



南京工业大学
NANJING TECH
UNIVERSITY

用户数据采集与关联分析

(结课作业)

吴志祥

18205185639

1030624832@qq.com



第一讲 课程导言与分词

学习使用在线NLP分词系统或微词云分词或清华大学分词演示系统。



第一讲 课程导言与分词

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



The image shows a JupyterLab interface. At the top, it says "jupyter Untitled8 Last Checkpoint: 54 seconds ago". Below this is a menu bar with "File", "Edit", "View", "Run", "Kernel", "Settings", and "Help". To the right of the menu bar is a "Trusted" button. Below the menu bar is a toolbar with icons for saving, adding, deleting, copying, pasting, running, and other actions. To the right of the toolbar, it says "JupyterLab" and "Python 3 (ipykernel)". In the center of the interface is a code cell with the following code:

```
[1]: print("Hello,world.Hello,'阮钰博'")
```

 Below the code cell is the output:

```
Hello,world.Hello,'阮钰博'
```

第一讲 课程导言与分词

完成课后作业（001-004，4份代码的运行）。

Jupyter001-word_cut_基本分词Last Checkpoint: 36 minutes ago

FileEditViewRunKernelSettingsHelp

JupyterLabPython 3 (ipykernel)

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。

1.基本分词

```
[32]: import jieba
[33]: seg_list1 = jieba.cut("曾经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。如果上天能够给我一个重新来过的
[34]: print(' '.join(seg_list1))
曾经/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/一个/重新/来/过/的/
[35]: seg_list2 = 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。")
[36]: print(' '.join(seg_list2))
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/。
```

2.加入词典，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

```
[37]: jieba.load_userdict('dict.txt')
[38]: 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")
[39]: 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

```
[40]: stopwords = [line.strip() for line in open('stop_words.txt', 'r', encoding='utf-8').readlines()]
[41]: seg_list_stopw = jieba.cut("曾经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。如果上天能够给我一个重新来过的
[42]: final = ''
[43]: #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
[44]: 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

完成课后作业（001-004，4份代码的运行）。

Jupyter

1002-word_count_科学家文本

Last Checkpoint: 38 minutes ago

File Edit View Run Kernel Settings Help

JupyterLab Python 3 (ipykernel)

```
[4]: print(''.join(seg_list_huang))  
  
Building prefix dict from the default dictionary ...  
Loading model from cache C:\Users\阮佐博\AppData\Local\Temp\jieba.cache  
Loading model cost 0.524 seconds.  
Prefix dict has been built successfully.  
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./。  
  
[5]: # 加入用户词典  
  
[6]: jieba.load_userdict('dict.txt')  
  
[7]: 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'。)。  
[8]: 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./。  
[9]: # 加入词典之后， 哪些词汇被分离出来了呢？  
[10]: # 使用停用词表  
[11]: stopwords = [line.strip() for line in open('stop_words.txt', 'r', encoding='utf-8').readlines()]  
[12]: stopwords = open('stop_words.txt', 'r', encoding='utf-8').read()  
stopwords = stopwords.split('\n')  
[13]: stopwords  
[13]: ['的', '了', '是', '啊', ',', '.', ':', ';', '。', '！', '，', '；', '：', '。', '！']  
[14]: 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'。)。  
[15]: final = ''  
[16]: for seg in seg_list_huang:  
    if seg not in stopwords:  
        final += seg + '  
[17]: 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

完成课后作业（001-004，4份代码的运行）。

```
jupyter 003-NER-企业年报-数字技术-安全管理 Last Checkpoint: 39 minutes ago
File Edit View Run Kernel Settings Help Trusted
JupyterLab Python 3 (ipykernel)

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

Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\阮仕博\AppData\Local\Temp\jieba.cache
Loading model cost 0.510 seconds.
Prefix dict has been built successfully.

[7]: words

[7]: ['\n',
      '爆发',
      ',',
      '企业',
      '管理',
      '年',
      ',',
      '主题',
      ',',
      '加强',
      'OEHS',
      '三',
      '体系',
      '建设',
      ',',
      '\n',
      '通过',
      '自动化',
      '\n']

[8]: # 2. 定义要统计的特定词汇
target_words = ['数字化', '智能化', '安全']

[9]: # 统计词频
word_counts = Counter(words)

[10]: # 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

[11]: # 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
'\n': 9次
```

第一讲 课程导言与分词

完成课后作业（001-004，4份代码的运行）。

jupyter004_使用大语言模型提取科技文献中的实体Last Checkpoint: 42 minutes ago

FileEditViewRunKernelSettingsHelp

Not Trusted

JupyterLabPython 3 (ipykernel)

```
[5]: # 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)
```

提取到的实体和专业术语:

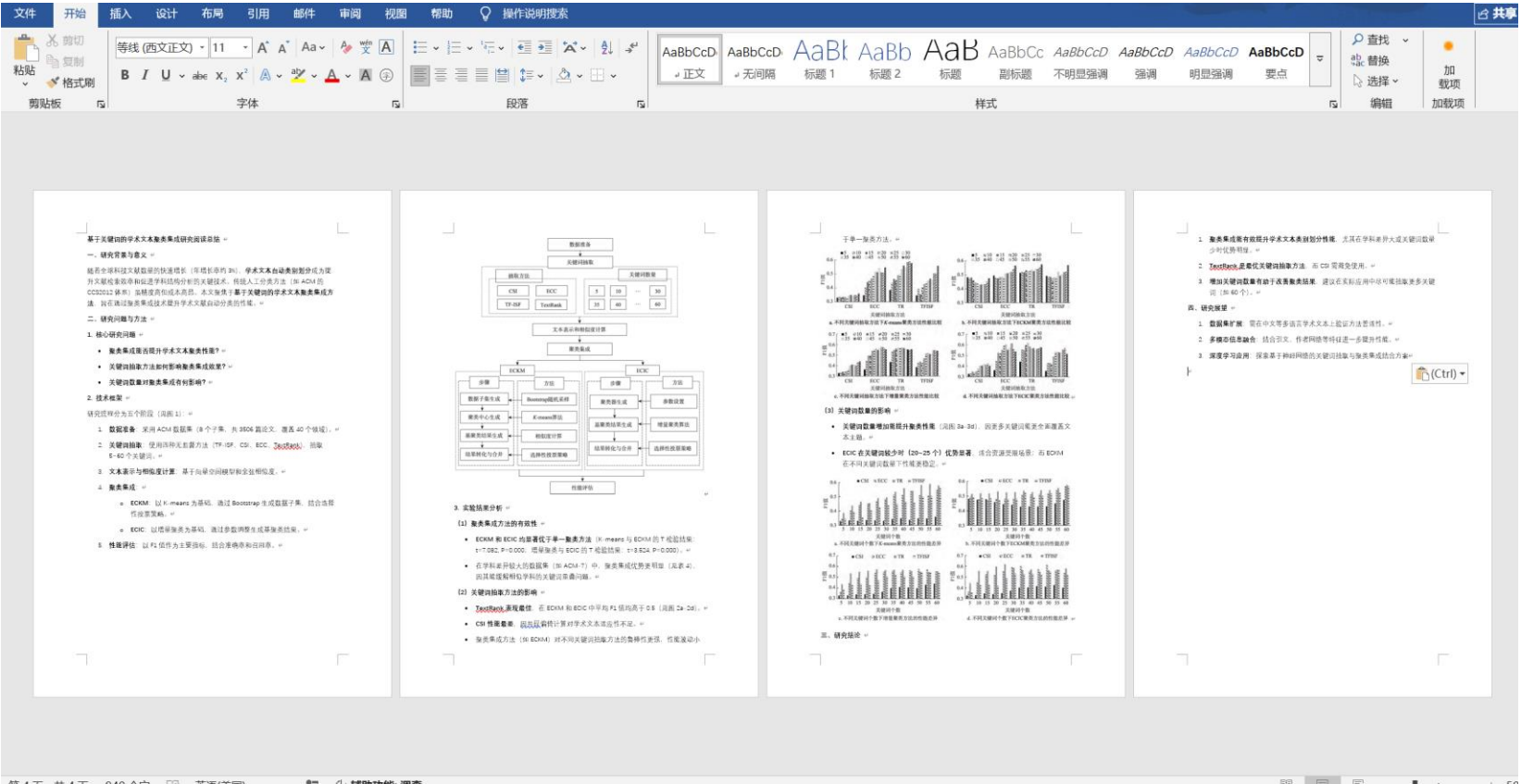
```
```json
{
 "理论": [
 "肿瘤免疫微环境",
 "T细胞耗竭",
 "免疫编辑理论"
],
 "方法": [
 "单细胞RNA测序",
 "细胞亚群聚类",
 "轨迹分析",
 "pseudotime推断",
 "细胞间通讯网络构建"
],
 "工具": [
 "Seurat",
 "Monocle3",
 "CellChat"
],
 "专业术语": [
 "TIME",
 "scRNA-seq",
 "非小细胞肺癌",
 "PD-1/PD-L1",
 "TGF-β途径",
 "免疫抑制信号通路",
 "个体化免疫治疗"
]
}
```
```

提问:

- 1, 使用deepseek开展工作的感觉如何?
- 2, 你觉得大语言模型的活干的怎么样?
- 3, 还是那个问题, 如果可以抽取实体, 那么如何识别关系呢? 你试试用大语言模型识别下关系?

第一讲 课程导言与分词

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）



第二讲 词频统计

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。

| | | |
|--|---|---|
| <div>一、研究背景与意义</div> <div><ul style="list-style-type: none">信息资源管理（Information Resource Management, IRM）是图书馆学、情报学与信息管理领域的重要研究方向，涵盖信息采集、组织、存储、检索、共享与利用的全过程。近十年，大数据、人工智能、云计算、开放获取等技术与理念的迅速发展，深刻影响了信息资源管理的理论、方法与应用场景。通过对 CNKI 收录的相关文献进行计量分析与主题挖掘，可以揭示该领域的研究热点演变、学科交叉趋势及未来发展方向。</div> <div>二、数据来源与研究方法</div> <div><div>1. 数据来源</div><div><ul style="list-style-type: none">中国知网（CNKI）期刊全文数据库检索时间范围：2014 年 1 月—2024 年 12 月检索式示例：SU="信息资源管理" OR SU="IRM"（可根据需要增加同义词）文献类型：学术期刊论文为主，辅以学位论文、会议论文</div><div>2. 研究方法</div><div><ul style="list-style-type: none">计量分析：年度发文量、机构分布、作者合作网络、期刊分布主题分析：高频关键词统计、共词分析、主题聚类（如 LDA 模型）趋势分析：基于时间线的主题热度变化、突现词检测（Burst Detection）可视化：时间线图、热力图、主题演化路径图</div></div> <div>三、总体发文趋势（2014-2024）</div> <div><ul style="list-style-type: none">2014-2016：发文量稳步增长，研究集中在传统的信息资源建设、数字图书馆、知识管理等领域。2017-2019：受大数据与“互联网+”政策推动，发文量显著增加，主题开始向数据治理、开放数据、智慧图书馆延伸。2020-2022：疫情催化数字化转型，在线教育、远程办公、公共卫生信息管理</div> | <div>成为新的研究热点，IRM 与应急管理交叉明显。</div> <div><ul style="list-style-type: none">2023-2024：人工智能生成内容（AIGC）、大模型、数据安全与隐私保护成为前沿主题，发文量趋于稳定或略有下降，但质量提升。</div> <div>四、主题变化分析</div> <div><div>1. 早期阶段（2014-2016）</div><div><ul style="list-style-type: none">高频关键词：数字图书馆、知识管理、信息组织、信息检索、元数据研究特点：以理论探讨和系统建设为主，技术应用相对有限。</div><div>2. 发展阶段（2017-2019）</div><div><ul style="list-style-type: none">新增高频词：大数据、数据挖掘、开放获取、数据共享、智慧图书馆研究特点：强调数据驱动的管理模式，关注多源异构数据的整合与利用。</div><div>3. 爆发阶段（2020-2022）</div><div><ul style="list-style-type: none">新增高频词：应急管理、疫情防控、在线教育、远程办公、公共卫生信息研究特点：IRM 在社会公共事件中的应用凸显，跨学科合作增多。</div><div>4. 前沿阶段（2023-2024）</div><div><ul style="list-style-type: none">新增高频词：人工智能、大模型、AIGC、数据安全、隐私保护、区块链研究特点：关注新技术对信息资源全生命周期管理的影响，尤其是伦理与法律问题。</div></div> <div>五、学科交叉与热点演化</div> <div><ul style="list-style-type: none">学科交叉：IRM 与计算机科学（AI、大数据）、公共管理（应急管理）、教育学（在线教育）、法学（数据合规）形成多学科融合。热点演化路径：<div><div>1. 信息资源建设 → 数据治理 → 智能管理 → 安全与合规</div><div>2. 传统图书馆 → 数字图书馆 → 智慧图书馆 → 全域知识服务平台</div></div></div> <div>六、研究结论</div> <div><div>1. 主题多元化：从传统的图书馆与信息管理扩展到社会治理、公共服务、技术创</div></div> | <div>新等多个领域。</div> <div><div>2. 技术驱动明显：大数据、AI、区块链等新技术不断重塑 IRM 的研究边界。</div><div>3. 应用场景拓展：从学术资源到公共卫生、教育、政府治理等广泛场景。</div><div>4. 安全与伦理关注度提升：数据隐私、信息安全成为近年来的重要议题。</div></div> <div>七、未来研究展望</div> <div><ul style="list-style-type: none">智能化管理：基于大模型的自动化信息组织与知识发现跨域融合：IRM 与智慧城市、数字政府建设的深度融合标准化与规范化：建立适应新技术环境的信息资源管理标准体系国际比较研究：借鉴国外先进经验，推动中国 IRM 理论与实践的全球化</div> |
|--|---|---|

第二讲 词频统计

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；

```
JupyterLab | Python 3 (ipykernel) | Not Trusted
File Edit View Run Kernel Settings Help
+ - [ ] [ ] [ ] [ ] [ ] Code
全义本词频统计的步骤
• 打开文本
• 分词
• 去除停用词（集合的方式）
• 利用字典，进行词-词频的存储
• 排序
• 展示（输出print）
• 
• 这个是非常简单的python程序

[1]: import jieba

[1]: 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

[2]: dele = ' ','!','?','的','“”','(',')',';','>','<','' # 手动设计一些停用词和符号

[4]: import jieba
jieba.add_word('国立交通大学') # 加入字典中没有的新词

Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\阮懿博\AppData\Local\Temp\jieba.cache
Loading model cost 0.502 seconds.
Prefix dict has been built successfully.

[5]: words = list(jieba.cut(article)) # 结巴分词出来的词汇

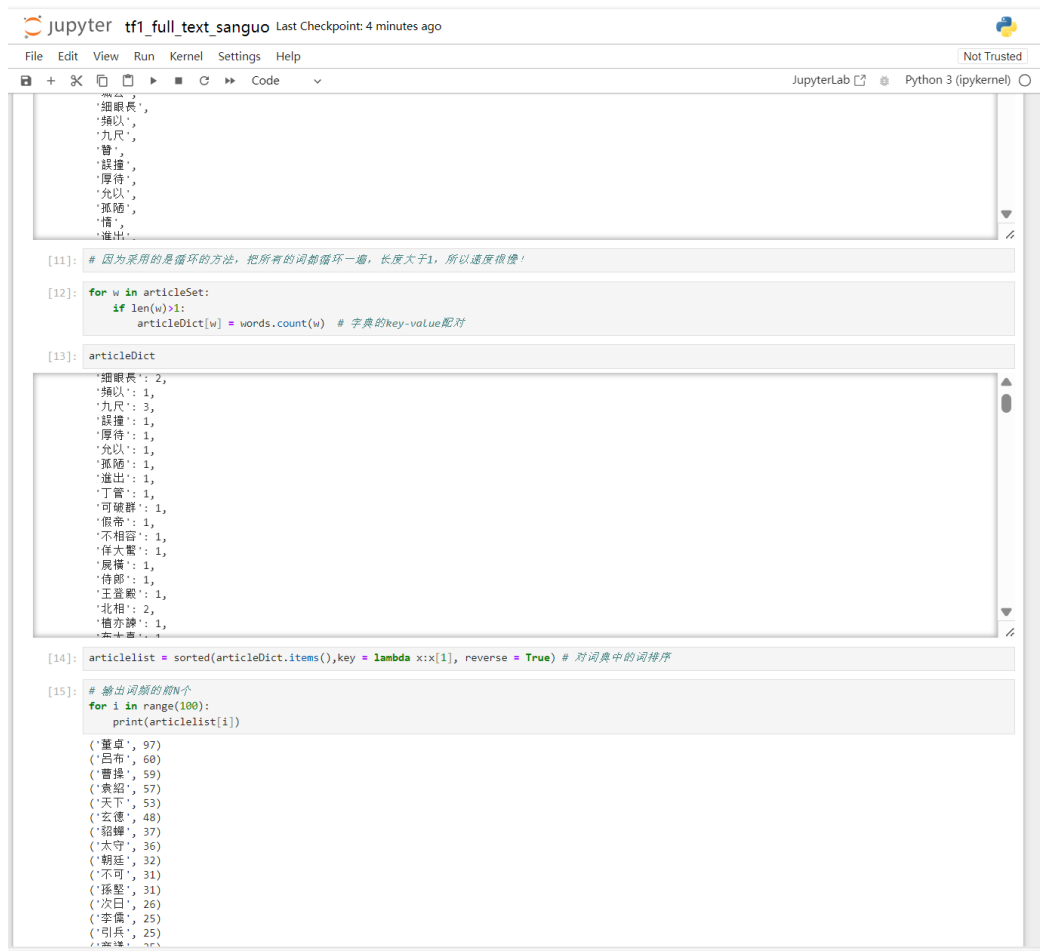
[6]: words

[6]: ['在',
      '核潜艇',
      '领域',
      ', ',
      '我国',
      '已',
      '形成',
      '一套',
      '完整',
      '的',
      '研究',
      ', ',
      '设计',
      ', ',
      '试验',
      ', ',
      '制造',
      ', ',
      '.']

字典
[7]: articleDict = {} # 这是一个字典，准备词-词频的保存
```

第二讲 词频统计

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；



```
jupyter tf1_full_text_sanguo Last Checkpoint: 4 minutes ago
File Edit View Run Kernel Settings Help Not Trusted
JupyterLab Python 3 (ipykernel)

[11]: # 因为采用的是循环的方法，把所有的词都循环一遍，长度大于1，所以速度很慢！

[12]: for w in articleSet:
      if len(w)>1:
          articleDict[w] = words.count(w) # 字典的key-value配对

[13]: articleDict

{'细眼': 2,
 '细以': 1,
 '九尺': 3,
 '該撞': 1,
 '厚待': 1,
 '允以': 1,
 '孤随': 1,
 '進出': 1,
 '丁管': 1,
 '可破群': 1,
 '假帝': 1,
 '不相容': 1,
 '伴大驚': 1,
 '展橫': 1,
 '侍郎': 1,
 '王登殿': 1,
 '北相': 2,
 '相亦諱': 1,
 '云云': 1}

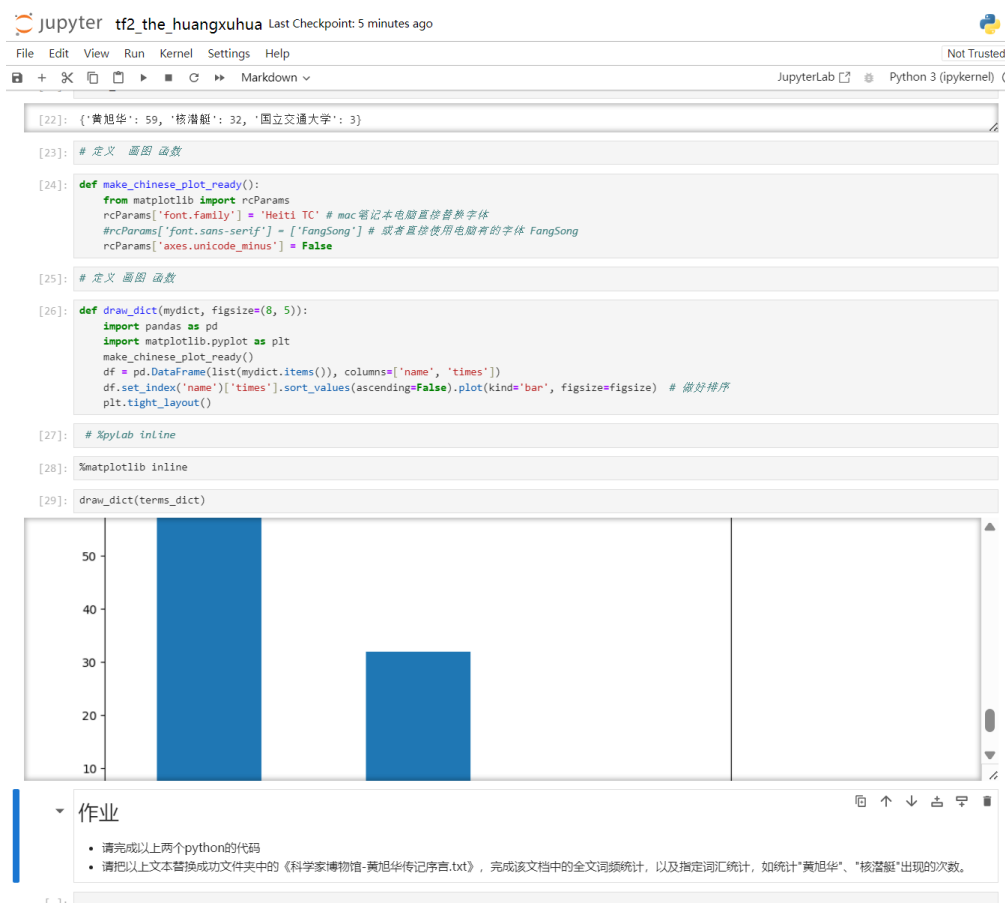
[14]: articlelist = sorted(articleDict.items(),key = lambda x:x[1], reverse = True) # 对词典中的词排序

[15]: # 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)
('曹操', 25)
```

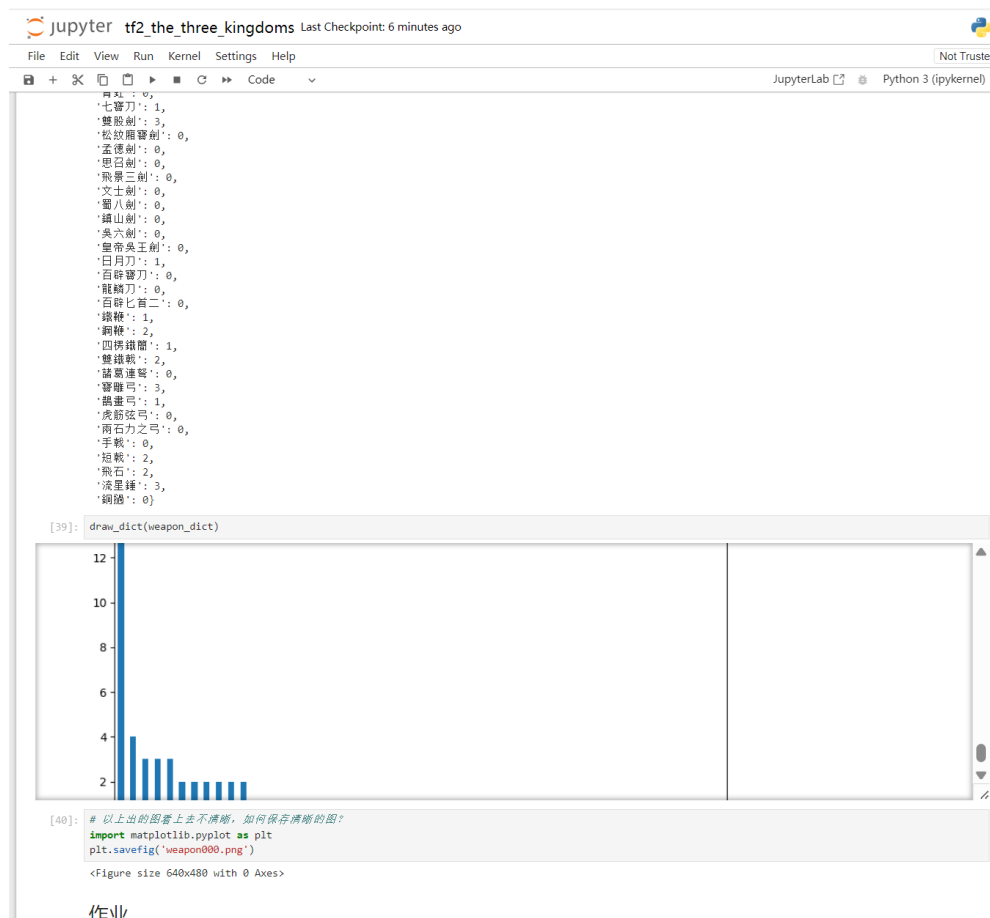
第二讲 词频统计

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；



第二讲 词频统计

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；



第二讲 词频统计

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”。

```
jupyter Untitled Last Checkpoint: 44 seconds ago

File Edit View Run Kernel Settings Help Trusted
JupyterLab Python 3 (pykernel)

[1]: import jieba
from collections import Counter

# 读取文本文件内容（请确保 huangxuhua.txt 和脚本在同一个目录，或提供正确路径）
file_path = '%script_dir/科学家博物馆-黄旭华传记序言.txt' # 替换为您的文件路径
with open(file_path, 'r', encoding='utf-8') as f:
    text = f.read()

# 使用 jieba 分词
words = jieba.lcut(text)

# 统计词频
word_freq = Counter(words)

# 打印词频最高的前10个词
print("=== 全文词频统计 (前10) ===")
for word, freq in word_freq.most_common(10):
    print(f'{word}: {freq}')

# 统计指定词频，如“黄旭华”
target_word = "黄旭华"
target_count = word_freq.get(target_word, 0)
print(f'=== 指定词统计 ===')
print(f'{target_word}出现的次数: {target_count}')

Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\ Administrator\AppData\Local\Temp\jieba.cache
Loading model cost 0.569 seconds.
Prefix dict has been built successfully.

=== 全文词频统计 (前10) ===
的: 169
. : 115
' : 101
* : 61
黄旭华: 53
了: 42

: 40
探索性: 32
探索: 29
面: 23
号: 22
要求: 22
指标: 21
时: 19
工作: 17
高: 17
成本: 15
" : 14
小组: 14
院士: 13
" : 13
专业: 13
进行: 13
数量: 12
次: 12
机制: 12
金: 12
要求: 12
研: 11
二: 11
年: 11
以: 10
和: 10
状况: 10
也: 9
原因: 8
等: 8
和: 8
为: 8
这: 8
发展: 8
介绍: 8
思想: 7
也: 7
历史: 7
传记: 7
人数: 7
及其: 7
研究: 6
统计: 6

=== 指定词统计 ===
"黄旭华"出现的次数: 53
```

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。

形状与主题暗示：头部轮廓 暗示内容可能和 “人才、智慧、科研工作者” 相关。

核心关键词:

最大的几个词是“核潜艇”、“研制”、“船舶工业”、“研究”、“专家”——直接点明主题领域：核潜艇的研发、船舶工业的技术研究，以及相关领域的专家。

其他高频词如“设计”、“制造”、“科技”、“获奖（特等奖、进步奖）”等，进一步说明是在讲核潜艇从设计到制造的科研过程，以及其中取得的科技成就与荣誉。

背景信息词:

地域：“中国”、“广东省（海丰县、揭阳县、汕尾市）”——提示人物或成果和这些地区有关联。

单位/机构：“中船（中国船舶集团）”、“总公司”、“造船系”、“国防科工委”——指向船舶工业系统内的单位、院校和专业领域。

荣誉/身份：“院士”、“创始人”、“总工程师”、“劳动模范”——体现人物在行业内的地位与贡献。



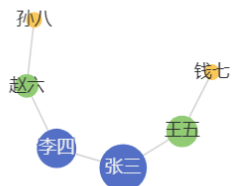
第三讲 词云与可视化

2. 使用Echarts，制作3个以上图，其中一个必须是“关系”，图的概念越明确越好。

社交网络关系图

展示用户之间的互动关系

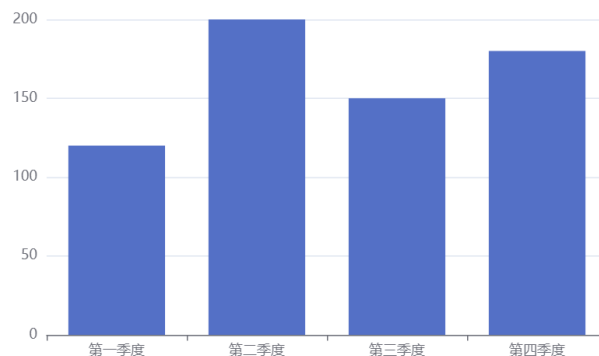
朋友 同事 家人



2023年季度销售额

单位：万元

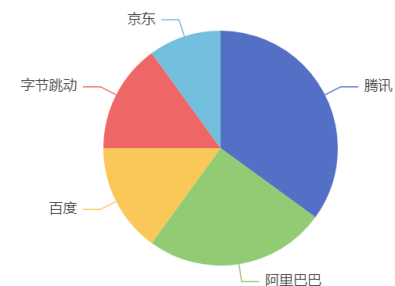
销售额



腾讯
阿里巴巴
百度
字节跳动
京东

2023年市场份额

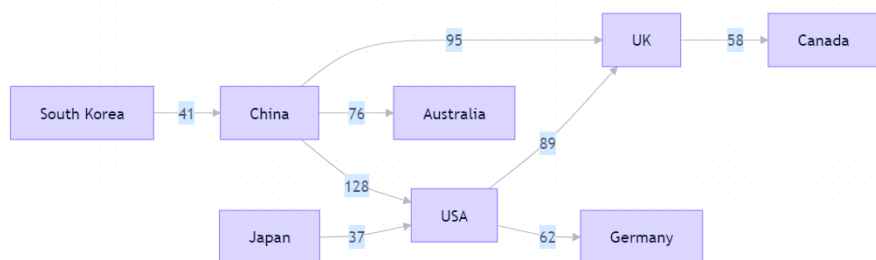
主要科技公司市场份额占比



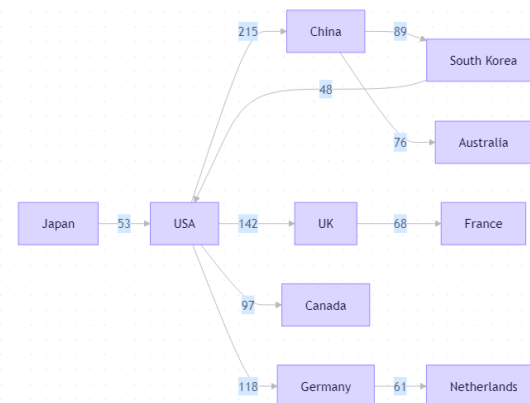
第三讲 词云与可视化

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。

“碳中和”研究的国家合作网络图谱



“生成式人工智能”（**Generative AI**）
研究的国际合作网络图谱（**2019–2024**）



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）。



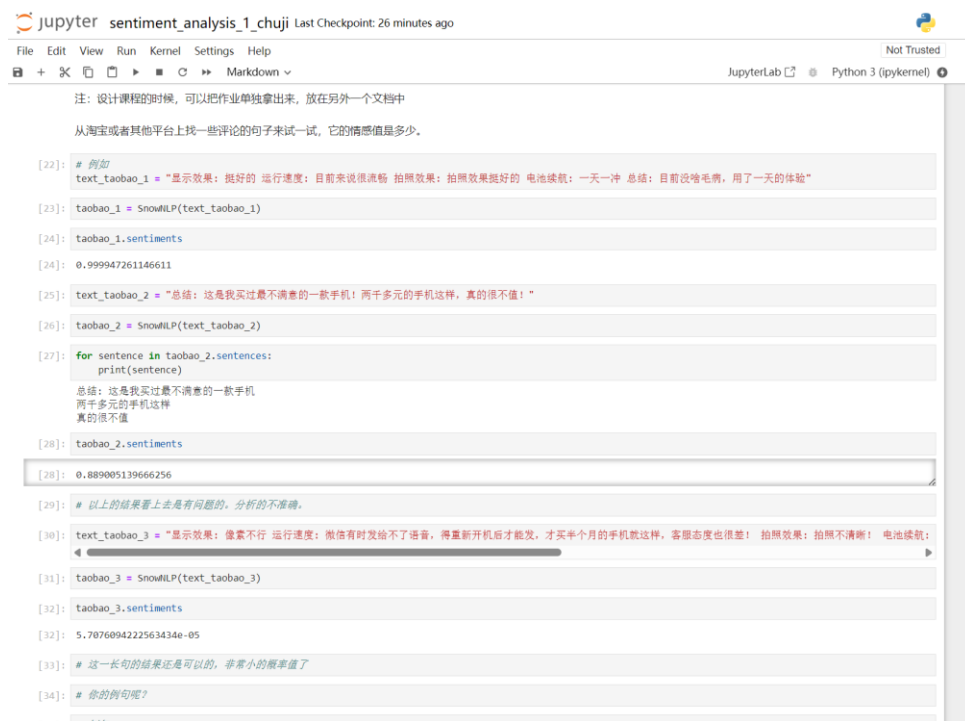
第四讲 情感分析

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。



The screenshot shows a JupyterLab notebook titled "sentiment_analysis_1_chuji". The notebook contains several code cells and their outputs. The first cell shows a comment about design courses and a task to analyze sentiment from Taobao reviews. The second cell shows the loading of a SnowNLP model. The third cell shows the sentiment analysis of a review about a phone. The fourth cell shows the sentiment analysis of another review about a phone. The fifth cell shows a loop that prints the sentences of a review. The sixth cell shows the sentiment analysis of a review about a phone. The seventh cell shows a comment about the results. The eighth cell shows the sentiment analysis of a review about a phone. The ninth cell shows a comment about the results. The tenth cell shows a comment about the results. The eleventh cell shows a comment about the results. The twelfth cell shows a comment about the results.

```
[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"

[23]: taobao_1 = SnowNLP(text_taobao_1)

[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：
      <

[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。

jupyter sentiment_analysis_2_timeline Last Checkpoint: 27 minutes ago

File Edit View Run Kernel Settings Help

Not Trusted

JupyterLab Python 3 (ipykernel)

在图中，我们观察到许多正负评价的情感数据的时间。同时，我们也得观察到部分负数据的时间点。为便于比较情感数据的时间点，我们进行对比，被Python分析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。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，不是因为没有男朋友，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，不是因为没有男朋友，而是感觉哪哪人都多，所以特意避开，这次实在鬼饯A餐厅了，所以赶在正日子也出来了，从下午四点多时候我看排号就排到一百多了，我从家开车过去得堵的一个小时，我一番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，到的时候我看看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，就从我们到那里到等位区开始，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，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，就坏得让服务员下的，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的，要不就别赶节日去，太火爆了！

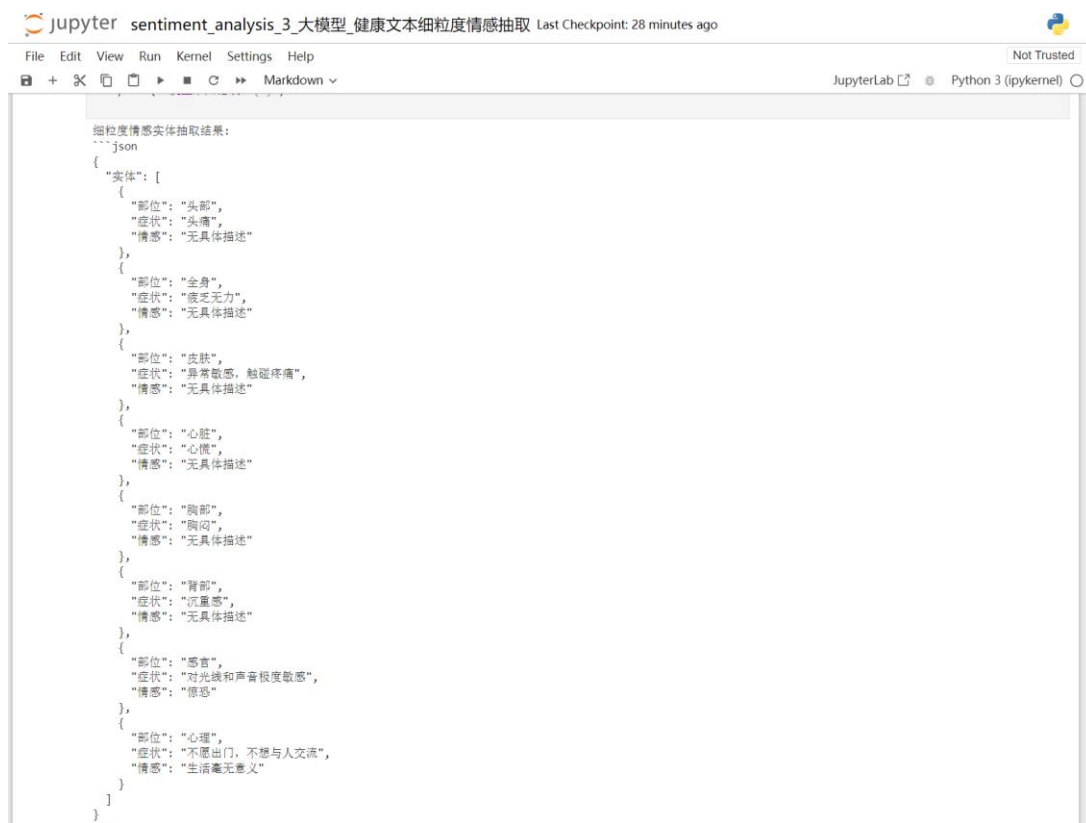
分析

- 通过阅读，你可以发现这位顾客确实有了一次比较糟糕的体验——等候的时间太长了，以至于使用了“悲剧”一词；另外还提及服务不够周到，以及环境吵闹等因素。正是这些词汇的出现，使得分析结果数值非常低。
- 好在顾客很通情达理，而且对该分店的人性化做法给予了正面的评价。
- 从这个例子，你可以看出，虽然情感分析可以帮助你自动化处理很多内容，然而你不能完全依赖它。
- 自然语言的分析，不仅要考虑表达强烈情感的关键词，也需要考虑到表述方式和上下文等诸多因素。这些内容，是现在自然语言处理领域的研究前沿。我们期待着早日应用到科学家们的研究成果，提升情感分析的准确度。
- 不过，即便目前的情感分析自动化处理不能达到非常准确，却依然可以帮助你快速定位到那些可能有问题的异常点(anomalies)，从效率上，比人工处理要高出许多。
- 你读完这条评论，长出了一口气。总结了经验教训后，你决定将人性化的服务贯彻到底。你又想到，可以收集用户等候时长数据，用数据分析为等待就餐的顾客提供更为合理的等待时长预期。这样就可以避免顾客一直等到很晚了。
- 祝贺你，经理！在数据智能时代，你已经走在了正确的方向上。

```
[ ]:
```

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。

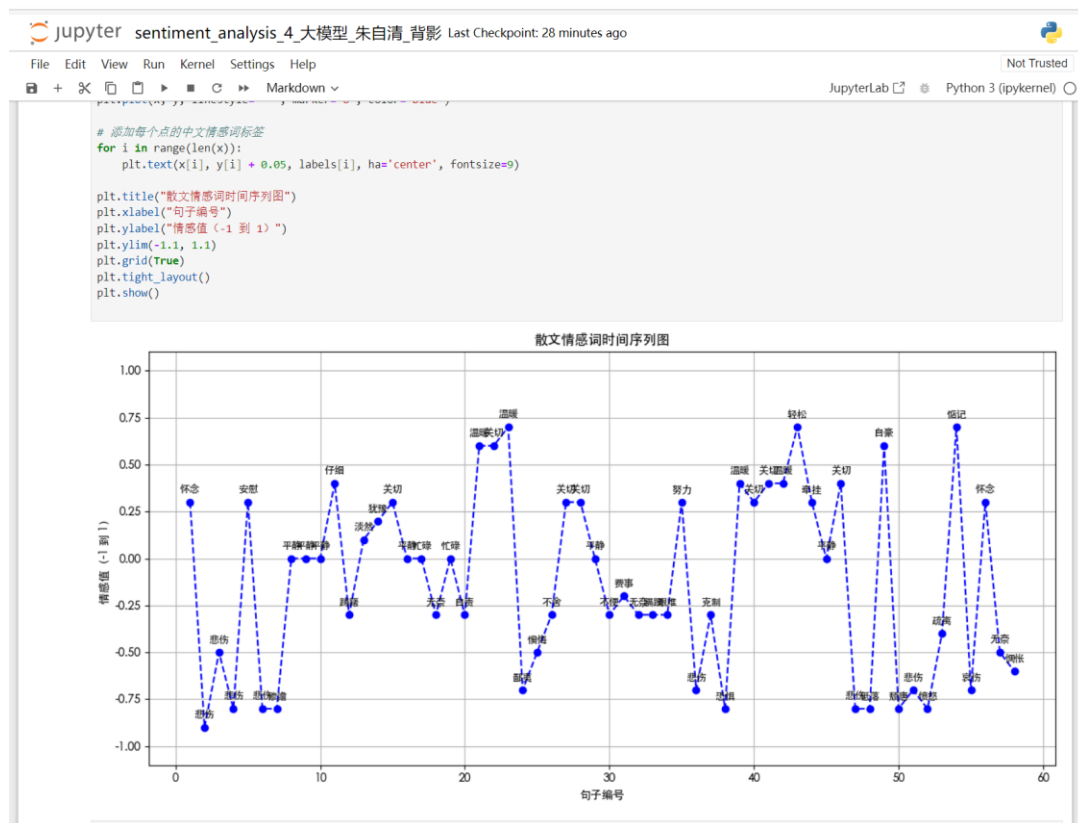


The screenshot shows a JupyterLab window titled "sentiment_analysis_3_大模型_健康文本细粒度情感抽取". The interface includes a menu bar (File, Edit, View, Run, Kernel, Settings, Help) and a toolbar with icons for file operations and execution. The main area displays a JSON output of sentiment analysis results. The JSON structure is as follows:

```
--- 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"
    }
  ]
}
```

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。



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。

截至2026年初，美团持续推动其“美团大脑”（Meituan Brain）的发展，这是一套融合多模态数据、深度学习与自然语言处理技术的智能决策系统，旨在为本地生活服务平台提供智能化基础设施。在最新的进展中，美团大脑的知识图谱生态构建取得了显著突破，呈现出以下几个核心特点：

一、最新进展概述

1. 多源异构数据的深度融合

美团大脑已整合来自外卖、到店、酒店旅游、生鲜零售、即时配送等多个业务线的结构化与非结构化数据，涵盖商家、商品、用户、地理位置、行为日志、评论文本等。最新版本引入了更多外部数据源（如政府公开数据、第三方地图信息、社交媒体内容），提升了图谱的广度与深度。

2. 动态知识图谱的实时更新机制

传统知识图谱多为静态或半静态，而美团大脑现已实现基于流式计算框架（如Flink + Kafka）的动态图谱更新，能够实时捕捉商家营业状态变化、用户偏好迁移、突发事件（如疫情、天气）对消费行为的影响，确保图谱的时效性与准确性。

3. 多模态知识表示与推理

美团大脑引入多模态学习技术，将文本、图像、语音、位置信号等融合建模。例如，通过视觉模型识别菜品图片中的食材与风格，结合用户评论的情感分析，自动丰富菜品实体的属性标签。同时，基于图神经网络（GNN）的推理引擎支持复杂查询，如“适合情侣约会且最近评价高的日料店”。

4. 知识图谱驱动的个性化推荐与决策

最新应用中，美团大脑不仅用于搜索排序和推荐系统，还深入到商家运营决策中。例如，通过分析区域竞争格局、用户流动趋势和供应链数据，为商家提供选址建议、定价策略和营销时机预测。

5. 开放平台与生态协同

美团已逐步开放部分知识图谱能力，通过API接口服务于中小商家、第三方开发者及研究机构，推动本地生活服务生态的智能化升级。例如，商家可通过接口获取自身店铺在知识图谱中的语义画像，优化线上展示。

第六讲 知识图谱理念

二、分析与评价

优势与创新点：

- 场景驱动的深度优化：美团大脑紧密结合本地生活服务的复杂场景，知识图谱设计并非通用型，而是针对“吃住行游购娱”全链路进行定制化建模，具有高度的实用性和落地性。
- 实时性与动态性领先：相比传统电商或搜索引擎的知识图谱，美团大脑在实时更新和事件响应方面表现突出，尤其在应对突发公共事件（如极端天气、节假日高峰）时展现出强大的适应能力。
- 多模态融合提升语义理解：通过整合视觉、语言、行为等多维度信息，美团大脑在实体消歧、情感理解、意图识别等方面表现优异，显著提升了用户体验。

挑战与局限：

- 数据隐私与合规风险：随着图谱规模扩大和外部数据引入，如何在保障用户隐私（如GDPR、中国《个人信息保护法》）的前提下实现数据共享与模型训练，仍是重大挑战。
- 图谱质量与噪声控制：多源数据融合不可避免地引入噪声，如何自动化清洗与验证知识三元组的质量，仍需更强大的自监督学习与人工反馈机制。
- 生态开放性不足：尽管已有开放计划，但核心知识图谱的访问仍受限，第三方开发者难以深度参与，可能限制生态创新活力。

第六讲 知识图谱理念

三、总结

美团大脑的知识图谱生态构建已从“支撑内部业务”迈向“赋能行业生态”，其最新进展体现了本地生活服务智能化的前沿水平。其在多模态融合、实时更新、场景化推理等方面的创新值得肯定，但在数据治理、开放协作和长期可持续性上仍有提升空间。未来，若能进一步加强跨平台知识共享、提升图谱的可解释性与透明度，美团大脑有望成为全球本地生活服务领域最具影响力的AI基础设施之一。

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）；



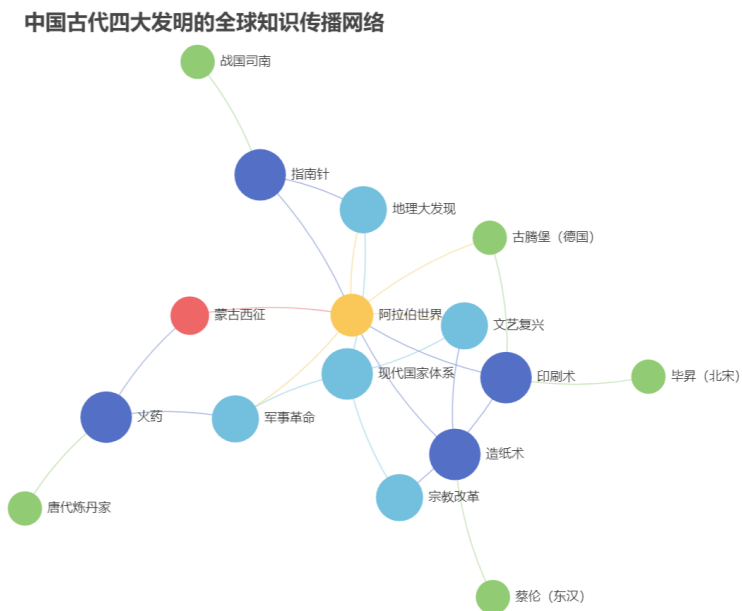
第六讲（2） 知识图谱工具

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；



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）（仅信管）。

黄旭华人物知识图谱

