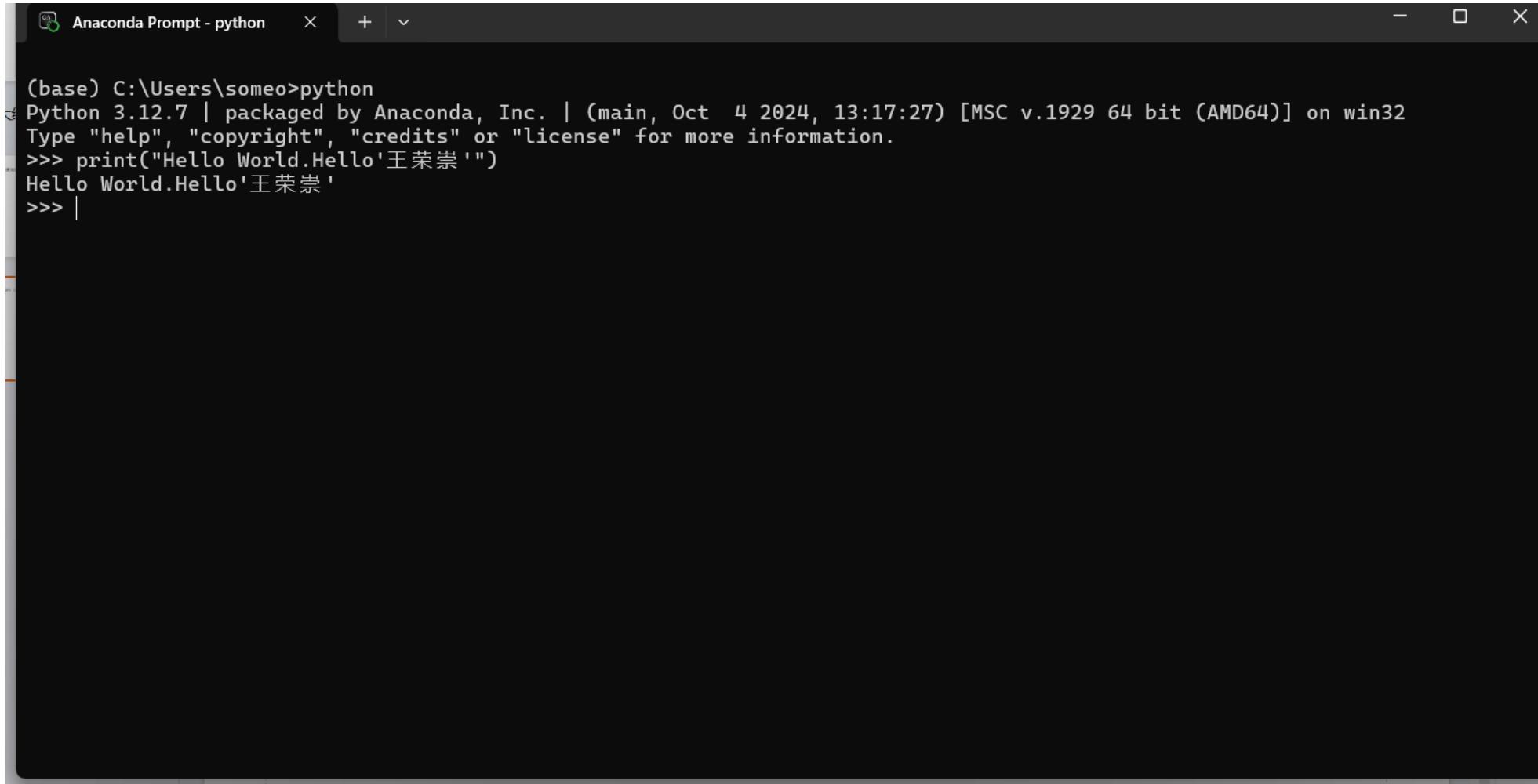


第一讲

(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'") ;

A screenshot of the Anaconda Prompt window titled "Anaconda Prompt - python". The window shows a Python 3.12.7 interactive session. The user has typed the following command:

```
(base) C:\Users\someo>python
Python 3.12.7 | packaged by Anaconda, Inc. | (main, Oct 4 2024, 13:17:27) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello World.Hello'王荣崇'")
Hello World.Hello'王荣崇'
>>> |
```

The output of the print statement is visible in the terminal.

3.完成课后作业（**001-004**, 4份代码的运行）。

001.

jupyter 001-word_cut_基本分词 Last Checkpoint: 9 minutes ago

File Edit View Run Kernel Settings Help

JupyterLab Python 3 (ipykernel)

[28]: `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\$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\$。

[29]: `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。")`

[30]: `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，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

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

[32]: `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。")`

[33]: `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

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

[35]: `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")`

[36]: `final = ''`

[37]: `#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`

[38]: `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/

002

jupyter 002-word_cut_科学家文本 最后检查点: 10分钟前

档案 编辑 景观 跑 内核 设置 帮助

拖拽至此上传

JupyterLab Python 3 (ipykernel)

```
[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/。/1994/年/当选/为/中国工程院院士/。')
[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', '、', '、', '。', '、', '停用']
[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/。/1994/年/当选/为/中国工程院院士/。')
[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/
```

003

jupyter 003-NER-企业年报-数字技术-安全管理 Last Checkpoint: 12 minutes ago

Not Trusted

File Edit View Run Kernel Settings Help

JupyterLab Python 3 (ipykernel)

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

[15]: # 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

[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
```

Jupyter 004_使用大语言模型提取科技文献中的实体 Last Checkpoint: 13 minutes ago

File Edit View Run Kernel Settings Help Not Trusted JupyterLab Python 3 (ipykernel)

```
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路径",
 "免疫治疗",
 "免疫抑制信号通路",
 "个体化免疫治疗"
]
}
```

```

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）**。

基于学术论文全文的研究方法句自动抽取研究阅读总结

基于学术论文全文的研究方法句自动抽取研究，旨在通过自然语言处理技术，精准识别并抽取论文中描述自身所用方法及引用前人方法的句子，实现学术方法论知识的结构化转化。其技术演进历经规则驱动、机器学习、深度学习至大模型融合四阶段，当前以BERT+BiLSTM为核心的深度学习模型为主流。研究虽为科研效率提升、学科方法体系梳理提供关键支撑，但仍面临中文语料稀缺、跨领域适配性弱、隐含方法句识别难等挑战。未来需聚焦多学科高质量数据集构建，融合大模型优化低资源场景性能，结合科学哲学完善方法界定，进一步拓展技术应用价值。

该研究聚焦学术文献中研究方法句的自动识别与抽取，依托自然语言处理技术，将分散于论文全文的方法论描述（含论文自身采用方法、引用前人方法两类）转化为结构化信息，核心价值是为科研工作者快速获取适配方法、梳理学科方法体系提供支撑，推动学术知识细粒度挖掘与跨学科扩散。

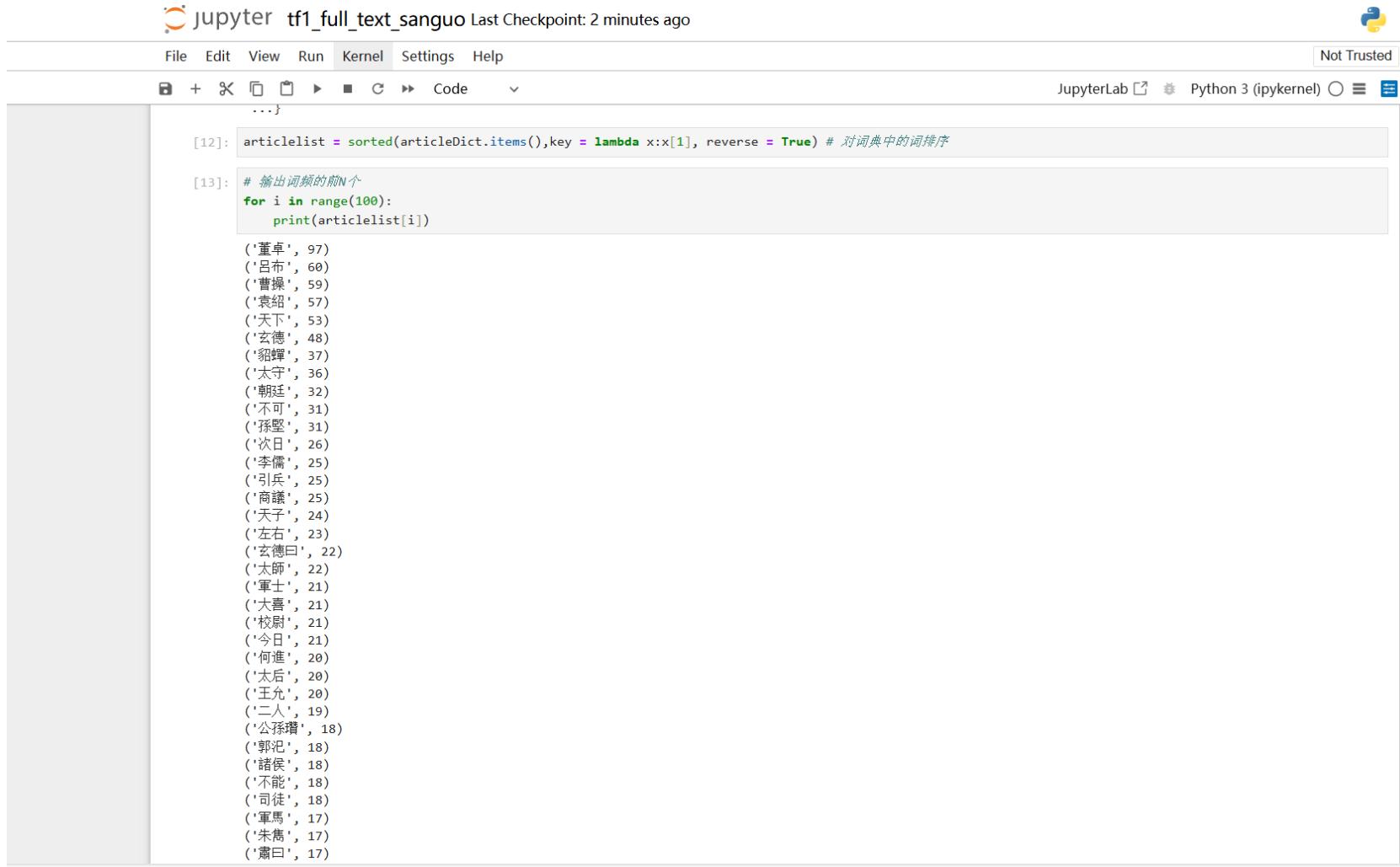
第二讲 词频统计

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。

近十年（2014-2024年）间，基于CNKI数据库的文献分析显示，“信息资源管理”领域经历了从传统文献资源管理向数据智能与生态化治理的深刻转型：早期（2014-2017年）研究集中于图书馆、档案馆的数字化与信息系统建设；中期（2018-2021年）随着大数据技术普及，重心转向数据治理、知识图谱与用户行为分析；近期（2022-2024年）则全面融入人工智能（尤其是生成式AI与多模态处理）、隐私计算及合规安全体系，形成技术智能驱动的新范式。研究主线从数据库、云计算逐步演进至区块链、元宇宙等可信技术，学科边界从图书情报学拓展至与计算机科学、管理学、法学的深度融合，尤其在数据资产化、健康信息管理、智慧城市等场景形成跨学科热点。与此同时，国家《网络安全法》《数据安全法》等政策与数据要素市场化战略，强力推动了研究向公共数据开放、平台治理等实践议题倾斜，而算法伦理、信息公平等社会性关注也显著升温。整体上，该领域已从机构化的资源保管，演进为以数据要素为核心、智能技术为引擎、多学科交叉为特征、兼顾价值创造与系统治理的综合性方向，呈现出“数据要素化—技术智能化—治理体系化”的清晰发展路径。

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；

全文本



The screenshot shows a Jupyter Notebook interface with the title "jupyter tf1_full_text_sanguo Last Checkpoint: 2 minutes ago". The menu bar includes File, Edit, View, Run, Kernel, Settings, Help, and a "Not Trusted" status indicator. The toolbar has icons for file operations like New, Open, Save, and Run. The bottom status bar shows "JupyterLab" and "Python 3 (ipykernel)". The code cell contains two lines of Python code:

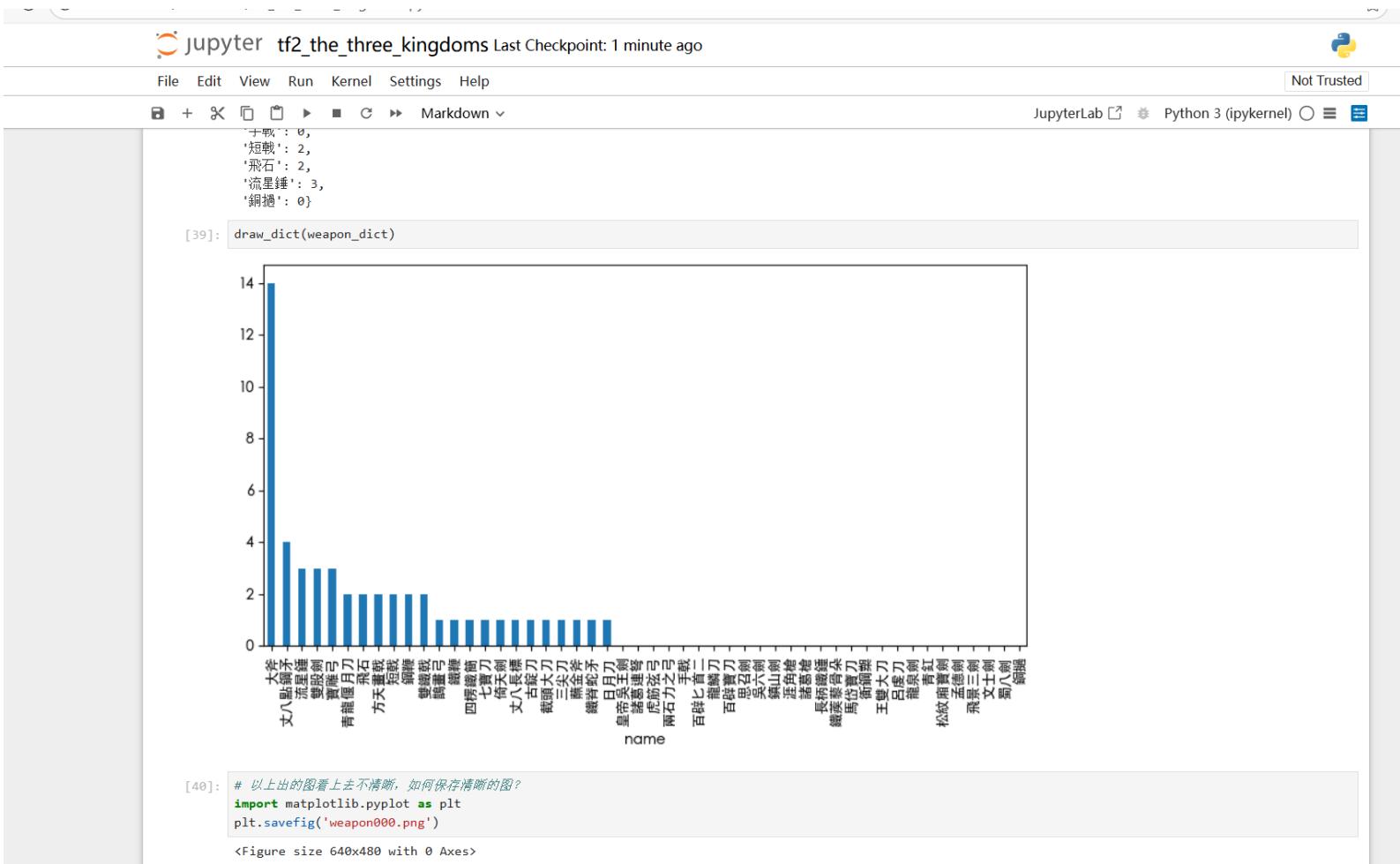
```
[12]: articlelist = sorted(articleDict.items(),key = lambda x:x[1], reverse = True) # 对词典中的词排序
[13]: # 输出词频的前N个
       for i in range(100):
           print(articlelist[i])
```

The output of the code is a list of tuples representing word frequencies, starting with ('董卓', 97) and ending with ('肅曰', 17). The list is truncated at 100 items.

| 词 | 频率 |
|-------|----|
| '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 |
| '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 |
| '公孫瓈' | 18 |
| '郭汜' | 18 |
| '諸侯' | 18 |
| '不能' | 18 |
| '司徒' | 18 |
| '軍馬' | 17 |
| '朱雋' | 17 |
| '肅曰' | 17 |

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



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”

i

1)

The screenshot shows a Jupyter Notebook interface with the following details:

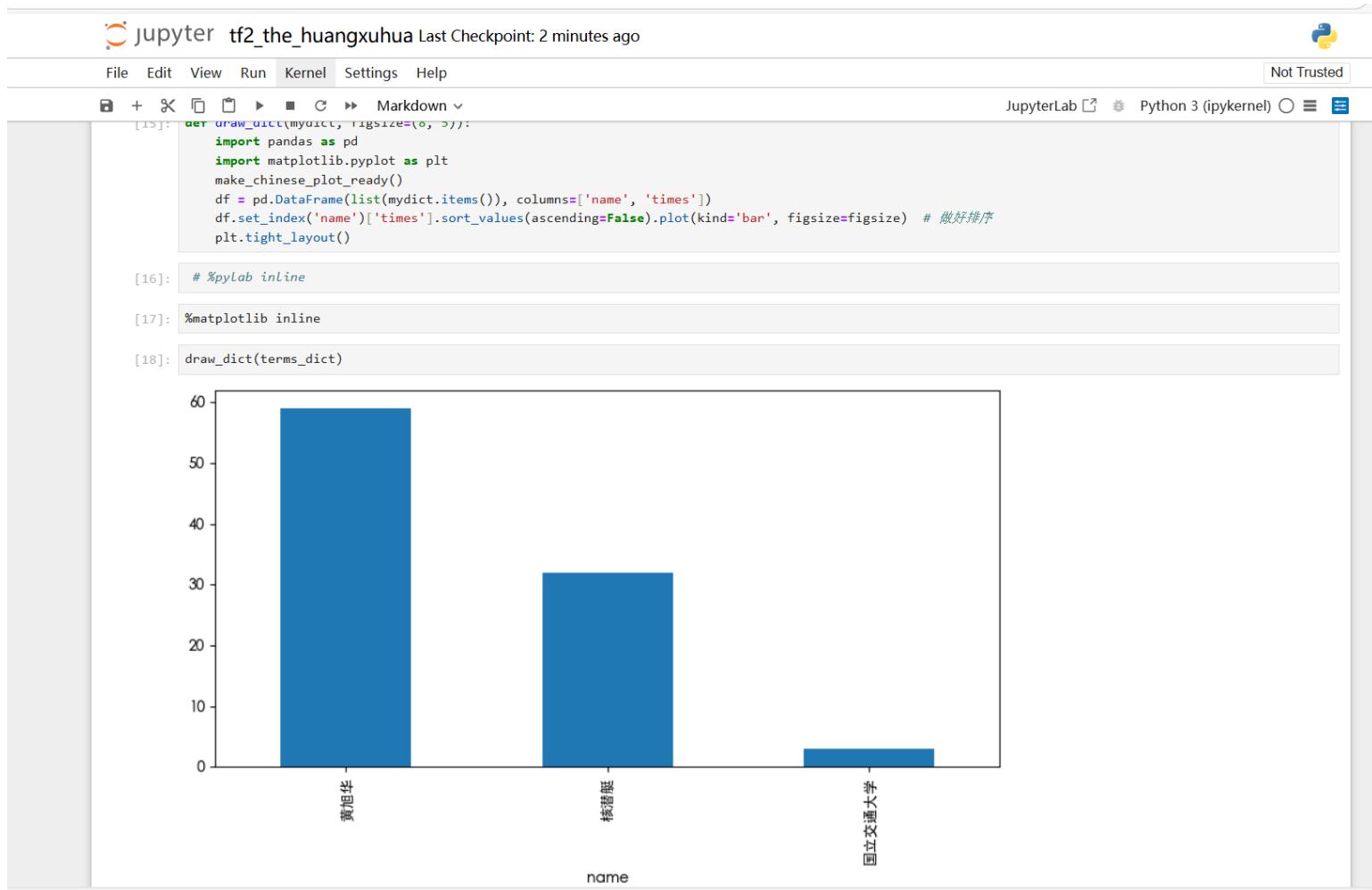
- Header:** The title is "jupyter tf1_full_text_huangxuhua Last Checkpoint: 1 minute ago".
- Toolbar:** Includes File, Edit, View, Run, Kernel, Settings, Help, and a "Not Trusted" status.
- Code Cells:** Several cells are visible:
 - [1]: `import jieba`
 - [2]: `article = open('科学家博物馆-黄旭华传记序言.txt','r',encoding = '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`
 - [3]: `delete = {'.','!','?','的','“','”','（','）','‘','’,'《','》','《','》'} # 手动设计一些停用词和符号`
 - [7]: `jieba.add_word('国立交通大学') # 加入字典中没有的新词`
 - [8]: `words = list(jieba.cut(article)) # 结巴分词出来的词汇`
 - [9]: `words` (This cell displays the list of words from the previous cell.)
- Output Cell [9]:** Shows the list of words:

```
[9]: ['在', '核潜艇', '领域', '、', '我国', '已', '形成', '一套', '完整', '的', '研究', '、', '设计', '、', '试验', '、', '制造', '。']
```
- Section Headers:** Two sections are labeled:
 - 字典** (Dictionary) under the heading "集合 (数据结构的一种)"
 - 集合 (数据结构的一种)**
- Code Cells at the Bottom:**
 - [10]: `articleDict = {} # 这是一个字典, 准备词-词频的保存`
 - [11]: `articleSet = set(words)-delete`

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)



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）。

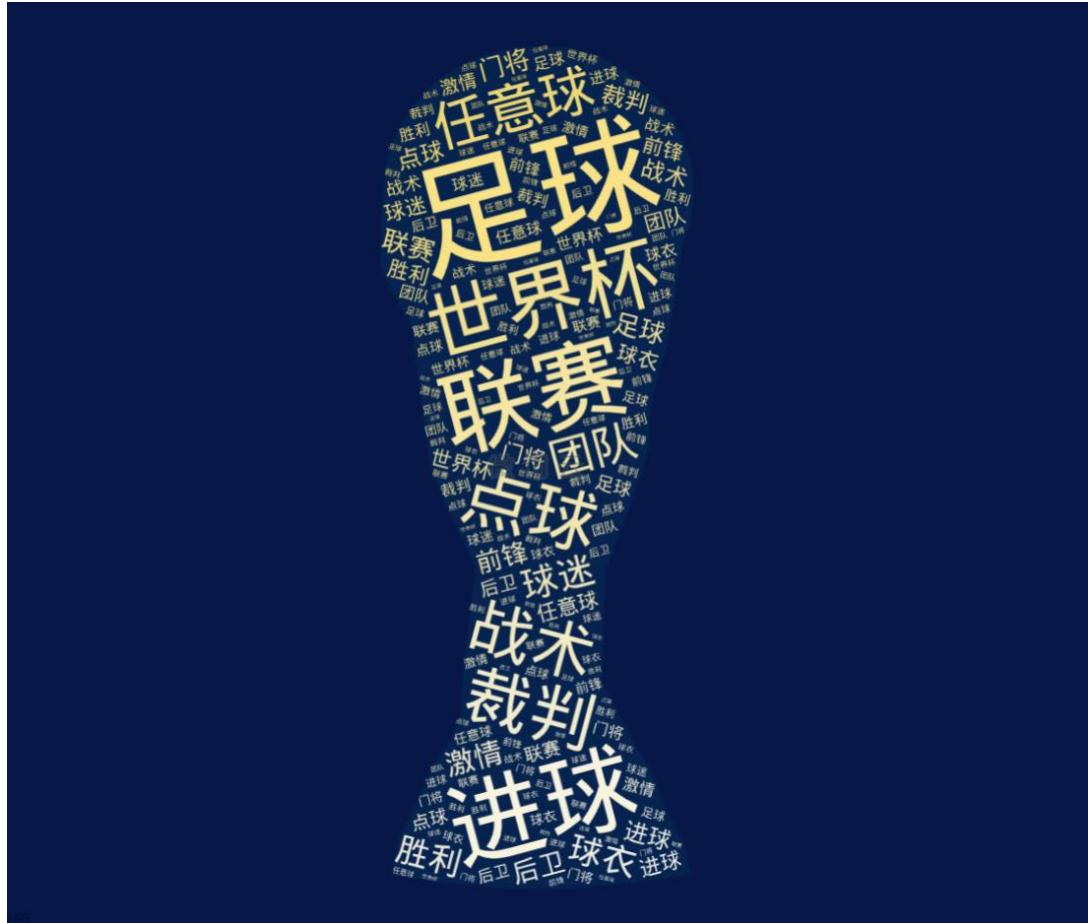
这篇题为《Long live the scientists: Tracking the scientific》的论文，由Wang及其合作者于2018年发表，为我们理解现代科学家的职业生涯与生命历程之间的关系，提供了一幅基于大数据的、细致而有时又反直觉的画卷。研究的出发点在于探究一个看似简单的问题：作为拥有高学历和社会经济地位的精英群体，科学家是否单纯地因为其社会优势而享有更长的寿命？他们的学术成就、所处领域以及地域背景，又会如何具体地雕刻其生命的长度？

研究发现：科学家平均寿命高于大众，但内部差异显著。高产出、高影响力的科学家更长寿，印证了“成就-长寿”关联。生命科学领域学者寿命最长，物理学等领域相对较短。反直觉的是，来自欧美发达科研体系的科学家寿命优势不明显，甚至更短，揭示了学术界高压竞争对健康的潜在损耗——此即“资源诅咒”假说。研究警示，在追求科学卓越的同时，必须关注科研生态的健康可持续性。

综合来看，Wang等人的这项研究远不止是一份关于科学家寿命的统计报告。它通过科学计量学与社会科学交叉的透镜，深刻揭示了在追求知识创新的宏大叙事背后，科学家作为个体的生命体验所承受的结构性张力。论文肯定了学术成就对寿命的积极价值，但同时也发出了严肃的警示：当前全球主流的、以高强度竞争和短期产出为核心的科研生态，可能正在损害其最重要资产——科研人员本身——的长期健康与可持续性。它促使科研机构、政策制定者以及整个学术共同体反思：在竞逐科学巅峰的同时，我们是否构建了一个足以滋养而非耗竭创造者的环境？这项研究因此不仅是对过去的测量，更是对未来更健康、更人文化的科学事业的一种呼唤。

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。

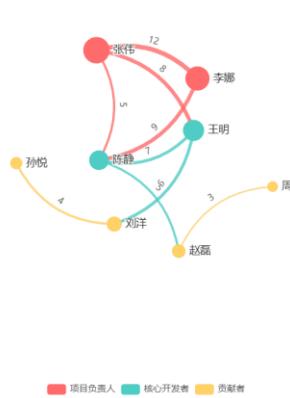


这个足球主题词云做成了大力神杯形状，以“足球”为核心关键词，围绕其聚合了足球领域的核心元素：
赛事维度：包含“世界杯”“联赛”等顶级赛事标签；
比赛要素：覆盖“进球”“点球”“任意球”等技术动作，以及“裁判”“战术”“团队”等赛事相关角色与策略；
场景与情感：融入“球迷”“激情”“胜利”等观赛 / 比赛氛围类词汇，还有“球衣”“门将”“前锋”等装备、位置类术语。

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。

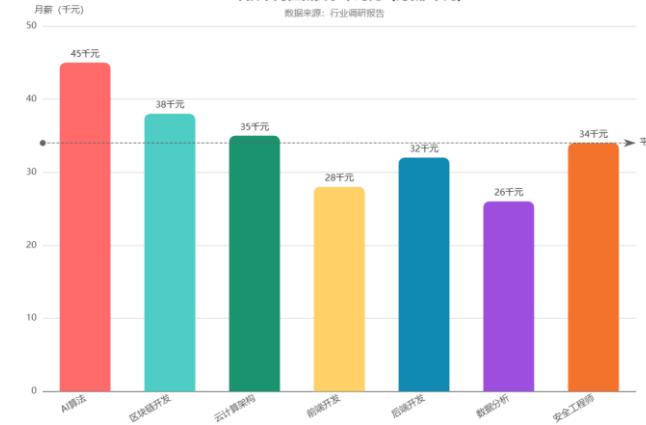
开源项目成员协作关系网络

节点大小=代码提交量 | 连线粗细=协作频率



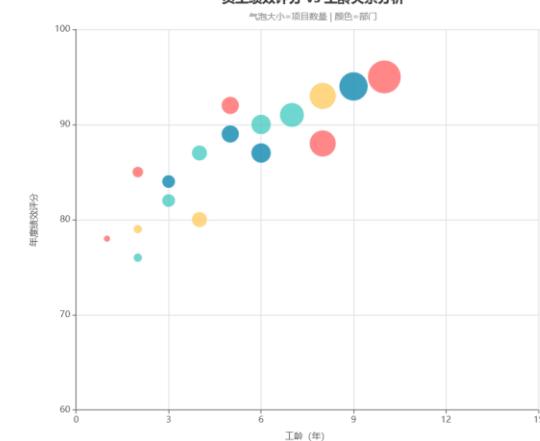
2025年技术岗位薪资水平对比 (月薪/千元)

数据来源：行业调研报告

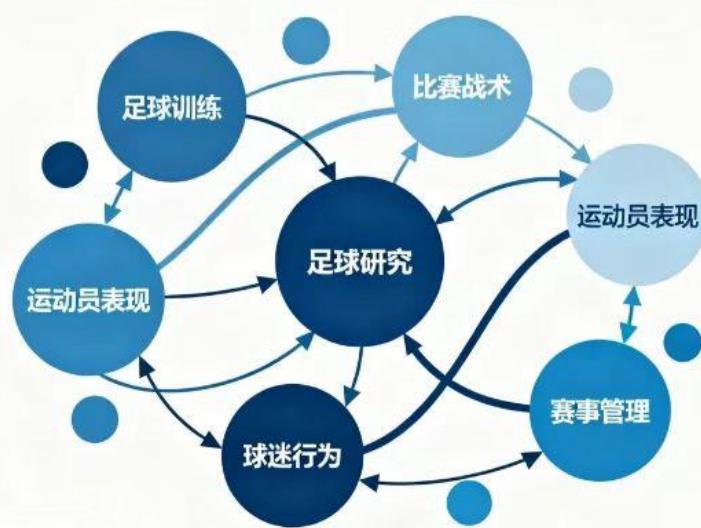


员工绩效评分 vs 工龄关系分析

气泡大小=项目数量 | 颜色=部门



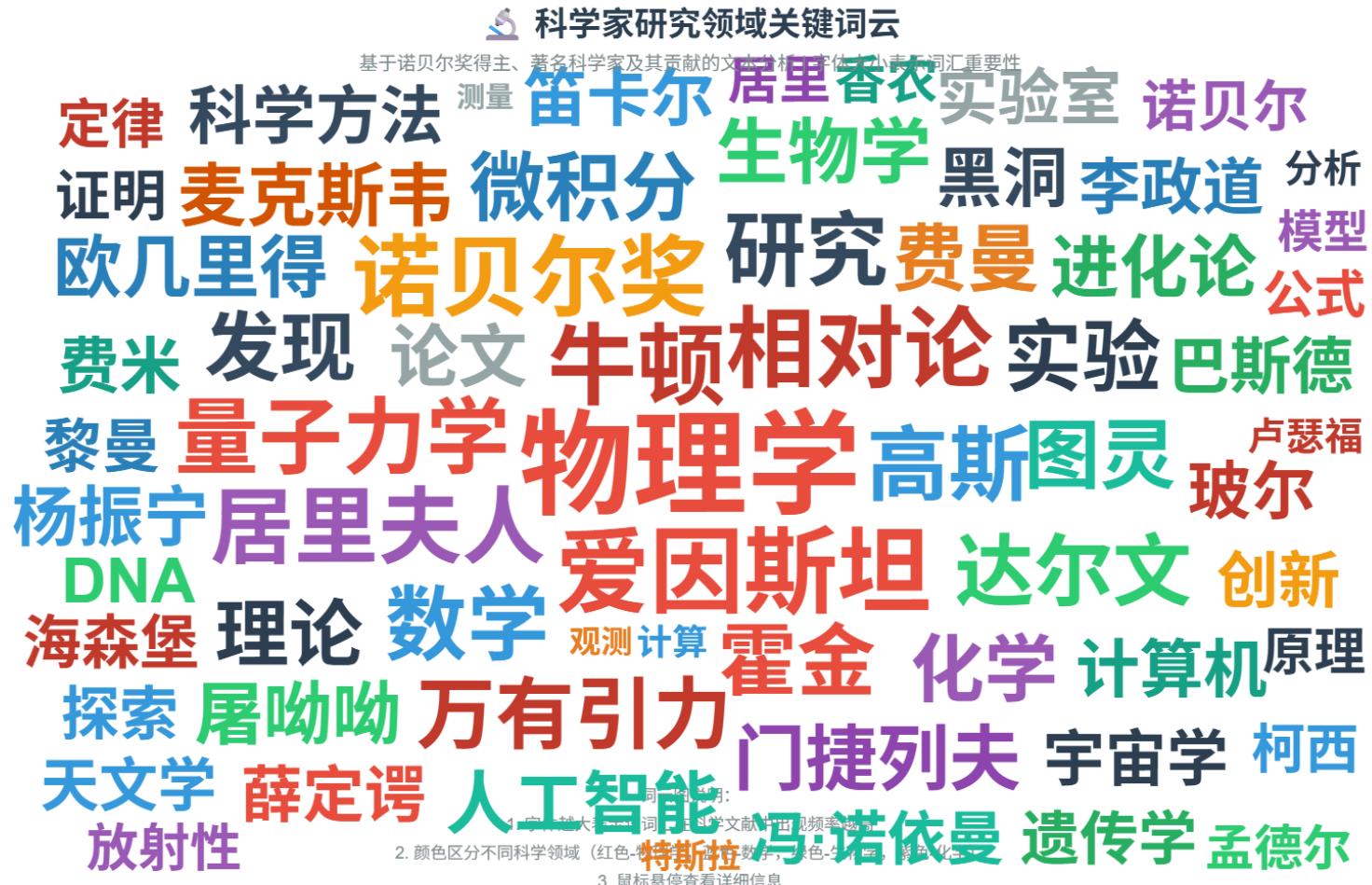
3. 使用Gehpi、VOSViewer、CiteSpace...其中任意一款工具，绘制任意你感兴趣的图谱1-2张。



| 足球运动研究领域关键词共现



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；

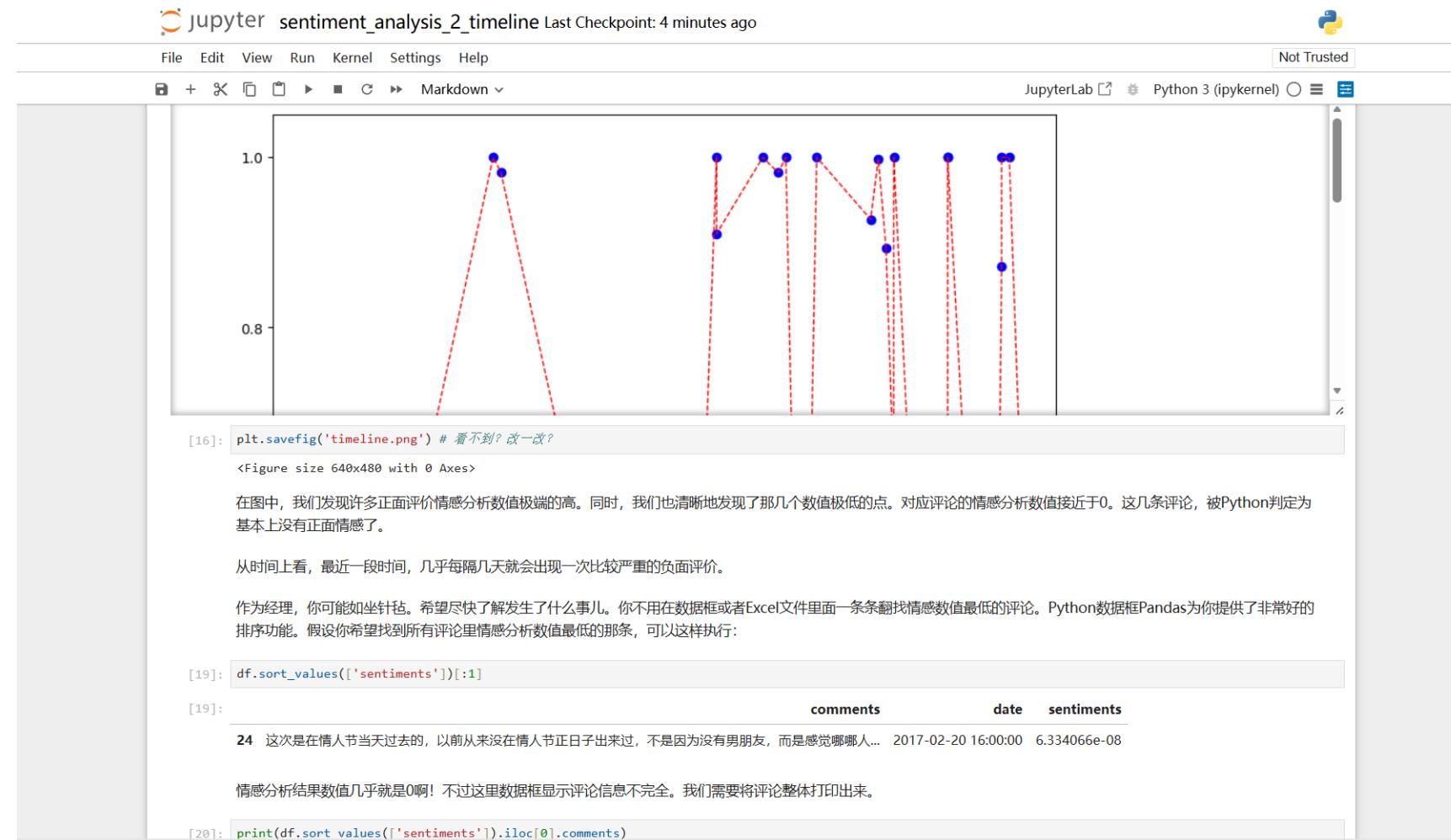


2. 完成sentiment_analysis_1-sentiment_analysis_4，4份代码。做截图，并简要做代码运行总结分析。

The screenshot shows a Jupyter Notebook interface with the title "jupyter sentiment_analysis_1_chuji Last Checkpoint: 3 minutes ago". The notebook has a "Not Trusted" status. The code cell [25] contains a text string about a不满的 phone. Cell [26] imports the SnowNLP library. Cell [27] prints the sentences from the text. Cell [28] prints the sentiment score of 0.889005139666256. Cell [29] is a comment. Cell [30] contains a long text string. Cell [31] imports the SnowNLP library. Cell [32] prints the sentiment score of 5.707609422256343e-05. Cell [33] is a comment. Cell [34] is a question. Cell [37] is a comment. Cell [38] is a comment. Cell [39] is a comment.

```
[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]: 5.707609422256343e-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？
[37]: # 例句1
[ ]:
[38]: # 例句2
[ ]:
[39]: # 查看原博客的地址，了解下有时候分析结果不准确的原因。
```

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。



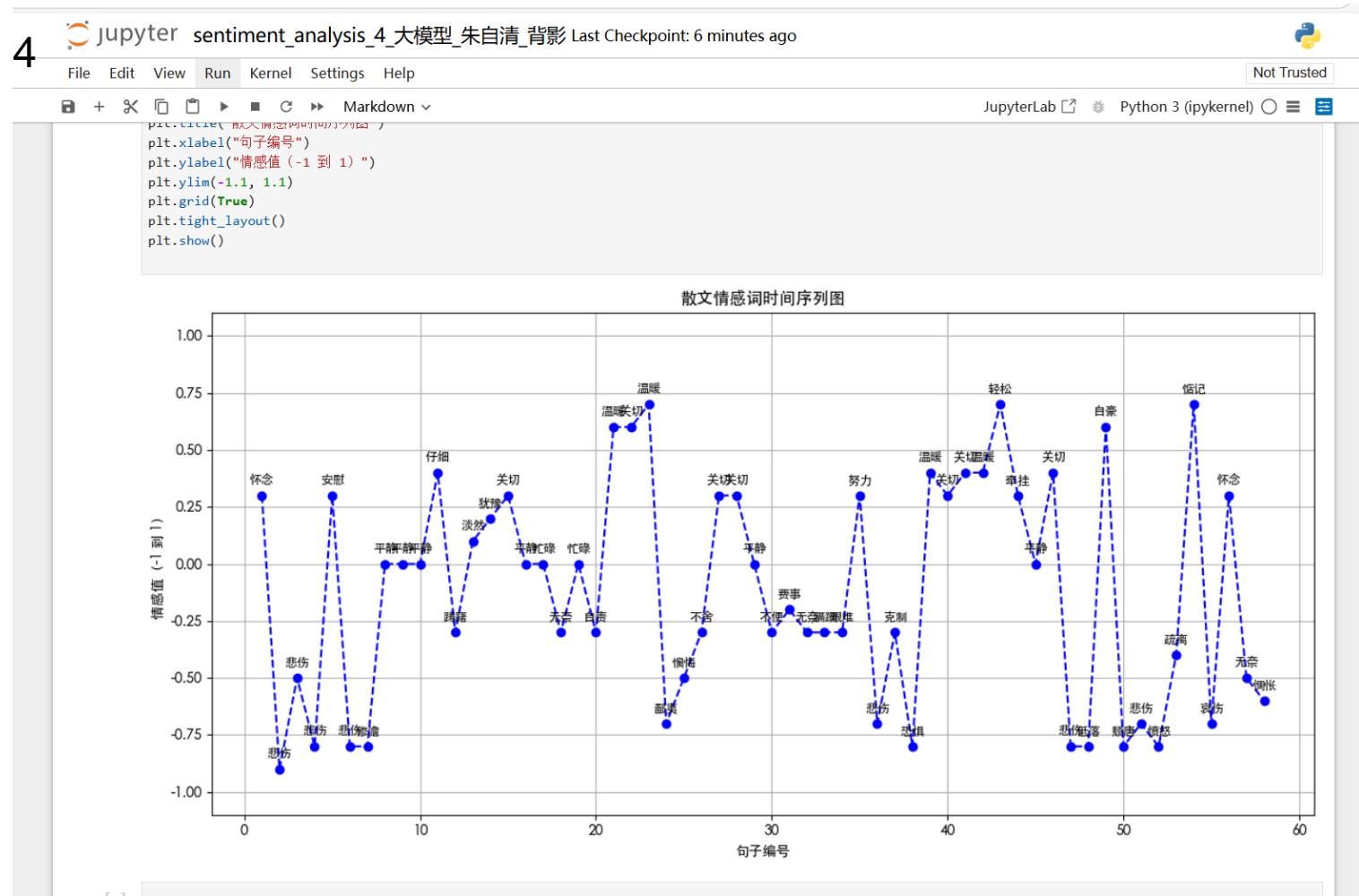
2.完成sentiment_analysis_1-sentiment_analysis_4，4份代码。做截图，并简要做代码运行总结分析。

The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** jupyter sentiment_analysis_3_大模型_健康文本细粒度情感抽取 Last Checkpoint: 5 minutes ago
- Kernel:** Python 3 (ipykernel)
- Cell Number:** 3
- Cell Type:** Markdown
- Content:** Displays a JSON object representing fine-grained entity extraction 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": "无具体描述"
 }
]
}
```\n
```

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。



第五讲 新媒体数据分析

- 略了...

第六讲 知识图谱理念

1. 实际产业案例分析：“阿里商品大脑”

阿里商品大脑：AI驱动的知识图谱新 纪元

最新进展、案例分析与未来展望



日期: 2025年12月31日

核心升级与技术深潜

重构前：挑战重重

- 商品信息杂乱无章，缺乏统一标准
- 人工录入错误率高，信息准确性差
- 严重影响用户体验与智能推荐效率

重构后：焕然一新

- AI Agent 自动学习与撰写商品信息
- 构建统一、规范、高质量的新一代商品库，并引入创新的LLM-KERec模型
- 为后续AI应用奠定坚实数据基础

LLM-KERec模型核心

实体抽取器：精准抽取商品、用户等实体。

互补图构建：利用大语言模型推理实体间互补关系。

E-E-I权重决策模型：实现“千人千面”的个性化推荐。

推荐点击率提升

10%

总结、展望与实战应用

我的评价



优势

AI与知识图谱深度融合，构建强大竞争优势，显著提升用户体验与商业效率。



挑战

模型的复杂性和数据安全是未来需要持续关注和解决的核心问题。

未来趋势



更深度的AI融合

AI将大规模应用于商品内容生成、营销文案创作等。



多模态交互

融合文本、图像、语音，提供更自然的购物体验。

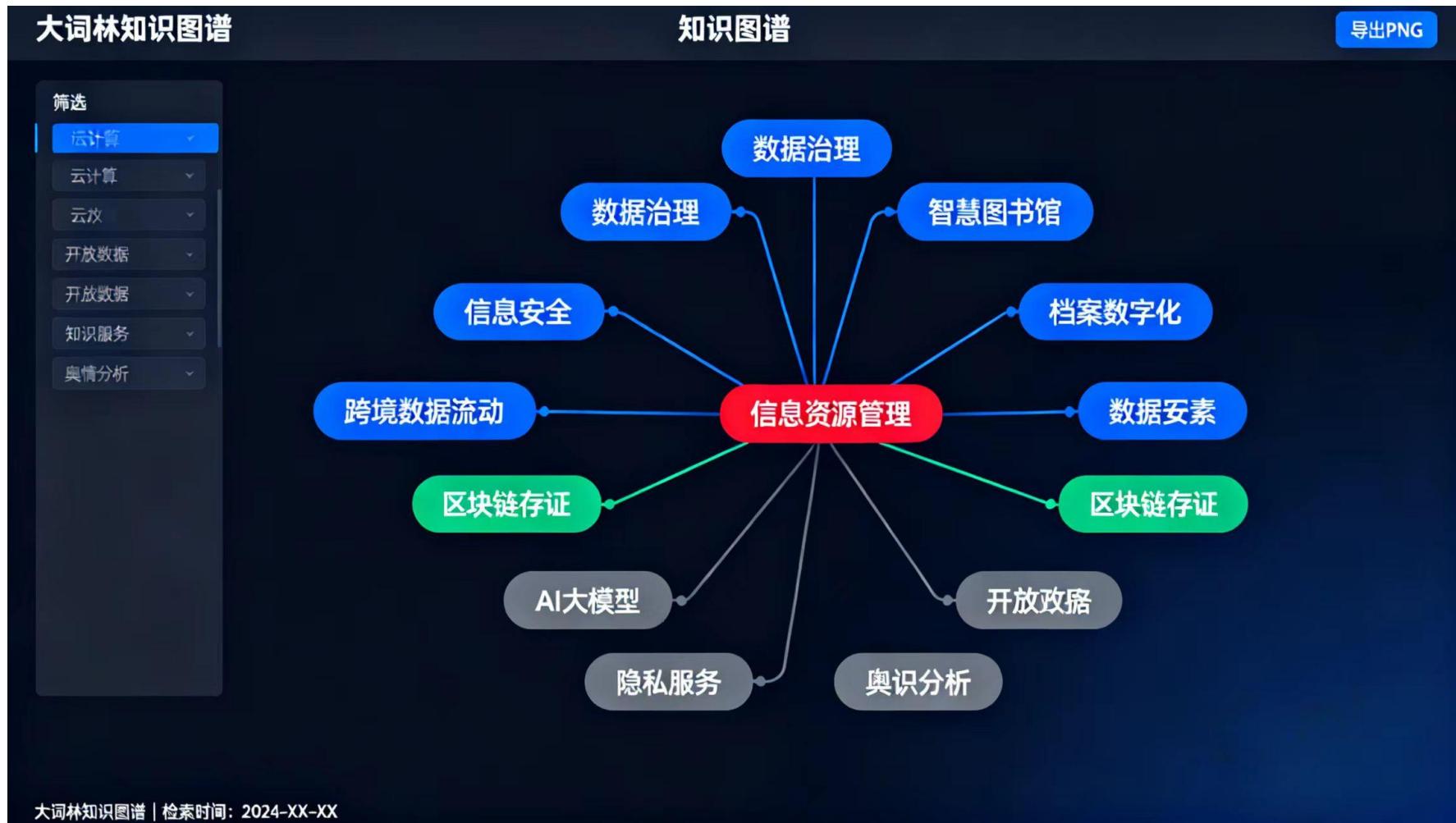


个性化与普惠

实现极致个性化推荐，赋能更多中小商家。

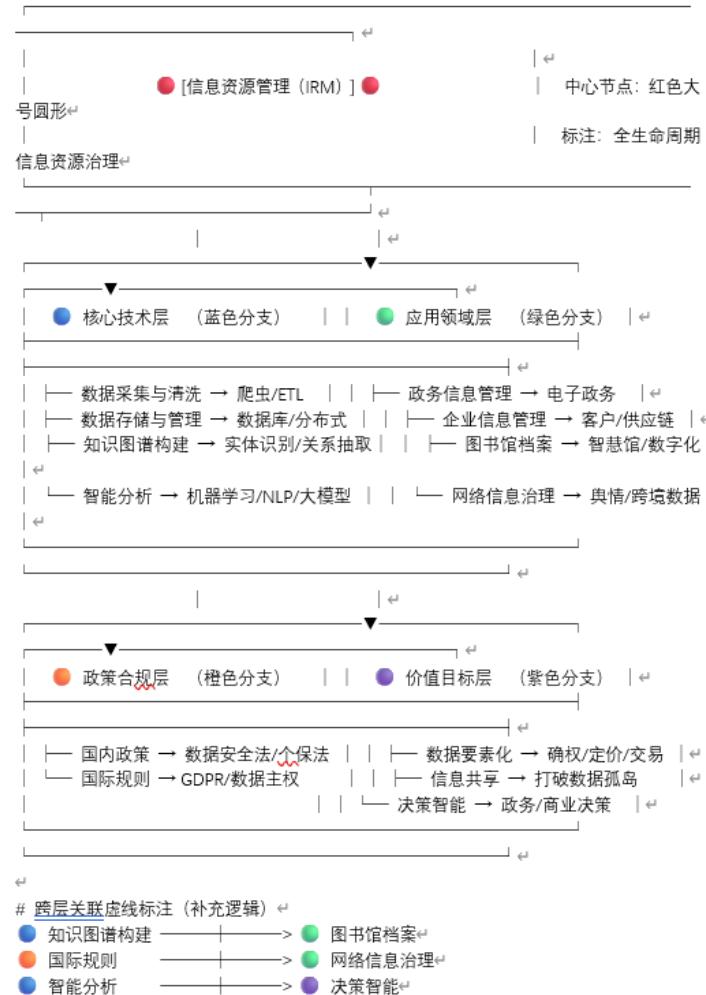
这次升级不仅是技术突破，更是阿里电商向AI全面转型的重要信号。

(2) 1. 使用PPT中知识图谱链接平台，检索、截图（大词林等，可用的）；

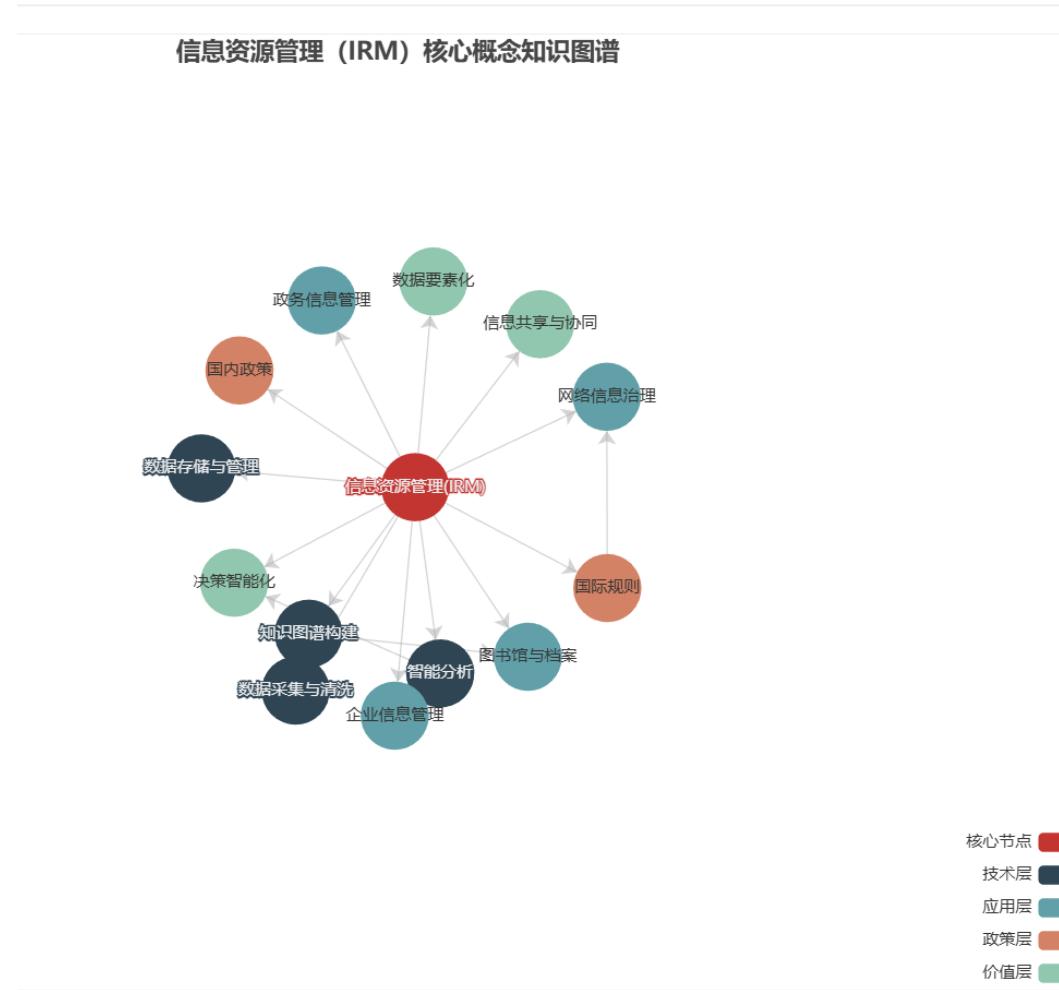


2. 使用白板建模绘制一个你感兴趣的“知识图谱”，可以是人物关系，也可以是事物关系，或者概念之间的关系等等，并解释你绘制的图谱；

这份以信息资源管理（IRM）为核心的“知识图谱”，构建了“技术支撑 - 场景应用 - 政策约束 - 价值导向”的四层闭环逻辑，其中核心技术层涵盖数据采集与清洗、存储与管理、知识图谱构建、智能分析等底层工具，是IRM落地的基础支撑；应用领域层包含政务信息管理、企业信息管理、图书馆与档案管理、网络信息治理四大核心场景，明确了IRM的实践载体；政策合规层以国内政策法规和国际规则标准为约束边界，划定了IRM的合法合规范围；价值目标层聚焦数据要素化、信息共享协同、决策智能化三大方向，指明了IRM的最终意义，同时图谱通过“知识图谱构建→图书馆档案”“国际规则→网络信息治理”“智能分析→决策智能”等跨层关联，进一步体现出技术赋能应用、政策规范场景、技术服务价值的动态联动关系，完整呈现了IRM从“怎么做”到“做什么”“凭什么做”“为什么做”的全链路逻辑。



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）。

