南京工業大學
NANJING TECH UNIVERSITY

用户数据采集与关联分析

（结课作业）

许成文

202321054025

信管2301

第一讲 课程导言与分词

1. 学习使用在线NLPIR分词系统或微词云分词或清华大学分词演示系统（案例演示截图）；

2. 安装python（anaconda）（编写输出“Hello World. Hello ‘你的姓名’”）；

THULAC：一个高效的中文词法分析工具包

欢迎使用THULAC中文分词工具包demo系统

黄旭华，1926年3月12日出生于广东省汕尾市，原籍广东省揭阳市。1949年毕业于上海交通大学。历任北京海军核潜艇研究室副总工程师、中船重工集团公司核潜艇总体研究设计所研究员、名誉所长。1994年当选为中国工程院院士

〖测试 Try〗

黄旭华_np ，_w 1926年_t 3月_t 12日_t 出生_v 于_p 广东省汕尾市_ns ，_w 原籍_n 广东省_ns 揭阳市_ns 。_w 1949年_t 毕业_v 于_p 上海交通大学_ni 。_w 历任_v 北京_ns 海军_n 核潜艇_n 研究室_n 副总_j 工程师_n 、_w 中_f 船_n 重工_j 集团公司_n 核潜艇_n 总体_n 研究_v 设计所_n 研究员_n 、_w 名誉_n 所长_n 。_w 1994年_t 当选_v 为_v 中国_ns 工程院_n 院士_n

```
In [1]: print("Hello World. Hello '你的姓名'")
        Hello World. Hello '你的姓名'
```

第一讲 课程导言与分词

3.完成课后作业（001-004，4份代码的运行）001

1.基本分词

```
在[69]: import jieba

在[70]: seg_list1 = jieba.cut("LSTM (Long Short-Term Memory) 是长短期记忆网络，是一种时间递归神经网络，适合于处理和预测时间序列中间隔和延迟相对较长的重

在[71]: print('$'.join(seg_list1))
```

LSTM$ ($Long$ $Short$-$Term$ $Memory$) $是$长短期记忆网络$，$是$一种$时间递归神经网络$，$适合$于$处理$和$预测$时间$序列$中$间隔$和$延迟$相对$较长$的$重要$事件$。

```
在[72]: seg_list2 = jieba.cut("黄旭华，1926年3月12日出生于广东省汕尾市，原籍广东省揭阳市，1949年毕业于上海交通大学，历任北京海军核潜艇研究室副总工程师、

在[73]: print('@'.join(seg_list2))
```

黄旭华@，@1926@年@3@月@12@日出@生于@广东省@汕尾市@，@原籍@广东省@揭阳市@，@1949@年@毕业@于@上海交通大学@，@历任@北京@海军@核潜艇@研究室@副@总工程师@、@中船重工集团公司@核潜艇@总体@研究@设计所@研究员@、@名誉@所长@，@1994@年@当选@为@中国工程院院士@。

2.加入词典，是针对第二个片段的，希望是能够完整把“长短期记忆网络”这个术语整体分割出来

```
在[74]: jieba.load_userdict('dict.txt')

在[75]: seg_list_dict = jieba.cut("LSTM (Long Short-Term Memory) 是长短期记忆网络，是一种时间递归神经网络，适合于处理和预测时间序列中间隔和延迟相对较长

在[76]: print('/'.join(seg_list_dict))
```

LSTM/ (/Long/ /Short/-/Term/ /Memory/) /是/长短期记忆网络/，/是/一种/时间递归神经网络/，/适合/于/处理/和/预测/时间/序列/中/间隔/和/延迟/相对/较长/的/重要/事件/。

3.加入停用词，针对第一个片段，希望的结果是，结果中不会出现“的、是”等虚词

```
在[46]: stopwords = [line.strip() for line in open('stop_words.txt', 'r', encoding='utf-8').readlines()]

在[43]: seg_list_stopw = jieba.cut("曾任有一份真诚的爱情摆在我的面前，我没有珍惜，等到失去的时候才追悔莫及，人世间最痛苦的事情莫过于此。如果上天能够给我

在[44]: final = ''

在[45]: #这是一行注释，进行分词结果的过滤
for seg in seg_list_stopw:
    if seg not in stopwords:
        final += seg + '/' #添加，累加

在[47]: print(final)
```

曾经/有/一份/真诚/爱情/摆在/我/面前/我/没有/珍惜/等到/失去/时候/才/追悔莫及/人世间/最/痛苦/事情/莫过于此/如果/上天/能够/给/我/一个/重新/来/过/机会/我会/对/那个/女孩子/说/三个/字/：/ '/我爱你/' /如果/事业/给/这份/爱/加上/一个/期限/我/希望/一万年/

第一讲 课程导言与分词

3.完成课后作业（001-004，4份代码的运行）002

功勋科学家-黄旭华-传记文本分词

```
In [1]: # 简单分词
In [6]: import jieba
```

现在，可以开启你的小组项目的第一个小小任务啦！就是对一小段有关"功勋科学家"的文本进行分词处理。

```
In [9]: seg_list_huang = jieba.cut('黄旭华，1926年3月12日出生于广东省汕尾市，原籍广东省揭阳市，1949年毕业于上海交通大学，历任北京海军核潜艇研究室副总工
In [10]: print('/'.join(seg_list_huang))
黄旭华/，/1926/年/3/月/12/日出/生于/广东省/汕尾市/，/原籍/广东省/揭阳市/，/1949/年/毕业/于/上海交通大学/，/历任/北京/海军/核潜艇/研究室/副/
总工程师/、/中船重工集团公司/核潜艇/总体/研究/设计所/研究员/、/名誉/所长/，/1994/年/当选/为/中国工程院院士/。
In [11]: # 加入用户词典
In [12]: jieba.load_userdict('dict.txt')
In [13]: seg_list_huang = jieba.cut('黄旭华，1926年3月12日出生于广东省汕尾市，原籍广东省揭阳市，1949年毕业于上海交通大学，历任北京海军核潜艇研究室副总工
```

```
In [14]: print('/'.join(seg_list_huang))
黄旭华/，/1926/年/3/月/12/日出/生于/广东省/汕尾市/，/原籍/广东省/揭阳市/，/1949/年/毕业/于/上海交通大学/，/历任/北京/海军/核潜艇/研究室/副/
总工程师/、/中船重工集团公司/核潜艇/总体/研究/设计所/研究员/、/名誉/所长/，/1994/年/当选/为/中国工程院院士/。
In [15]: # 加入词典之后，哪些词汇被分出来了呢？
In [16]: # 使用停用词表
In [17]: # stopwords = [line.strip() for line in open('stop_words.txt','r', encoding='utf-8').readlines()]
In [18]: stopwords = open('stop_words.txt','r', encoding='utf-8').read()
         stopwords = stopwords.split('\n')
In [19]: stopwords
Out[19]: ['的', '了', '是', '啊', '，', '。', '、', '；', '：', '停用']
In [20]: seg_list_huang = jieba.cut('黄旭华，1926年3月12日出生于广东省汕尾市，原籍广东省揭阳市，1949年毕业于上海交通大学，历任北京海军核潜艇研究室副总工
In [21]: final = ''
```

```
In [22]: for seg in seg_list_huang:
             if seg not in stopwords:
                 final+= seg+'/'
In [23]: print(final)
黄旭华/1926/年/3/月/12/日出/生于/广东省/汕尾市/原籍/广东省/揭阳市/1949/年/毕业/于/上海交通大学/历任/北京/海军/核潜艇/研究室/副/总工程师/中船
重工集团公司/核潜艇/总体/研究/设计所/研究员/名誉/所长/1994/年/当选/为/中国工程院院士/
```

第一讲 课程导言与分词

3.完成课后作业（001-004，4份代码的运行）003

In [1]:
```
import jieba
from collections import Counter
```

In [2]:
```
# 文本
text = '''
落实“企业管理年”主题，加强QEHS三体系建设，
通过自动化、数字化、智能化升级改造加快新一代信息技术与企业生产经营融合，
打造精益制造能力，提升精细化管理水平，助力公司从“制造”升级为“智造”，
从而提高经营效率和效益；通过集团一体化智数管理平台建设与运营，
丰富企业供应链管理、生产工艺控制等管理工具，不断增强生产经营过程数据获取与分析能力，
强化全过程一体化管理，提高自动化、数字化、智能化的供应链管理能力，
为体系安全稳定运行与管理水平提升保驾护航，致力打造安全智能化工厂；
打造助剂互联网技术合作和商务合作平台，构建具有国际竞争力的供应链体系。
'''
```

In [3]:
```
# 1. 分词处理
words = jieba.lcut(text)
```

```
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\XU~1\AppData\Local\Temp\jieba.cache
Loading model cost 1.043 seconds.
Prefix dict has been built successfully.
```

In [5]:
```
words
```

Out[5]:
```
['\n',
 '落实',
 '“',
 '企业',
 '管理',
 '年',
 '”',
 '主题',
 '，',
 '加强',
 'QEHS',
 '三',
 '体系',
 '建设',
 '，',
 '\n',
 '通过',
 '自动化',
 '、',
```

In [6]:
```
# 2. 定义要统计的特殊词汇
target_words = ['数字化', '智能化', '安全']
```

In [7]:
```
# 统计词频
word_counts = Counter(words)
```

In [8]:
```
# 输出特定词汇的词频统计结果
print("特定词汇词频统计结果：")
for word in target_words:
    print(f"'{word}'：{word_counts[word]}次")
```

```
特定词汇词频统计结果：
'数字化'：2次
'智能化'：3次
'安全'：2次
```

In [9]:
```
# 输出所有词汇的词频（按频率降序）
print("\n所有词汇词频统计（前20个）：")
for word, count in word_counts.most_common(20):
    print(f"'{word}'：{count}次")
```

```
所有词汇词频统计（前20个）：
'，'：13次
'
'：9次
'管理'：5次
'、'：5次
'与'：4次
'“'：3次
'企业'：3次
'”'：3次
'体系'：3次
'智能化'：3次
'经营'：3次
'打造'：3次
'能力'：3次
'供应链'：3次
'建设'：2次
'通过'：2次
'自动化'：2次
'数字化'：2次
'升级'：2次
'生产'：2次
```

第一讲 课程导言与分词

3.完成课后作业（001-004，4份代码的运行）004

In [1]:
```
import requests
import json

# 定义DeepSeek API的URL和headers
DEEPSEEK_API_URL = "https://api.deepseek.com/v1/chat/completions"
API_KEY = "yourkey"  #自己复制过来
```

In [2]:
```
# 准备prompt和论文文本
paper_text = """
随着肿瘤免疫微环境（Tumor Immune Microenvironment，TIME）研究的深入，
T细胞耗竭（T cell exhaustion）被认为是限制免疫治疗效果的关键机制之一。
本研究基于免疫编辑理论，提出了一种基于单细胞RNA测序（scRNA-seq）的T细胞状态动态识别方法。
具体而言，我们使用Seurat与Monocle3等生物信息学工具对50例非小细胞肺癌患者的肿瘤样本进行细胞亚群聚类和轨迹分析，
结合pseudotime推断T细胞从激活到耗竭的转化过程。此外，借助CellChat软件构建细胞间通讯网络，
进一步识别可能诱导T细胞耗竭的免疫抑制信号通路，如PD-1/PD-L1和TGF-β路径。研究结果揭示了T细胞功能衰竭的关键节点，并为个体化免疫治疗提供了潜在靶点
"""

prompt = f"""
请从以下科技论文文本中提取包含理论、方法、工具的实体或专业术语，以json字典的格式输出：

{paper_text}
"""
```

In [3]:
```
# 准备请求数据
data = {
    "model": "deepseek-chat",
    "messages": [
        {"role": "user", "content": prompt}
    ],
    "temperature": 0.3
}

headers = {
    "Content-Type": "application/json",
    "Authorization": f"Bearer {API_KEY}"
}

# 发送请求
response = requests.post(DEEPSEEK_API_URL, headers=headers, data=json.dumps(data))
```

In [4]:
```
# 处理响应
if response.status_code == 200:
    result = response.json()
    try:
        entities = result['choices'][0]['message']['content']
        print("提取到的实体和专业术语：")
        print(entities)
    except KeyError:
        print("无法解析API响应，原始响应：")
        print(result)
else:
    print(f"请求失败，状态码：{response.status_code}")
    print(response.text)
```

```
请求失败，状态码：401
{"error":{"message":"Authentication Fails, Your api key: ****rkey is invalid","type":"authentication_error","param":null,"code":"invalid_req
uest_error"}}
```

第一讲 课程导言与分词

4.阅读压缩文件中（“实体抽取论文-换成PDF”）中的其中一篇论文，并做阅读总结（《基于学术论文全文的研究方法句自动抽取研究》）

本文针对如何从学术论文中自动提取研究方法句这一任务，进行了系统性研究。作者创新地将研究方法句分为两类：“论文使用方法句”和“论文引用方法句”。前者指论文本身采用的研究方法，后者指论文中所引用的他人研究方法。这一分类有助于构建方法知识库和跟踪方法的发展演变。

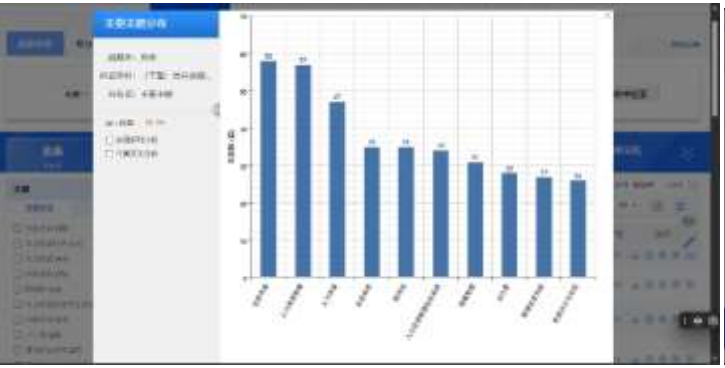
在技术实现上，作者构建了一个基于深度学习的句子分类模型，并进行了多组对比实验。在词向量表示方面，比较了BERT和word2vec的效果；在特征提取层，测试了卷积神经网络（CNN）、双向长短期记忆网络（BiLSTM）以及带注意力机制的BiLSTM（Att-BiLSTM）等不同结构；此外，还比较了“单层次”（直接分为三类）和“两层次”（先判断是否为方法句，再分类）两种模型框架。实验结果显示，基于BERT词向量的单层次BiLSTM模型表现最好，整体准确率达到93.42%，Macro-F1值达到81.57%。

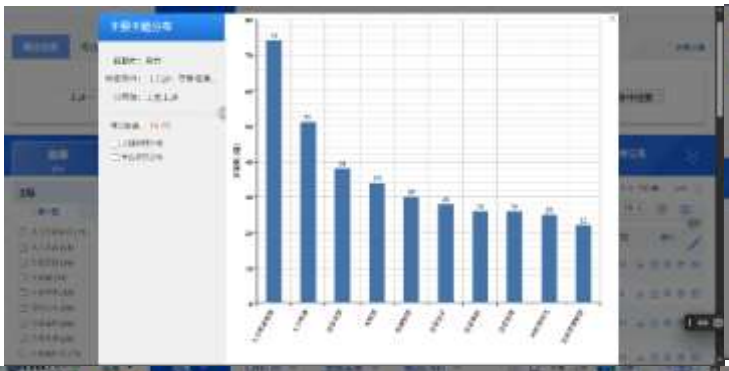
在应用分析方面，作者使用该模型对《情报学报》2009–2018年间的1170篇论文进行了自动抽取，共得到15276句论文使用方法句和5655句论文引用方法句。通过对这些句子的分布进行时间序列分析，发现该期刊在2009–2013年间，注重研究方法的论文比例有所上升；而2015年之后，研究方法句的比例整体呈下降趋势，引用方法句的比例也有所降低。作者推测，这可能反映出该期刊近年来更加关注情报学的基础理论和学科体系建设。

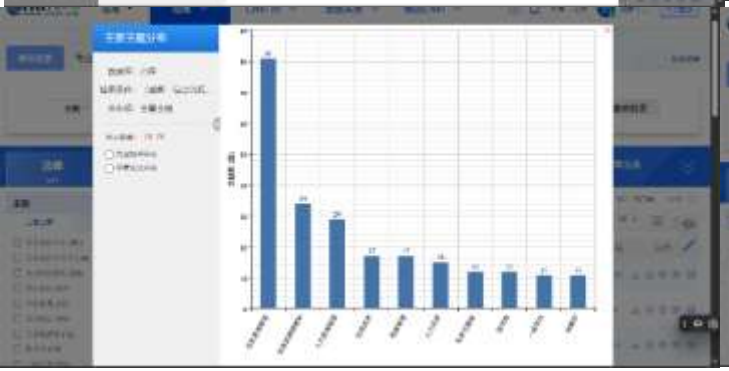
综上，本研究不仅验证了深度学习模型（尤其是BERT+BiLSTM）在学术文本细粒度信息抽取中的有效性，还通过实际数据抽取与分析，为观察学科发展趋势提供了一种基于方法描述的量化视角。未来可进一步探索半监督学习以降低标注成本，并将模型推广到更多期刊及学科领域。

第二讲 词频统计

1. 基于CNKI数据库统计分析2014-2024年（近10年），“信息资源管理”变化趋势。







在研究前期（2014-2018年左右），关键词多集中于“信息组织”“信息检索”“图书馆”“知识管理”等传统技术与机构视角。

进入2019年后，研究热点明显向“大数据”“人工智能”“数据治理”“数据安全”等技术驱动领域迁移。

近年来（2022-2024年），“智慧服务”“数字化转型”“数据要素”“高质量发展”等与国家战略和经济社会发展紧密结合的应用性、战略性关键词显著增多，反映出该领域的研究正从技术与管理层面向赋能经济社会发展的价值层面深化拓展。

第二讲 词频统计

2.完成ppt中的程序运行，包括全文词频统计，指定类型词频统计；代码一

```
In [14]: # 输出词频的前N个
for i in range(100):
    print(articlelist[i])
```

```
('黄旭华', 53)
('核潜艇', 32)
('采集', 29)
('学术', 22)
('资料', 21)
('工作', 17)
('成长', 15)
('小组', 14)
('院士', 13)
('进行', 13)
('专业', 13)
('我国', 12)
('技术', 12)
('研制', 12)
('工程', 11)
('访谈', 10)
('介绍', 8)
('科学', 8)
('主要', 8)
('第一代', 8)
('人生', 7)
('历史', 7)
('传记', 7)
('思想', 7)
('及其', 7)
('要求', 6)
('一生', 6)
('研究', 6)
('成就', 6)
('传主', 6)
('过程', 6)
('按照', 6)
('设计', 6)
('重点', 5)
('黄旭', 5)
('求学', 5)
('精神', 5)
('先后', 5)
('实现', 5)
('实物', 5)
('保密', 4)
('09', 4)
('重要', 4)
('完整', 4)
('中国', 4)
('船舶', 4)
('完成', 4)
('反映', 4)
('依据', 4)
('系统', 4)
('描述', 4)
('时间', 4)
('学习', 4)
('成熟', 4)
('轨迹', 4)
('任务', 4)
('照片', 4)
('其中', 4)
('事件', 4)
('还原', 4)
('国立交通大学', 3)
('节点', 3)
('关于', 3)
('撰写', 3)
('同时', 3)
('处理', 3)
('分析', 3)
('生活', 3)
('历程', 3)
('贡献', 3)
('章节', 3)
('叙述', 3)
('制造', 3)
('背景', 3)
('计划', 3)
('回顾', 3)
('自己', 3)
('曲折', 3)
('大学', 3)
('事业', 3)
('一个', 3)
('李世英', 3)
('特点', 3)
('人员', 3)
('清单', 3)
('本章', 3)
('突破', 3)
('包括', 3)
('直接', 3)
('客观', 3)
('国家', 3)
('阶段', 3)
('代表', 3)
('经过', 3)
('实施', 3)
('形成', 3)
('地下党', 3)
('提升', 3)
('参加', 3)
('展示', 3)
```

第二讲 词频统计

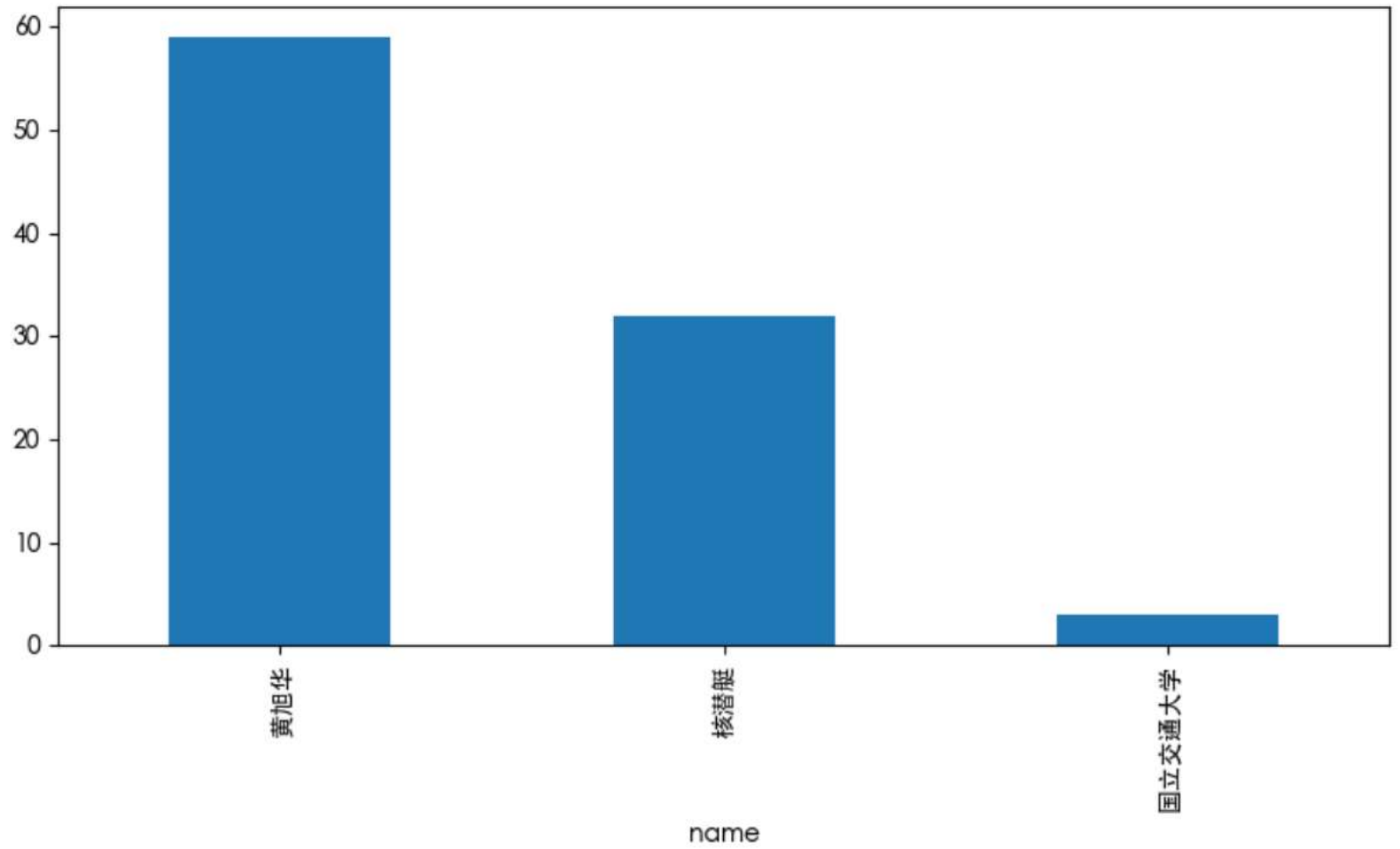
2.完成ppt中的程序运行，包括全文词频统计，指定类型词频统计；代码二

In [13]:
```
# 输出词频的前N个
for i in range(100):
    print(articlelist[i])
```

('董卓', 97)
('吕布', 60)
('曹操', 59)
('袁紹', 57)
('天下', 53)
('玄德', 48)
('貂蟬', 37)
('太守', 36)
('朝廷', 32)
('不可', 31)
('孫堅', 31)
('次日', 26)
('李儒', 25)
('引兵', 25)
('商議', 25)
('天子', 24)
('左右', 23)
('玄德曰', 22)
('太師', 22)
('軍士', 21)
('大喜', 21)
('校尉', 21)
('今日', 21)
('何進', 20)
('太后', 20)
('王允', 20)
('二人', 19)
('公孫瓚', 18)
('郭汜', 18)
('諸侯', 18)
('不能', 18)
('司徒', 18)
('軍馬', 17)
('朱雋', 17)
('肅曰', 17)
('如此', 17)
('盧植', 17)
('百姓', 17)
('何人', 17)
('百官', 16)
('李肅', 16)
('張濟', 16)
('張角', 15)
('大事', 15)
('文醜', 15)
('張飛', 15)
('何故', 15)
('如何', 15)
('留王', 15)
('皇甫嵩', 15)
('袁術', 14)
('只見', 14)
('刺史', 14)
('正是', 14)
('張梁', 14)
('因此', 14)
('此人', 14)
('冀州', 14)
('劉表', 14)
('曹嵩', 13)
('英雄', 13)
('五百', 13)
('不得', 13)
('將軍', 13)
('樊稠', 13)
('是夜', 13)
('五千', 13)
('不到', 13)
('李蒙', 12)
('人馬', 12)
('性命', 12)
('華雄', 12)
('三十', 11)
('大臣', 11)
('叔父', 11)
('趕來', 11)
('一人', 11)
('星夜', 11)
('一日', 11)
('社稷', 11)
('張讓', 11)
('丞相', 11)
('蔡瑁', 11)
('主公', 11)
('相府', 11)
('相見', 10)
('分解', 10)
('不敢', 10)
('一面', 10)
('中郎將', 10)
('不知', 10)
('乘勢', 10)
('三人', 10)
('只得', 10)
('不見', 10)
('何太后', 10)
('大呼', 10)
('忽見', 10)
('再拜', 10)
('下文', 10)

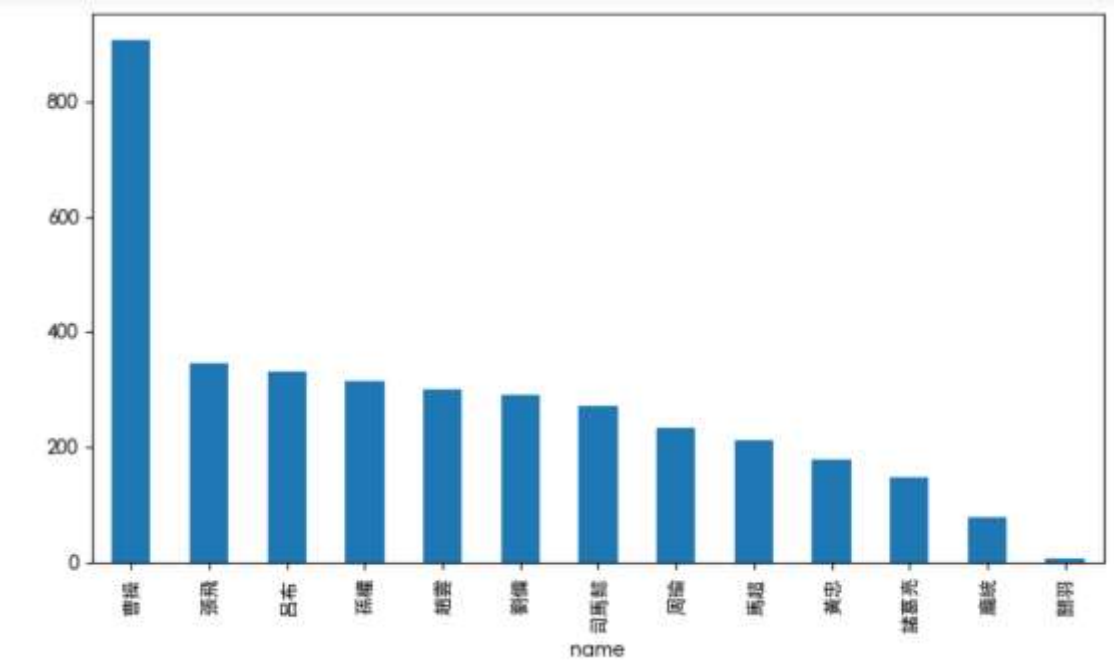
第二讲 词频统计

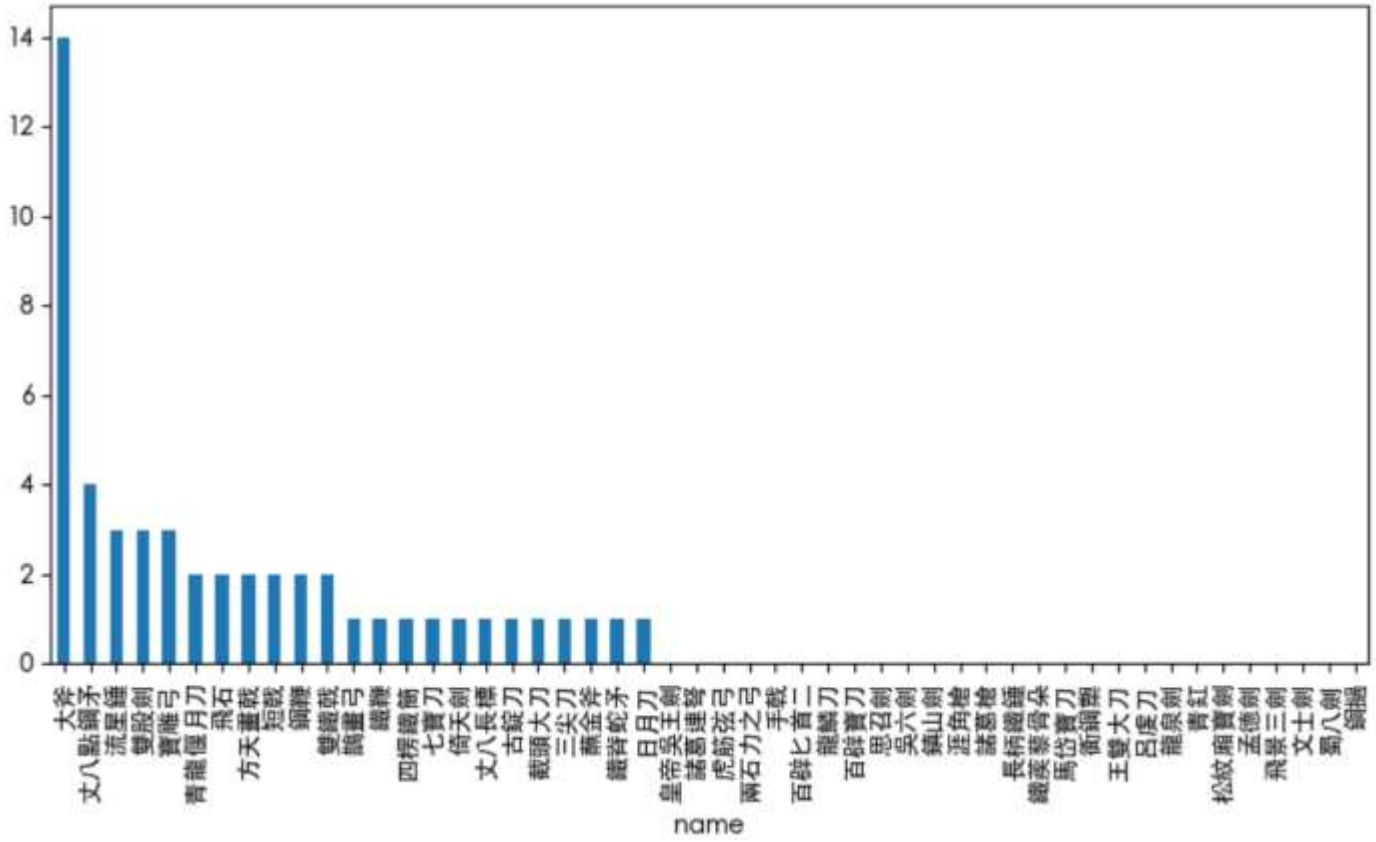
2.完成ppt中的程序运行，包括全文词频统计，指定类型词频统计；代码三



第二讲 词频统计

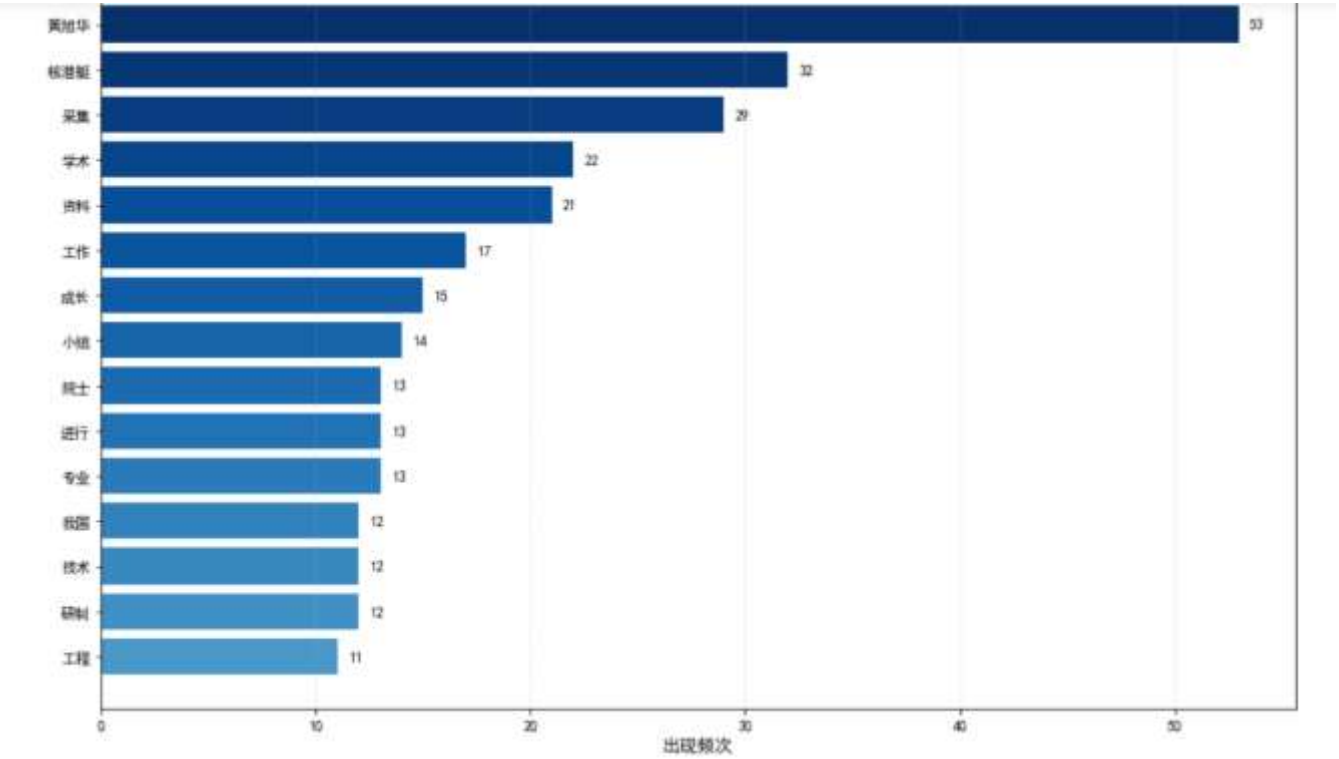
2.完成ppt中的程序运行，包括全文词频统计，指定类型词频统计；代码四

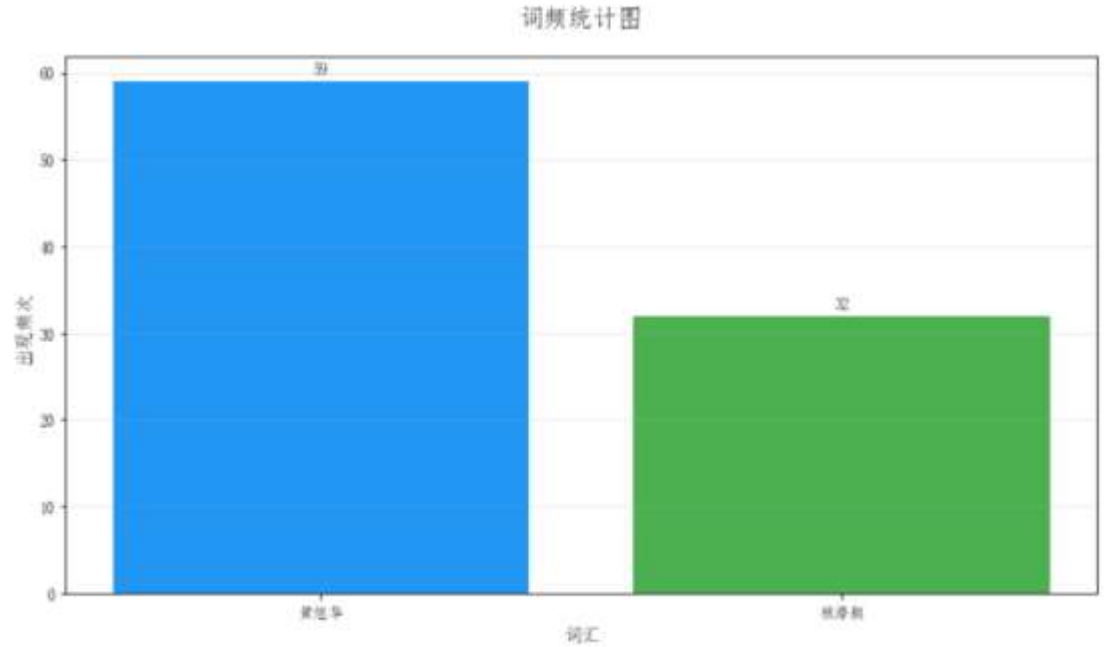




第二讲 词频统计

3.链接功勋科学家： 把ppt中的文本换成功勋科学家黄旭华院士的传记序言文本（文件夹中，科学家博物馆-黄旭华传记序言.txt），1）统计全文词频；2）统计指定词频，如“黄旭华”；





=== 统计摘要 ===
黄旭华：59次
核潜艇：32次

第二讲 词频统计

4.阅读论文“2018-Wang 等 - Long live the scientists Tracking the scientific”，并做阅读总结。

这篇论文研究科学家在历史上的影响力。作者不数论文引用数，而是统计科学家名字在谷歌图书中被提到的次数。

研究发现牛顿和爱因斯坦去世后名声依然很大，说明伟大科学家的影响力能持续数百年。还有“老乡效应”——科学家在自己国家更出名。比如英国书里牛顿被提得多，美国德国书里爱因斯坦更受欢迎。科学家主要因重大成就被记住：牛顿因万有引力，爱因斯坦因相对论。但他们的故事、趣闻也让人们记住他们。

研究列了现代物理学家名声排行榜，爱因斯坦排第一，牛顿第三。这说明老科学家的影响力到今天依然在。

这研究告诉我们，评价科学家不能只看论文引用，还要看他们对社会文化的长远影响。好的科学家就像思想上的“长寿者”——身体不在了，智慧还能影响后世几百年。

虽然方法有局限，比如英文书多其他语言书少，但这给了我们一个新角度：在大数据时代，我们能看到科学家更完整的历史地位。

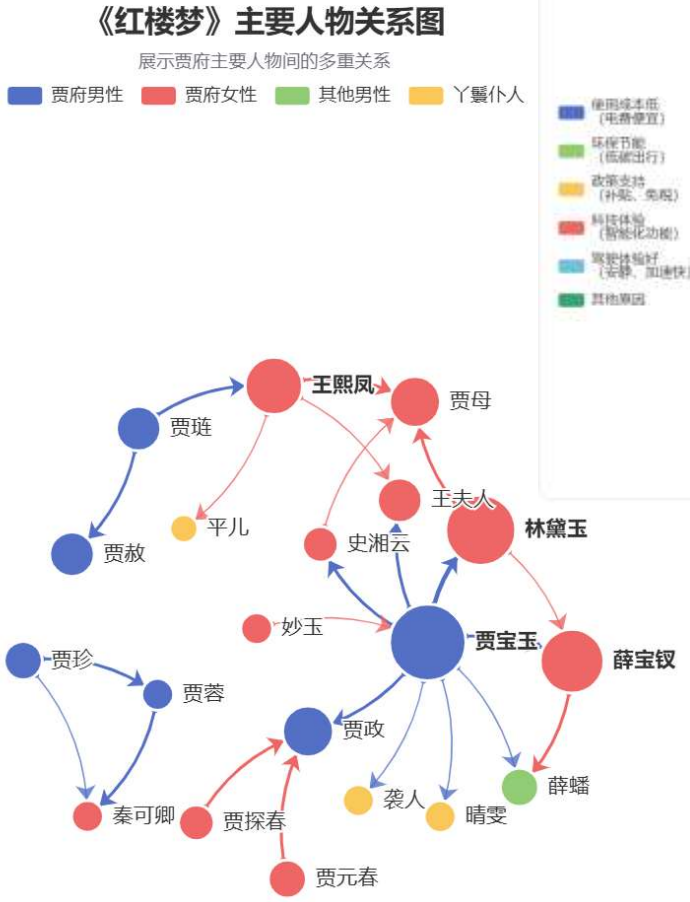
第三讲 词云与可视化

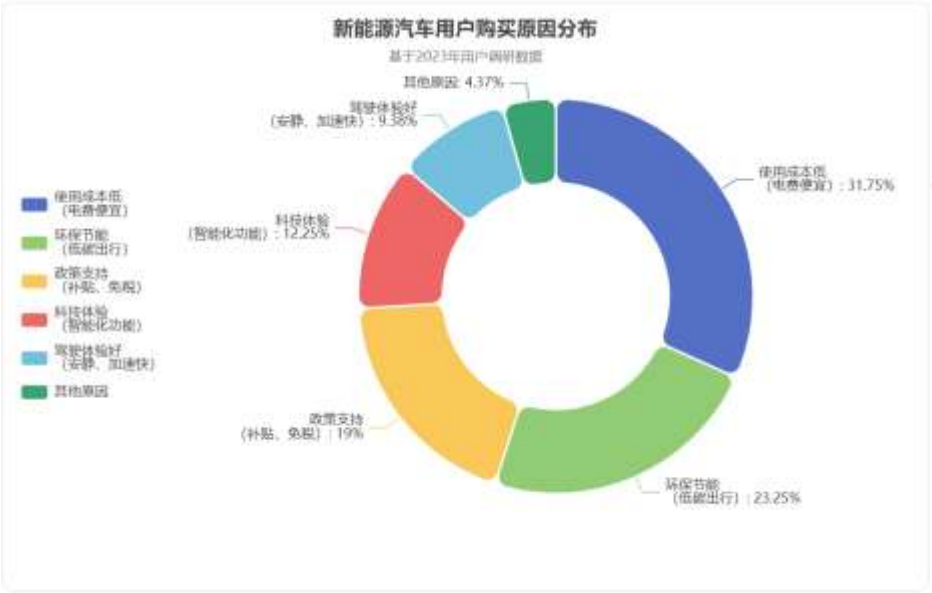
1.用任意一款词云工具，制作一个好看的词云（内容合理即可），并对词云图有一段话的解释。

根据用户调研词云图的分析结果，新能源汽车在推广过程中面临的主要阻力明确指向了充电基础设施建设的不足。图中频繁出现的"充电桩难找"、"小区无法安装"、"充电等待时间长"等关键词，生动展现了消费者在实际使用中遭遇的切实困难。当前充电网络的覆盖密度和布局合理性仍有明显欠缺，特别是在老旧居民区和跨城出行场景中，这种短板直接引发了用户的里程焦虑和便利性担忧。

第三讲 词云与可视化

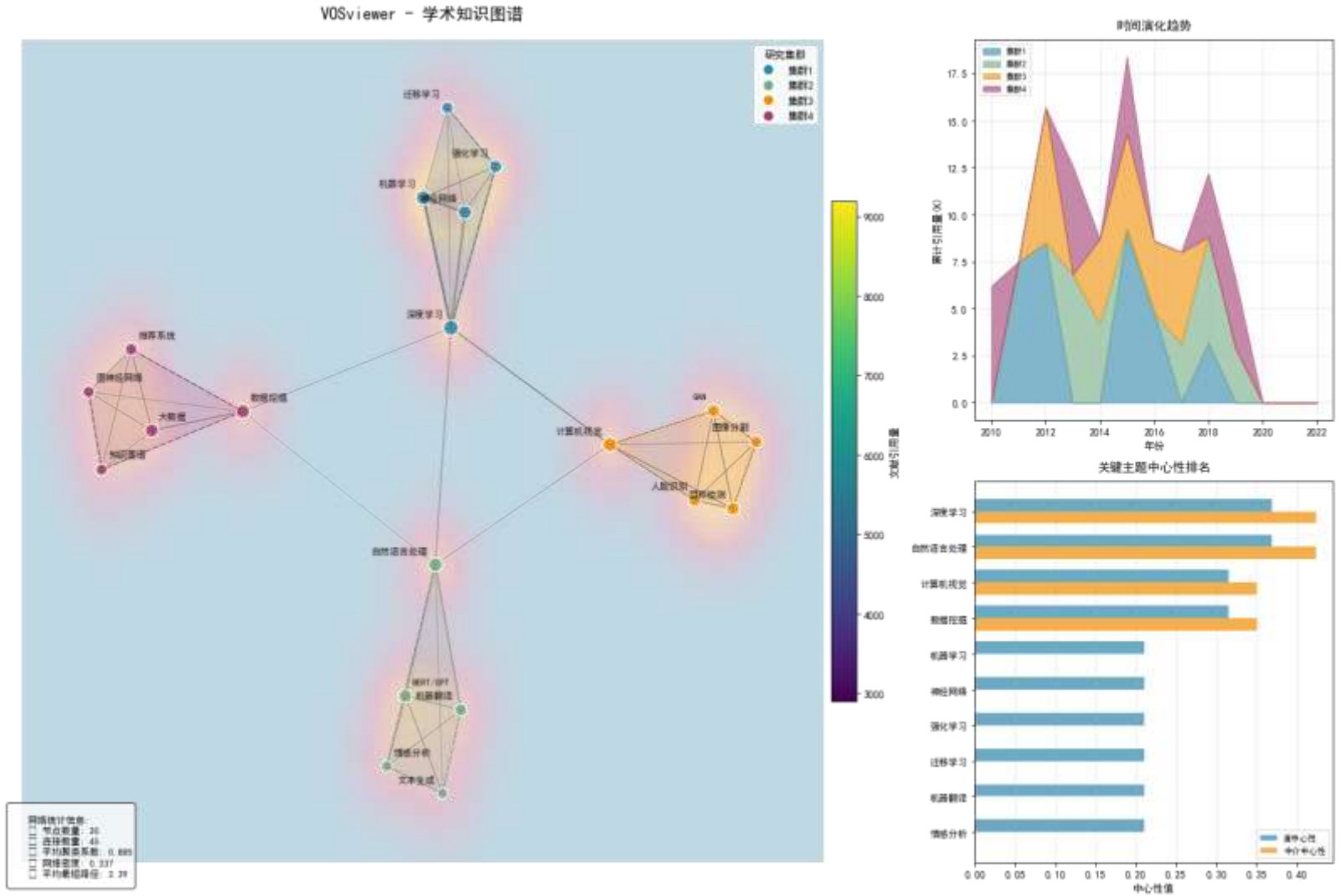
2.使用Echarts，制作3个以上图，其中一个必须是“关系”。





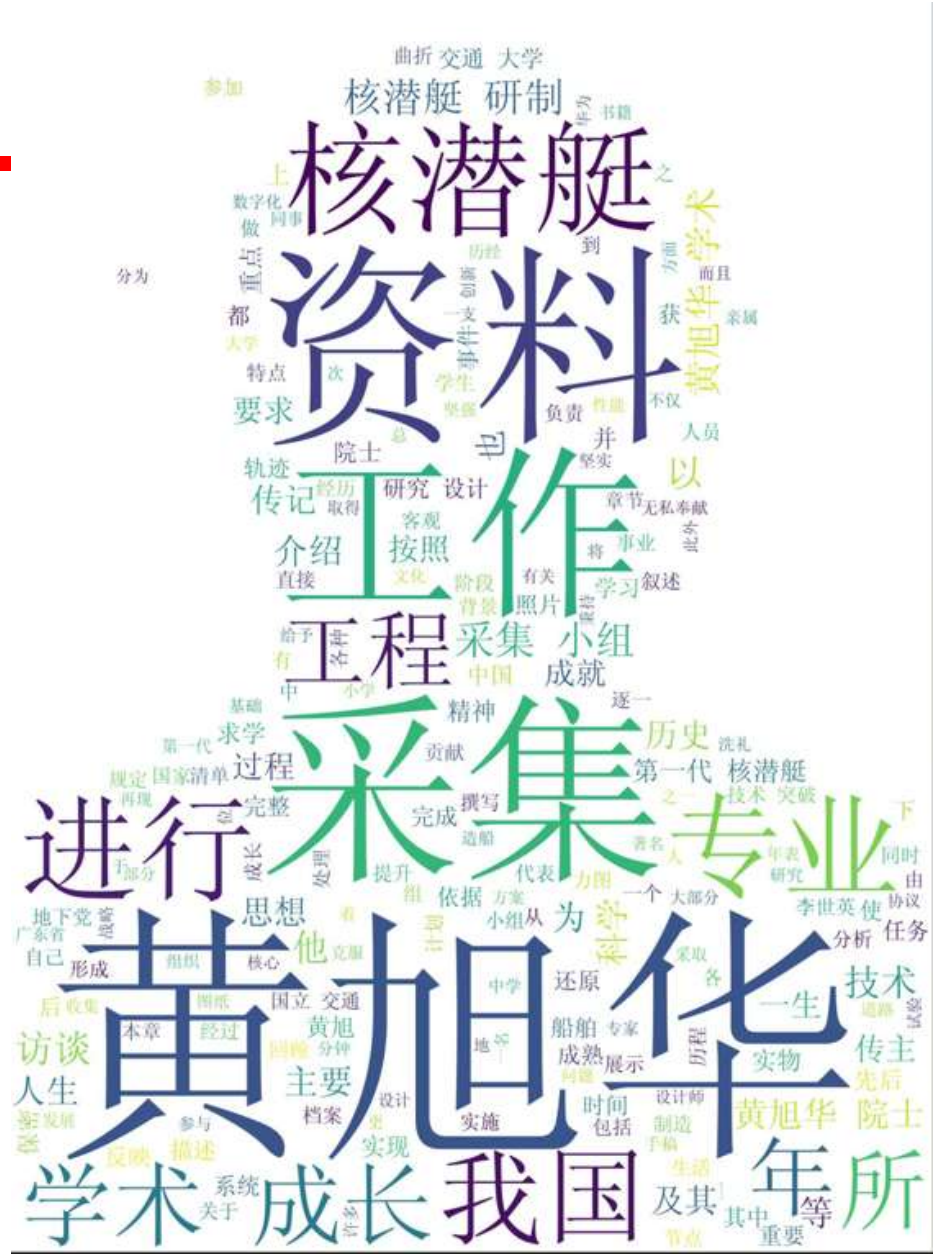
第三讲 词云与可视化

3. 使用 Gehpi 、VOSViewer 、CiteSpace…其中任意一款工具，绘制任意你感兴趣的图谱1-2张。



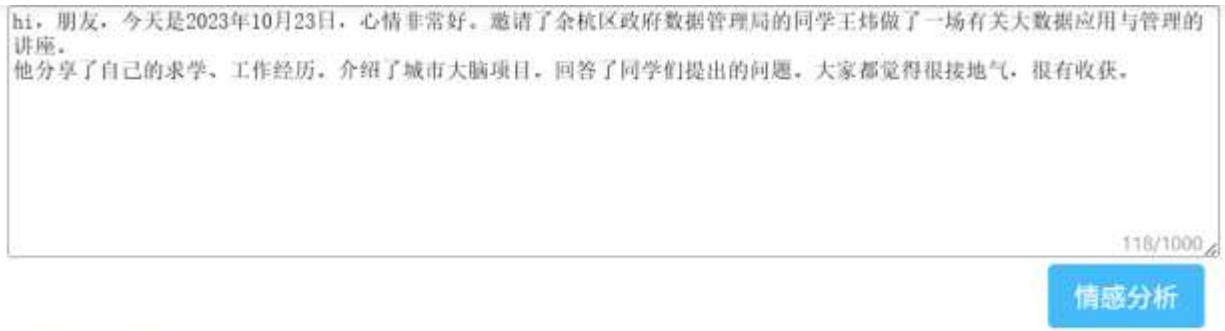
第三讲 词云与可视化

4.采用给的程序，实现一段科学家文本的词云图绘制，越清晰越好（生成的词云图要单独拿出来）。



第四讲 情感分析

1.使用PPT给的情感分析平台（或其它平台），对文本情感进行分析，并截图；



第四讲 情感分析

2. 完成sentiment_analysis_1-sentiment_analysis_4，4份代码。做截图，并简要做代码运行总结分析。1

```
In [22]: # 例如
text_taobao_1 = "显示效果：挺好的 运行速度：目前来说很流畅 拍照效果：拍照效果挺好的 电池续航：一天一冲 总结：目前没啥毛病，用了一天的体验"

In [23]: taobao_1 = SnowNLP(text_taobao_1)

In [24]: taobao_1.sentiments
Out[24]: 0.999947261146611

In [25]: text_taobao_2 = "总结：这是我买过最不满意的一款手机！两千多元的手机这样，真的很不值！"

In [26]: taobao_2 = SnowNLP(text_taobao_2)

In [27]: for sentence in taobao_2.sentences:
             print(sentence)
总结：这是我买过最不满意的一款手机
两千多元的手机这样
真的很不值

In [28]: taobao_2.sentiments
Out[28]: 0.889005139666256

In [29]: # 以上的结果看上去是有问题的，分析的不准确。

In [30]: text_taobao_3 = "显示效果：像素不行 运行速度：微信有时发给不了语音，得重新开机后才能发，才买半个月的手机就这样，客服态度也很差！ 拍照效果：拍照

In [31]: taobao_3 = SnowNLP(text_taobao_3)

In [32]: taobao_3.sentiments
Out[32]: 5.7076094222563434e-05
```

这是一个双语情感分析代码，使用snownlp处理中文、textblob处理英文，让初学者直观比较不同语言NLP处理的差异。代码通过商品评论等生活化示例将抽象概念具体化，采用Jupyter Notebook的交互式设计支持边运行边修改，配合可视化增强学习效果。

成功实现了“快速体验”目标，用最少代码展现NLP应用魅力，能有效激发学习兴趣。但教学仅停留在工具调用层面，未涉及算法原理或模型评估，展示的通用模型在实际业务场景中精度有限。

总体而言，这是一个精心设计的“兴趣触发器”，适合作为机器学习入门实践课，建议在此基础上进一步探究情感分析的核心原理和高级应用。

第四讲 情感分析

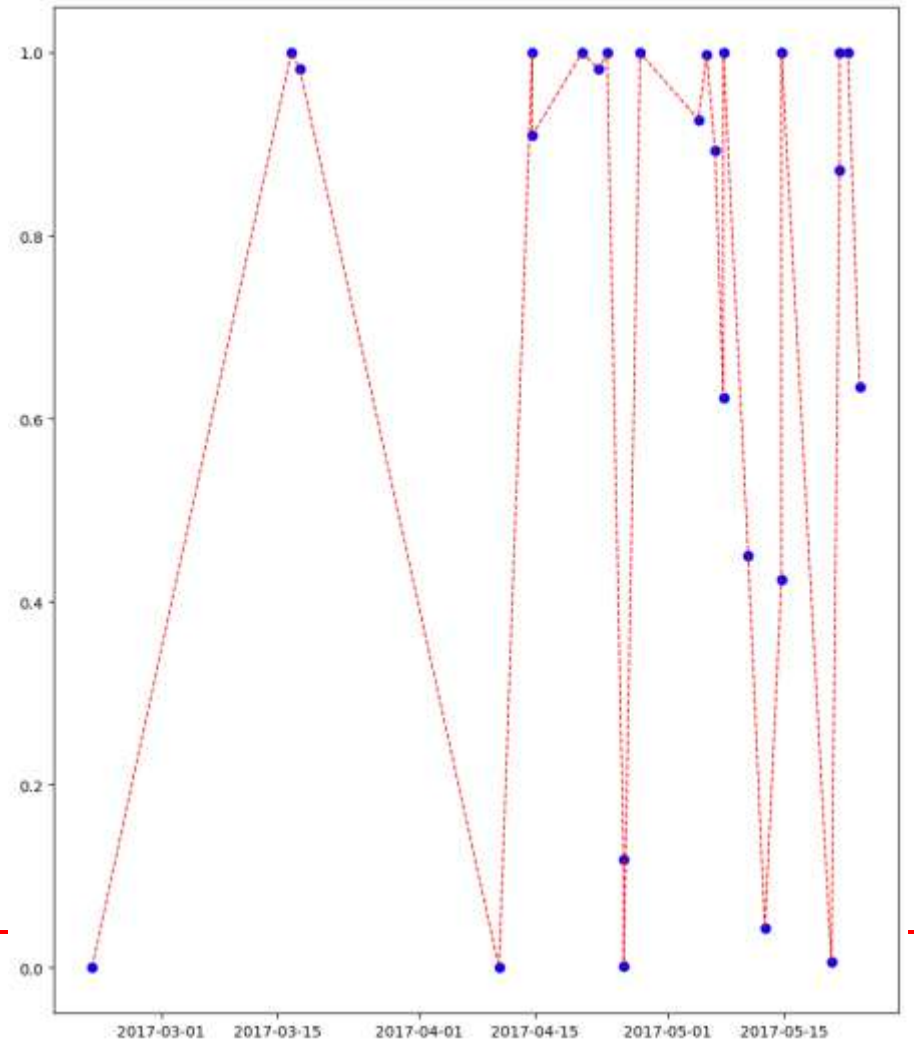
2.完成sentiment_analysis_1-sentiment_analysis_4，4份代码。做截图，并简要做代码运行总结分析。2

这是一个餐厅评论情感分析代码，完整实现了数据处理、情感计算到可视化展示的全流程。

代码通过四个步骤完成：1）pandas读取Excel评论数据 2）snownlp进行单句情感分析 3）批量生成情感值列 4）ggplot绘制时间序列趋势图。情感值范围0-1，数值越高代表越积极。

分析发现情感中位数高于平均值，表明多数评论积极，但少数极端负面评价拉低了整体评分。可视化图表可清晰识别负面评论集中的时间段。

代码结构清晰实用，既适合教学演示也可用于实际项目，同时指出了时间格式处理和通用模型精度等注意事项。



第四讲 情感分析

2.完成sentiment_analysis_1-sentiment_analysis_4，4份代码。做截图，并简要做代码运行总结分析。3

细粒度情感实体抽取结果：

```json
{
  "实体": [
    {
      "部位": "头部",
      "症状": "头痛",
      "情感": "无具体描述"
    },
    {
      "部位": "全身",
      "症状": "疲乏无力",
      "情感": "无具体描述"
    },
    {
      "部位": "皮肤",
      "症状": "异常敏感，触碰疼痛",
      "情感": "无具体描述"
    },
    {
      "部位": "心脏",
      "症状": "心慌",
      "情感": "无具体描述"
    },
    {
      "部位": "胸部",
      "症状": "胸闷",
      "情感": "无具体描述"
    },
    {
      "部位": "背部",
      "症状": "沉重感",
      "情感": "无具体描述"
    },
    {
      "部位": "感官",
      "症状": "对光线和声音极度敏感",
      "情感": "惊恐"
    },
    {
      "部位": "心理",
      "症状": "不愿出门，不想与人交流",
      "情感": "生活毫无意义"
    }
  ]
}
```

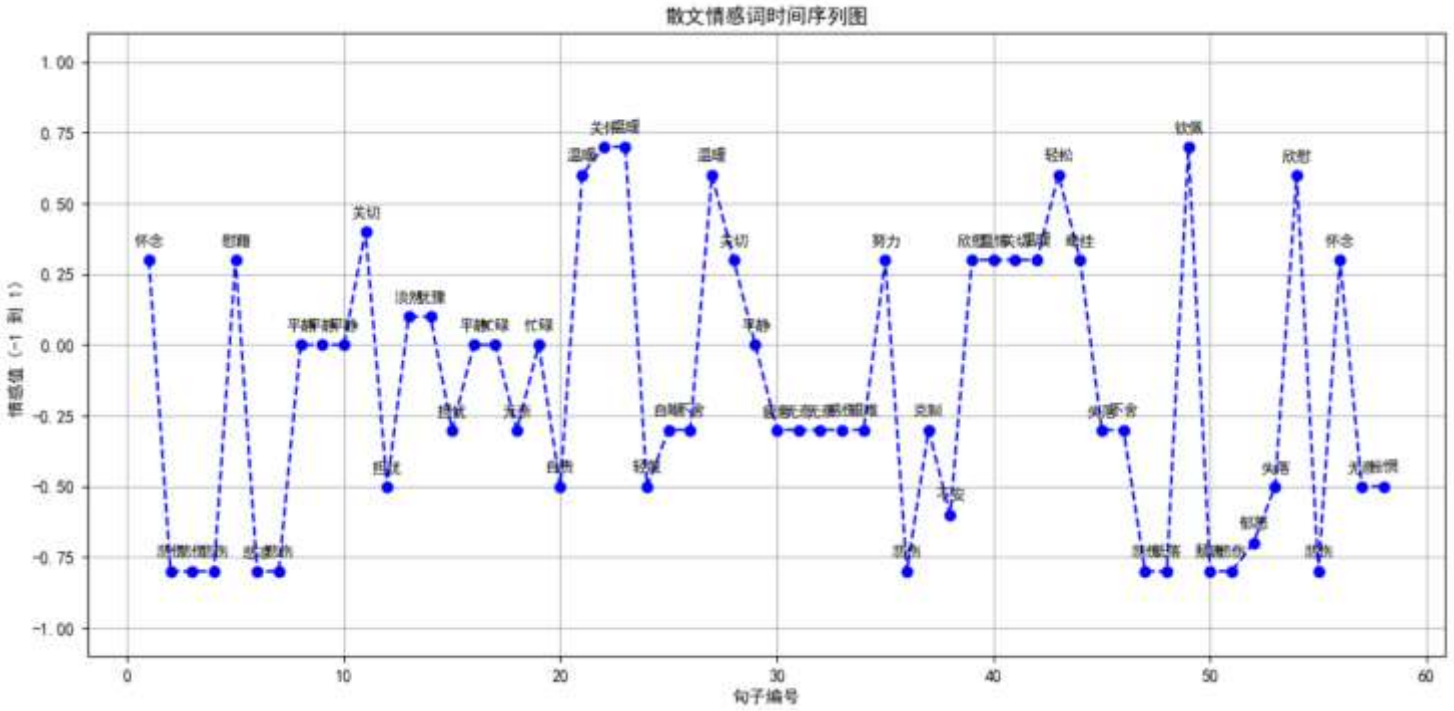
这是一个利用大语言模型对“心理-身体”病痛文本进行深度解析的代码。其核心目标是实现细粒度信息抽取，不仅识别具体症状（如焦虑、头痛），还要提取症状属性（强度、频率）及其语义关系（因果、伴随等）。

项目通过设计结构化指令，引导大模型从文本中抽取出结构化信息。例如，从“一紧张就头疼得睡不着”这句话中，可提取出“紧张”（心理）、“头疼”和“失眠”（身体）三个实体，并建立“紧张→头疼→失眠”的关联网络。

这项技术能将非结构化的患者自述转化为可计算分析的知识图谱，为心理健康评估和临床研究提供数据支持，提升诊疗的精准度。

第四讲 情感分析

2.完成sentiment_analysis_1-sentiment_analysis_4，4份代码。做截图，并简要做代码运行总结分析。4



这段代码实现了一个完整的中文文本情感分析可视化流程。它首先对输入的文本进行预处理，按照句号、感叹号等标点进行分句并为每个句子编号。随后，代码调用大语言模型的API，对每个编号句子进行深度分析，提取其中明确或隐含的情感词或情感实体，并为每个提取出的情感元素计算一个量化的情感得分。

最后，基于句子序号和情感得分，代码绘制了一个时间序列散点图。图中横轴是句子编号，纵轴是情感得分，每个散点代表一个情感词并用其文本作为标签。所有散点按句子顺序用虚线连接起来，直观地展示了文本情感强度随内容推进而演变的波动轨迹。整个流程系统地将非结构化的文本情感转化为可视化的趋势图谱。

第六讲 知识图谱理念

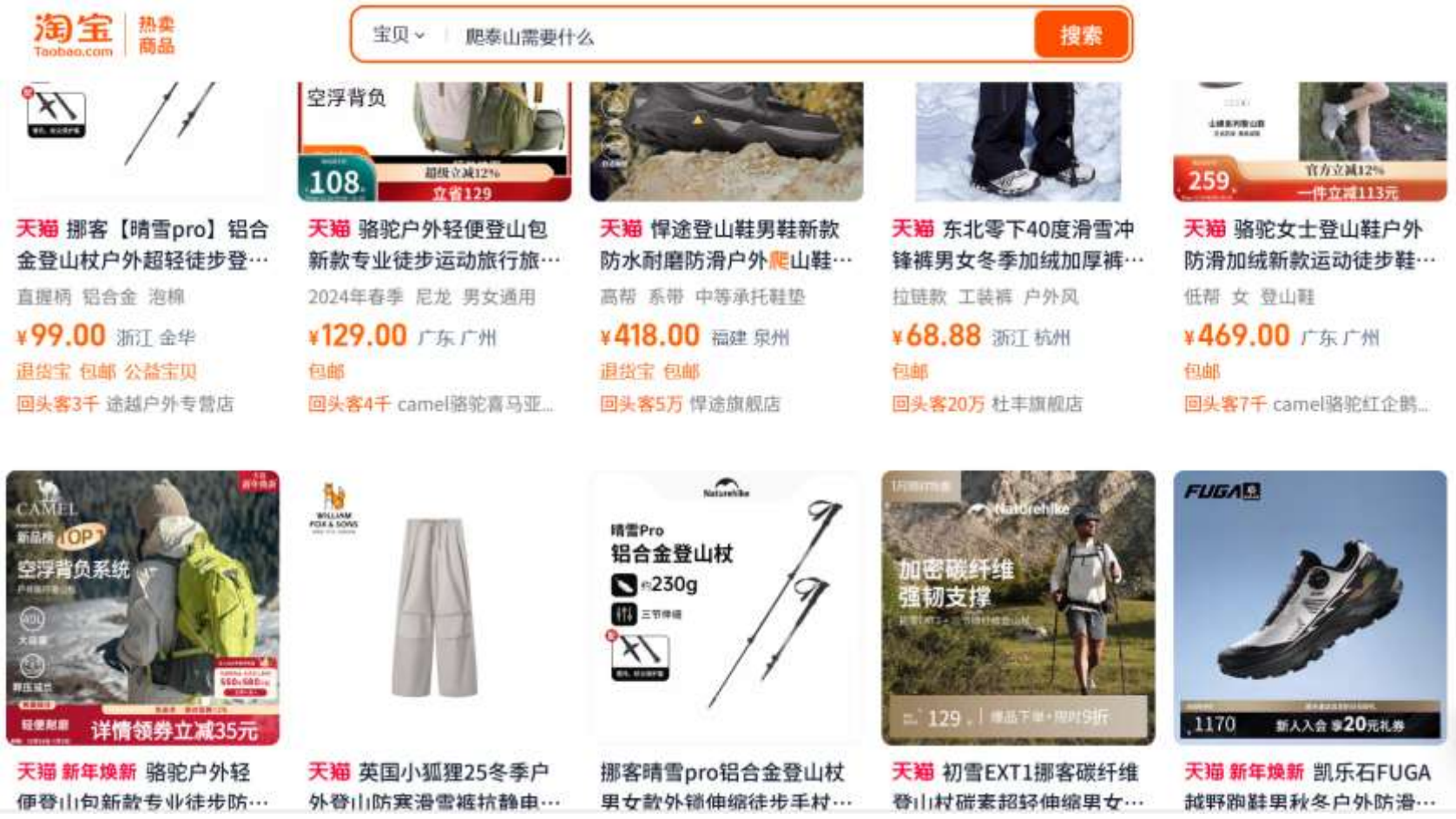
1.实际产业案例分析：对“阿里商品大脑”知识图谱生态构建，进行简要介绍与分析（爬泰山需要什么）

1. 思考：需求的本质升级

用户查询“爬泰山需要什么”，本质是请求一个**复杂活动的场景解决方案**，而非简单商品列表。这要求平台能理解地理环境、运动负荷、季节天气等复合知识，标志着需求从“找商品”升级为“规划一件事”。

第六讲 知识图谱理念

1.实际产业案例分析：对“阿里商品大脑”知识图谱生态构建，进行简要介绍与分析（爬泰山需要什么）



2. 推理：从场景到精准匹配

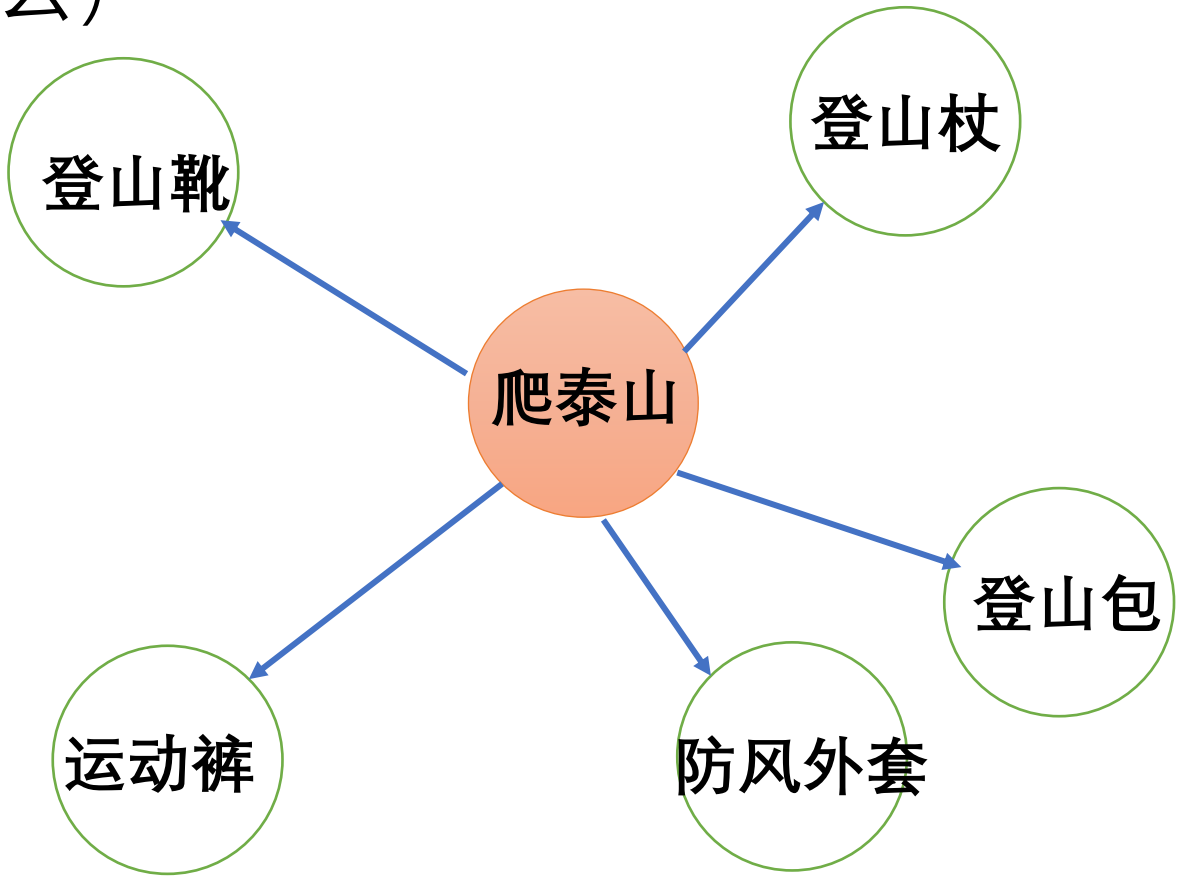
“阿里商品大脑”通过融合**地理、运动、商品属性**的多维知识图谱进行智能推理。它将“泰山”的陡峭石阶与“登山鞋”的防滑系数关联，将长耗时与“能量补给”关联，从而从海量商品中精准筛选并排序出个性化装备方案。

第六讲 知识图谱理念

1.实际产业案例分析：对“阿里商品大脑”知识图谱生态构建，进行简要介绍与分析（爬泰山需要什么）

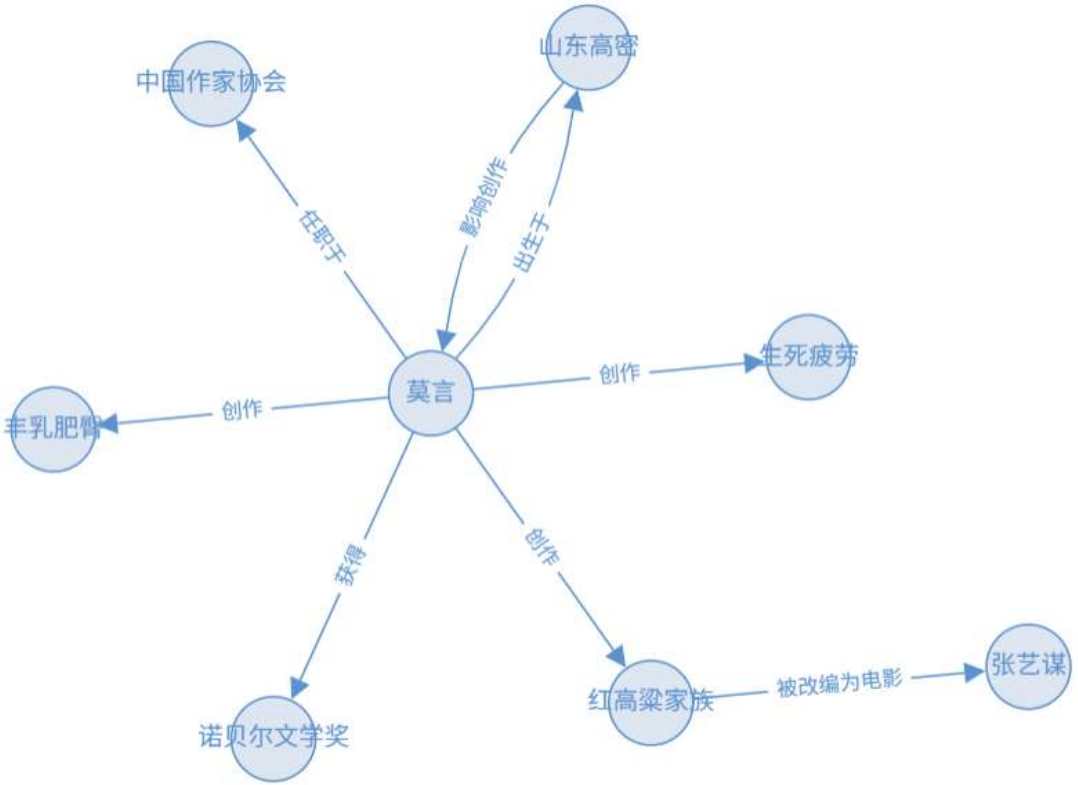
3. 构建：动态的知识生态

这构建了一个**自我优化的生态系统**：精准推荐直接提升转化与体验；需求预测指导供应链在景区周边智能备货；而用户的每一次真实消费与反馈，又作为新数据回流，持续训练图谱，使其更懂“爬山”乃至各类真实生活场景，最终成为驱动产业效率与体验升级的核心引擎。



第六讲（2） 知识图谱工具

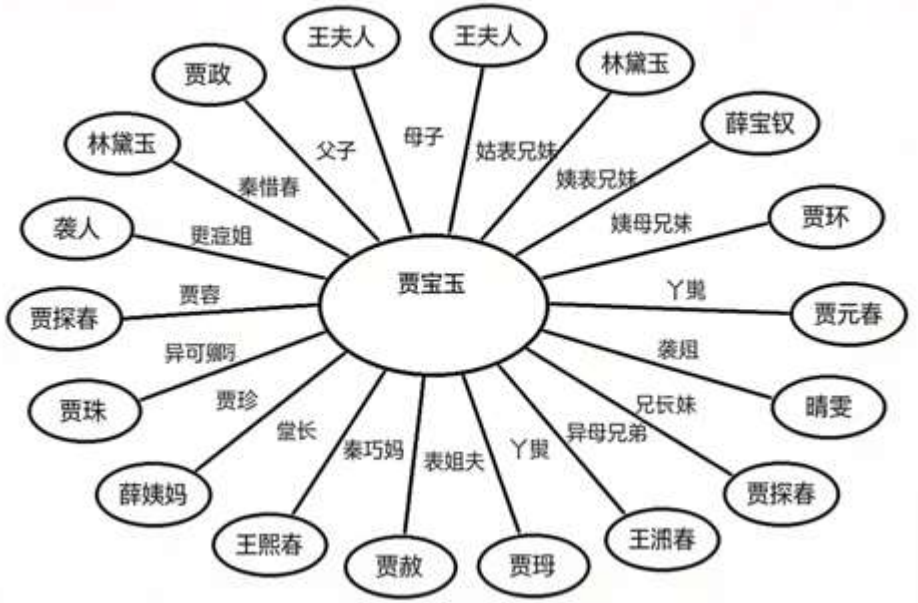
1.使用PPT中知识图谱链接平台，检索、截图（大词林等，可用的）；



第六讲（2） 知识图谱工具

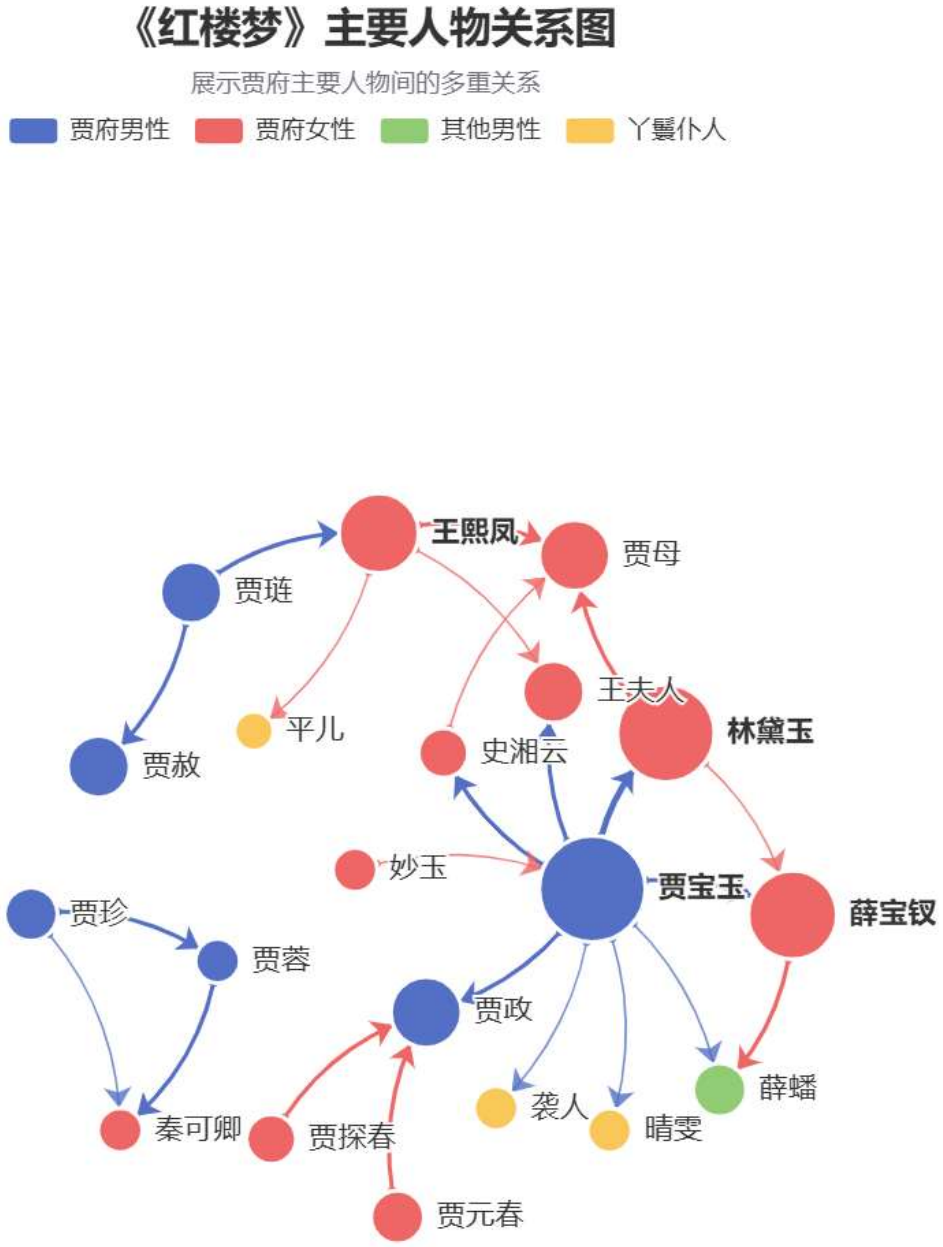
2.使用白板建模绘制一个你感兴趣的“知识图谱”，可以是人物关系，也可以是事物关系，或者概念之间的关系等等，并解释你绘制的图谱；

这是《红楼梦》核心人物图谱：核心是贾宝玉，他是贾政与王夫人之子，姑表妹是林黛玉、姨表姐是薛宝钗，贴身丫鬟有袭人与晴雯，异母弟妹是贾探春、贾环。贾政的妻子是王夫人，妾室赵姨娘育有探春、贾环，长女贾元春是皇妃。王夫人的姐妹是薛姨妈（宝钗之母），侄女是王熙凤（贾琏之妻）。贾母是贾政之母，女儿贾敏是黛玉母亲，侄孙女是史湘云。宁国府的贾珍是族长，子贾蓉、媳秦可卿，妹是贾惜春。



第六讲（2） 知识图谱工具

3.使用echarts中的关系图，绘制作业2）中的“知识图谱”。



第六讲（2） 知识图谱工具

4.使用Neo4j（可在线版本），编程绘制一款（简单）知识图谱（内容不限）。