

实验报告

作业一：图谱数据采集和关联



姓名：陈静

学号：201628013727024

日期：2017-06-01

目录

[一、实验目的 2](#_Toc484187581)

[二、实验任务 2](#_Toc484187582)

[三、实验内容 2](#_Toc484187583)

[3．1 数据爬取与解析 3](#_Toc484187584)

[3．2 数据处理及存储 4](#_Toc484187585)

[3．3 数据可视化 7](#_Toc484187586)

[四、实验结果 8](#_Toc484187587)

[五、实验总结 10](#_Toc484187588)

[附件：文件清单 11](#_Toc484187589)

# 一、实验目的

知识图谱，也称为科学知识图谱，它通过将应用数学、[图形学](https://baike.so.com/doc/6529350-6743086.html)、信息可视化技术、[信息科学](https://baike.so.com/doc/6616895-6830689.html)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](https://baike.so.com/doc/95374-100731.html)以及整体知识架构达到多学科融合目的的现代理论。为学科研究提供切实的、有价值的参考。

本次实验旨在通过数据采集、解析、可视化等操作来初步了解构建图谱的过程和基本方法，掌握基本的图谱相关的知识。

# 二、实验任务

任务1：编写一个网络爬虫，爬取某百科网站的特定领域数据，要求爬取文档的规模不少于1000篇；

任务2：解析百科文档的结构化数据内容，构建实体间的关联关系，并设计一个存储结构或表示方法，将数据存储到图数据库或关系数据库中；

任务3：利用D3.js或ECharts.js等可视化工具包，可视化展示构建的实体关联关系，要求能够查询和展示以某个实体为中心，N跳以内的网络拓扑结构。

# 三、实验内容

本文选择的主题是明星关系图谱制作，主要是因为最普遍的知识图谱就是人物关系，所以我也想先从这个入手来学习知识图谱的相关内容，而且网上关于明星的个人资料很丰富。抓取的网页来自百度百科，具体内容为百度百科中明星个人网页中的关系部分，然后将数据存入图数据库中，最终用Echarts工具包来完成可视化，并支持查询某个明星N跳以内的关系图谱。

本次实验是按照三个任务展开的，下面具体介绍每个任务的实现思路和方法。

## 3．1 数据爬取与解析

工具：Python 2.7、urllib2、Beautiful Soup

整个爬取程序分为五个部分，分别是主调用程序、链接管理、网页读取、网页解析和结果输出。

链接管理是管理已经抓取过的和待抓取的网页，既要将相关的链接存储下来还要保证同一页面不会重复抓取。

网页读取是利用Python的urllib2库完成的，通过urlopen和read函数来获取网页内容，为了避免爬虫超时，可以加入延时避免过于频繁的爬取某网页和加入超时处理，本次抓取设置的超时时间为10秒。

网页解析的工作是借助Beautiful Soup库来完成的，Beautiful Soup是用Python写的一个HTML/XML的解析器。需要解析的内容如图3-1所示。



图3-1 红色边框内的内容为具体爬取内容

通过查看元素，可知需要爬取的内容对应的html内容如图3-2所示。



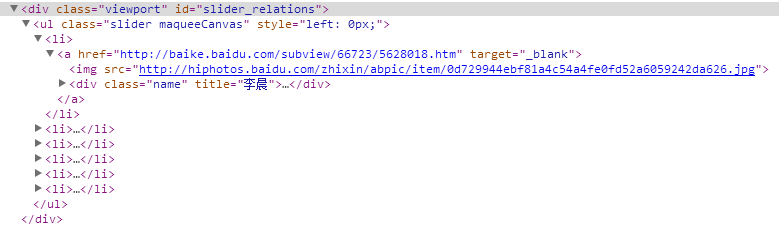


图3-2 需要解析的元素

在这里经过解析后需要抓取的内容有三个：主页的主人公（范冰冰），与其有关的人物及具体关系（男友李晨、母亲张传美……）和相关人物的网页链接（李晨：http://baike.baidu.com/subview/66723/5628018.htm ……），而利用Beautiful Soup我们可以很轻易地找到这些内容，例如抓取相关链接只要找到这些链接的组成规则，就可以通过简单正则化的表达规则来找到（http://baike.baidu.com/subview/\d+/\d+\.htm）。而其它内容通过查找相应的class和title即可。

结果输出是将爬取到的主人公、主人公网页链接和与其相关的人物关系及名字存为本地的html文件。

主调用程序就是协调以上各个部分的工作，本次实验是以范冰冰的个人网页作为第一个网址开始的，进而读取网页内容、解析网页内容，将相关的链接存入待爬取的链接中，将相关的内容保存到数组中，继续读取待爬取链接中的链接，循环上述过程，直到满足终止条件，将所有数据输出为本地的html文件。

本次实验的终止条件为爬取的网页数量达到1200条或待爬取的链接为空，由于某些明星的个人网页较老，其链接规范不满足之前写下的正则表达，所以不会被加入待爬取链接中，因此本次实验在爬取到1010条的时候停止了。

另外，由于爬取的数据基本是中文，所以在整个过程尤其是在输出的时候需要注意编码问题。

## 3．2 数据处理及存储

工具：Python 2.7、Neo4j、GraphDatabase库

原始数据中是有重复项的，这是因为有的明星的网页链接不止一个，即同一个人对应着不同的链接地址，所以我在Python中通过一个小程序，将重复项去除了，最终剩下1000条数据，并将html文件转成了txt文件。

下一步是将数据存入到图数据库中，本次实验用的是Neo4j，在这里数据将通过节点与节点间的关系表达出来。为了更加方便地将数据导入到neo4j图数据库中，再次将txt中的数据提取出来并生成json格式，这里用到了Python里的json库。数据被分成了两个json文件，一个含有节点信息，即含有主人公姓名和网页链接，另一个含有关系信息，即含有主人公姓名和与之相关的人物。两个json文件的内容分别如图3-3和图3-4所示。

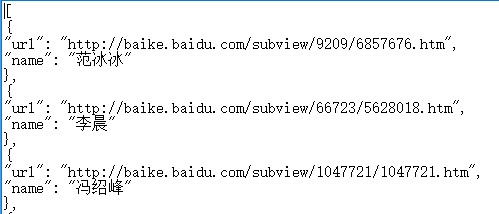


图3-3 节点json文件

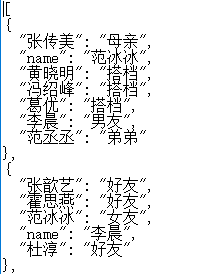


图3-4 关系json文件

从图3-4中可以看出，在关系json文件里，除去主人公的姓名，我是以人物姓名为键，关系为键值的，这是因为姓名是唯一的，而关系却不是，也就是说同一个关系下可能对应着多个人名，例如范冰冰的搭档有黄晓明、冯绍峰和葛优三人，那么在“搭档”这个键下需要存储三个键值，这样在读取json文件时会比上述的表示方法麻烦一点。

做完上述的准备工作，就可以把数据导入到Neo4j图数据库里了。利用GraphDatabase库中的driver和session产生一个会话控制，方便在Python中对Neo4j进行操作。分别读取上边生成的节点json文件和关系json文件，使用"CREATE (a:Persons {name: name })"和""CREATE (node1)-[relation: relation]->(node2)"语句在图数据库中生成节点和关系。此处需要注意的是，不要重复建立相同名字的人物节点。可在http://localhost:7474/中查看结果，导入后的效果如图3-5所示。

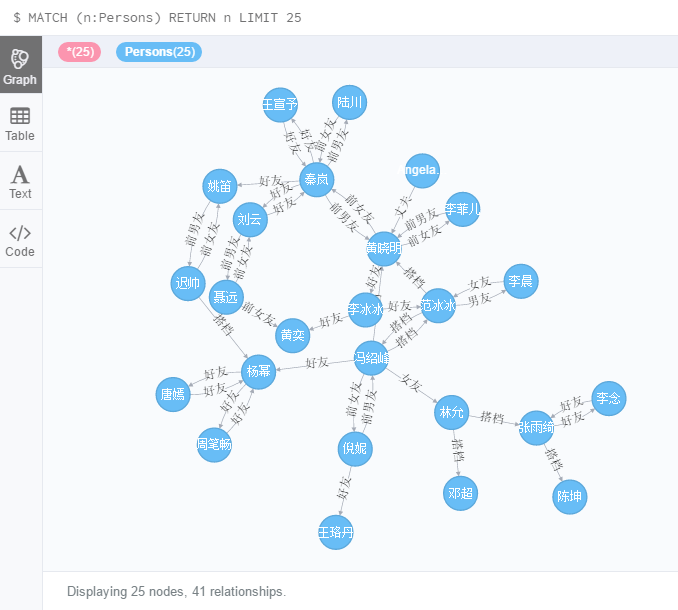


图3-5 Neo4j图数据库

在数据库的左边Relationship Types一栏可以看到有各种关系，任意点击一个即可看到具有该关系的各种人物，例如点击“同学”，就会产生如图3-6所示的结果。

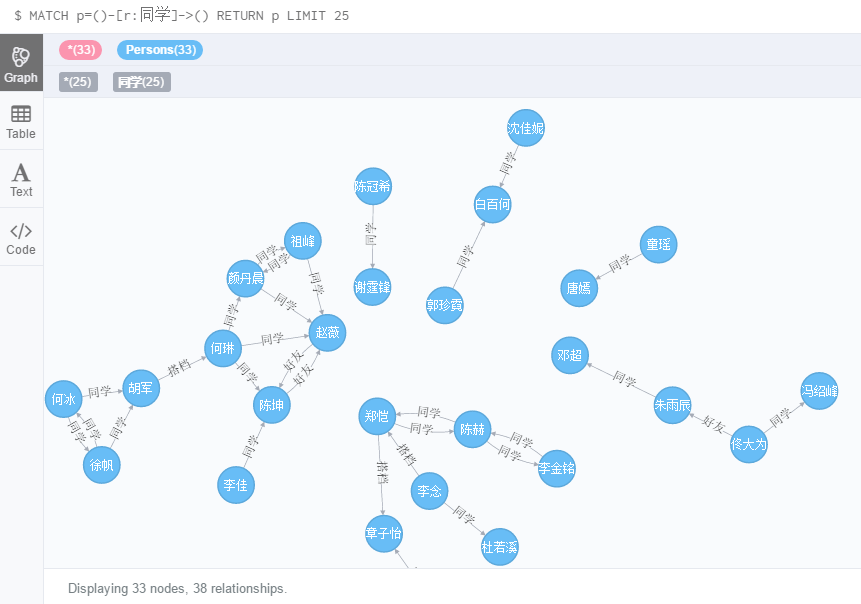


图3-6 具有“同学”关系的节点

## 3．3 数据可视化

工具：Python 2.7、Flask、Echarts.js

Flask是一个使用 Python 编写的轻量级 Web 应用框架，本次实验将利用它搭建一个小型服务器。我一共构造了三个URL（都是本地访问的），分别是http://127.0.0.1:5000/nodes、http://127.0.0.1:5000/index1和http://127.0.0.1:5000/index2，它们上边分别承载了所有数据（节点和关系）、可实现N跳以内关系查询的实例和所有数据的可视化，可以通过访问这三个网址分别查看。

利用render\_template()函数可以引入可视化的脚本文件，即html文件。在html文件中利用ajax()到网址“/nodes”获取数据，该网址上的数据可以直接从第二步构建的数据库中获得，并通过对会话控制对数据库进行操作以得到想要的数据，但是这里我并没有采用这种方法，我是通过读取本地的json文件（与第二步中的json文件稍有不同，为方便可视化，这里将形式改了些）将数据以json形式传到对应网址上。这也就意味着在实现N跳结构时，我不能用到数据库中现有的对数据的操作，而是需要自己对数据进行取舍。在html文件中，有两个文本获取框和一个按钮，需要输入的文本分别是明星的名字和想要查询的跳数，当点下确定按钮后，将会呈现出该明星N跳以内的关系图谱。实现N跳的代码是写在html文件中的，通过for循环来获取输入名字的明星的相关数据并最终呈现出来。可视化的结果将在下一部分展示。

# 四、实验结果

在Flask框架下一共实现了两个可视化结果，一个是全体数据的可视化，可通过访问http://127.0.0.1:5000/index2进行查看，结果如图4-1所示。

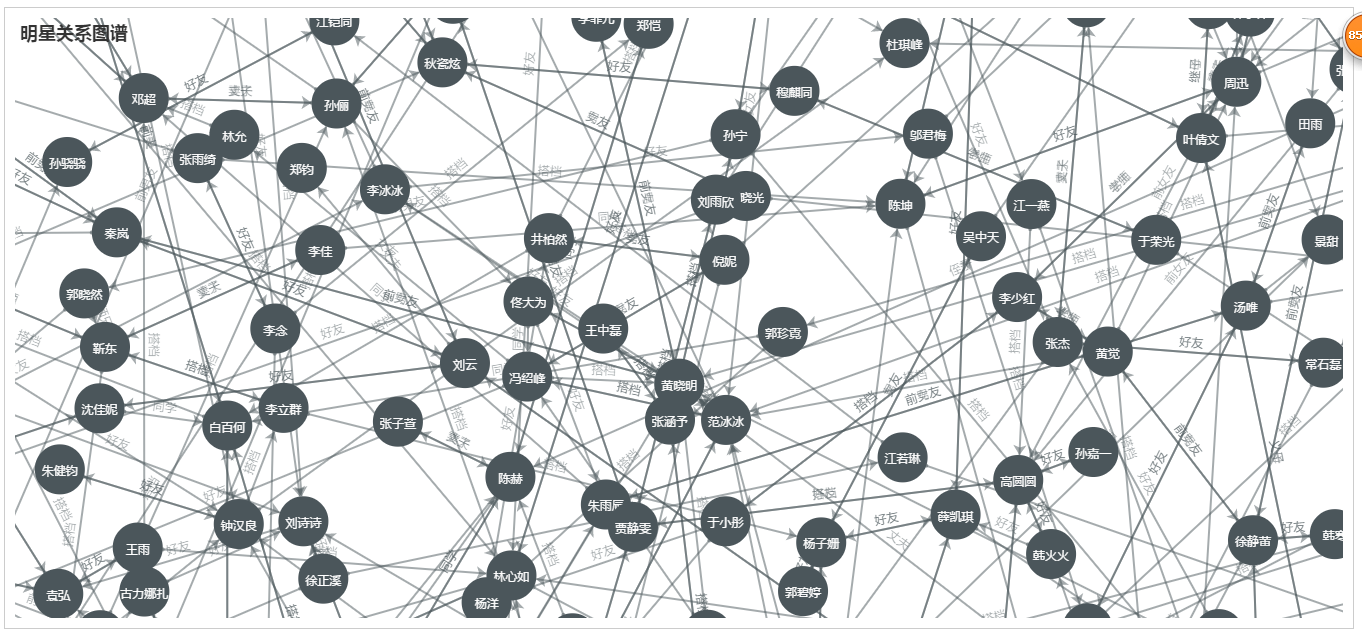


图4-1 所有数据的可视化

从上图中可以看出，这些关系是错综复杂的，是一个强大的关系网络，这也导致可视化的结果看上去并不清晰美观，当把鼠标放到某一个明星的名字上时，会显示出该明星的网页链接和与他（她）有直接关系的人物，效果如图4-2所示。

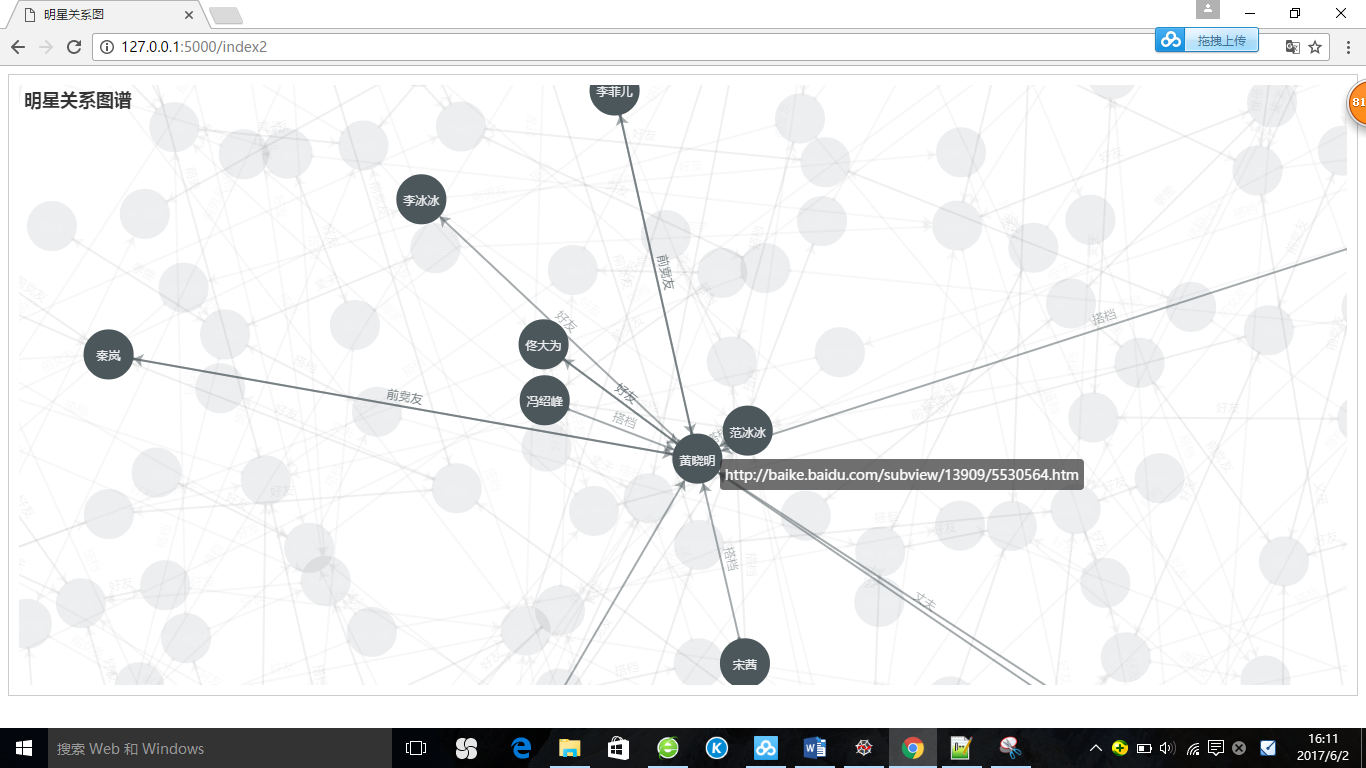


图4-2 单个明星的关系

另一个是实现N跳以内的查询的可视化，可通过访问http://127.0.0.1:5000/index1进行查看，结果如图4-3所示。

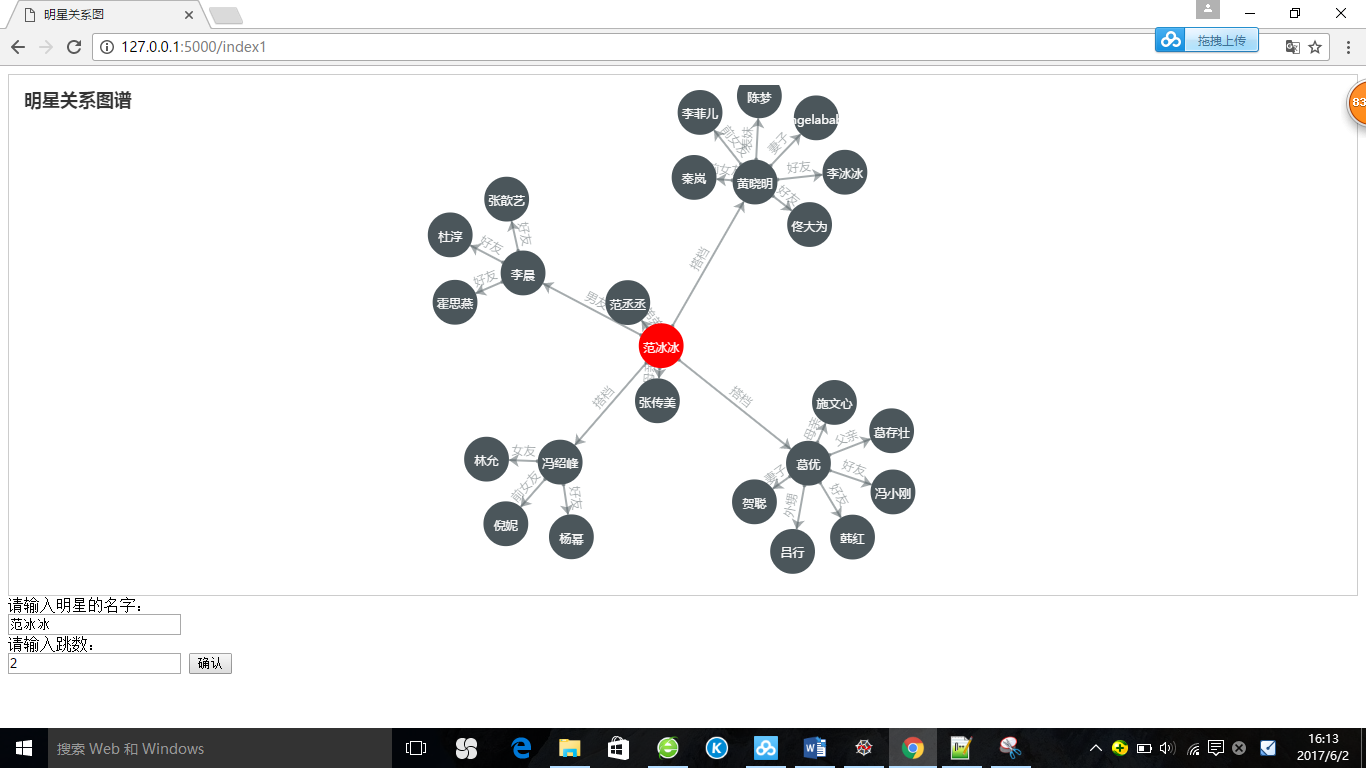


图4-3 范冰冰2跳的关系图谱

可以输入不同的明星和跳数进行查询，例如输入“谢娜”，跳数为“3”，结果如图4-4所示。

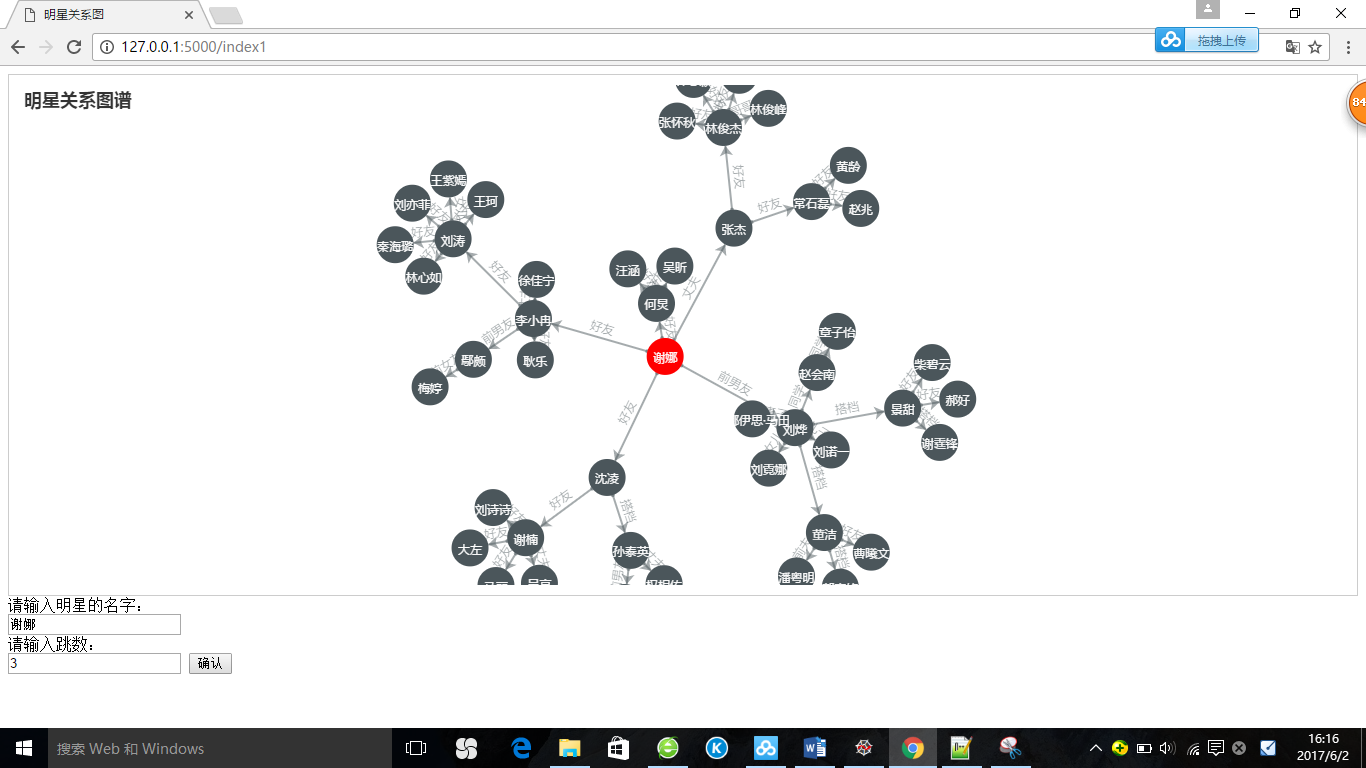


图4-4 谢娜3跳的关系图谱

该查询只体现了单边的关系，即没有双向的箭头，也没有展示兄弟节点间的关系，这是为了使得呈现的结果较为清楚和美观，不会有混乱的视觉效应。

# 五、实验总结

在本次实验中，我完成了数据爬取、数据存储和数据可视化的任务，熟悉了Python语言、JavaScript语言以及Neo4j数据库操作等等，将构建知识图谱的基本流程走了一遍，学习到了很多的东西，而且在不停地制造错误、消除错误的过程中收获了很多经验。

本次实验完成得比较简单，可扩展性还很大，例如爬取的数据内容可以更丰富些，可视化展示的内容可以再多些，形式可以更美观些，这些都需要在以后继续加强训练。

# 附件：文件清单

baike\_spider 爬取数据的程序包

爬取到的原始数据

 清洗数据、去除重复项的程序

 清洗后的数据

 生成节点和关系json文件的程序

 节点和关系json文件

将json格式的数据导入到图数据库的程序

my\_flask 可视化的程序包