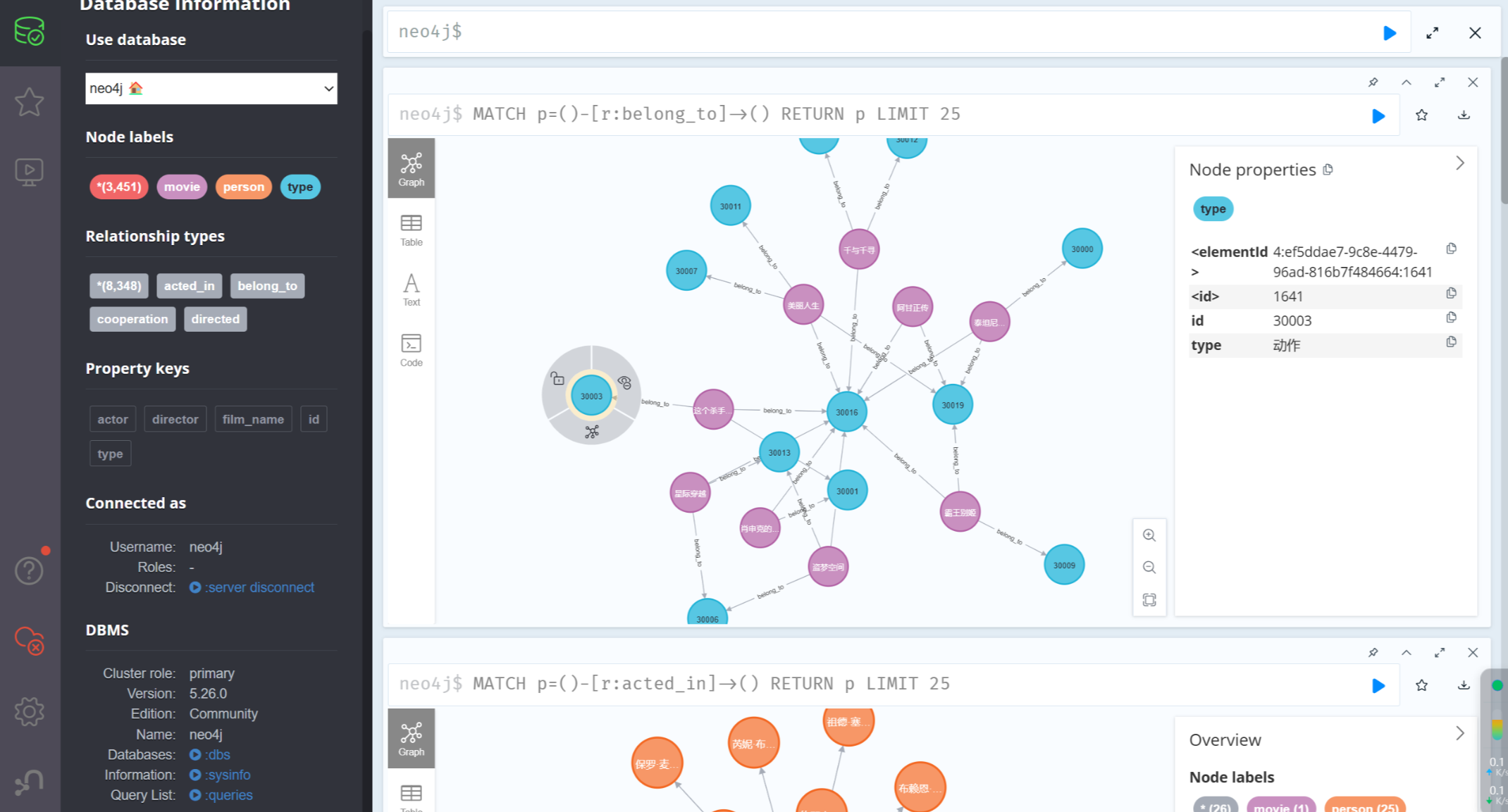
左侧栏目可以查看结点和结点关系。



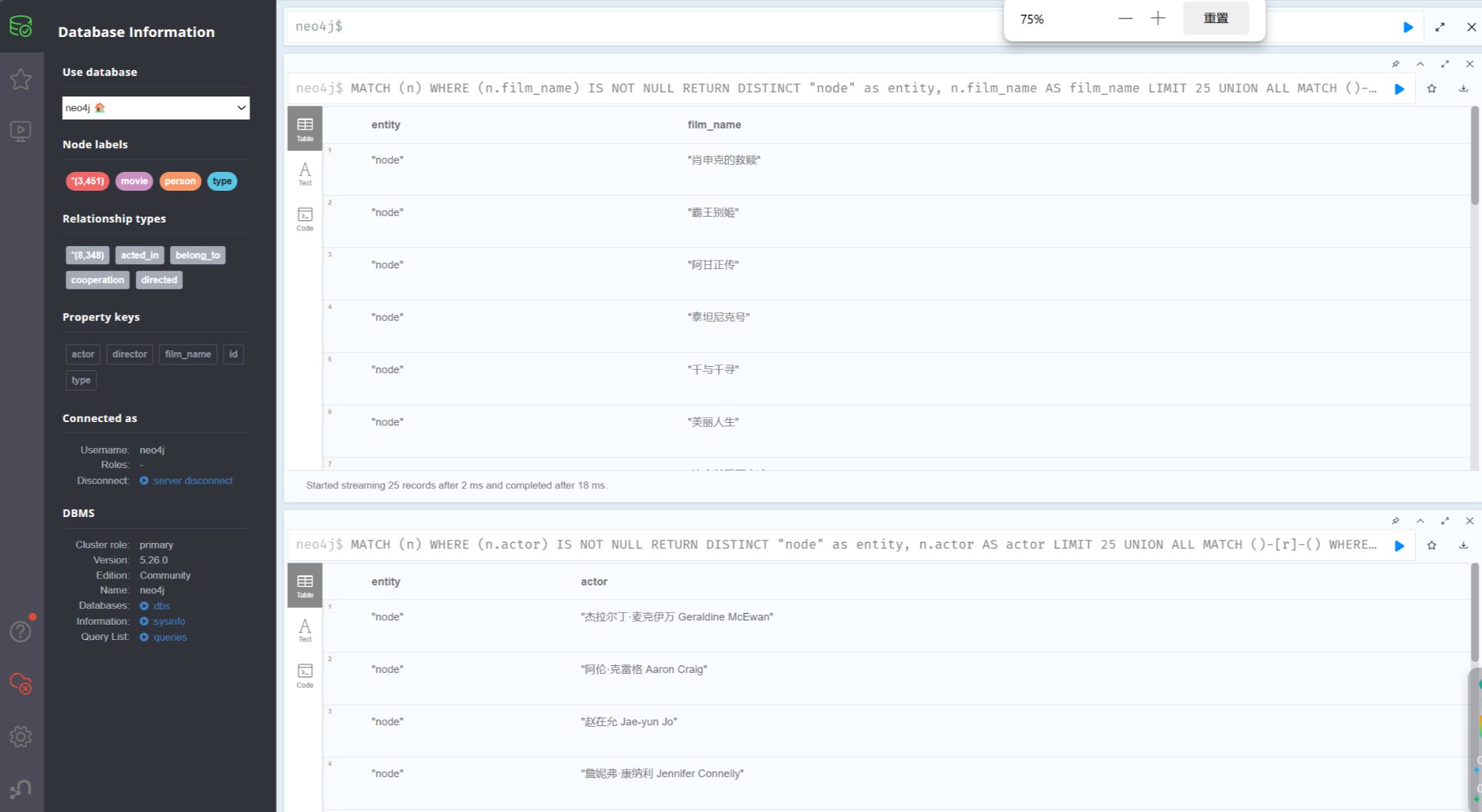
我们可以查看部分acted\_in结点关系：



belong\_to关系：



以及节点的属性：



# 实验二：十五数码问题求解与路径规划

任务描述：十五数码问题（也称为滑块拼图问题），它在一个4x4的网格上进行，包含数字1到15以及一个空格。目标是通过滑动数字（包括空格旁边的数字滑入空格）来恢复初始的排列顺序（通常是1到15按顺序排列，空格在右下角）。这个问题比八数码问题更复杂，因为状态空间更大，求解难度更高。实现宽度优先搜索（BFS）、深度优先搜索（DFS）、A\*算法来求解十五数码问题。

实验要求：

设计一个4x4的网格来表示十五数码问题的状态。定义合法的移动操作，即数字可以滑入空格的位置。实现一个函数来生成给定状态的所有可能后续状态。

选择一种搜索算法（BFS、DFS或A\*）实现并管理搜索成本。

实现一个有效的状态存储机制，以避免重复访问相同的状态。

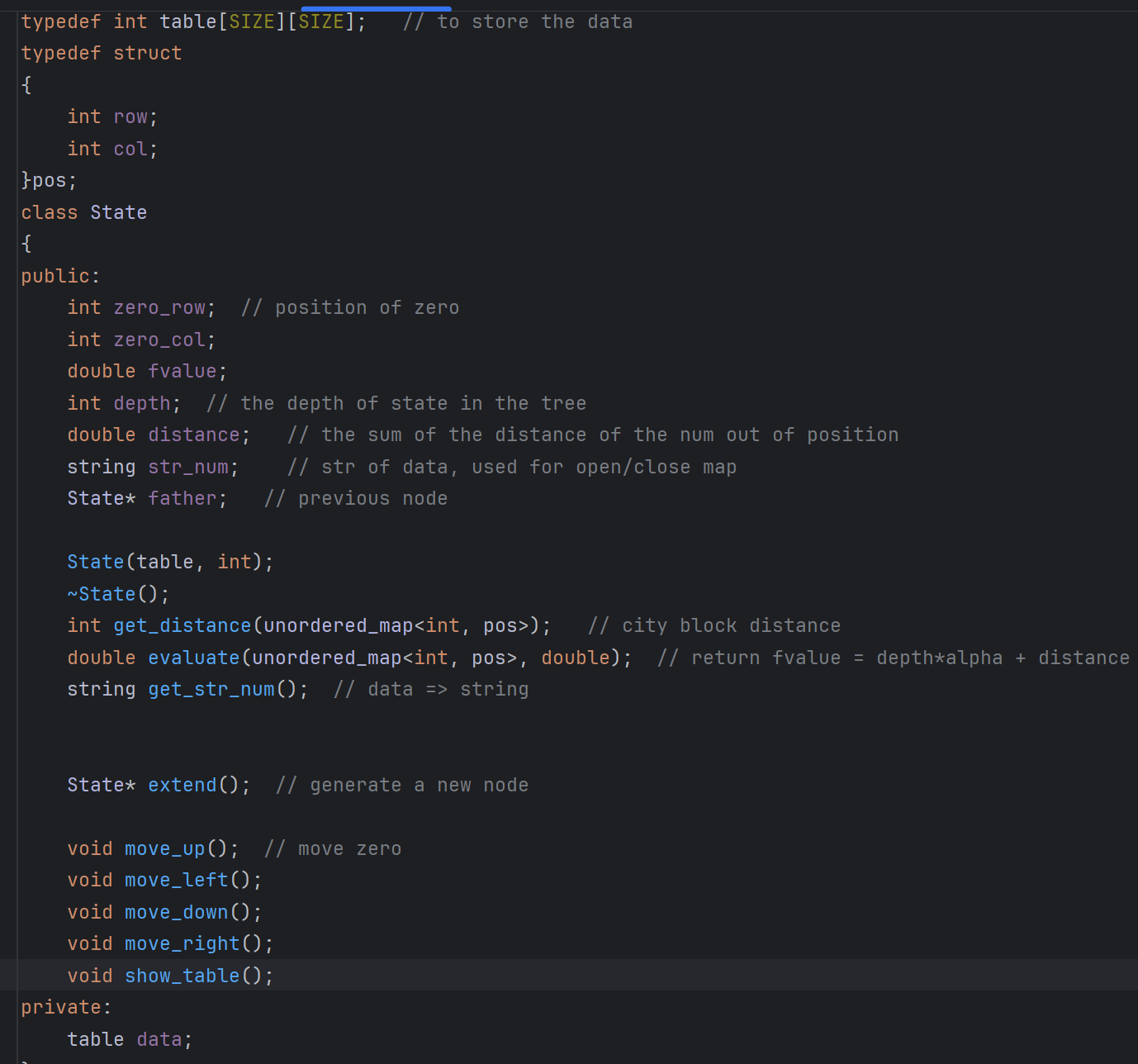
要求路径记录与可视化。

测试与评估：选择几个不同的初始状态进行测试。记录求解时间、路径长度和内存使用情况等指标。分析算法的性能，并讨论可能的改进方向。

## 实验设计

A\*算法通过维护Open和Close两个表，逐步扩展估价函数最小的节点，寻找从初始节点到目标节点的最优路径。在15数码问题中，算法从Open表中选取当前估价函数最小的节点进行扩展，生成新的状态节点，并计算其估价函数 G=Dis+depth×α。扩展出的新节点若在Open或Close表中，则比较估价函数，保留较优节点；若不在表中，则将其加入Open表。处理完成的节点移入Close表，避免重复计算。循环持续至找到目标节点或达到限制条件，最终输出解路径或失败结果。

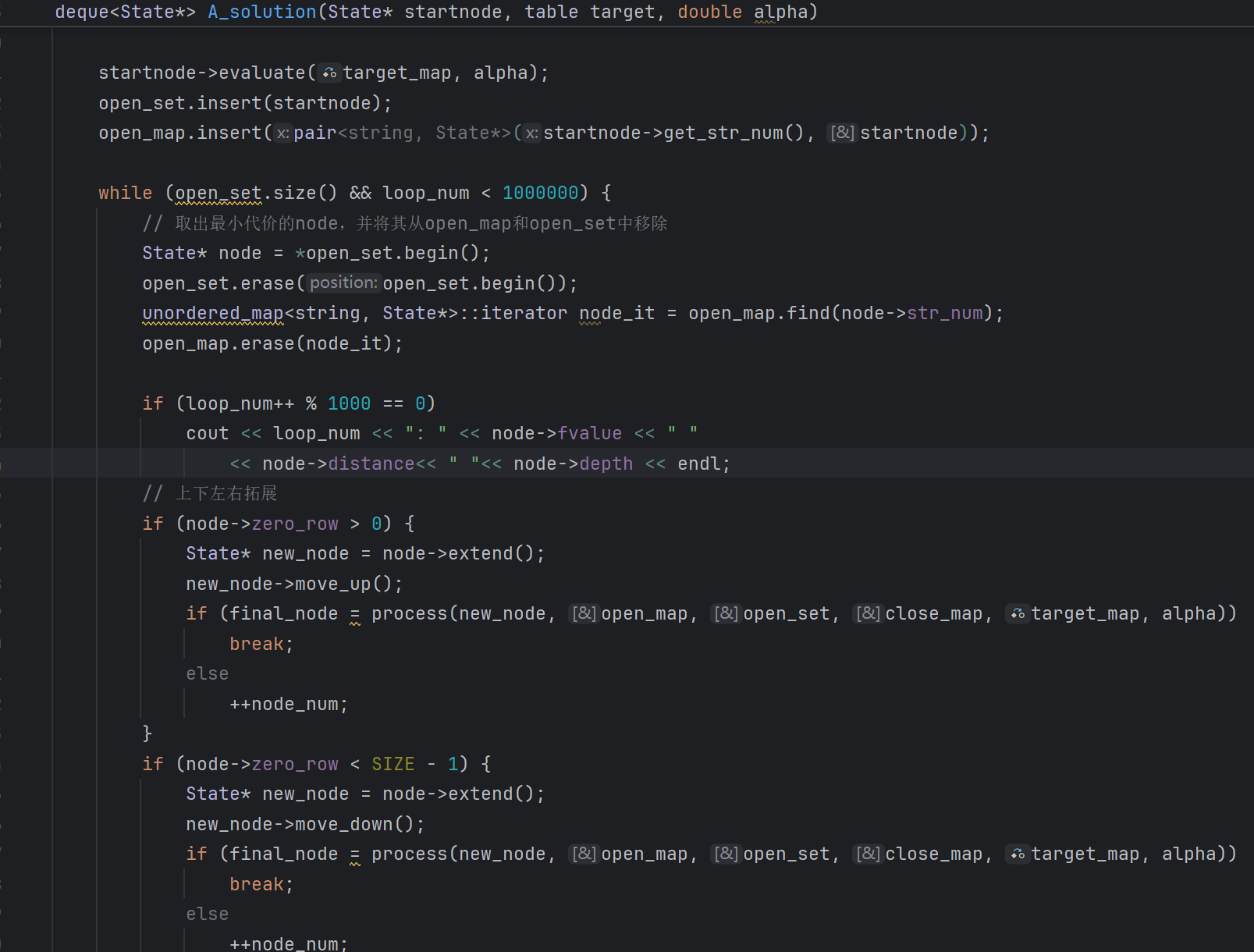
定义数据结构如下：



process函数用于处理新生成的状态节点new\_node，判断其是否已存在于open\_map或close\_map中，并根据评估值（fvalue）决定如何更新状态或插入表中。首先，计算new\_node的fvalue，如果它已存在于open\_map且当前评估值更优，则更新open\_map并在open\_set中替换旧节点；如果存在于close\_map且评估值更优，则将其从close\_map移至open\_map和open\_set中。若new\_node未在任一表中且目标状态已达成（distance == 0），则返回解节点；否则，将其插入open\_map和open\_set中，并继续探索其他节点。函数返回解节点或空指针以指示搜索是否完成。核心代码如下：

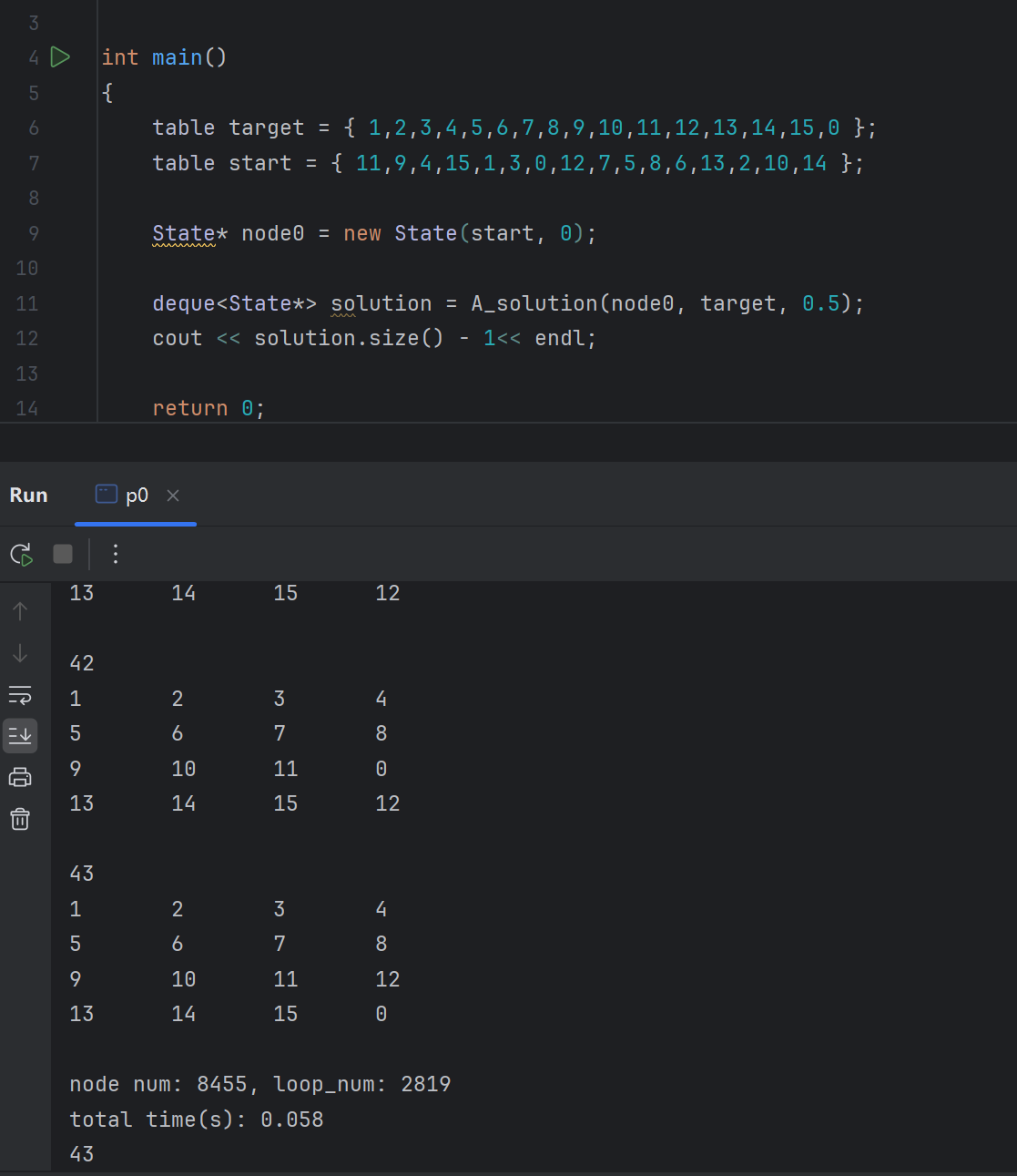


主程序A\_star\_solution函数实现了A\*算法的主逻辑，用于从初始节点startnode搜索到目标状态target，并返回通往解节点的路径。首先初始化数据结构，包括open\_set（按fvalue排序的优先队列）、open\_map和close\_map（用于快速查找节点）、以及记录目标状态信息的target\_map。将起始节点加入open\_set和open\_map后，算法在while循环中不断从open\_set中取出当前评估值最小的节点，扩展其上下左右四个方向的状态。每个新节点通过process函数处理，决定是否更新到open\_map或close\_map，或直接作为解节点终止搜索。若找到目标状态（distance == 0），构建从解节点到初始节点的路径链表并返回，同时输出路径长度、搜索节点数以及运行时间等信息。核心代码如下：



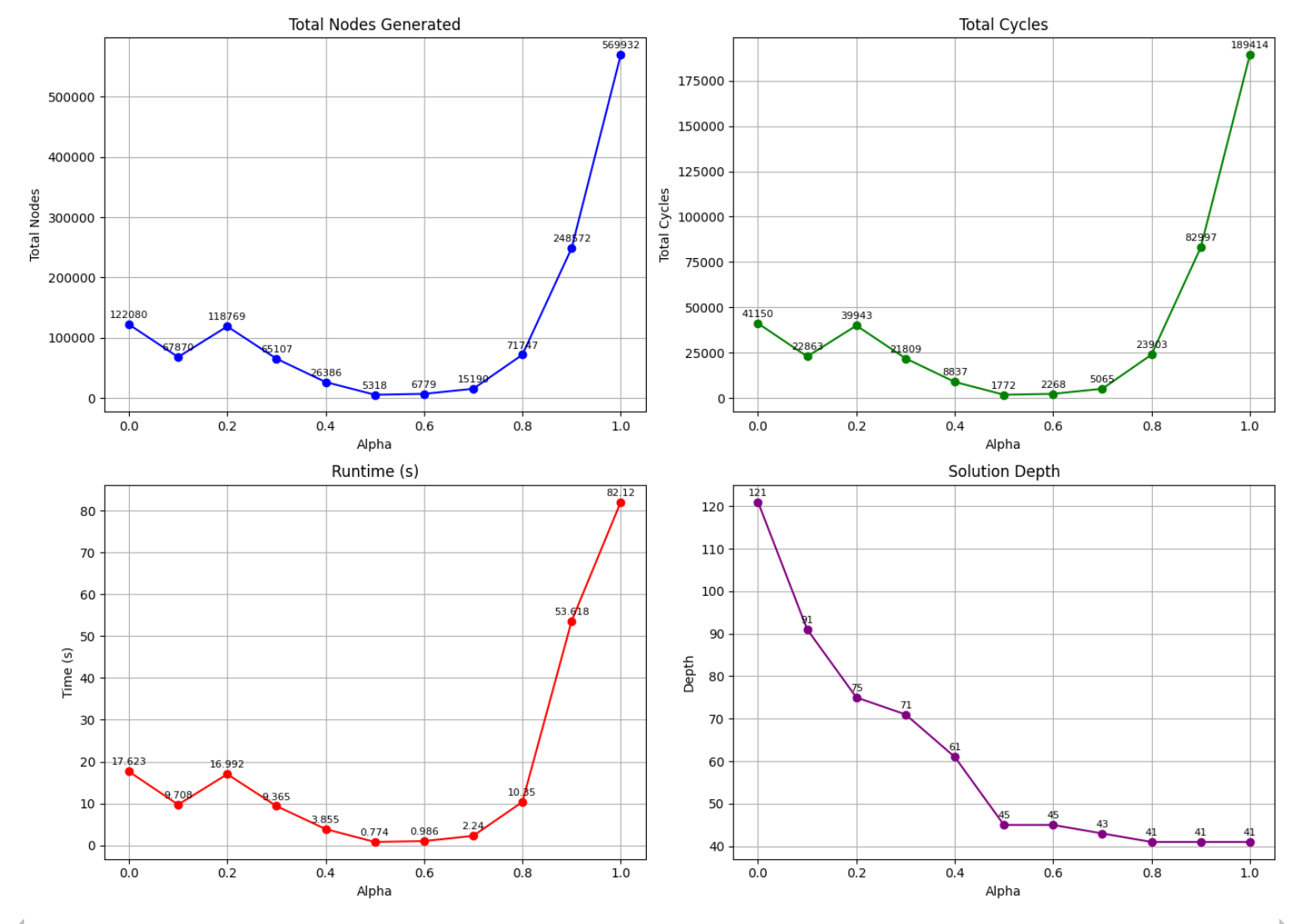
## 实验结果

运行结果：



可以看到给定了初始状态和终止状态，输出结果包括了运行时间，生成结点的数量和循环次数。并记录了每一步格子移动的结果。

此时是[α](https://baijiahao.baidu.com/s?id=1777054396260032004)=0.5，而[α](https://baijiahao.baidu.com/s?id=1777054396260032004)是一个可以调整的参数，经过实验，[α](https://baijiahao.baidu.com/s?id=1777054396260032004)的取值会影响生成结点的个数、循环次数和运行时间和解的深度。如下图所示。



# 实验四：基于情感分析的社交媒体数据分析

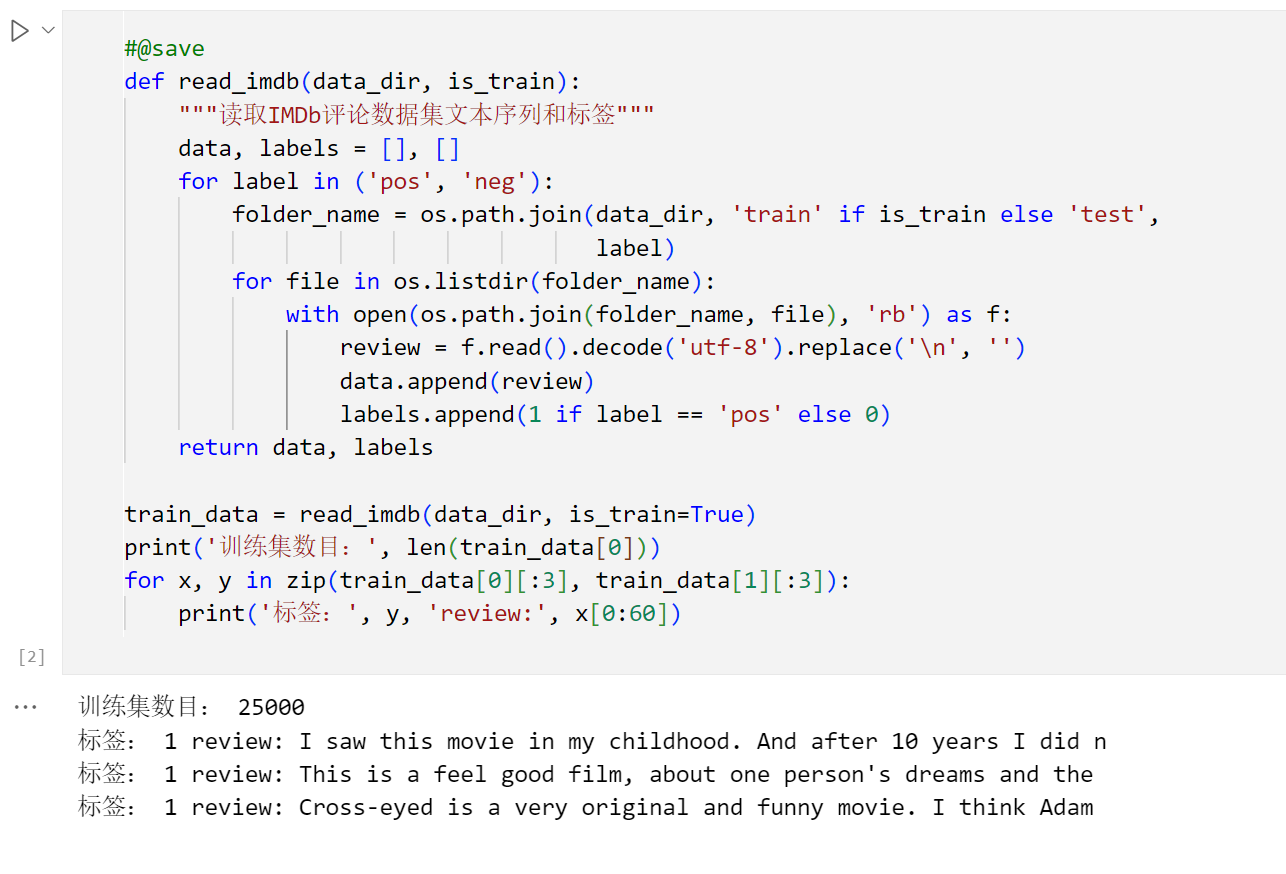
**任务描述**：利用机器学习算法对社交媒体数据进行情感分析，识别用户情感倾向（如正面、负面或中性）。

**实验要求**：

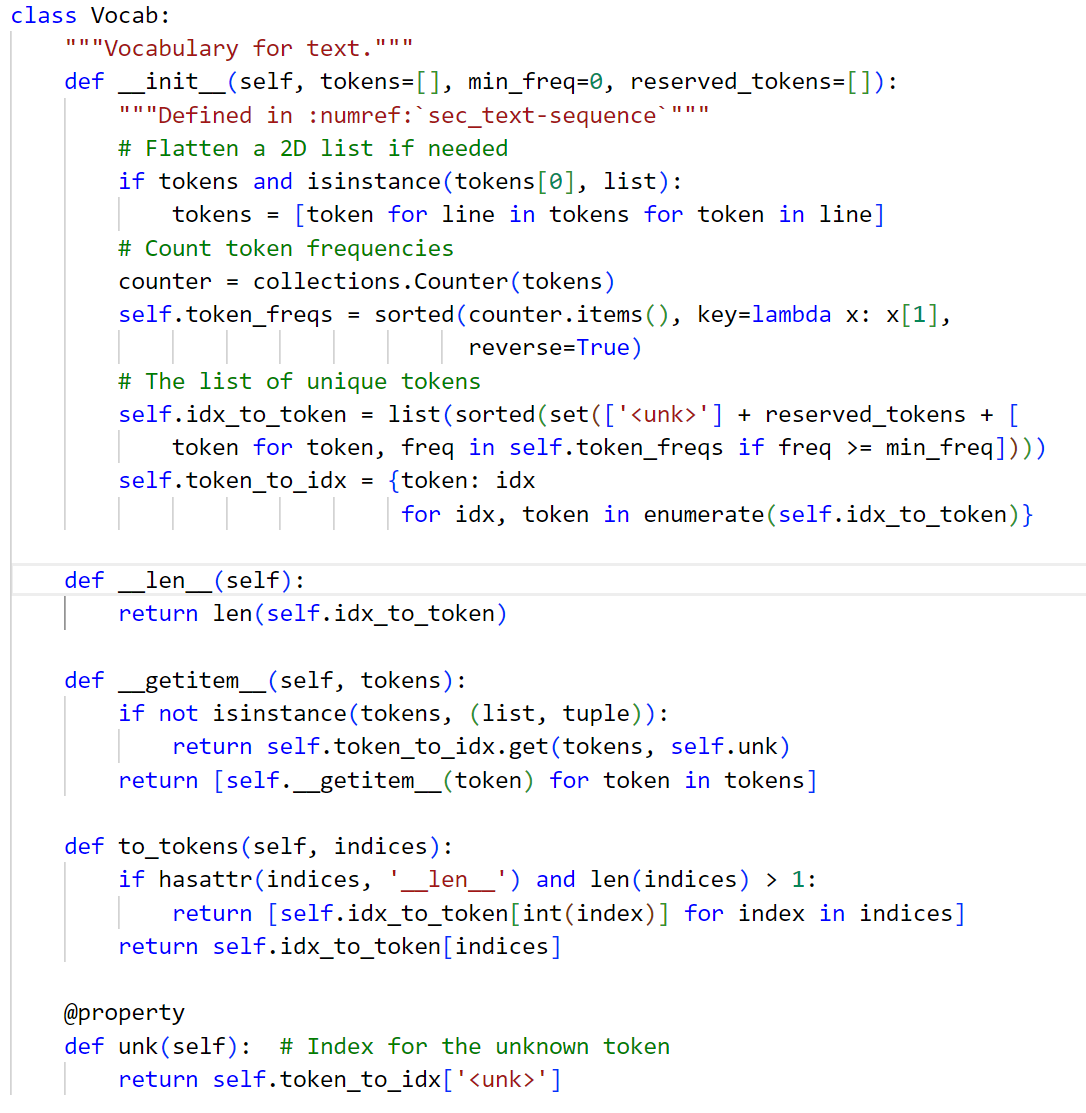
数据预处理：对文本进行清洗（去除特殊字符、停用词、分词等）。使用朴素贝叶斯、支持向量机、LSTM或其他算法建立情感分类模型。使用交叉验证评估模型性能，比较不同算法在情感分类任务上的效果。给出情感分析结果（如正负情感比例、情感趋势等）。讨论情感分析结果对市场营销、舆情监测的实际应用。数据集不指定，自行下载相关数据集进行实验。

## 实验设计

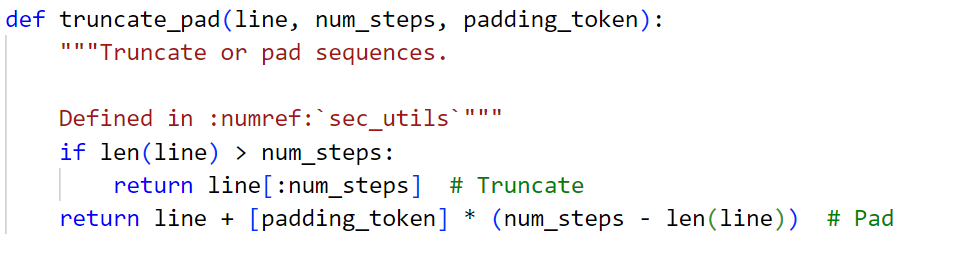
首先，下载并提取路径../data/aclImdb中的IMDb评论数据集。接下来，读取训练和测试数据集。每个样本都是一个评论及其标签：1表示“积极”，0表示“消极”。



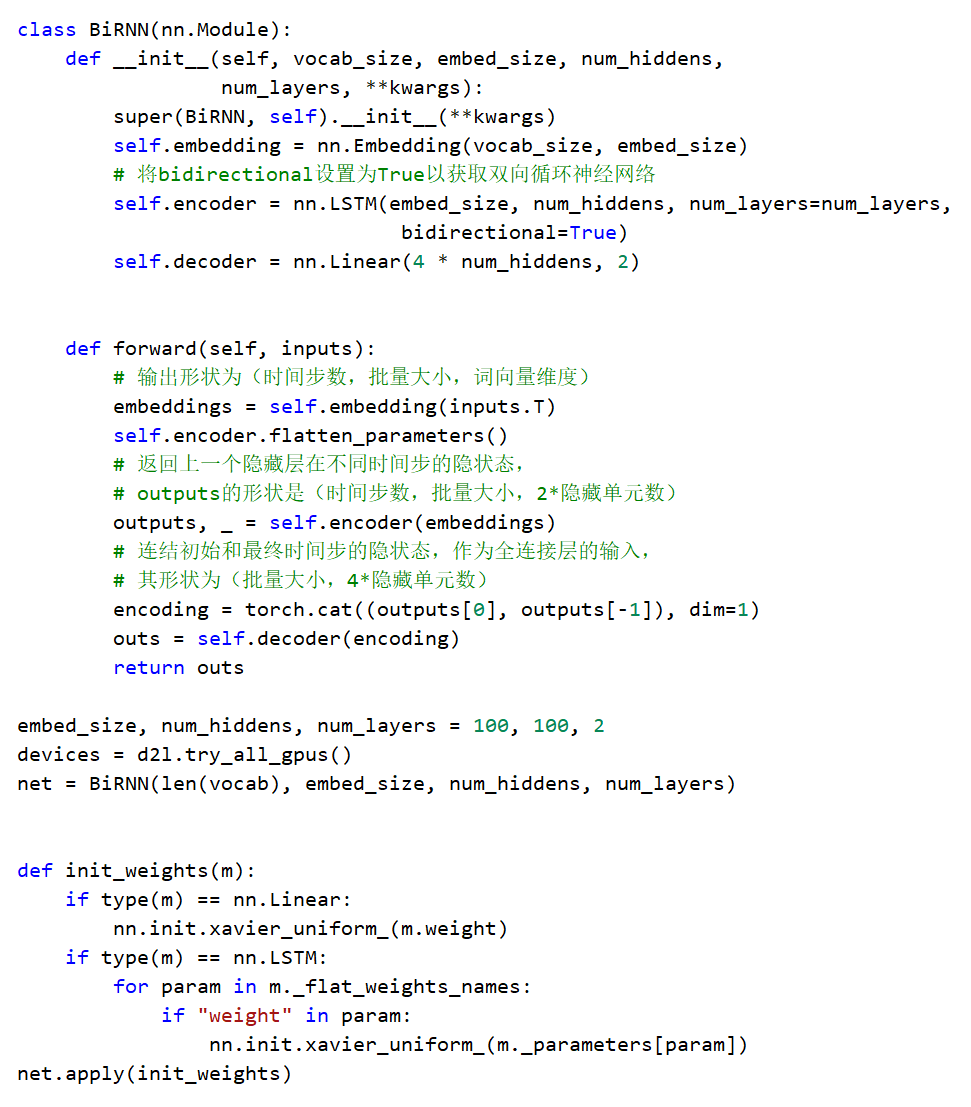
接下来对文本数据进行处理。将每个单词作为一个词元，过滤掉出现不到5次的单词，我们从训练数据集中创建一个词表。



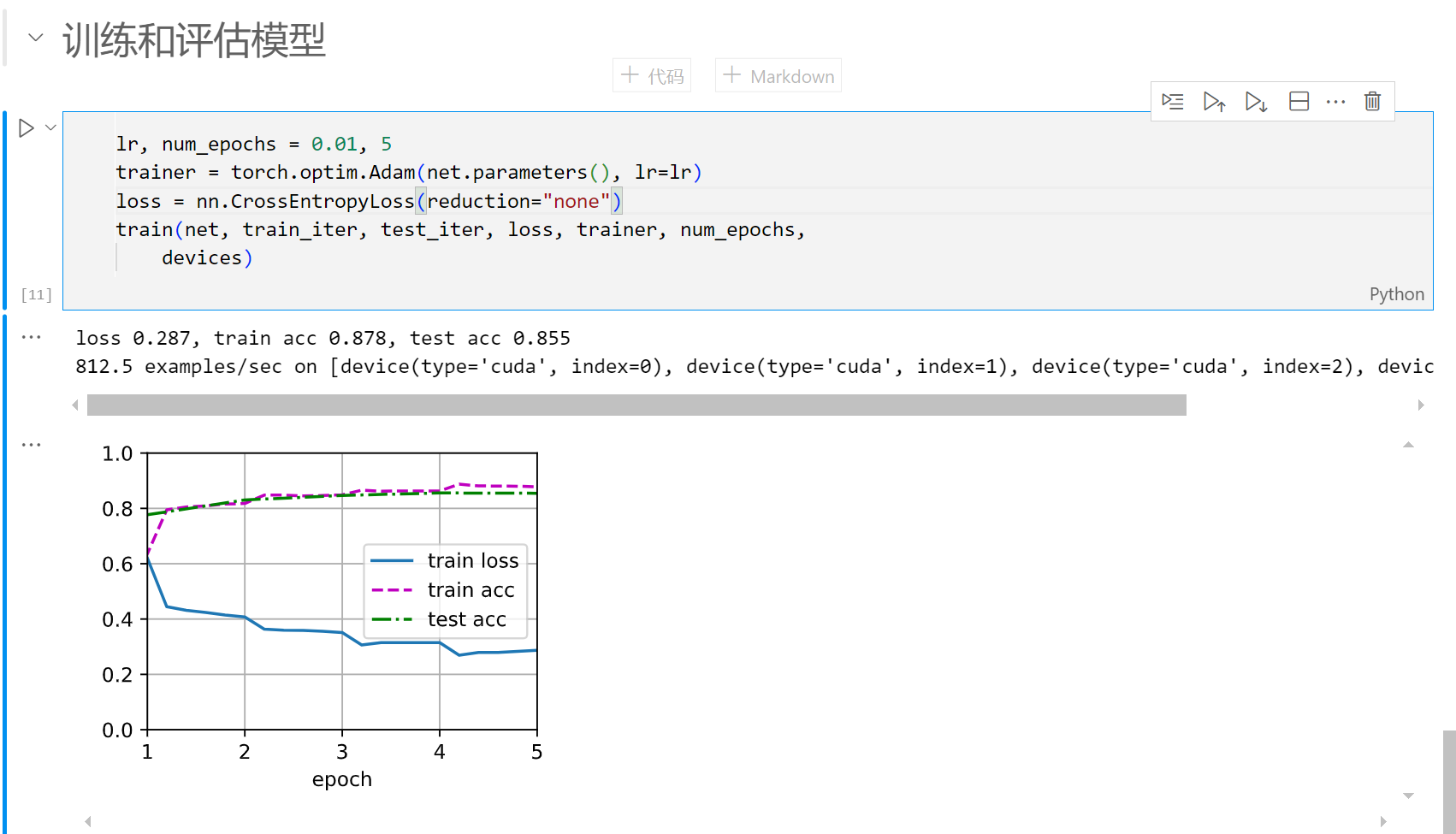
由于评论的长度各不相同。为了每次处理一小批量这样的评论，我们通过截断和填充将每个评论的长度设置为500。



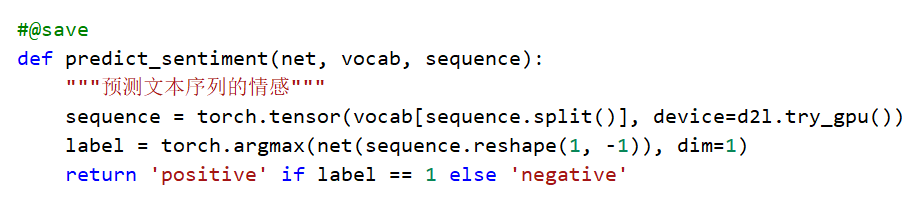
创建模型。使用LSTM，双向长短期记忆网络在初始和最终时间步的隐状态（在最后一层）被连结起来作为文本序列的表示。然后，通过一个具有两个输出（“积极”和“消极”）的全连接层（self.decoder），将此单一文本表示转换为输出类别。模型如下：



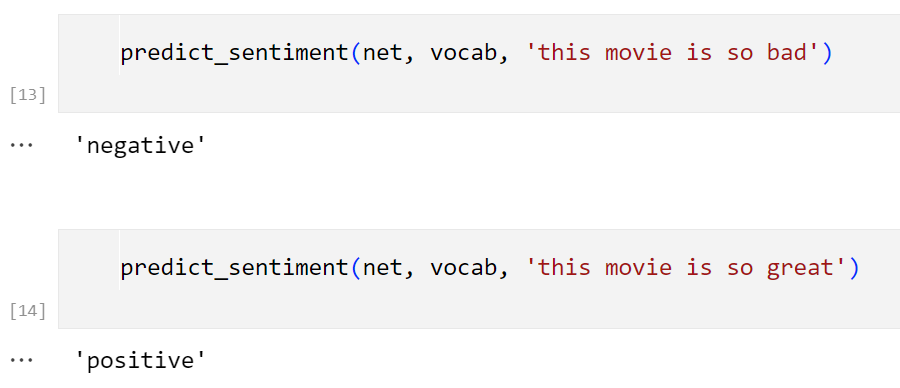
## 实验结果分析



预测代码：



预测结果：



# 总结与展望

本组选取了三个实验，本篇报告侧重实验一和实验二的内容。完整的源码见小组仓库：