

名侦探柯南知识图谱及其问答

小组成员：王泽宇、张龙姣、黄梓淇、林思仪

1. 知识图谱构建

1.1 数据来源

数据爬取

百度百科中对《名侦探柯南》系列相关人物和作品都有较为详尽的词条描述，本项目的知识主要来自于此网站。我们通过爬取百科词条，对所需知识内容进行提取，分别爬取结构化的人物基本信息和非结构化的文本信息，用于后续操作。在编写爬虫程序时，我们主要基于python的requests库获取页面源码，随后通过xpath路径解析的方法定位得到所需内容。首先是对非结构化信息进行爬取。我们选择分集剧情和主要人物介绍作为非结构化文本的爬取目标。在“分集剧情”条目中，我们可以对每集题目和主要剧情进行爬取，同时还可以在“角色介绍”条目中获取对每个角色的基本介绍。将爬取的数据写入txt文档，我们得到了53名主要人物及1068集动画的分集剧情。整理好的txt文档展示如下。





江户川柯南

外表看似小孩，其真实身份是高中生侦探——工藤新一。

和青梅竹马的同学毛利兰一起去游乐园玩，目击到黑衣男子的可疑交易现场，被灌下开发中的药物，变成了小学生的身体。那天以后，为了隐藏真实身份，化名江户川柯南，在青梅竹马的毛利兰家寄住的同时，日复一日解决了许多案件。一切都是为了恢复自己的身体。 [25]

其次是结构化信息的爬取。以上是“名侦探柯南”词条的角色介绍部分。从上图中可以发现，发现每一名主要人物都有对应词条的超链接。我们找好源码中超链接部分的 xpath，就可得到需要爬取的子页面列表。在每个角色对应的子页面中，都有如下格式的一个描述人物基本信息的表格。我们将这部分内容进行爬取，再将爬到的内容整理成 json 格式文件，就可得到描述人物关系的结构化三元组信息。

中文名	江户川柯南	性别	男
外文名	江戸川 コナン(日文名) Conan Edogawa(英文名)	登场作品	名侦探柯南(漫画及其衍生作品) 魔术快斗(动画)
别名	工藤新一(本名)、亚瑟·平井(临时化名)	生日	5月4日(福尔摩斯和莫里亚蒂教授一同坠入瀑布的日期)
饰演	藤崎直		[1]
配音	高山南(日本)、蒋笃慧(中国台湾)、冯友薇(中国台湾)、方雪莉(中国台湾)、丘梅君(中国台湾)、曾允凡(中国台湾)、卢素娟(中国香港)、周恩恩(中国香港)、陆惠玲(中国香港)、黄玉娟(中国香港)、李世荣(中国大陆)、郝幽玥(中国大陆)、Alison Viktorin(美国)	年龄	真实年龄17岁 [2]
		虚拟人物血型	不详，与小兰相同 [3]
		体重	18 kg [4]
		真实身份	高中生侦探工藤新一 [5]
		就读	帝丹小学1年B班 [5]

其中，用于爬取人物信息表格的爬虫程序如下：

```
#对每个子网页进行爬取，摘录结构化信息
for i in range(len(names)):
    # 提前获取好的人名信息，同时也是三元组的其中一个实体
    try:
        # 获取页面信息
        r=requests.get(preurl+subpages[i],headers={'User-Agent' : random.choice(my_headers)})
        code=r.encoding
        content=r.content
        rhtml=str(content,'utf-8')
        soup=bs(rhtml,'html.parser')
        script=soup.find_all('script')
        page=r.text
        tree=etree.HTML(page)
        # 百科词条开头的表格
        dt_path='''//div[@class="basic-info J-basic-info cmn-clearfix"]//
            dt[@class="basicInfo-item name"]/text()'''
        # 爬取表格左列的关系信息
        dd_path='''//div[@class="basic-info J-basic-info cmn-clearfix"]//
            dd[@class="basicInfo-item value"]'''
        # 爬取表格右列的具体内容，由于可能有换行等信息，还需进一步处理
        dt=[i.replace('\xa0','') for i in tree.xpath(dt_path)] #收集表格左列内容 (即关系)
        dd=tree.xpath(dd_path)
        values=[]
        for i in range(len(dd)):
            values.append(tree.xpath(dd_path)[i].xpath('string().').replace('\n',''))
        # 对爬到的表格右列进行处理，将换行内容替换为' '，保证表格左列与右列的一一对应，
        # 并且将整理后的内容进行存储 (三元组的另一实体)
    except:
        print('error')
```

整理好的 json 文件实例如下。

```
"工藤新一":
{
  "中文名": "工藤新一",
  "日文名": "工藤 新一",
  "英文名": "Jimmy Kudo",
  "别名": "江户川柯南 (化名), “平成年代的福尔摩斯”等",
  "饰演": "小栗旬 (真人版第1、2部) 沟端淳平 (真人版第3、4部, 真人连续剧版)",
  "性别": "男",
  "登场作品": "漫画《名侦探柯南》及其衍生作品《青山刚昌短篇集》《魔术快斗》《犬夜叉》动画",
  "生日": "5月4日",
  "年龄": "17岁",
  "身高": "174 cm",
  "就读": "帝丹高中2年B班"
},
```

数据清洗

(1) 根据爬取的数据建立实体集合后, 统计从半结构化数据中获取的属性列表, 对同一含义的属性进行合并, 清洗非中英文字符, 并去除出现频率过低的属性, 保留以下 15 种属性, 清洗后的实体及其属性可查看文件 ./data/new_property_clean.csv

或 ./data/new_property_json.csv

生日:string, 毕业院校:string, 性别:string, 职业:string, 就读:string, 警衔:string, 身份:string, 单位:string, 年龄:string, 忌日:string, 登场作品:string, 真实身份:string, 初登场:string

(2) 根据爬取的数据建立实体集合后, 过滤头实体和尾实体不属于实体集合的关系数据。手动将一部分双向关系拆成单项关系, 如将“父子”关系转换为“父亲”和“子女”。将具有双向含义的关系, 如“同学”、“同事”交换头实体和尾实体, 生成新的三元对, 补充缺少数据。

(3) 将爬取的关系数据转换为 cnSchema 支持的概念, 但由于我们的项目主要处理《名侦探柯南》世界观下的知识图谱, cnSchema 中现有的概念中缺少一部分关系的描述, 因此我们新增一部分关系概念, 共定义下列 18 种关系

['亦敌亦友', '被喜欢', '晚辈', '合作', '师生', '父母', '前同事', '子女', '同学', '长辈', '敌人', '情侣', '兄弟姐妹', '前任', '配偶', '好友', '喜欢', '同事']

1.2 文本深度关系抽取

在 1.1 节中通过爬虫程序爬取了半结构化的人物信息, 从爬取的人物名称列表中抽取《名侦探柯南》知识图谱中的实体, 并从爬取的半结构化数据构造知识图谱中的人物关系。除此之外, 我们尝试从 1.1 中爬取的分集剧情和人物文本介绍中抽取人物关系。

模型训练

参考项目 <http://openkg.cn/dataset/a-harry-potter-kg>, 深度关系抽取模型采用 Chinese-BERT-wwm 中文预训练模型, 并使用 <https://github.com/taorui-plus/OpenNRE> 提供的人物关系数据集进行训练, 该数据集共包含父母、夫妻、祖父、师生、同门等十四个人物关系, 训练得到的模型保存在 ./ckpt/people_chinese_bert_softmax.pth.tar 中

从文本中进行关系抽取

(1) 实体匹配

对于读入的文本进行分句，并在文本子句中匹配实体名称。在文本中人物实体出现的形式可能是完整姓名，也可能是人物别称，我们爬取了

<http://htmfiles.englishhome.org/Japsurnames/Japsurnames.htm> 中的日本姓氏表，对于人物实体的中文姓名分别取出他们的姓和名（如“工藤新一”拆分为“工藤”和“新一”），并对于其中不常用的称呼进行手动裁剪，根据别名补充一部分先验知识。别称与相应的实体 id 存储在 ./data/names 文件中。

对于可能的别称进行匹配，获取实体在分句中的位置，将匹配到的实体和所在分句位置存储在列表里。由于可能存在同姓氏的情况，我们对于匹配到的字符串进行了去重，保证取出的字符串不是之前取出的字符串的子串。

```
id_loc0 = []
id_list0 = []
for id in subname2id:
    if id in sentence:
        loc = [(item.start(), item.end()-1) for item in re.finditer(id, sentence)]
        id_list0.append(id)
        id_loc0.append(loc[0])
id_loc = []
id_list = []
for i in range(0, len(id_list0)):
    isNew=True
    id=id_list0[i]
    for j in range(0, len(id_list0)):
        if(i == j):
            continue
        if(id in id_list0[j]):
            if(id != id_list0[j] or j > i):
                isNew = False
                continue
    if(isNew):
        id_list.append(id)
        id_loc.append(id_loc0[i])
```

(2) 关系预测

对于出现实体数量大于 2 的分句，预测在这一分句中出现的任意两个实体之间的关系。最终关系列表中仅保留预测置信度大于 0.95 的预测结果。

```
relation_list = []
for data in tqdm(new_data):
    text = data['text']
    t_pos = data['t']['pos']
    h_pos = data['h']['pos']
    rela = model.infer(data)
    if(rela[1]>0.95 and rela[0]!='unknown'):
        print(text, [text[t_pos[0]:t_pos[1]+1], text[h_pos[0]:h_pos[1]+1], rela])
        relation_list.append([text[t_pos[0]:t_pos[1]+1], text[h_pos[0]:h_pos[1]+1], rela])
```

预测示例如下：

1. 分句 data['text']：江户川柯南之后寻求阿笠博士的帮助，在被青梅竹马毛利兰询问自己名字时，化名为江户川柯南。

实体：毛利兰 t_pos: (23, 25)

实体：工藤新一 h_pos: (0, 4)

预测结果：['毛利兰', '江户川柯南', ('情侣', 0.9562152028083801)]

2. 分句 data['text']：江户川柯南在阿笠博士的提议下，寄居于小兰的父亲毛利小五郎家中，解决各种案件的同时秘密调查黑衣组织。

实体：兰（别称对应毛利兰） t_pos: (19, 19)

实体：毛利小五郎 h_pos: (23, 27)

预测结果：['毛利小五郎', '兰', ('父母', 0.9952424764633179)]

3. 分句 data['text']：灰原哀，日本漫画《名侦探柯南》及其衍生作品中的角色，帝丹小学一年 B 班学生、少年侦探团成员，原黑衣组织科学家。

实体：灰原哀 t_pos: (0, 2)

实体：柯南（别称对应江户川柯南） h_pos: (15, 16)

预测结果：['灰原哀', '柯南', ('好友', 0.9857942461967468)]

但在人物介绍分句中同时出现多个人物实体时，对于预测结果产生了干扰，存在一部分推断错误的关系需要后续进行手动清洗。

（3）清洗关系预测结果

基于模型的预测结果中同一对实体间可能存在多种关系，我们按照预测关系出现次数和预测关系的置信度两个数据确定两个实体的关系，优先选择出现预测出现次数最多的关系，出现次数相同时则选择预测置信度较高的关系。清洗后一部分预测结果如下所示

京极真, 铃木园子, 情侣, 0.9990885257720947

京极真, 毛利兰, 夫妻, 0.9921330809593201

羽田秀吉, 世良真纯, 兄弟姐妹, 0.9439747333526611

服部平次, 江户川柯南, 好友, 0.9980144500732422

服部平藏, 服部平次, 父母, 0.9996293783187866

远山和叶, 服部平次, 好友, 0.982006311416626

服部平次, 工藤新一, 好友, 0.9943497776985168

远山和叶, 江户川柯南, 父母, 0.9980663657188416

远山银司郎, 远山和叶, 父母, 0.9994105100631714

妃英理, 毛利小五郎, 夫妻, 0.9956905245780945

工藤优作, 工藤新一, 好友, 0.9704906940460205

工藤有希子, 工藤优作, 夫妻, 0.9478380084037781

通过 1.1 节中直接爬取的关系更加完善精准，基于文本的预测的有效关系数量较少，准确度偏差，因此仅将文本抽取的结果手动清洗后进行一部分补充，主要采用 1.1 节中爬取的关系进行构建知识图谱。最终构建关系展查看文件./data/web_relation_new.csv。

2. 数据存储

数据库

我们选用 [neo4j 图数据库](#) 来存储我们的名侦探柯南人物关系知识图谱。

neo4j 版本：4.4.16

JDK 版本：11.0.17

Schema

Node:

点类型: People

点属性:

[peopleId:ID(People), 中文名:string, 别名:string, 生日:string, 毕业院校:string, 性别:string, 职业:string, 就读:string, 警衔:string, 身份:string, 单位:string, 年龄:string, 忌日:string, 登场作品:string, 真实身份:string, 初登场:string]

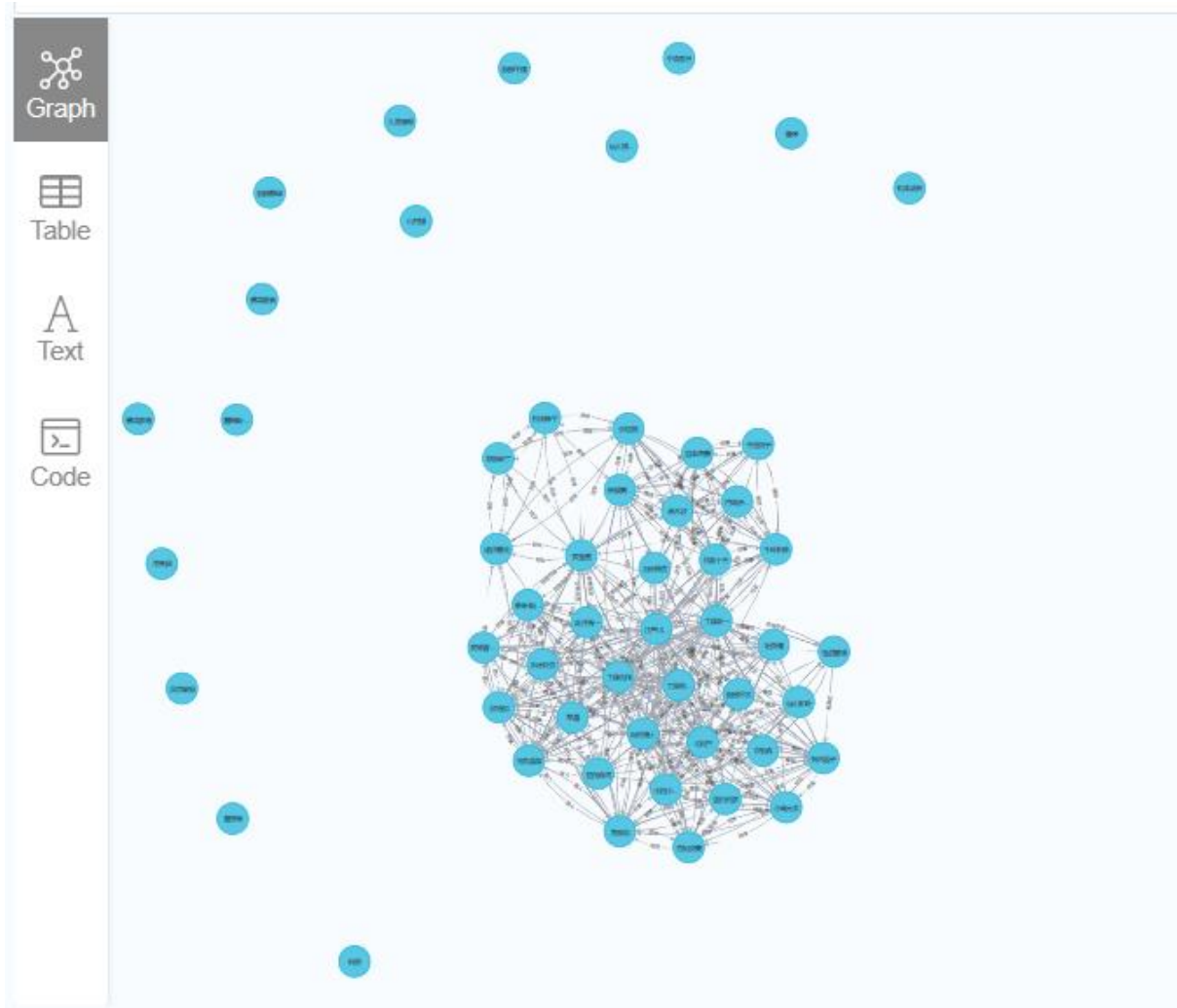
Edge:

边类型:

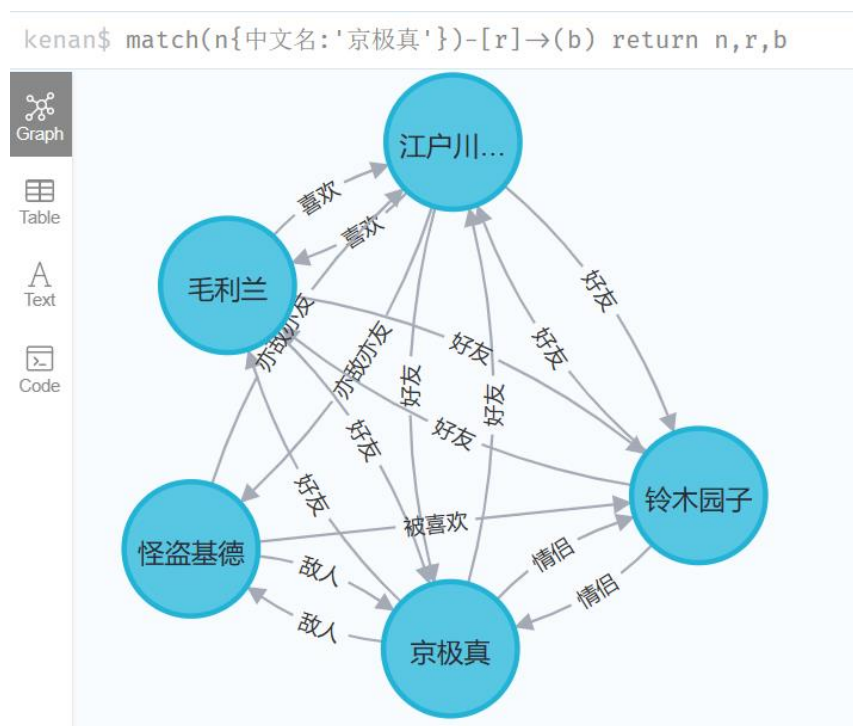
['亦敌亦友', '被喜欢', '晚辈', '合作', '师生', '父母', '前同事', '子女', '同学', '长辈', '敌人', '情侣', '兄弟姐妹', '前任', '配偶', '好友', '喜欢', '同事']

数据可视化展示:

全局展示: `match (n) return n`



展示京极真的相关关系: `match(n{中文名:'京极真'})-[r]->(b) return n,r,b`



展示京极真的属性:

Node properties

People

<id>	11	
peopleId	11	
中文名	京极真	
别名	蹴击贵公子	
年龄	18岁	
性别	男	
毕业院校	杯户高中	
登场作品	漫画《名侦探柯南》及其衍生作品	

数据存储

我们使用 neo4j 执行脱机数据库的完整备份, 将数据库转储到称为 `<database>.dump` 的单文件存档中。

文件名: `kenan.dump`

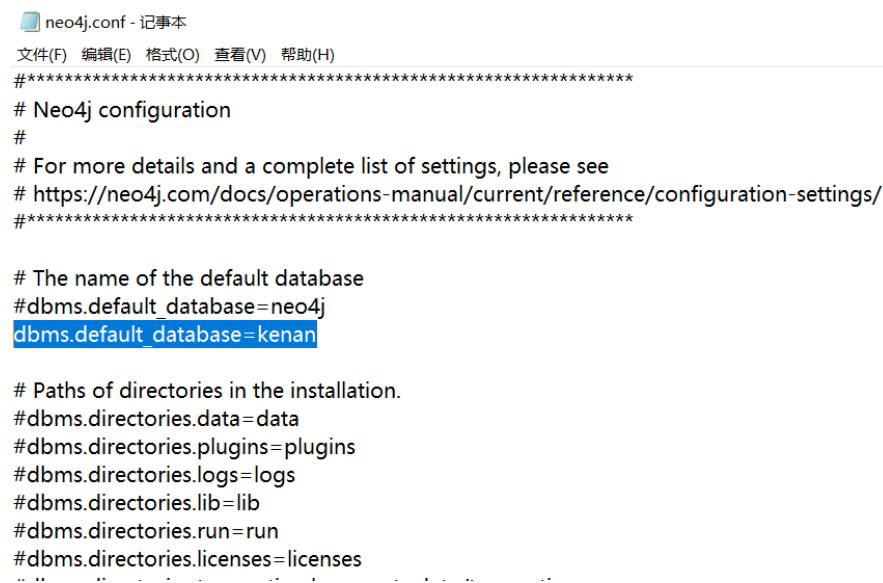
导入方式:

1. 保持 neo4j 服务关闭
2. 进入 neo4j 安装目录下的 bin 目录, 或设置环境变量
3. 执行命令: `neo4j-admin.bat load --from=??? --database=kenan --force=???` 填入 dump 文件路径

4. 修改 neo4j 安装目录下 conf 目录内文件 neo4j.conf，添加如下一行：

```
dbms.default_database=kenan
```

如图：



```
neo4j.conf - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
*****
# Neo4j configuration
#
# For more details and a complete list of settings, please see
# https://neo4j.com/docs/operations-manual/current/reference/configuration-settings/
*****

# The name of the default database
#dbms.default_database=neo4j
dbms.default_database=kenan

# Paths of directories in the installation.
#dbms.directories.data=data
#dbms.directories.plugins=plugins
#dbms.directories.logs=logs
#dbms.directories.lib=lib
#dbms.directories.run=run
#dbms.directories.licenses=licenses
..
..
```

5. 打开 neo4j 服务，如执行命令 neo4j.bat console

注：以上命令为 win 环境下格式，在 linux 下不需要 .bat 后缀，如 neo4j.bat console 变为 neo4j console

3. 知识图谱应用：基于语义分析的知识图谱问答

简介

基于上述的结构化数据和图数据库，我们在知识图谱上进行了问答这一应用。用户可以输入一些语句，问答程序将对输入语句进行关键词分析和语义分析，在知识图谱的数据库中进行对应的查找，最后生成结果返回答案给用户。

算法具体流程

1. 接收用户问句作为输入。
2. 对用户问句进行关键词分析。用 AC 自动算法进行匹配，从问句中识别出是否存在人物姓名、关系、属性、组织等关键词。
3. 根据问句中识别出的关键词，对问句进行归类。
4. 确定问句类型，结合对应的关键词，连接 neo4j 图数据库，并在数据库中寻找对应的结果。
5. 将数据库中得到的结果返回给用户。

目前支持的问句形式

1. 查询人物的属性：包括中文名, 初登场, 别名, 单位, 就读, 年龄, 性别, 毕业院校, 生日, 登场作品, 真实身份, 职业, 警衔。例如：工藤新一的生日是几号？
2. 查询人物的关系对象，包括兄弟姐妹, 前任, 同事, 同学, 喜欢, 合作, 好友, 子女, 情侣, 敌人, 父母。例如，毛利兰的父母是谁？
3. 查询组织所属成员，包括警察、黑衣组织等。例如，黑衣组织有哪些成员？

程序具体实例

```
→ python3 main.py
```

Q: 工藤新一的生日是多少?

A: 工藤新一的生日是 5月4日

Q: 灰原哀的身份是什么?

A: 灰原哀的身份是 小学生、变小前是黑衣组织科学家

Q: 毛利兰的父母是谁?

A: 毛利兰的父母是妃英理, 毛利小五郎

Q: 吉田步美的同学有哪些 ?

A: 吉田步美的同学是 灰原哀, 江户川柯南, 小岛元太, 圆谷光彦

Q: 黑衣组织的成员有哪些?

A: 黑衣组织的成员有 琴酒, 伏特加, 基安蒂, 科恩, 贝尔摩德