

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NLPIR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 screenshot shows a Jupyter Notebook interface. At the top, there's a toolbar with icons for file operations, a plus sign for new cells, and other options. To the right of the toolbar is a "Trusted" badge and a Python 3 (ipykernel) button. The main area contains a code cell with the following content:

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

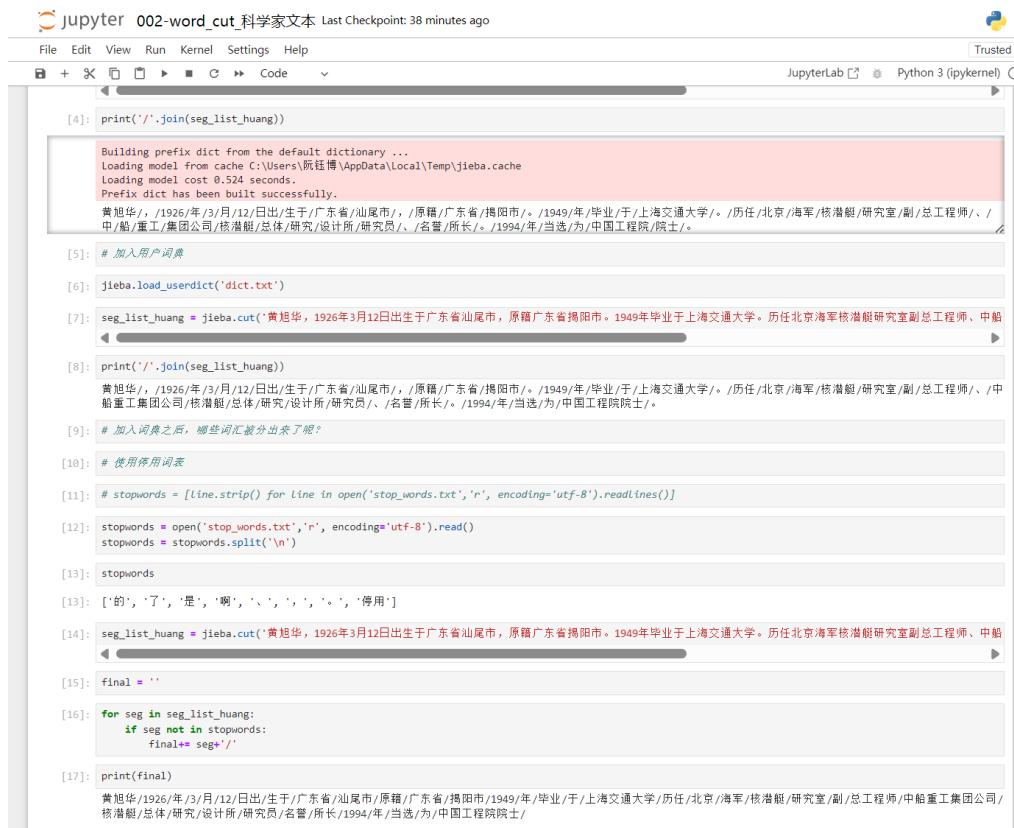
The screenshot shows a Jupyter Notebook interface with the title "jupyter 001-word_cut_基本分词 Last Checkpoint: 36 minutes ago". The notebook contains four code cells, each demonstrating a different aspect of word segmentation:

- 1. 基本分词**: This cell imports the `jieba` module and uses it to segment a Chinese sentence into words. The output shows the original sentence followed by its segmented version.
- 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**: This cell loads a user dictionary and segments the sentence again, successfully splitting the term "长短期记忆网络" into its components.
- 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”、“等虚词**: This cell reads stop words from a file and performs segmentation, resulting in a list where common stop words like "的" and "等" are removed.
- 4. 对于第四个片段，进行分词结果的过滤**: This cell adds a filtering step to the fourth snippet's code, which is described as "filtering the segmented results".

The code cells are numbered [32] through [44]. The output of each cell is displayed below the code, showing the results of the word segmentation process.

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）。



The screenshot shows a Jupyter Notebook interface with the title "jupyter 002-word_cut_科学家文本 Last Checkpoint: 38 minutes ago". The notebook contains the following Python code:

```
[4]: print('/'.join(seg_list_huang))

Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\阮钰博\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。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 = ''

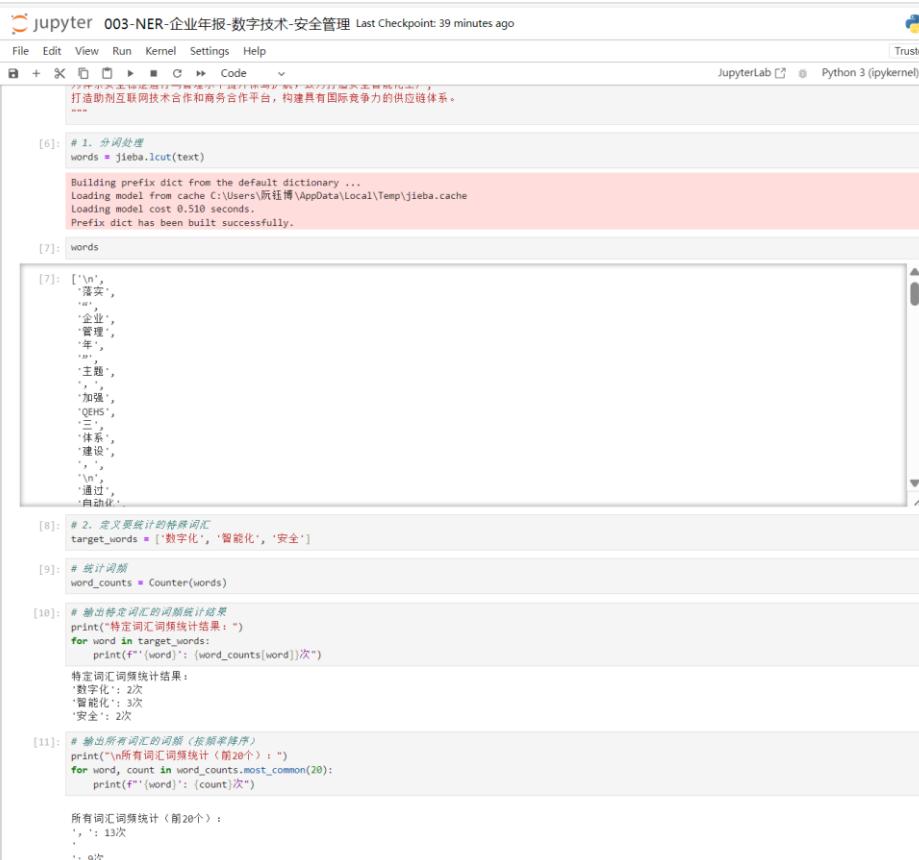
[16]: for seg in seg_list_huang:
    if seg not in stopwords:
        final+= seg+'

[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/
```

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）。



The screenshot shows a Jupyter Notebook interface with the title "jupyter 003-NER-企业年报-数字技术-安全管理 Last Checkpoint: 39 minutes ago". The notebook has four cells:

- Cell [6]:

```
# 1. 分词处理
words = jieba.lcut(text)

Building prefix dict from the default dictionary ...
Loading model from cache C:/Users/阮征博/AppData/Local TempData/jieba.cache
Loading model cost 0.510 seconds.
Prefix dict has been built successfully.
```
- Cell [7]:

```
words
```

Output:

```
['\n', '，', '企业', '管理', '，', '，', '主题', '，', '加强', 'QHSE', '三', '体系', '建设', '，', '，', '通过', '自动化', '']
```
- Cell [8]:

```
# 2. 定义要统计的高频词汇
target_words = ['数字化', '智能化', '安全']
```
- Cell [9]:

```
# 统计词频
word_counts = Counter(words)
```
- Cell [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')

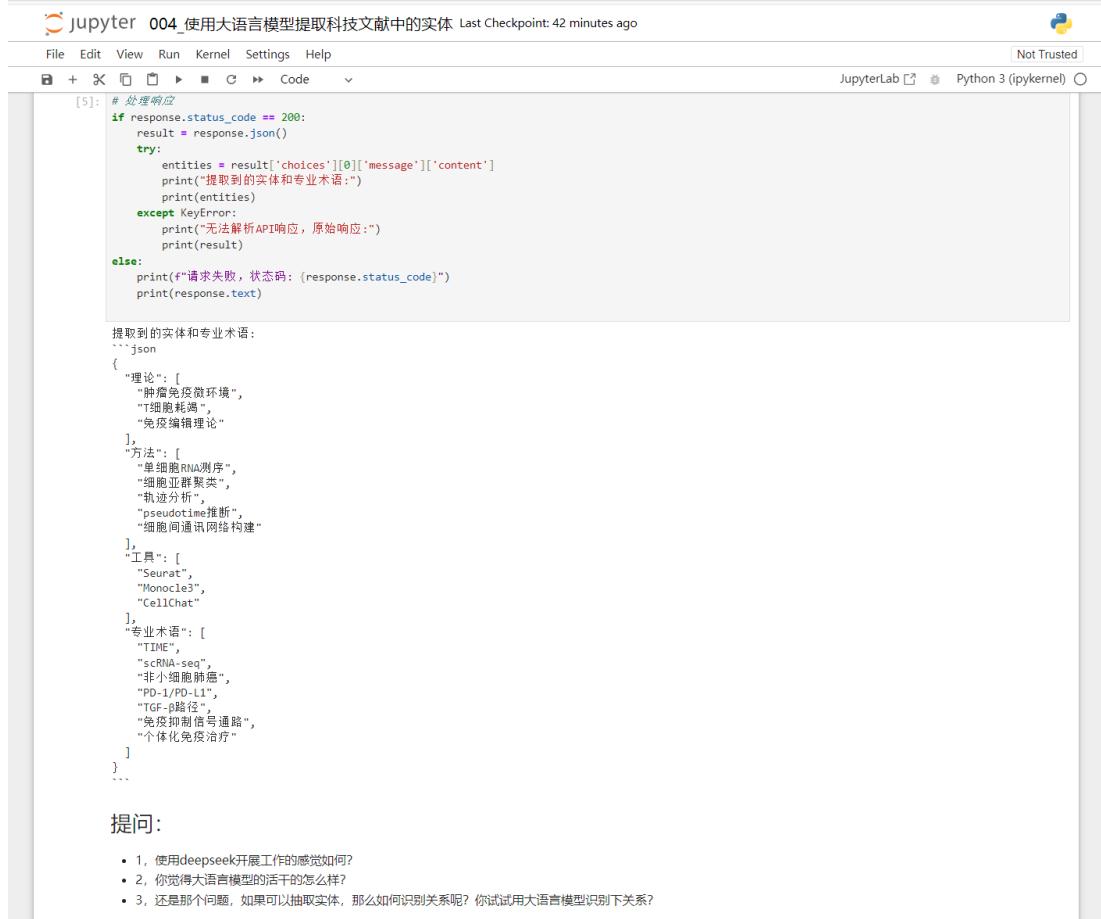
特定词汇词频统计结果：
数字化：2次
智能化：3次
安全：2次
```
- Cell [11]:

```
# 输出所有词汇的词频（按频率降序）
print('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
```

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 004_使用大语言模型提取科技文献中的实体 Last Checkpoint: 42 minutes ago
File Edit View Run Kernel Settings Help
Not Trusted
JupyterLab Python 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("请求失败, 状态码: " + str(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": [
 "TTIME",
 "scRNA-seq",
 "非小细胞肺癌",
 "PD-1/PD-L1",
 "TGF-β路径",
 "免疫抑制信号通路",
 "个性化免疫治疗"
]
}
```
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干活的怎么样?
- 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）

基于关键词的学术文本聚类集成研究综述

一、研究背景与意义

随着学术论文数量的快速增长，传统的手工分类方法已无法满足学术研究和信息传播的需求。因此，自动化的学术文本分类技术（如 ACM 的 CCS2012 分类）应运而生。本文将探讨基于关键词的学术文本聚类方法，并在通过聚类或文本特征分析来提升学术文本聚类性能。

二、核心问题与方法

1. 核心研究问题

- 聚类是否能提升学术文本聚类性能？
- 关键词抽取出方法对聚类效果有影响吗？
- 关键词数量对聚类效果有影响吗？

2. 模型框架

研究模型分为三个阶段（图 1）：

1. 数据准备：采用 ACM 数据集（8 个子集，共 3504 篇论文，涉及 40 个领域）。
2. 关键词抽取：使用两种无监督方法（TF-IDF, CS, ECC, ~~SVD~~），采集 6-50 个关键词。
3. 文本聚类与综合度计算：基于向量空间模型和余弦相似度。

3. 聚类方法

- ECCM：以 k-means 为基础，通过 Bootstrap 生成簇质心，结合迭代和投票策略。
- EDC：以层次聚类为基础，通过参数调整生成簇质心。
- 性能评估：以 F1 作为主要指标，结合准确率和召回率。

4. 实验结果分析

(1) 聚类方法对聚类效果的影响

- ECCM 和 EDC 的聚类效果与单一聚类方法（k-means 与 ECCM 的 T 统计结果： $t=1.02, P=0.05$ ，表明聚类与 EDC 的 T 值显著； $t=1.82, P<0.0001$ ，表明聚类效果大于 EDC）。
- 在学系类别较大的数据集（如 ACM 分类）中，聚类集成优势更明显（见表 2），因其能同时利用学系的关键词来聚类。

(2) 关键词抽取方法的影响

- Testbook 和 EDC 在 EDC 和 ECC 中平均 F1 是相同的（见图 2a-D）。
- CS 性能最好，但其结果与学术文本主题并不一致。
- 聚类集成方法（如 ECCM）对不同关键词抽取方法鲁棒性更强，性能波动小。

5. 研究结论

1. 聚类集成能有效提升学术文本聚类性能，尤其在学科类别大或关键词数量少时效果佳。

2. Testbook 是最优关键词抽取方法，而 CS 需要使用。

3. 增加关键词数量有助于改善聚类质量，建议在实际应用中尽可能抽取更多关键词（如 40 个）。

四、结论与展望

1. 面向多语种扩展：需在中文等多语言学术文本上验证方法普适性。
2. 多模态融合结合引文、作者信息等特征进一步提升性能。
3. 深度学习应用：探索基于神经网络的关键词抽取与聚类集成结合方案。

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。

一、研究背景与意义

- 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。^④
- 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。^⑤
- 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。^⑥

二、数据来源与研究方法

- 数据来源^⑦
 - 中国知网（CNKI）期刊全文数据库^⑧
 - 检索时间范围：2014年1月—2024年12月^⑨
 - 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）^⑩
 - 文献类型：学术期刊论文为主，辅以学位论文、会议论文^⑪
- 研究方法^⑫
 - 计量分析：年度发文量、机构分布、作者合作网络、期刊分布^⑬
 - 主题分析：高频关键词统计、共词分析、主题聚类（如LDA模型）^⑭
 - 趋势分析：基于时间线的主题热度变化、突现词检测（Burst Detection）^⑮
 - 可视化：时间线图、热力图、主题演化路径图^⑯

三、总体发文趋势（2014-2024）

- 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。^⑰
- 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。^⑱
- 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

成为新的研究热点，IRM 与应急信息管理交叉明显。^⑲

- 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。^⑳

四、主题变化分析

1. 早期阶段（2014-2016）

- 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^㉑
- 研究特点：以理论探讨和系统建设为主，技术应用相对有限。^㉒

2. 发展阶段（2017-2019）

- 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^㉓
- 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。^㉔

3. 爆发阶段（2020-2022）

- 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^㉕
- 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。^㉖

4. 前沿阶段（2023-2024）

- 新增高频词：人工智能、大模型、AIGC、数据安全、隐私保护、区块链^㉗
- 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。^㉘

五、学科交叉与热点演化

- 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。^㉙
- 热点演化路径：^㉚

1. 信息资源建设 → 数据治理 → 智能管理 → 安全与合规^㉛

2. 传统图书馆 → 数字图书馆 → 智慧图书馆 → 全域知识服务平台^㉜

新等多个领域。^㉝

- 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。^㉞

- 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。^㉟

- 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。^㉟

七、未来研究展望

- 智能化管理：基于大模型的自动化信息组织与知识发现^㉟
- 跨域融合：IRM 与智慧城市、数字政府建设的深度融合^㉟
- 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^㉟
- 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^㉟

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；

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

- Title Bar:** jupyter tf1_full_text_sanguo Last Checkpoint: 4 minutes ago
- Kernel:** Python 3 (ipykernel)
- Code Cells:**
 - [11]: A list of words from a set, including '细眼長', '賴以', '九尺', '曾', '設壘', '厚待', '允以', '孤陋', '情', and '進出'.
 - [12]: A for loop that iterates over words in articleSet. It checks if the length of the word is less than or equal to 1. If true, it compares the word's count in words with its value in articleDict.
 - [13]: Prints the articleDict, which is a dictionary mapping words to their counts. The output includes '細眼長': 2, '賴以': 1, '九尺': 3, '設壘': 1, '厚待': 1, '允以': 1, '孤陋': 1, '進出': 1, '丁晉': 1, '可破群': 1, '假帝': 1, '不相容': 1, '佯大聲': 1, '晨構': 1, '侍郎': 1, '王益堅': 1, '北相': 2, '惟亦諫': 1, and '董卓'.
 - [14]: Sorts the items in articleDict by value in descending order using a lambda function.
 - [15]: Prints the top 100 words from the sorted list. The output shows the top words and their counts, starting with ('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), ('李備', 25), ('引兵', 25), and ('孟獲', 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；

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

- Kernel:** Python 3 (ipykernel)
- Cell 22:** Output: {'黄旭华': 59, '核潜艇': 32, '国立交通大学': 3}
- Cell 23:** # 定义 画图 函数
- Cell 24:** def make_chinese_plot_ready():
 from matplotlib import rcParams
 rcParams['font.family'] = 'Heiti TC' # mac笔记本电脑直接替换字体
 #rcParams['font.sans-serif'] = ['FangSong'] # 或者直接使用电脑有的字体 FangSong
 rcParams['axes.unicode_minus'] = False
- Cell 25:** # 定义 画图 函数
- Cell 26:** def draw_dict(mydict, figsize=(8, 5)):
 import pandas as pd
 import matplotlib.pyplot as plt
 make_chinese_plot_ready()
 df = pd.DataFrame(list(mydict.items()), columns=['name', 'times'])
 df.set_index('name')['times'].sort_values(ascending=False).plot(kind='bar', figsize=figsize) # 做好排序
 plt.tight_layout()
- Cell 27:** # %pylab inline
- Cell 28:** %matplotlib inline
- Cell 29:** draw_dict(terms_dict)
- Output (Figure):** A bar chart titled '作业' showing the frequency of terms. The y-axis ranges from 0 to 50. The x-axis lists terms. The chart has two bars: one for '黄旭华' at approximately 59 and one for '核潜艇' at approximately 32.
- Assignment Section:** A sidebar labeled '作业' contains the following bullet points:
 - 请完成以上两个python的代码
 - 请把以上文本替换成功文件夹中的《科学家博物馆-黄旭华传记序言.txt》，完成该文档中的全文词频统计，以及指定词汇统计，如统计“黄旭华”、“核潜艇”出现的次数。

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 tf2_the_three_kingdoms Last Checkpoint: 6 minutes ago

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

```
[38]: weapon_dict = {
    '月牙戟': 0,
    '七星刀': 1,
    '雙頭劍': 3,
    '松紋龍寶劍': 0,
    '孟德劍': 0,
    '思呂劍': 0,
    '飛雲三劍': 0,
    '文士劍': 0,
    '蜀八劍': 0,
    '鎮山劍': 0,
    '吳六劍': 0,
    '皇帝免王劍': 0,
    '日月刀': 1,
    '百辟寶刀': 0,
    '龍睛刀': 0,
    '百辟七首刀': 0,
    '銀牌': 1,
    '銅牌': 2,
    '四佛掛劍': 1,
    '雙雄戰': 2,
    '藍冕道劍': 0,
    '寶雞弓': 3,
    '鵝垂弓': 1,
    '虎筋蛇弓': 0,
    '兩石力之弓': 0,
    '手鍊': 0,
    '短鍊': 2,
    '飛石': 2,
    '流星錘': 3,
    '鋼鎚': 0
}
```

[39]: draw_dict(weapon_dict)

[40]: # 以上出的图看上去不清晰, 如何保存清晰的图?
import matplotlib.pyplot as plt
plt.savefig('weapon000.png')
<Figure size 640x480 with 0 Axes>

作业||

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”。

The screenshot shows a Jupyter Notebook interface with the following code in cell [1]:

```
[1]: import jieba  
from collections import Counter  
  
# 读取文本文件 (读操作 huangxhua.txt 读取操作同一目录, 路径相对路径)  
file_path = "科学家博物馆-黄旭华传记序言.txt" # 路径为您的文件路径  
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)  
  
# 打印前50个最常见的词  
print("前50个最常见的词如下:")
for word, freq in word_freq.most_common(50):
    print(f'{word}: {freq}')  
  
# 统计指定词, 并输出结果  
target_word = "黄旭华"
target_count = word_freq.get(target_word, 0)
print(f'<{target_word}>词频统计 ***')
print(f'<{target_word}>出现的次数: ({target_count})')  
  
Building profile dict from the default dictionary ...
Loading model from cache C:\Users\倪伟峰\AppData\Local\Temp\jieba.cache
Loading model cost 0.509 seconds.
Profile dict has been built successfully.
*** 文本处理完成: (约100) ***
的: 169
. : 135
. : 103
. : 61
黄旭华: 53
了: 42
. : 40
星海: 32
科学: 29
定: 23
等: 22
要求: 22
科学: 22
等: 21
对: 19
工作: 17
声: 17
研究: 15
. : 14
小话: 14
新华社: 13
. : 13
专业: 13
运行: 13
语言: 12
. : 12
特别: 12
类: 12
. : 12
. : 11
. : 11
工建: 11
. : 11
. : 10
. : 10
. : 10
. : 10
. : 9
. : 8
. : 8
. : 8
. : 8
. : 8
. : 8
. : 8
. : 7
. : 7
. : 7
. : 7
. : 7
. : 7
. : 6
设计: 6  
  
*** 词频统计 ***
<黄旭华>出现的次数: 53
```

第二讲 词频统计

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

- 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。

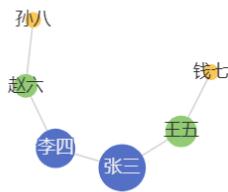
第三讲 词云与可视化

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

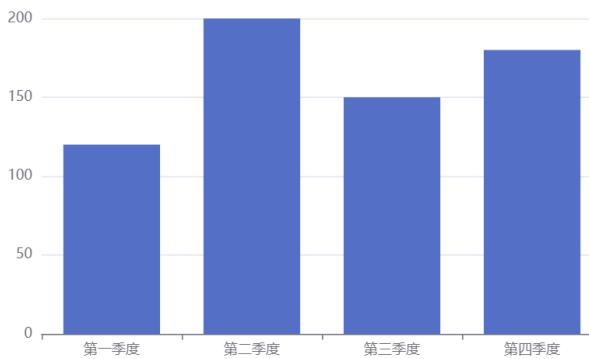
展示用户之间的互动关系

朋友 同事 家人



2023年季度销售额

单位：万元



腾讯

阿里巴巴

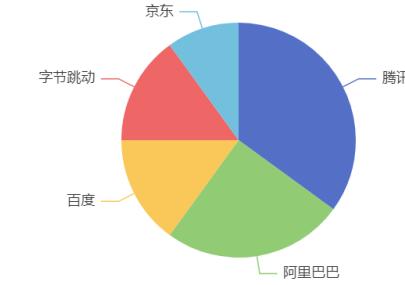
百度

字节跳动

京东

2023年市场份额

主要科技公司市场份额占比

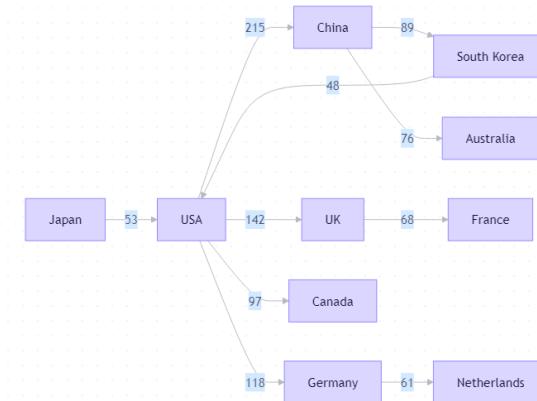
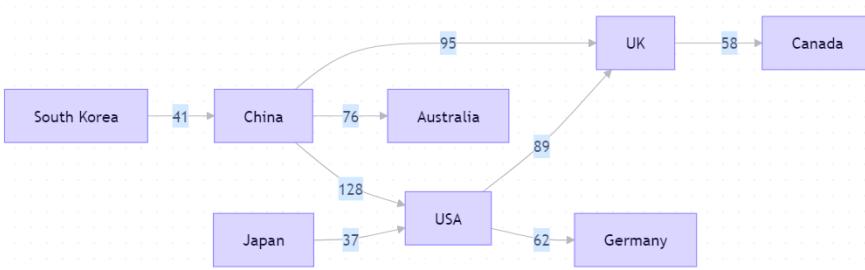


第三讲 词云与可视化

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

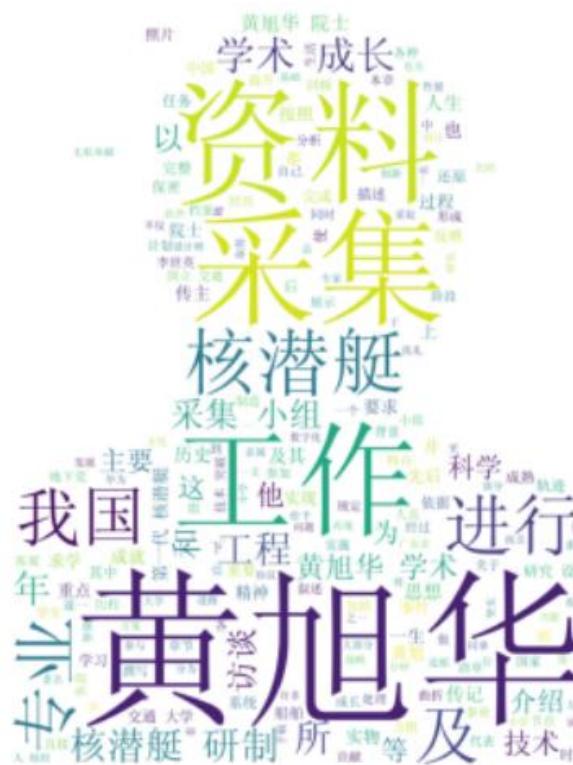
“碳中和”研究的国家合作网络图谱

“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 Jupyter Notebook interface with the title "sentiment_analysis_1_chuji". The notebook has a "Not Trusted" status. The code cell [22] contains a comment about testing on Taobao reviews. Cells [23] through [28] show the processing of a single review, resulting in a sentiment score of 0.999947261146611. Cells [29] through [34] show the processing of another review, resulting in a sentiment score of 0.889005139666256. The code uses the SnowNLP library to analyze text.

```
# 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"
taobao_1 = SnowNLP(text_taobao_1)
taobao_1.sentiments
0.999947261146611

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!"
taobao_2 = SnowNLP(text_taobao_2)
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

taobao_2.sentiments
0.889005139666256

# 以上的结果看上去是有问题的, 分析的不准确.

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:
taobao_3 = SnowNLP(text_taobao_3)
taobao_3.sentiments
5.707609422256343e-05

# 这一长句的结果还是可以的, 非常小的概率值了

# 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。

The screenshot shows a Jupyter Notebook interface with the title 'jupyter sentiment_analysis_2_timeline Last Checkpoint: 27 minutes ago'. The notebook has a 'Not Trusted' status. It displays a Markdown cell with text about sentiment analysis and a Python cell with code and its output.

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

| comments | date | sentiments |
|--|------|------------|
| 24 这次是在情人节当天过去的，以前从来没在情人节正日子出来过，不是因为没有男朋友，而是感觉哪人... 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)
```

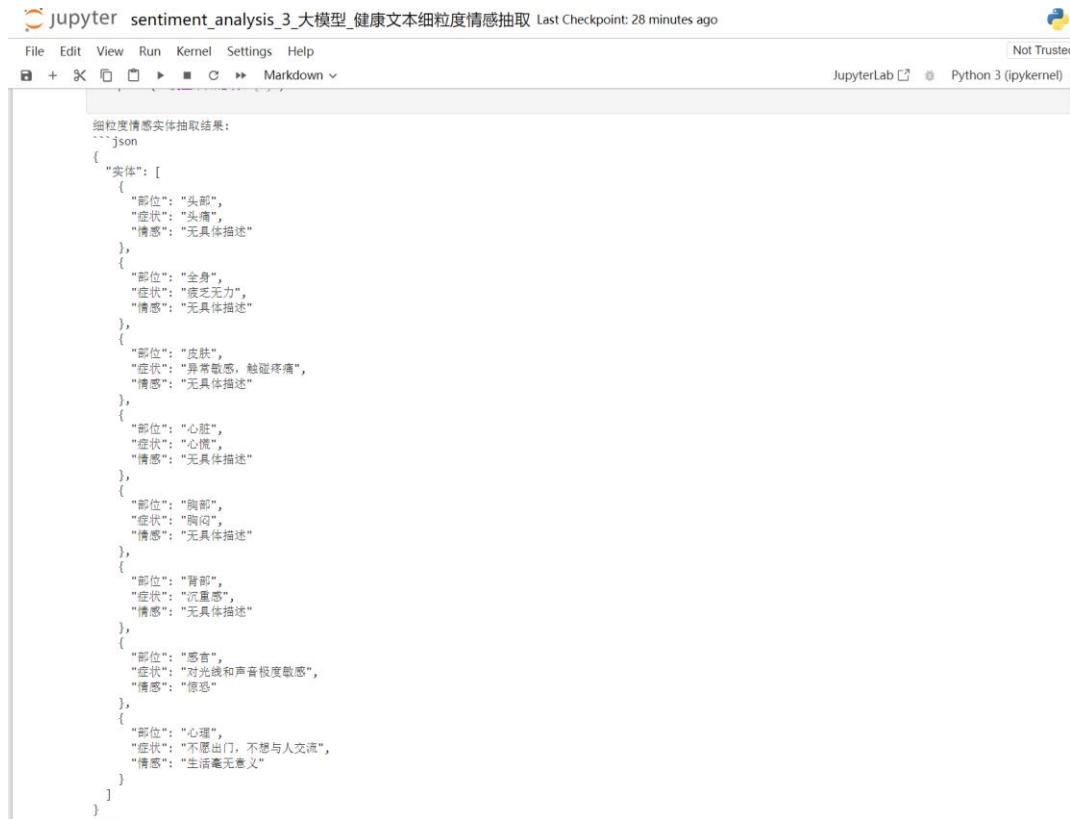
这次是在情人节当天过去的，以前从来没在情人节正日子出来过。不是因为没有男朋友，而是感觉哪人都多，所以特意错开，这次实在是A餐厅了，所以赶在正日子也出来了。从下午四点多的时候我就看到一百多了，我从家开车过去得堵的话一个小时，我一看提前两个小时就在网上先排着号了，差不多我们是六点半到的，到那的时候我看号码前面还有才三十多号，我想着肯定没问题了。等一会就能吃上的，没想到悲剧了，就从我们到那坐到那位区开始，大约是十分二十分一叫号，中途多次我都想走了，哈哈，哎，等到最后早上九点才吃上的，服务员感觉也没以前清淡时候到了，不过这肯定的，一人负责好几桌，今天节假日这么多人，肯定是很累的，所以大多也就是我自己跑腿，没让服务员给弄太多，进店让服务员下的，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，而且当次就可以使用，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，要不然别赶节日去，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，不仅要看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。



jupyter sentiment_analysis_3_大模型_健康文本细粒度情感抽取 Last Checkpoint: 28 minutes ago

File Edit View Run Kernel Settings Help

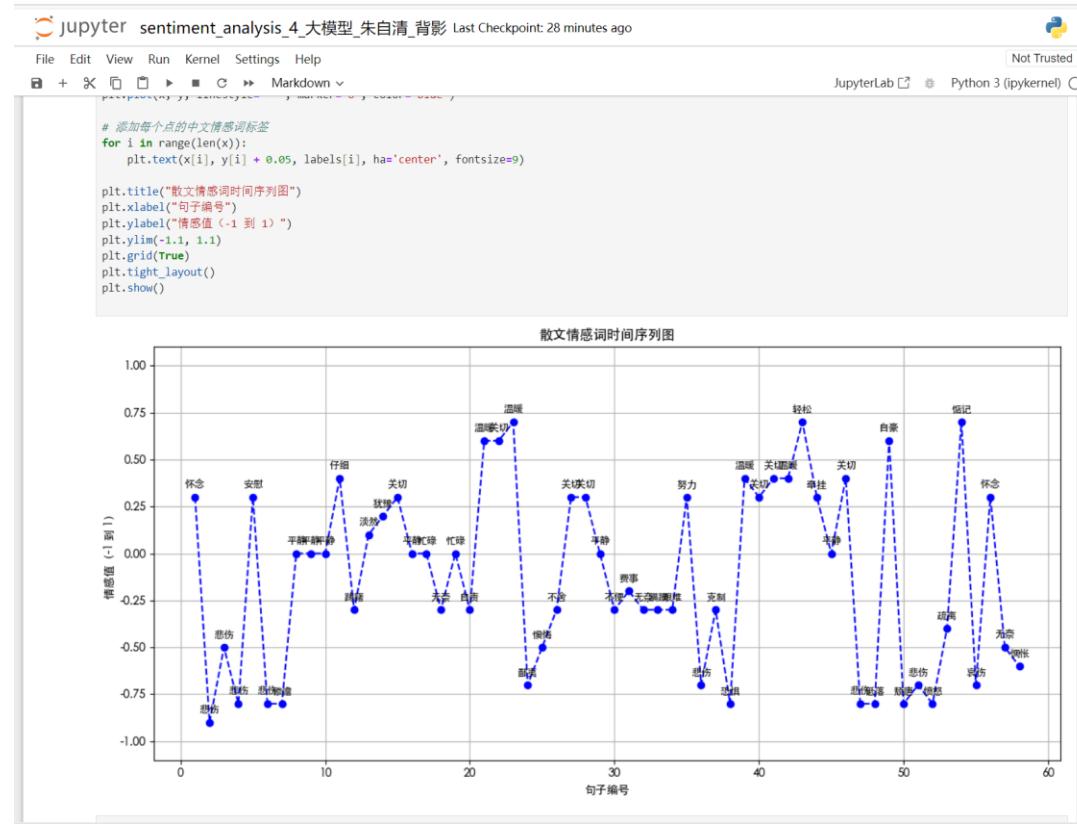
Not Trusted

JupyterLab Python 3 (ipykernel)

```
細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": "疲惫无力",
 "情感": "无具体描述"
 },
 {
 "部位": "皮肤",
 "症状": "异常敏感, 躺睡疼痛",
 "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、<sup>1</sup>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）；



## 可视化



通过Neo4j Bloom，通过可视化的力量沉浸于数据之中。通过丰富且友好的用户界面，深入了解Neo4j图数据中实体内外的每一个连接、关系和属性。

## 探索

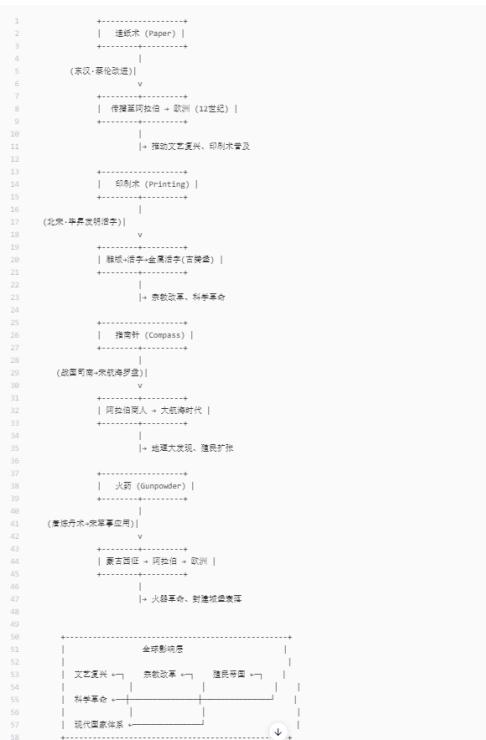
通过Neo4j Bloom，调查并探索推动组织内有意义成果的数据。Bloom易用的图解设计和强大的图表分析，帮助勾勒出数据所传达的美丽画面。

## 协作

Neo4j Bloom通过无代码搜索到故事的设计提升了跨团队协作。使用基于密码的自定义搜索短语和近自然语言搜索功能简化复杂查询。定义、保存并分享图表视角，并为每个视图分配基于角色的访问权限。

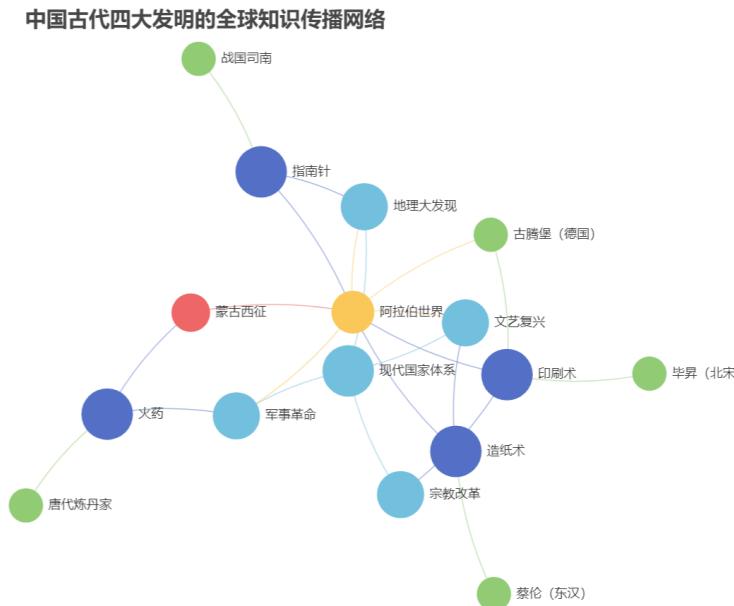
# 第六讲（2） 知识图谱工具

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

