



南京工业大学  
NANJING TECH  
UNIVERSITY

# 用户数据采集与关联分析

## (结课作业)

任俊杰



# 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’”**）；
3. 完成课后作业（**001-004，4份代码的运行**）。
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（1页PPT即可）（**仅信管**）。
5. 谈一谈在营销学科/领域，文本、文本分词以及实体的内涵。例如：客户关系管理中，文本分析的价值。（**仅营销**）

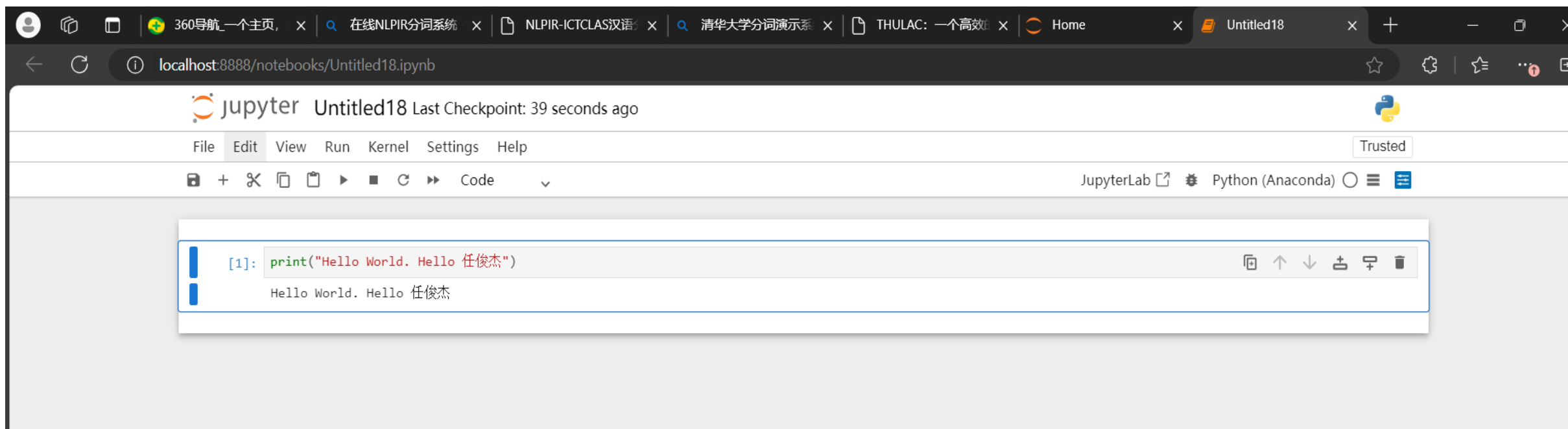
# 第一讲 课程导言与分词

## 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）；



# 第一讲 课程导言与分词

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



The screenshot shows a web browser window with multiple tabs. The active tab is titled 'Untitled18' and shows a JupyterLab interface. The address bar indicates the URL is 'localhost:8888/notebooks/Untitled18.ipynb'. The JupyterLab header includes the 'jupyter' logo, the notebook name 'Untitled18', and the last checkpoint time 'Last Checkpoint: 39 seconds ago'. Below the header is a menu bar with 'File', 'Edit', 'View', 'Run', 'Kernel', 'Settings', and 'Help'. A toolbar contains various icons for file operations and execution. The main area displays a code cell with the following content:

```
[1]: print("Hello World. Hello 任俊杰")
```

Below the code cell, the output is displayed:

```
Hello World. Hello 任俊杰
```

# 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；



The screenshot shows a JupyterLab interface with a notebook titled 'Untitled18'. The notebook contains the following code cells:

```
[19]: seg_list = jieba.cut("使用了停用词表之后啊，效果就好看很多了，什么啊、了、是之类的词就不见了")

[20]: final = ''

[21]: for seg in seg_list:
        if seg not in stopwords:
            final += seg+' '

[22]: print (final)
使用*词表*之后*效果*就*好看*很多*什么*之类*词*就*不见*

[23]: from snownlp import SnowNLP

[24]: s = SnowNLP(u'质量不大好')

[25]: print("，".join(s.words))
质量,不大,好

[26]: ss = jieba.cut('质量不大好')

[27]: print("，".join(ss))
质量,不大好

[28]: s1 = SnowNLP(u"吴志祥是南京工业大学青年教师，他对那种二次元小魔仙是无感的，这怎么行？")

[29]: print("，".join(s1.words)) # 因为snownlp擅长处理英文
吴,志,祥,是,南,京,工,业,大,学,青,年,教,师,, ,他,对,那,种,二,次,元,小,魔,仙,是,无,感,的,, ,这,怎,么,行,？
```

# 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；



```
jupyter Untitled19 Last Checkpoint: 3 minutes ago
File Edit View Run Kernel Settings Help
JupyterLab Python (Anaconda)

Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\ruguo\AppData\Local\Temp\jieba.cache
Loading model cost 0.707 seconds.
Prefix dict has been built successfully.
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@。

[5]: jieba.load_userdict('dict.txt')

[6]: 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的重要事件。")

[7]: 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/。

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

[9]: 到失去的时候才追悔莫及，人世间最痛苦的事情莫过于此。如果上天能够给我一个重新来过的机会，我会对那个女孩子说三个字：‘我爱你’。如果非要给这份爱加上一个期限，我希望是，一万年”

[10]: final = ''

[11]: #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 + '/' #叠加，累加

[12]: 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/非要/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;



```
[6]: print(' '.join(seg_list_huang))
黄旭华/, /1926/年/3/月/12/日出/生于/广东省/汕头市/, /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/.

[7]: stopwords = open('stop_words.txt', 'r', encoding='utf-8').read()
stopwords = stopwords.split('\n')

[8]: stopwords

[8]: ['的', '了', '是', '啊', '、', ' ', ' ', ' ', ' ', ' ', ' ', '停用']

[9]: 汕头市, 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。)

[10]: final = ''

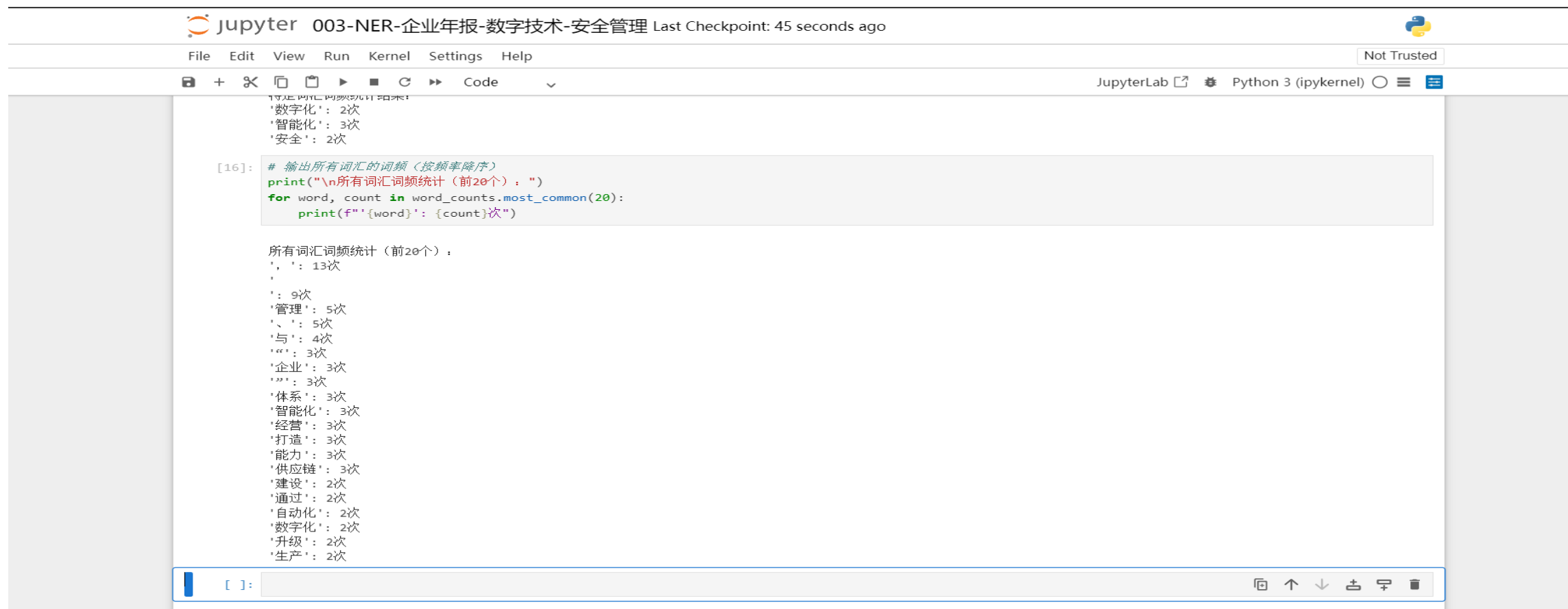
[11]: for seg in seg_list_huang:
    if seg not in stopwords:
        final+= seg+' '

[12]: print(final)
黄旭华/1926/年/3/月/12/日出/生于/广东省/汕头市/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;



The image shows a JupyterLab interface with a code cell and its output. The code cell contains a comment and a loop that prints the top 20 most common words from a word count dictionary. The output shows the results of this loop, listing words and their frequencies.

```
jupyter 003-NER-企业年报-数字技术-安全管理 Last Checkpoint: 45 seconds ago
```

File Edit View Run Kernel Settings Help

Not Trusted

JupyterLab Python 3 (ipykernel)

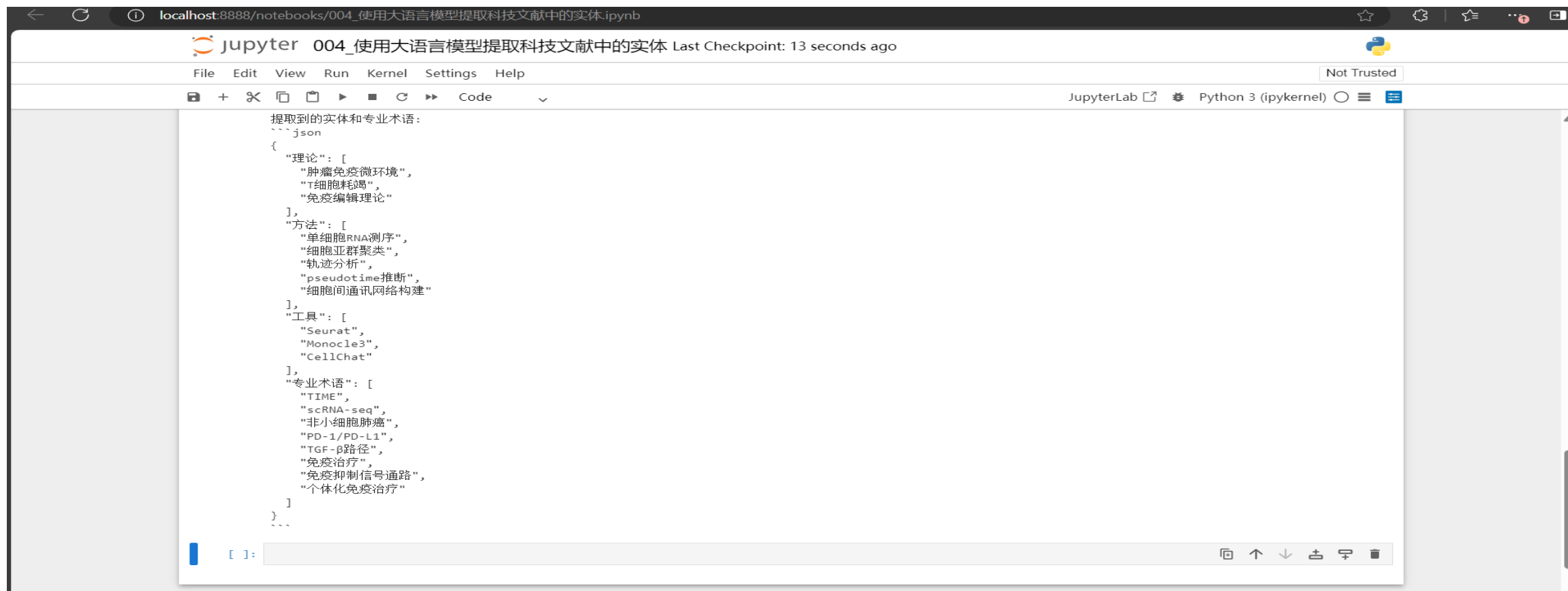
```
[16]: # 输出所有词汇的词频 (按频率降序)
print("\n所有词汇词频统计 (前20个): ")
for word, count in word_counts.most_common(20):
    print(f"{word}: {count}次")
```

```
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;

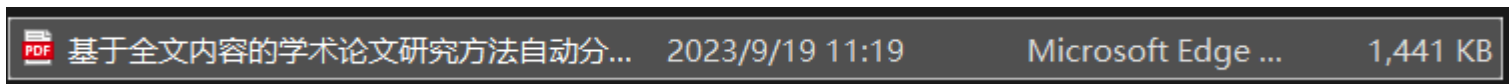


The screenshot shows a JupyterLab notebook titled "004\_使用大语言模型提取科技文献中的实体". The notebook is running on a local host (localhost:8888). The interface includes a menu bar (File, Edit, View, Run, Kernel, Settings, Help) and a toolbar with various icons. The main area displays a code cell with the following JSON output:

```
提取到的实体和专业术语:
```json
{
  "理论": [
    "肿瘤免疫微环境",
    "T细胞耗竭",
    "免疫编辑理论"
  ],
  "方法": [
    "单细胞RNA测序",
    "细胞亚群聚类",
    "轨迹分析",
    "pseudotime推断",
    "细胞间通讯网络构建"
  ],
  "工具": [
    "Seurat",
    "Monocle3",
    "CellChat"
  ],
  "专业术语": [
    "TIME",
    "scRNA-seq",
    "非小细胞肺癌",
    "PD-1/PD-L1",
    "TGF-β路径",
    "免疫治疗",
    "免疫抑制信号通路",
    "个体化免疫治疗"
  ]
}
```

# 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（1页PPT即可）（仅信管）。



阅读总结：

结论：

全文内容显著优于摘要：所有模型在全文数据上的表现均远超仅使用摘要的基线。最好的模型（**CC-NB**）在全文上的**F1**值达到**0.705**，而摘要仅为**0.656**。这有力证明了全文包含的丰富上下文信息对于精准识别研究方法至关重要。

模型性能差异：在所有模型中，朴素贝叶斯在分类器链（**CC-NB**）策略下表现最佳。这表明在考虑标签间潜在关联性的同时，**NB**算法在本任务中具有良好的适应性。相比之下，**ML-KNN**模型表现一般，暗示研究方法间的强依赖性可能不显著。

类别不平衡与特征表征能力的影响：研究发现，不同研究方法的分类效果差异巨大。样本量大且特征鲜明的“实验法”（**F1=0.836**）和“计量法”（**F1=0.803**）分类效果极佳；而“内容分析法”（**F1=0.460**）和“其他方法”（**F1=0.443**）则表现较差。这揭示了两个核心挑战：一是训练数据规模不足会导致泛化能力弱；二是不同研究方法本身的语言特征表征能力不同，有些方法（如内容分析）缺乏独特、稳定的关键词汇。

## 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”或“网络营销”或其他你感兴趣的主题变化趋势。
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；
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”；
4. 阅读论文“2018-Wang 等 - Long live the scientists Tracking the scientific”，并做阅读总结（1页PPT即可）。

## 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”或“网络营销”或其他你感兴趣的主题变化趋势。

### 一、从“传统文献管理”向“数字与数据驱动”转型（2014-2017）

此阶段研究仍以图书馆学、情报学和档案学三大支柱为基础，但已明显转向数字化议题。关键词如“数字图书馆”“元数据”“信息组织”“知识服务”高频出现。研究重点在于如何将纸质文献资源转化为结构化数字资产，并构建统一的资源描述与检索体系。同时，“大数据”概念开始渗透，部分学者尝试探讨信息资源在政府开放数据、智慧城市等场景中的整合路径。

### 二、学科融合加速，“健康信息学”“数据科学”兴起（2018-2020）

随着《“健康中国2030”规划纲要》等政策推动，健康信息学成为信息资源管理的重要分支。CNKI数据显示，关于“医院信息资源”“医疗知识库”“CHDK医药总库”等主题的论文显著增长。与此同时，数据管理与数据治理成为新热点，“数据资产”“数据生命周期”“数据质量”等术语频繁出现。学科边界进一步模糊，信息资源管理开始与公共卫生、生物医学、人工智能等领域深度交叉。

## 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”或“网络营销”或其他你感兴趣的主题变化趋势。

### 三、技术赋能深化，“AI+知识管理”成为核心范式（2021-2023）

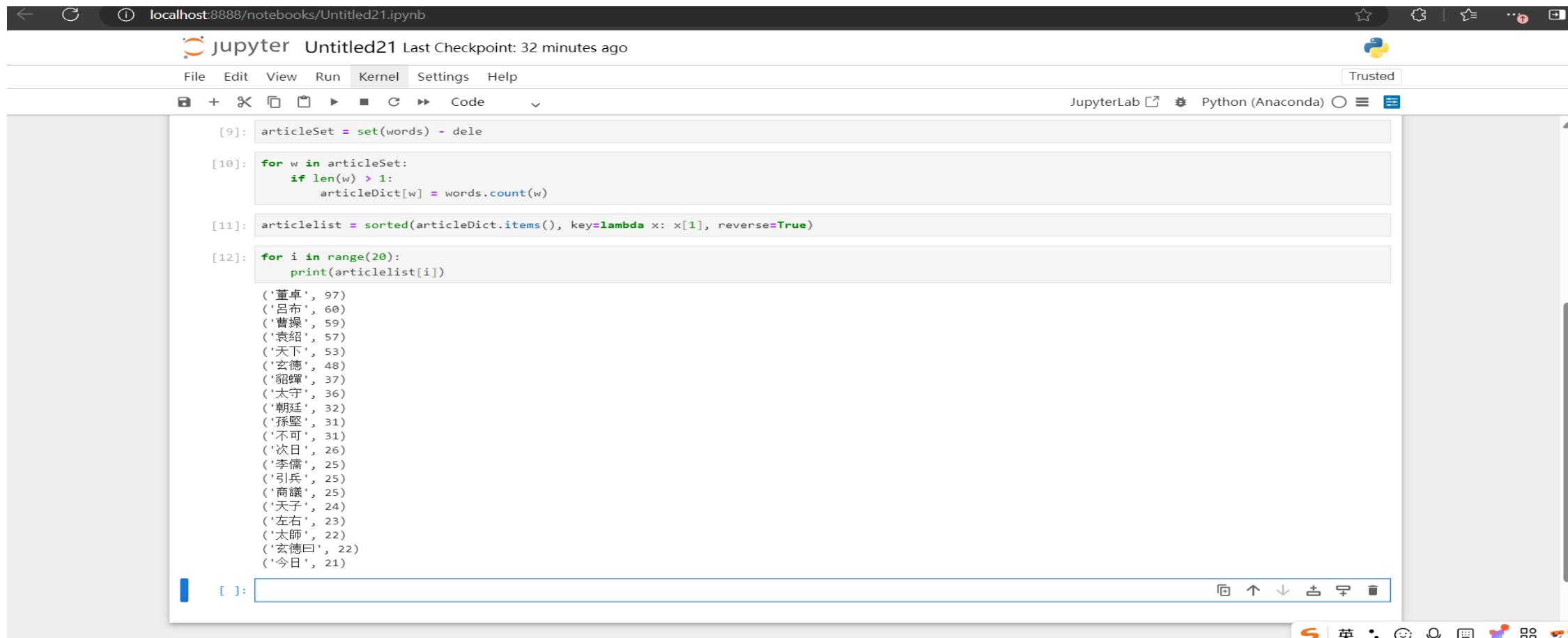
人工智能、自然语言处理和知识图谱技术被广泛应用于信息资源的智能组织与服务。研究焦点从“资源存储”转向“知识发现”与“智能推荐”。例如，利用BERT等模型对学术论文进行研究方法自动分类（如章成志等，2020），或构建中医药方剂知识网络（依托《中国医药卫生知识资源总库》）。此外，“信息素养教育”内涵扩展，新增“数据素养”“健康信息素养”“元素养”等子维度，反映社会对公民信息能力的新要求。

### 四、战略价值凸显，迈向“国家信息资源体系”建设（2024至今）

近年研究更强调信息资源的国家战略属性与制度化治理。关键词如“国家数据局”“数据要素市场”“信息资源资产化”“公共数据授权运营”集中涌现。2023年国家数据局成立后，学界迅速响应，探讨如何构建覆盖政府、企业、科研机构的全域信息资源协同管理体系。同时，信息资源管理被纳入新文科、交叉学科建设框架，其作为一级学科的地位得到强化，研究视野从技术操作层面上升至制度设计与生态构建层面。

# 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\*全文词频统计



The screenshot shows a JupyterLab interface with a notebook titled 'Untitled21'. The notebook contains four code cells. The first cell defines 'articleSet' as a set of words minus 'dele'. The second cell is a loop that iterates over 'articleSet' and updates 'articleDict' with word counts. The third cell sorts 'articleDict' by value in descending order. The fourth cell prints the first 20 items of the sorted list. The output shows a list of words and their counts, such as ('董卓', 97), ('吕布', 60), ('曹操', 59), etc.

```
[9]: articleSet = set(words) - dele

[10]: for w in articleSet:
      if len(w) > 1:
          articleDict[w] = words.count(w)

[11]: articlelist = sorted(articleDict.items(), key=lambda x: x[1], reverse=True)

[12]: for i in range(20):
      print(articlelist[i])

('董卓', 97)
('吕布', 60)
('曹操', 59)
('袁绍', 57)
('天下', 53)
('玄德', 48)
('貂蝉', 37)
('太守', 36)
('朝廷', 32)
('孙坚', 31)
('不可', 31)
('次日', 26)
('李儒', 25)
('引兵', 25)
('商議', 25)
('天子', 24)
('左右', 23)
('太師', 22)
('玄德曰', 22)
('今日', 21)
```

# 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\*指定类型词频统计



The image shows a JupyterLab interface with a code editor and a console. The code editor contains a Python script that reads a file named 'name.txt' and prints the first 50 characters. The console shows the output of the script, which is a list of names separated by vertical bars. The names are: 諸葛亮, 關羽, 劉備, 曹操, 孫權, 關羽, 張飛, 呂布, 周瑜, 趙雲, 龐統, 司馬懿, 黃忠, 馬超.

```
[2]: f_name = open('name.txt', encoding='utf-8')

[4]: with open('name.txt', encoding='gbk') as f_name:
      data_name = f_name.read()
      print(data_name[:50])

      諸葛亮|關羽|劉備|曹操|孫權|關羽|張飛|呂布|周瑜|趙雲|龐統|司馬懿|黃忠|馬超

[5]: f_name.close()

[6]: names = data_name.split('|')

[7]: print(names)

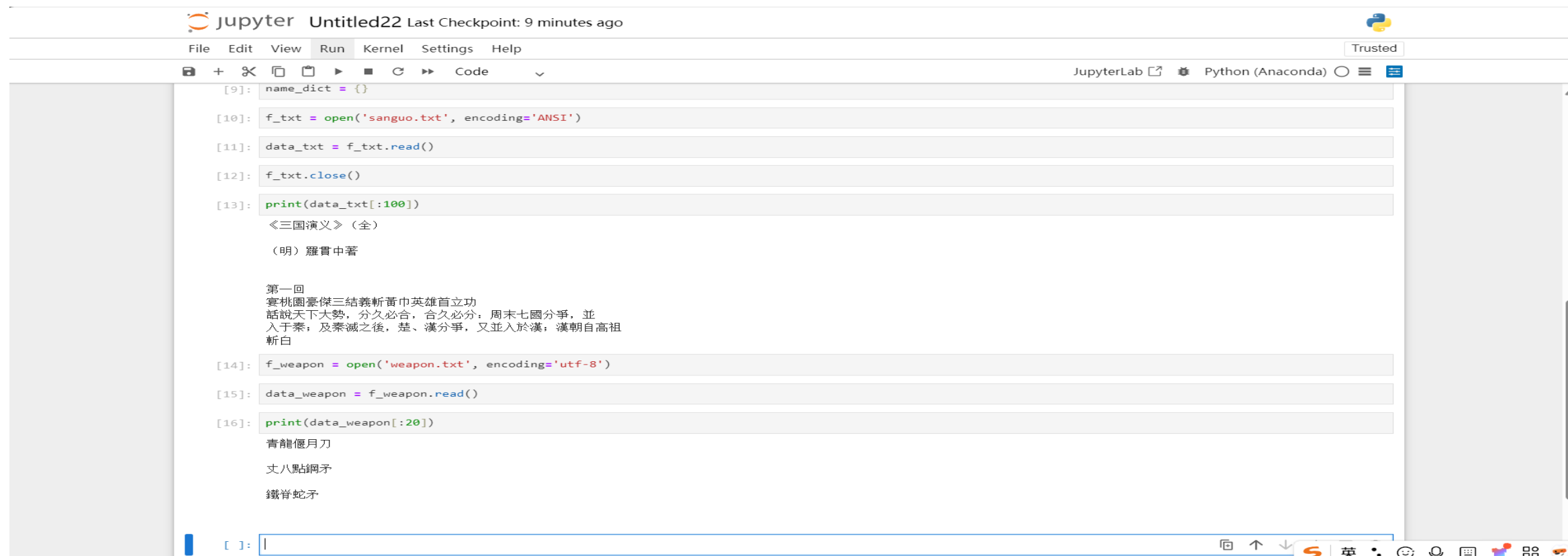
      ['諸葛亮', '關羽', '劉備', '曹操', '孫權', '關羽', '張飛', '呂布', '周瑜', '趙雲', '龐統', '司馬懿', '黃忠', '馬超']

[8]: names

      ['諸葛亮',
       '關羽',
       '劉備',
       '曹操',
       '孫權',
       '關羽',
       '張飛',
       '呂布',
       '周瑜',
       '趙雲',
       '龐統',
       '司馬懿',
       '黃忠',
       '馬超']
```

# 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\*指定类型词频统计



```
[9]: name_dict = {}

[10]: f_txt = open('sanguo.txt', encoding='ANSI')

[11]: data_txt = f_txt.read()

[12]: f_txt.close()

[13]: print(data_txt[:100])
《三国演义》（全）
（明）羅貫中著

第一回
宴桃園豪傑三結義斬黃巾英雄首立功
話說天下大勢，分久必合，合久必分；周末七國分爭，並
入于秦；及秦滅之後，楚、漢分爭，又並入於漢；漢朝自高祖
斬白

[14]: f_weapon = open('weapon.txt', encoding='utf-8')

[15]: data_weapon = f_weapon.read()

[16]: print(data_weapon[:20])
青龍偃月刀
丈八點鋼矛
鐵脊蛇矛

[ ]: |
```



## 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”；\*（1）

Jupyter Untitled23 Last Checkpoint: 5 minutes ago

File Edit View Run Kernel Settings Help Trusted

+ - ✂ 📄 🗑 ▶ ■ ↺ ▶ Code ▾

JupyterLab Python (Anaconda)

```
[1]: import jieba

[2]: article = open('科学家博物馆-黄旭华传记序言.txt','r',encoding='utf-8').read() # 打开并读取三国前10回 #出现乱码提示，就把ANSI改成utf-8

[4]: dele = {'.','!',' ','?','的','(',')','>','<',',',' '} # 手动设计一些停用词和符号

[5]: words = list(jieba.cut(article))

Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\ruguo\AppData\Local\Temp\jieba.cache
Loading model cost 0.674 seconds.
Prefix dict has been built successfully.

[6]: articleDict = {}

[7]: articleSet = set(words) - dele

[8]: for w in articleSet:
    if len(w) > 1:
        articleDict[w] = words.count(w)

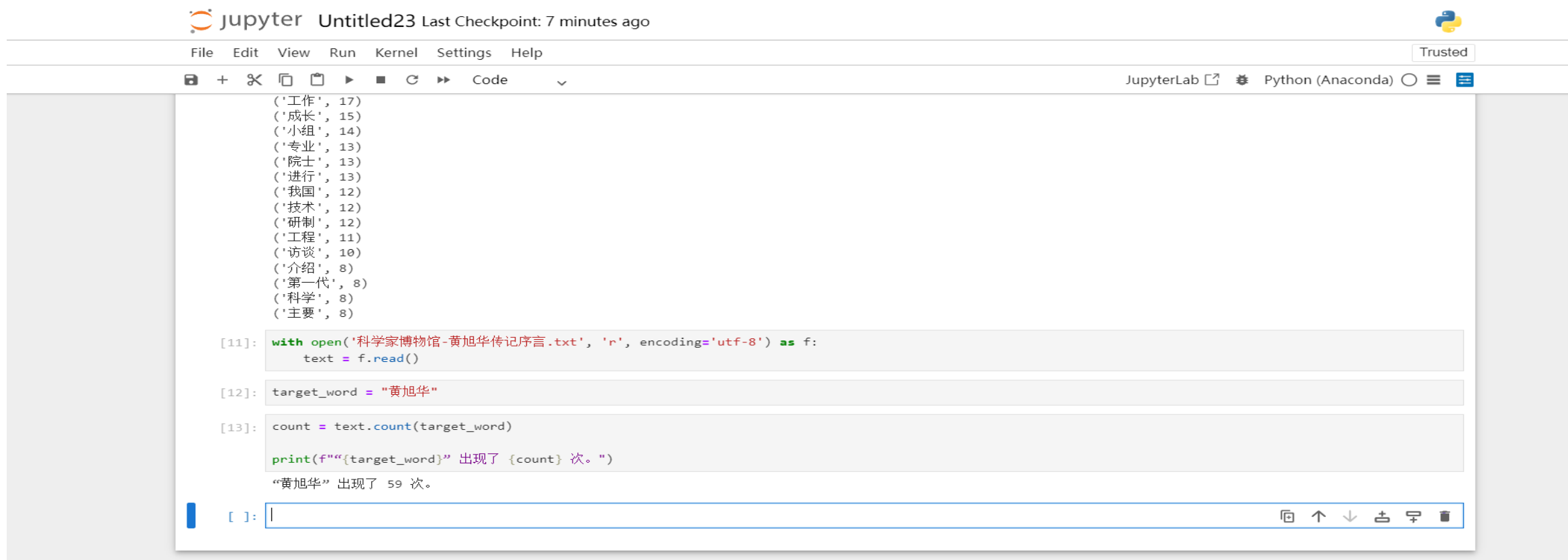
[9]: articlelist = sorted(articleDict.items(), key=lambda x: x[1], reverse=True)

[10]: for i in range(20):
    print(articlelist[i])

('黄旭华', 53)
('核潜艇', 32)
('采集', 29)
('学术', 22)
('资料', 21)
('工作', 12)
```

## 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”；\*（2）



The image shows a JupyterLab interface with a file named 'Untitled23'. The top bar indicates the last checkpoint was 7 minutes ago. The interface includes a menu bar (File, Edit, View, Run, Kernel, Settings, Help) and a toolbar with icons for file operations and code execution. The main area displays a list of words and their frequencies, followed by three code cells. The first cell opens the file '科学家博物馆-黄旭华传记序言.txt' in UTF-8 encoding. The second cell sets the target word to '黄旭华'. The third cell counts the occurrences of the target word and prints the result: '黄旭华' 出现了 59 次.

```
['工作', 17]
['成长', 15]
['小组', 14]
['专业', 13]
['院士', 13]
['进行', 13]
['我国', 12]
['技术', 12]
['研制', 12]
['工程', 11]
['访谈', 10]
['介绍', 8]
['第一代', 8]
['科学', 8]
['主要', 8]

[11]: with open('科学家博物馆-黄旭华传记序言.txt', 'r', encoding='utf-8') as f:
      text = f.read()

[12]: target_word = "黄旭华"

[13]: count = text.count(target_word)

      print(f"{target_word}" 出现了 {count} 次。")
      "黄旭华" 出现了 59 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（1页PPT即可）

阅读总结;

核心研究方法与数据:

研究创新性地结合了两种数据源：一是覆盖3600万册图书的谷歌图书，用于衡量科学家在公共及文化层面的知名度（通过其姓名在书籍中的出现频率）；二是索引9100万篇学术文献的谷歌学术，用于衡量其在学术界的影响力（通过被引次数）。此外，研究还利用谷歌Ngram Viewer按不同语言（如英式英语、美式英语、德语等）进行细分分析，并通过共现分析探究科学家与其标志性成就的关联度。

主要研究发现:

1. “伟人虽逝，声名永存”：研究证实，历史上最伟大的科学家并未被后世遗忘。他们的影响力跨越数百年，持续存在于人类的文化记忆中。例如，牛顿在17-19世纪享有极高声望，而爱因斯坦自20世纪中叶起在全球范围内的提及度已超越牛顿。
2. “同群偏好”效应显著：研究发现了强有力的“同群偏好”（own-group preference）证据。科学家在其本国或使用相同语言的群体中享有更高声誉。最典型的例子是，在英式英语语料库中，牛顿的提及度始终高于爱因斯坦，这与2005年英国皇家学会的民意调查结果一致；而在美式英语、德语等语料库中，爱因斯坦则更受推崇。
3. 声望源于标志性成就：共现分析揭示了科学家声望的具体来源。牛顿的声望主要与其万有引力定律和运动定律相关；而爱因斯坦的声望则高度集中于相对论（尤其是广义相对论）和量子理论。

# 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。
2. 使用Echarts，制作3个以上图，其中一个必须是“关系”，图的概念越明确（可解释，而不是自带的模板）越好。
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。
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

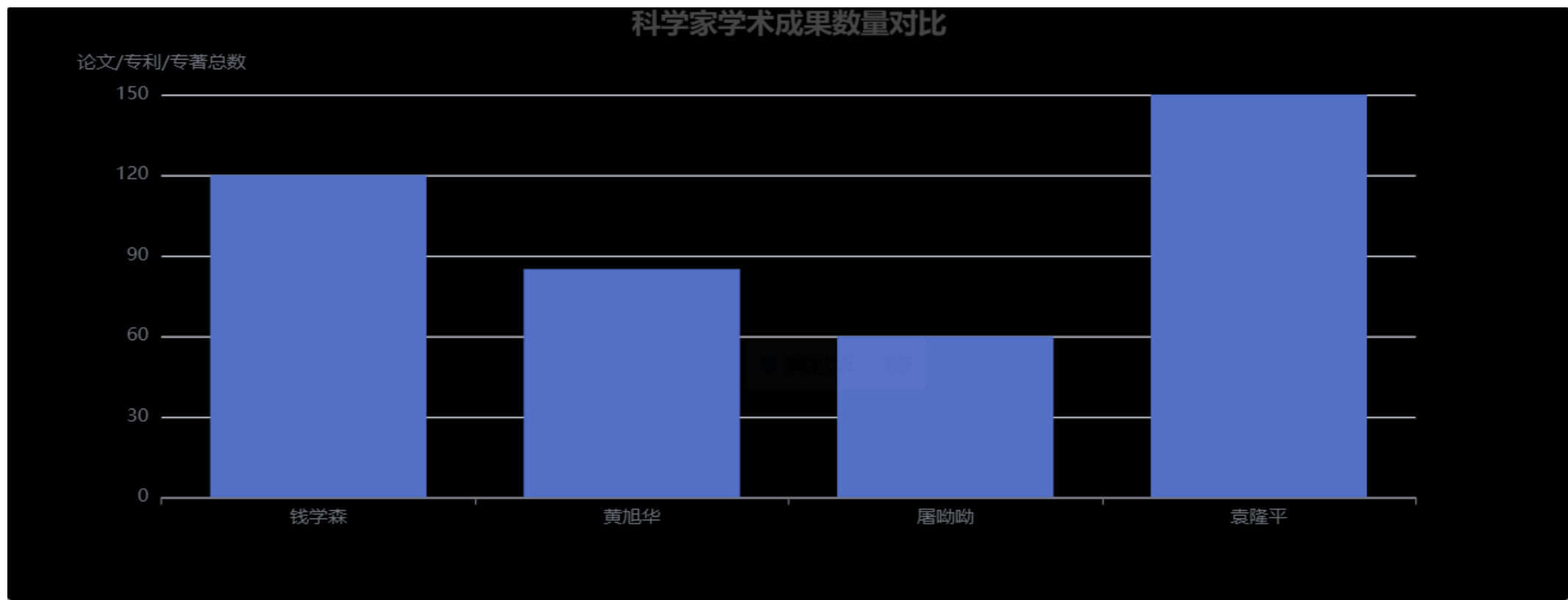
1. 用任意一款词云工具，制作一个好看的词云（内容合理即可），并对词云图有一段话的解释。



一段关于人格分裂的解释的文字生成的词云图中人格占最大最中间，说明这段话中“人格”这个词的占比最大。其他的词例如“患者”“差异”等都是文字中出现词频较少的词

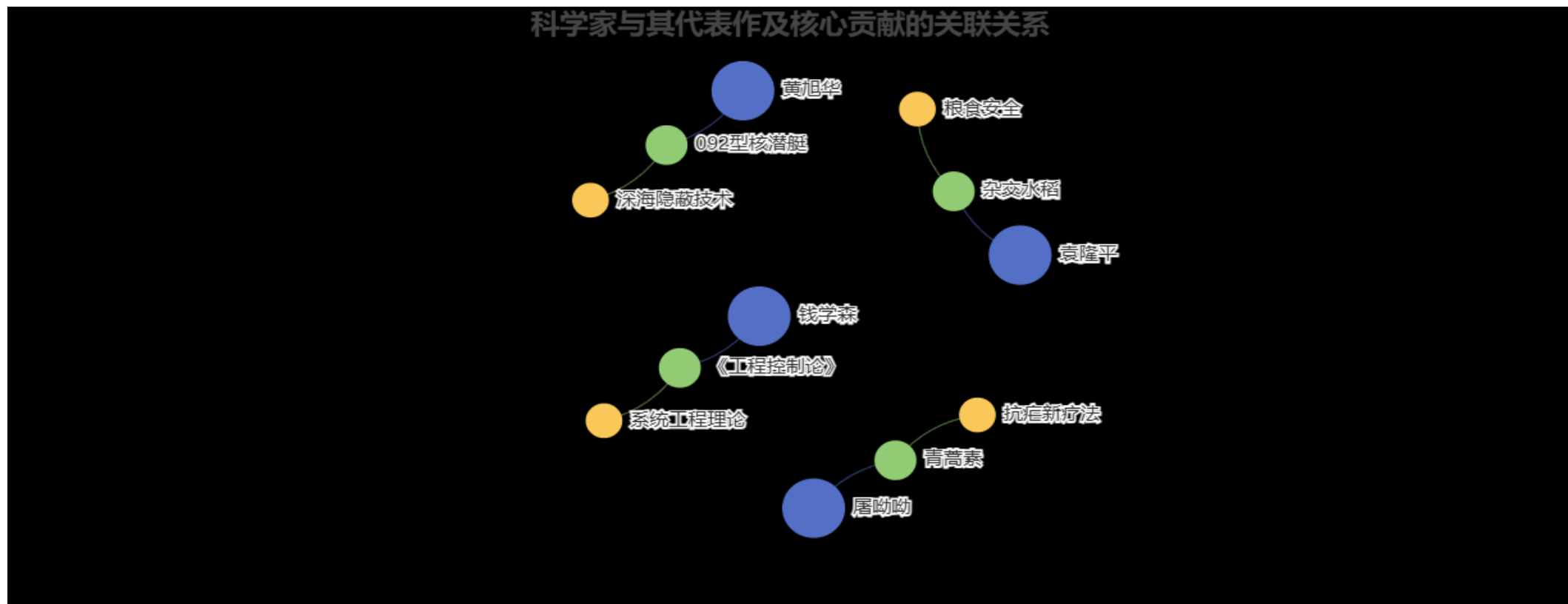
# 第三讲 词云与可视化

2. 使用Echarts，制作3个以上图，其中一个必须是“关系”，图的概念越明确（可解释，而不是自带的模板）越好。



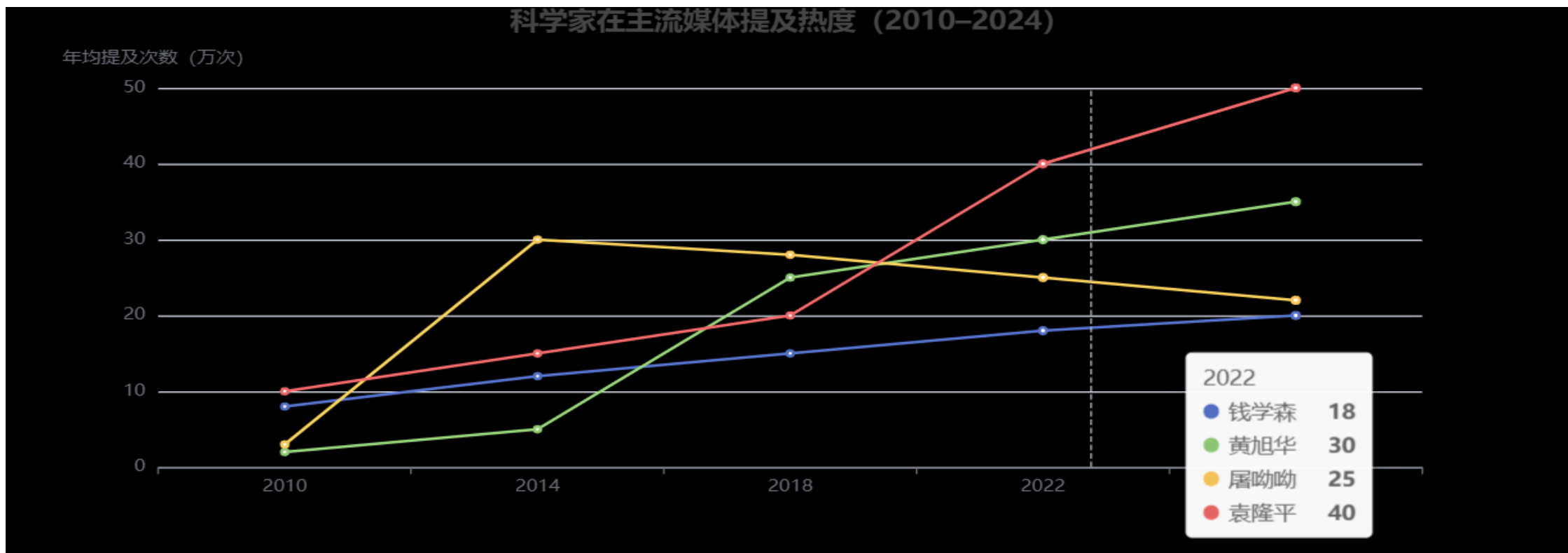
# 第三讲 词云与可视化

2. 使用Echarts，制作3个以上图，其中一个必须是“关系”，图的概念越明确（可解释，而不是自带的模板）越好。



# 第三讲 词云与可视化

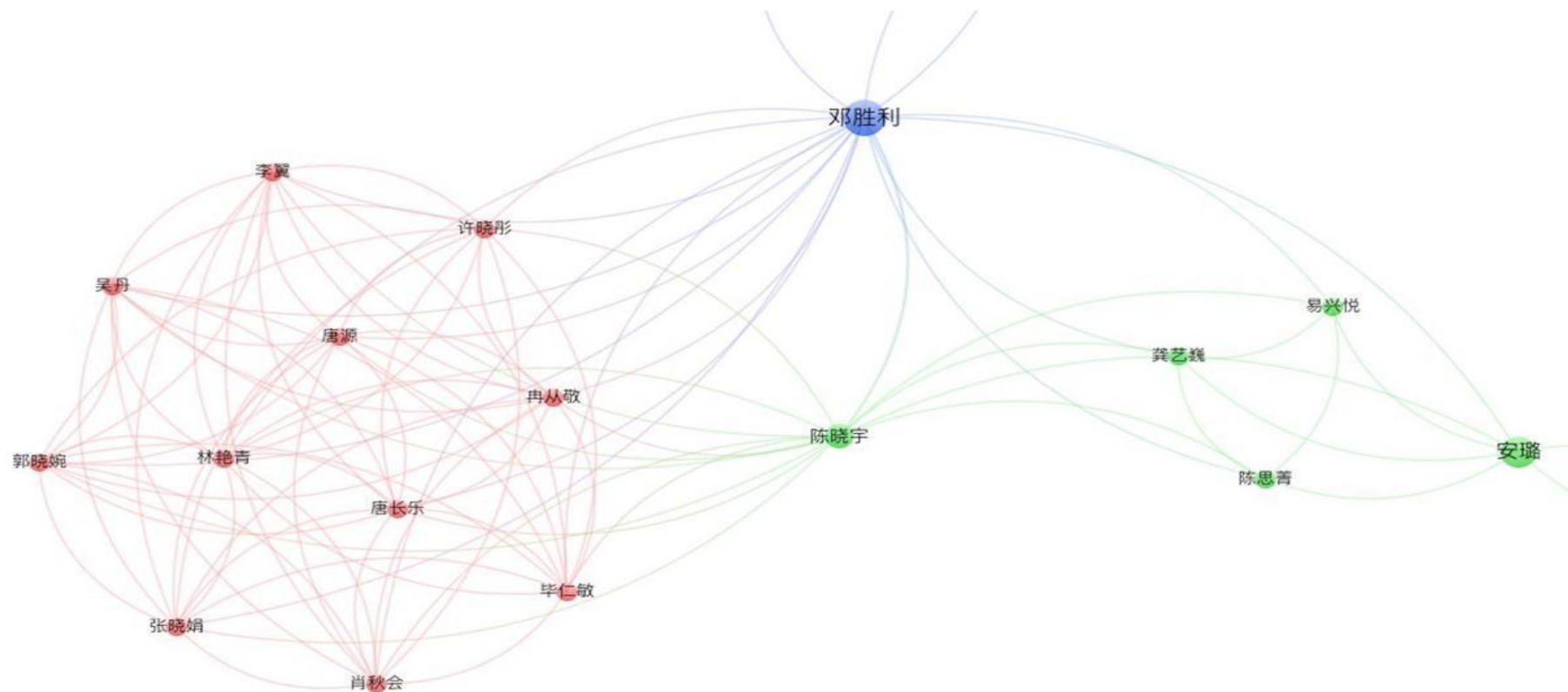
2. 使用Echarts，制作3个以上图，其中一个必须是“关系”，图的概念越明确（可解释，而不是自带的模板）越好。





# 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；
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. 谈一谈情感分析在营销学科/领域的应用以及价值；并且分析大语言模型（LLM）在该领域可能带来的新应用与新改变（**仅营销**）。

# 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



The screenshot shows a JupyterLab notebook titled "sentiment\_analysis\_1\_chuji" with the following code and output:

```
[24]: taobao_1.sentiments
[24]: 0.999947261146611

[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！"
[26]: taobao_2 = SnowNLP(text_taobao_2)
[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
[28]: taobao_2.sentiments
[28]: 0.889005139666256

[29]: # 以上的结果看上去是有问题的，分析的不准确。
[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：拍照不清晰！ 电池续航：手机不
[31]: taobao_3 = SnowNLP(text_taobao_3)
[32]: taobao_3.sentiments
[32]: 5.7076094222563434e-05
[33]: # 这一长句的结果还是可以的，非常小的概率值了
[34]: # 你的例句呢？
```

# 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

Jupyter sentiment\_analysis\_2\_timeline Last Checkpoint: 3 minutes ago

File Edit View Run Kernel Settings Help

Not Trusted

JupyterLab Python 3 (ipykernel)

```
[16]: plt.savefig('timeline.png') # 看不到? 改一改?
```

<Figure size 640x480 with 0 Axes>

在图中，我们发现许多正面评价情感分析数值极端的高。同时，我们也清晰地发现了那几个数值极低的点。对应评论的情感分析数值接近于0。这几条评论，被Python判定为基本上没有正面情感了。

从时间上看，最近一段时间，几乎每隔几天就会出现一次比较严重的负面评价。

作为经理，你可能如坐针毡。希望尽快了解发生了什么事儿。你不用在数据框或者Excel文件里面一条条翻找情感数值最低的评论。Python数据框Pandas为你提供了非常好的排序功能。假设你希望找到所有评论里情感分析数值最低的那条，可以这样执行：

```
[19]: df.sort_values(['sentiments'])[:1]
```

	comments	date	sentiments
24	这次是在情人节当天过去的，以前从来没有在情人节正日子出来过，不是因为没男朋友，而是感觉哪哪人...	2017-02-20 16:00:00	6.334066e-08

情感分析结果数值几乎就是0啊！不过这里数据框显示评论信息不完全。我们需要将评论整体打印出来。

```
[20]: print(df.sort_values(['sentiments']).iloc[0].comments)
```

这次是在情人节当天过去的，以前从来没有在情人节正日子出来过，不是因为没男朋友，而是感觉哪哪人都多，所以特意错开，这次实在是馋A餐厅了，所以赶在正日子也出来了，从下午四点多的时候我看排号就排到一百多了，我从家开车过去得堵的话一个小时，我一看提前两个小时就在网上先排着号了，差不多我们是六点半到的，到那的时候我看号码前面还有才三十多号，我想着肯定没问题了，等一会就能吃上的，没想到悲剧了，就从我们到那坐到等位区开始，大约是十分二十分一叫号，中途多次我都想走了，哈哈，哎，等到最后早上九点才吃上的，服务员感觉也没以前清闲时周到了，不过这肯定的，一人负责好几桌，今天节日这么多人，肯定是很累的，所以大多也都是我自己跑腿，没让服务员给弄太多，就虾滑让服务员下的，然后环境来说感觉卫生方面是不错，就是有些太吵了，味道还是一如既往的那个味道，不过A餐厅最人性化的就是看我们等了两个多小时，上来送了我们一张打折卡，而且当场就可以使用，这点感觉还是挺好的，不愧是A餐厅，就是比一般的要人性化，不过这次就是选错日子了，以后还是得提前预约，要不就别赶节日去，太火爆了！

# 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



The screenshot shows a JupyterLab environment with a file named 'sentiment\_analysis\_3\_大模型\_健康文本细粒度情感抽取'. The code in the editor includes error handling for network requests, JSON decoding, and general exceptions. Below the code, the output shows a JSON structure representing fine-grained sentiment analysis results for health text.

```
print(f"错误信息: {response.text}")

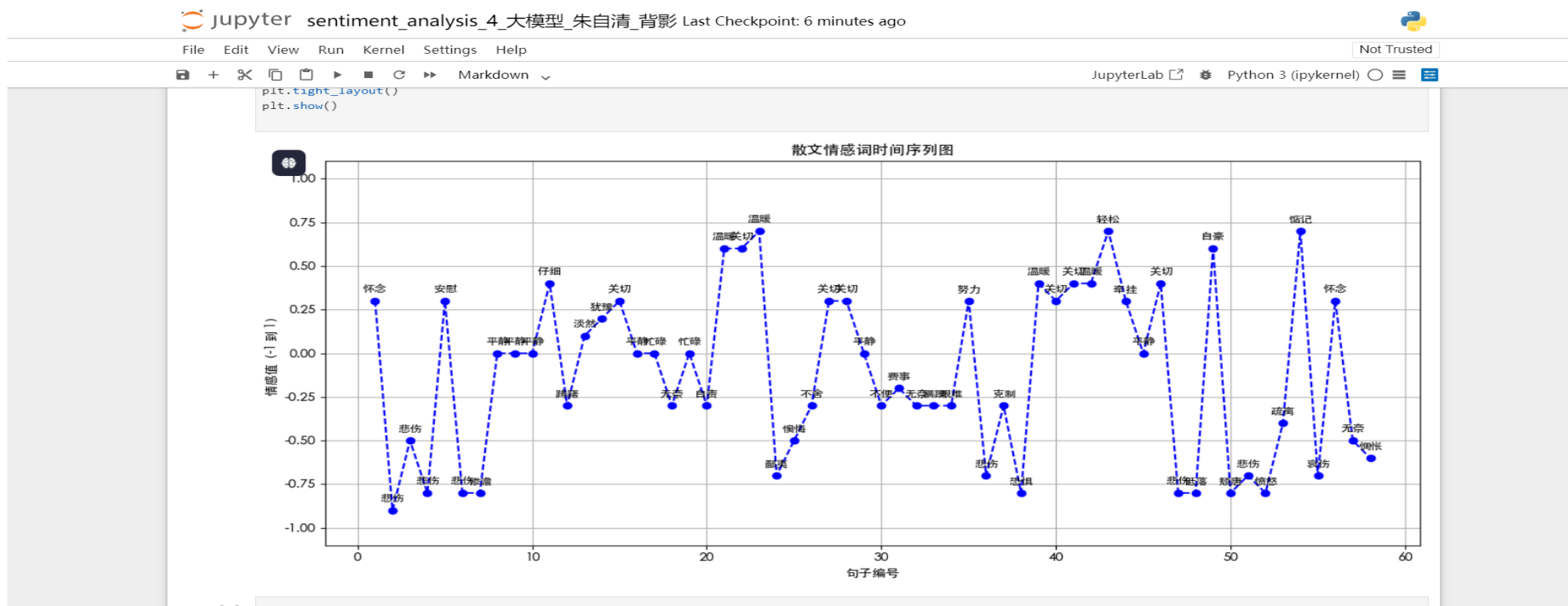
except requests.exceptions.RequestException as e:
    # 处理网络请求异常
    print(f"网络请求失败: {e}")
except json.JSONDecodeError as e:
    # 处理JSON解析异常
    print(f"JSON解析失败: {e}")
except Exception as e:
    # 处理其他异常
    print(f"发生未知错误: {e}")

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"
    }
  ]
}
```



# 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





# 第六讲 知识图谱理念

1. 实际产业案例分析：使用3-5页PPT对“阿里商品大脑”、“美团大脑”、“丁香医生知识图谱”、“领英知识图谱”...其中任意一家机构/公司最新的知识图谱生态构建，进行简要介绍与分析。要求：需要是最新进展（不能复制课程PPT中的内容）；可以是一个简单的案例；有自己的评价。自由发挥。（营销、信管，都可以结合自己的专业兴趣，自由选择分析对象）

## 美团大脑简介

### 美团大脑是什么？

美团大脑是美团点评基于其庞大的业务体系（包括外卖、酒店预订、电影票务等）构建的一个AI系统，旨在通过深度学习、自然语言处理(NLP)、知识图谱等多种技术手段提升用户体验和服务效率。

**核心功能：**(1)用户行为预测(2)智能推荐(3)商家管理优化

**最新进展：**美团大脑在2025年引入了增强现实(AR)和虚拟现实(VR)技术，用于改进用户界面体验，并增强了其知识图谱的深度和广度。

# 第六讲 知识图谱理念

详细介绍与分析：

美团大脑是美团构建的超大规模生活服务领域知识图谱系统，旨在支撑其“Food + Platform”战略。截至2025年，该图谱已覆盖超6亿实体（包括用户、商户、商品、地址、菜品等）和千亿级关系，核心创新在于深度融合多源异构数据——不仅整合交易日志、用户评论、菜单图片，还通过OCR、NLP和多模态模型从非结构化内容中抽取结构化知识（如“宫保鸡丁含花生”“某餐厅适合亲子聚餐”）。

其典型应用包括：智能推荐（基于“用户-场景-供给”三元组）、搜索意图理解（如识别“带娃吃饭”隐含的儿童座椅需求）、以及动态定价与调度优化。尤其在“到店+到家”融合场景中，知识图谱打通了线上行为与线下服务，实现跨业态关联推理。

值得肯定的是，美团大脑将知识图谱从“静态本体”升级为“动态认知引擎”，具备实时更新与因果推断能力。但挑战仍存：一是长尾商户信息稀疏导致冷启动问题；二是用户隐私与数据利用的边界需更透明。总体而言，美团大脑代表了知识图谱在本地生活服务领域的深度落地，其“场景驱动、闭环反馈”的构建范式具有行业示范意义。

# 第六讲 知识图谱理念

---

评价：

美团大脑这个知识图谱，说白了就是美团给自己建的一个“超级大脑”，用来搞懂用户到底想要啥、商家能提供啥、以及怎么把这两头高效地连起来。它不光记下你点了什么外卖、看了哪家酒店，还能从评论里扒出“这家店有宝宝椅”“那道菜特别辣”，甚至看菜单图片识别出菜品配料。这样一来，当你搜“带孩子吃饭”，它就能精准推有儿童设施的餐厅，而不是随便列一堆馆子。这玩意儿确实挺聪明，也真有用。比如送餐时间估得更准了，推荐的店也更合口味，背后都是知识图谱在默默算。而且它不是死的，今天新开一家店、明天某路堵车，它都能快速更新，反应很快。但也有让人嘀咕的地方。比如它知道得太多了——你搜过一次减肥餐，接下来一周都在推轻食，有点吓人；还有些小餐馆没数据，就很难被推荐，可能永远没机会冒头。总的来说，美团大脑技术很牛，让生活方便了不少，但怎么用好这份“聪明”，别让用户觉得被盯着，也别让小商家被落下，是它下一步得想明白的事。

## 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）；
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；
3. 使用echarts中的关系图，绘制作业2）中的“知识图谱”。
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）（仅信管）。

# 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）；

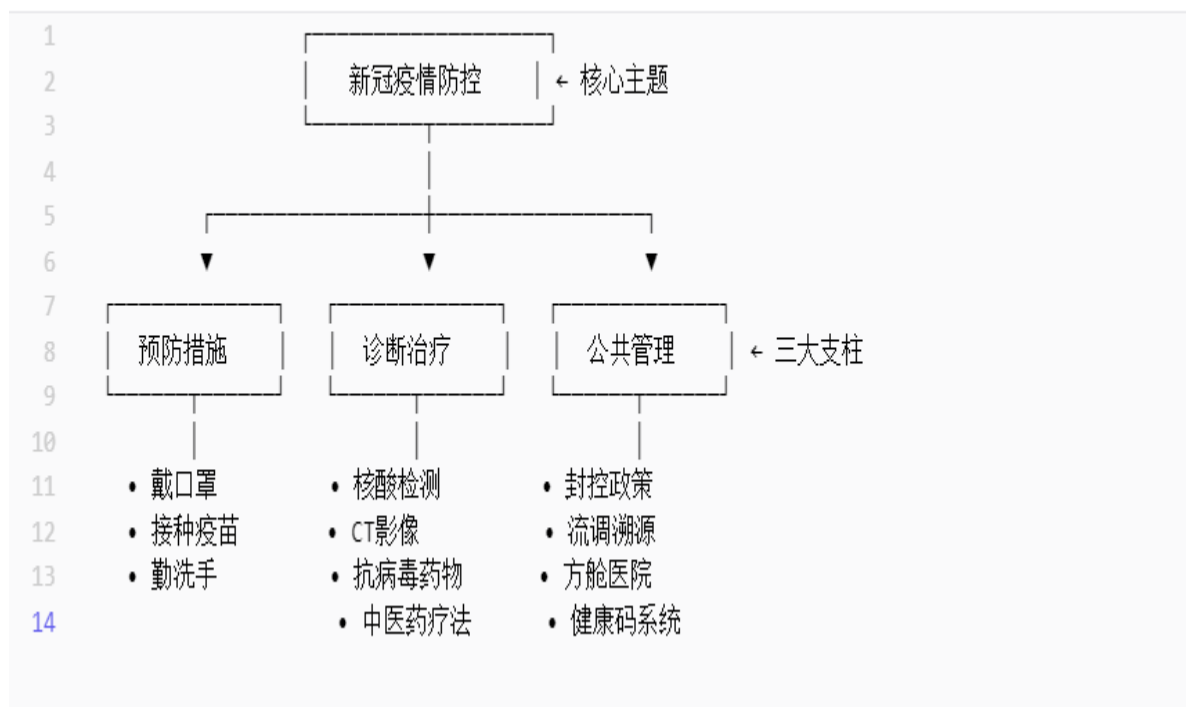
The screenshot shows a web browser window displaying the OpenConcepts dataset page. The browser's address bar shows the URL `data.openkg.cn/dataset/openconcept`. The page header includes the OpenKG.CN logo and navigation links such as '首页', '开放资源', '开放评测', '精选项目', '兴趣小组', '关于我们', and '致谢'. The main content area is titled '浙江大学—大规模细粒度中文概念图谱OpenConcepts'. On the left, there is a sidebar with the dataset name, a follower count of 5, and the institution '浙江大学' with its logo. The main text area contains an introduction and three numbered points:

- OpenConcepts 介绍** OpenConcepts (<http://openconcepts.openkg.cn/>) 是一个基于自动化知识抽取算法的大规模中文概念图谱。概念是人脑对事物的本质反应，能够帮助机器更好的理解自然语言。相较于传统的知识图谱，OpenConcepts包含大量中文细粒度概念，且具备自动更新、自动扩充的能力。比如对于“刘德华”这一实体，OpenConcepts不仅包含“香港歌手”、“演员”等传统概念，还具有“华语歌坛不老男歌手”、“娱乐圈绝世好男人”等细粒度标签。
- OpenConcepts构建** 构建知识图谱具有诸多挑战。早年的英文知识图谱如CyC、WordNet以及中文知识库如HowNet等大多通过专家手工构建，其构建成本非常高昂。ZJCG采取完全自动化构建的方式，基于海量的中文网页数据和若干开放的中文知识库通过自动化信息抽取、短语挖掘等自然语言处理技术，实现概念知识图谱的自动化构建。相较于传统的概念知识图谱，OpenConcepts的特点在于：（1）OpenConcepts包含大量的中文细粒度概念，这部分细粒度概念填补了中文细粒度知识的空白。（2）OpenConcepts是基于全自动化构建的方式，其整合了诸多自然语言处理算法并形成一套完整的知识抽取框架，具备自动化抽取、自动化扩展、自动化更新的能力。OpenConcepts的自动化构建主要分为两大模块，1) 概念知识的自动化抽取 2) 概念知识的融合。我们首先通过开放的知识库、百科InfoBox等结构化、半结构化数据抽取粗粒度的概念。对于细粒度的概念，我们采取短语挖掘和序列标注相结合的策略，通过实体-概念模板和无监督短语挖掘构造弱监督样本，并基于迭代的降噪学习训练基于序列标注的概念抽取模型 ([http://openconcepts.openkg.cn/concept\\_extract/](http://openconcepts.openkg.cn/concept_extract/))，在离线测试集上概念抽取模型准确率可达0.89，召回率可达0.85。然后，我们对抽取到的不同的实体和概念进行融合，并通过贝叶斯估计过滤掉低置信度的概念。此外，我们也构造人工规则约束对高层次的概念进行人工干预，保证准确率。
- OpenConcepts规模和用途** 本次，我们开源了OpenConcepts中的440万概念核心实体，以及5万概念和1200万实体-概念三元组。这些数据包括了常见的人物、地点等通用实体。我们的数据还在不断更新中。本次开源的数据可在openkg.cn获取，OpenConcepts能够为智能推荐、智能问答、人机对话等应用提供数据支持。

# 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；

text



这张知识图谱以“**新冠肺炎疫情防控**”为核心主题，构建了一个三层逻辑结构，清晰展现抗疫工作的系统性与协同性。

**第一层（中心）** 是总目标——有效控制疫情传播、保障人民生命健康。

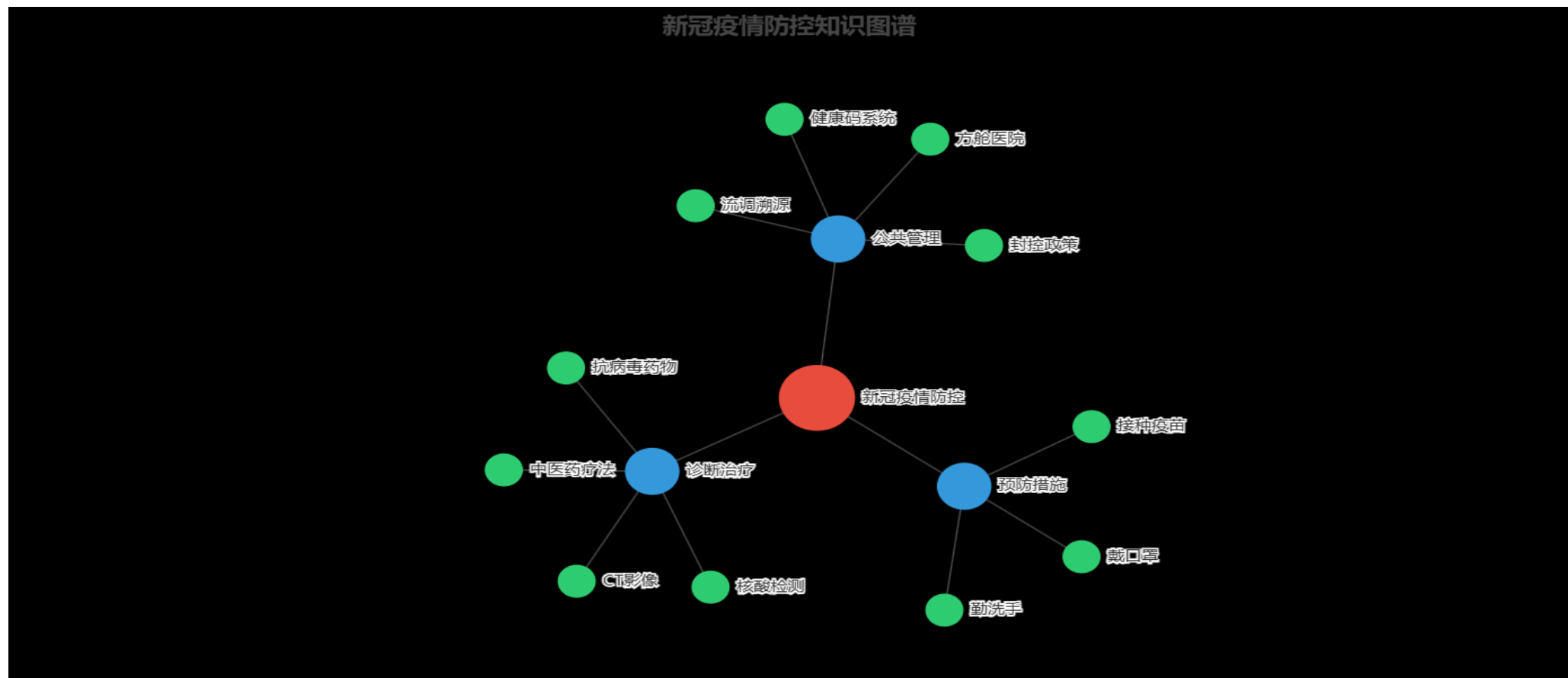
**第二层** 分为三大关键维度：**预防、诊疗、管理**，分别对应“防得住、治得好、控得稳”的策略逻辑。

**预防措施**：聚焦个体行为与免疫屏障（如疫苗），是从源头降低感染风险；**诊断治疗**：关注临床路径，涵盖从检测到用药的全链条医疗响应；**公共管理**：则体现国家治理能力，通过流调、封控、信息系统实现精准防控。**第三层** 列出具体的手段，每个都是真实落地的抗疫实践。例如，“健康码”连接了个人行动与政府监管，“方舱医院”解决了轻症收治难题。



## 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）

