南京工業大學
NANJING TECH UNIVERSITY

用户数据采集与关联分析

（结课作业）

冯世欣

信管2301

202321054001

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.

2.

```
[1]: print("hello world.Hello 冯世欣.")
```

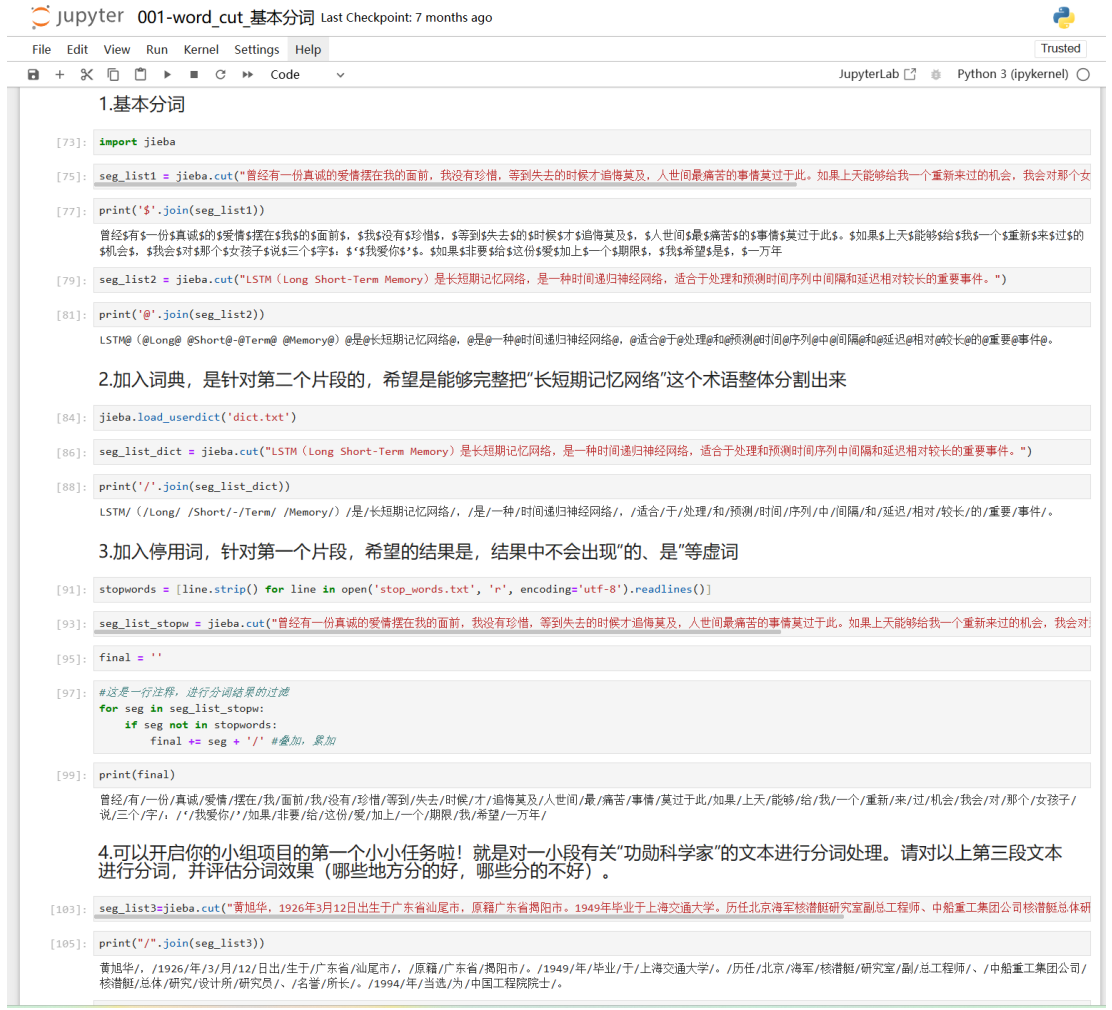
```
hello world.Hello 冯世欣.
```

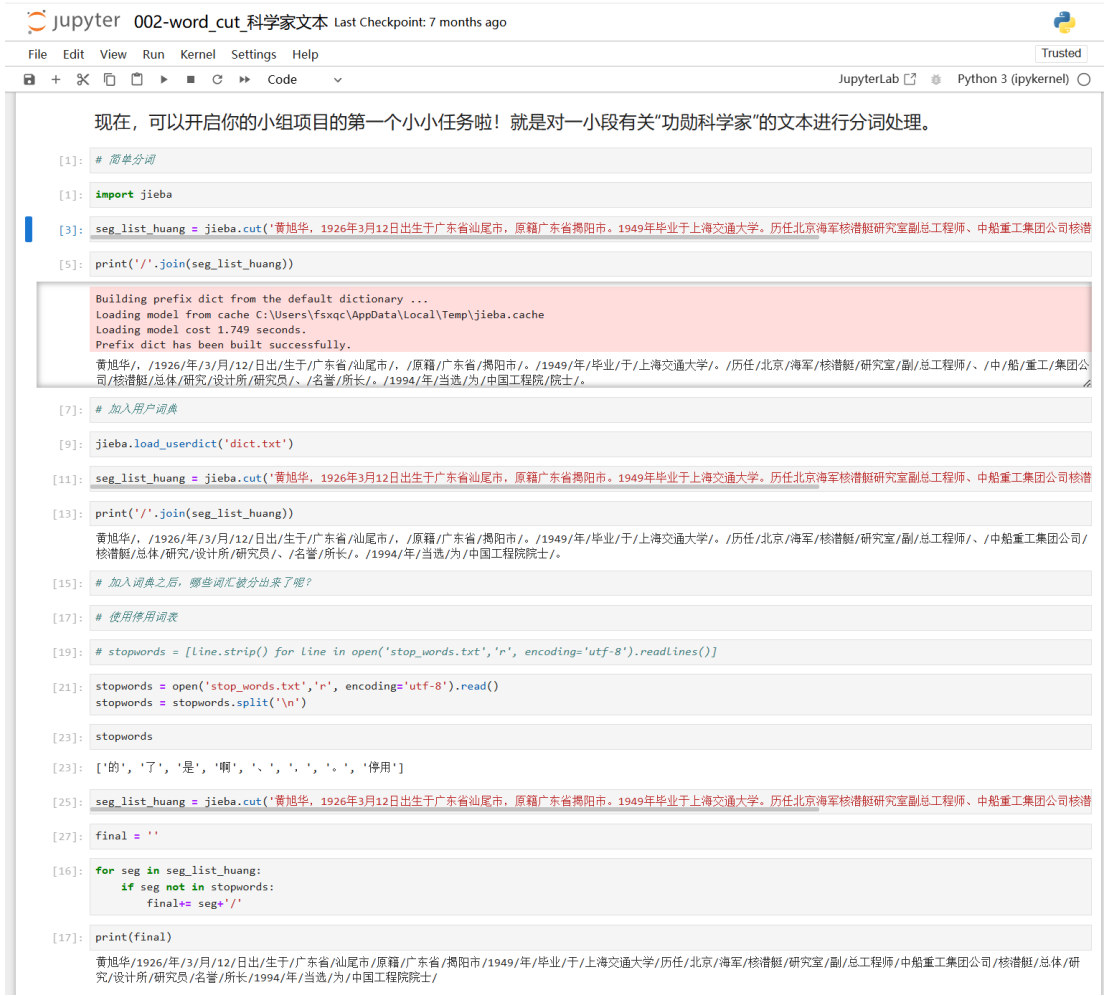
```
[3]: print("谢谢吴老师，教我们学习python")
```

```
谢谢吴老师，教我们学习python
```

第一讲 课程导言与分词

3.





第一讲 课程导言与分词

3.

jupyter 003-NER-企业年报-数字技术-安全管理 Last Checkpoint: 8 months ago

File Edit View Run Kernel Settings Help Trusted

Markdown JupyterLab Python 3 (ipykernel)

```
    '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',
    '。',
    '\n']
```

[9]:
```
# 2. 定义要统计的特殊词汇
target_words = ['数字化', '智能化', '安全']
```

[11]:
```
# 统计词频
word_counts = Counter(words)
```

[13]:
```
# 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
```

jupyter 004_使用大语言模型提取科技文献中的实体 Last Checkpoint: 7 months ago

File Edit View Run Kernel Settings Help Trusted

Markdown JupyterLab Python 3 (ipykernel)

网页版免费对话 ↗

[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 = "sk-fc8a8709232740a9a3b4467faea9fe29"  #直接复制过来
```

[3]:
```
# 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}
"""
```

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

[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)
```

第一讲 课程导言与分词

4. **关于《基于学术论文全文的研究方法实体自动识别研究》的阅读总结**

本研究旨在解决学术文献中研究方法难以被系统性量化分析的问题。作者指出，研究方法的规范性是学科成熟度的重要标志。为此，本文提出了从学术论文全文中自动识别两种研究方法实体的任务：即区分“论文使用方法”（本文实际采用的方法）与“论文引用方法”（本文提及或对比的方法），以期自动化地梳理学科内研究方法的应用与演进模式。

为实现此目标，研究构建了专门的标注数据集，并系统比较了八种基于深度学习的序列标注模型。实验深入探索了不同技术组合，包括词向量（通用、领域、字向量）与模型结构（BiLSTM、BiLSTM+CRF）。最终，结合字向量、BiLSTM和CRF的混合模型被证明为最优方案，其性能显著超越传统CRF模型，验证了深度学习在此任务上的有效性。

获得高性能模型后，作者将其应用于《情报学报》近十年论文，进行了大规模实证分析。统计结果揭示了一个鲜明现象：在情报学领域，“实验法”在研究方法的使用与引用上均占据绝对主导地位（占比超过80%）。同时，高频词分析也直观反映了该学科偏向量化与计算模型的现状。

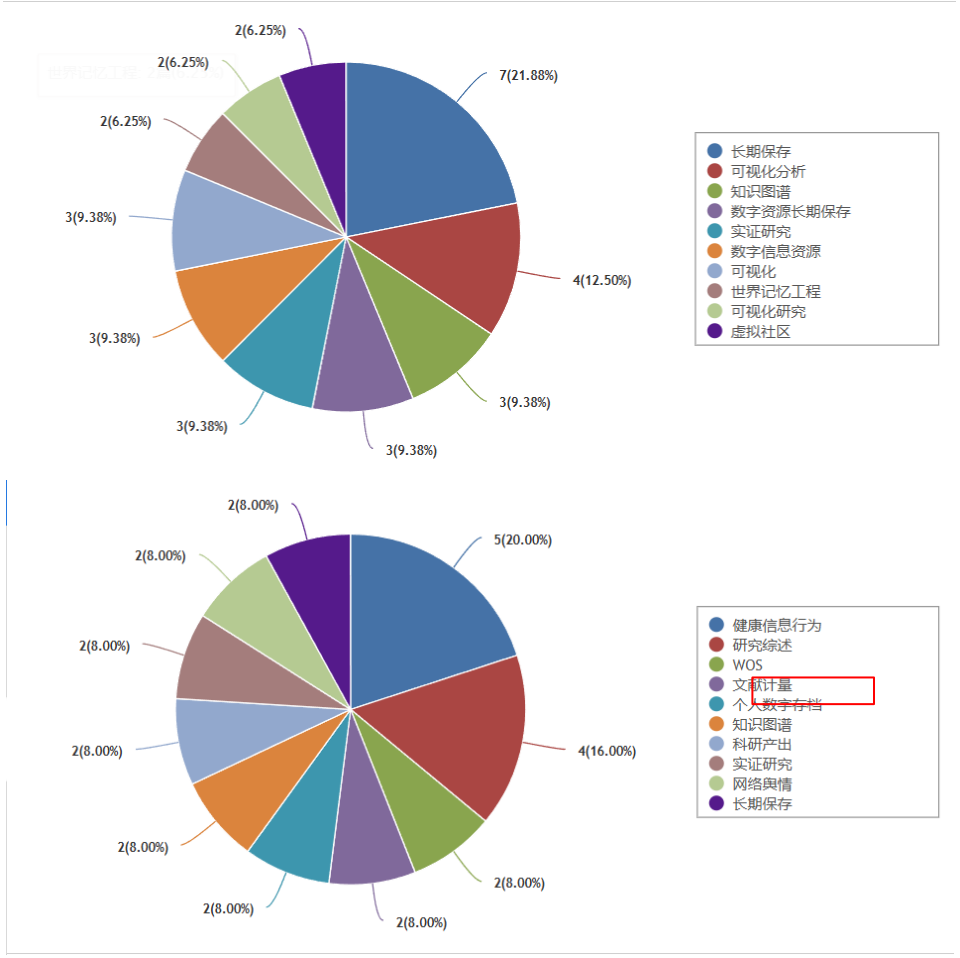
总结而言，本研究不仅验证了自动识别研究方法的技术路径，还揭示了重要的学科现象。文章最后展望未来，指出可通过引入半监督学习与规则方法，来克服当前在标注数据与识别性能上的局限，从而进一步深化研究。

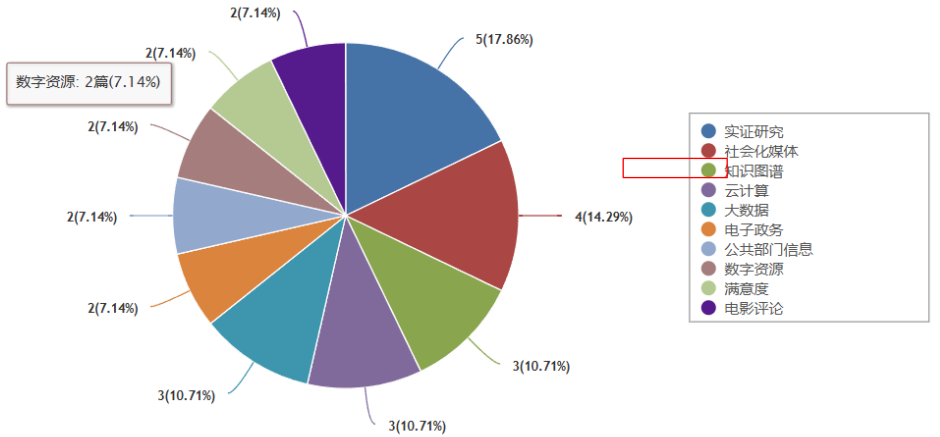
第二讲 词频统计

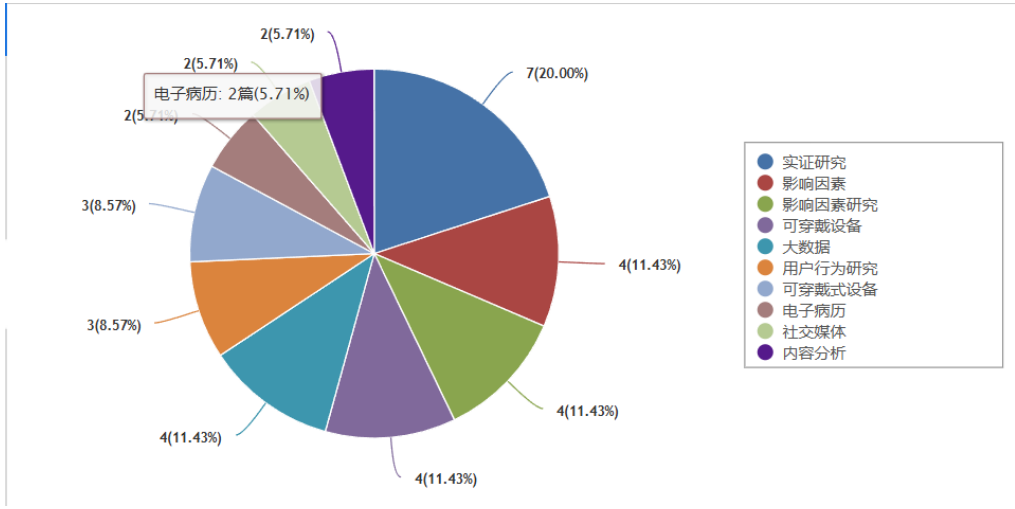
1. 基于CNKI数据库统计分析2014-2024年（近10年），“信息资源管理”或“网络营销”或其他你感兴趣的主题变化趋势。
2. 完成ppt中的程序运行，包括全文词频统计，指定类型词频统计；
3. 链接功勋科学家：把ppt中的文本换成功勋科学家黄旭华院士的传记序言文本（文件夹中，科学家博物馆-黄旭华传记序言.txt），1）统计全文词频；2）统计指定词频，如“黄旭华”；
4. 阅读论文“2018-Wang 等 - Long live the scientists Tracking the scientific”，并做阅读总结（1页PPT即可）。

第二讲 词频统计

1. 2014年-2017年

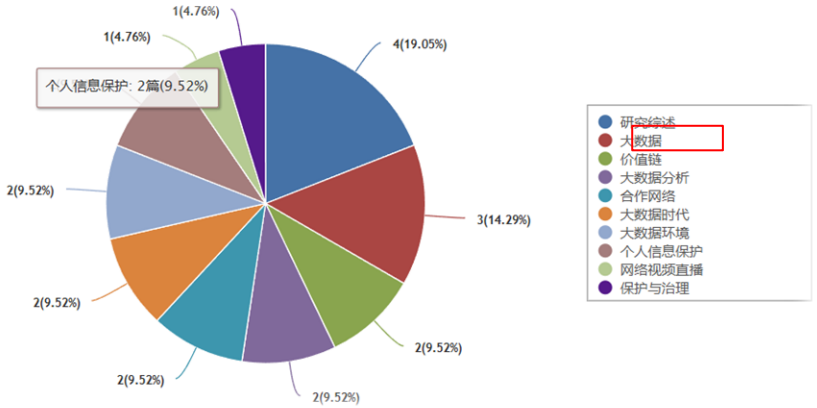


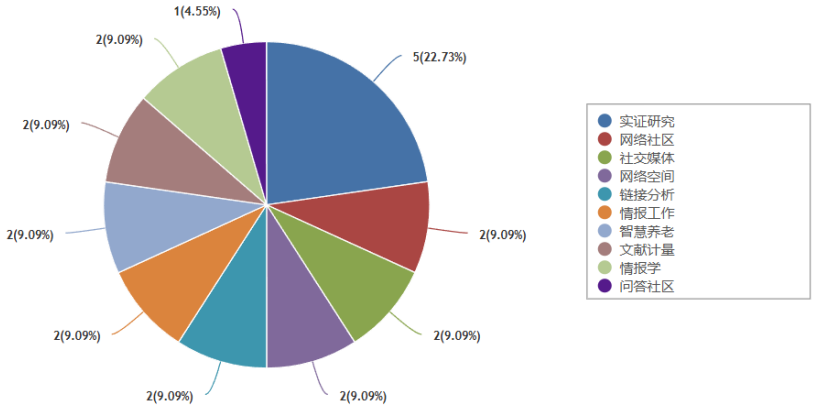


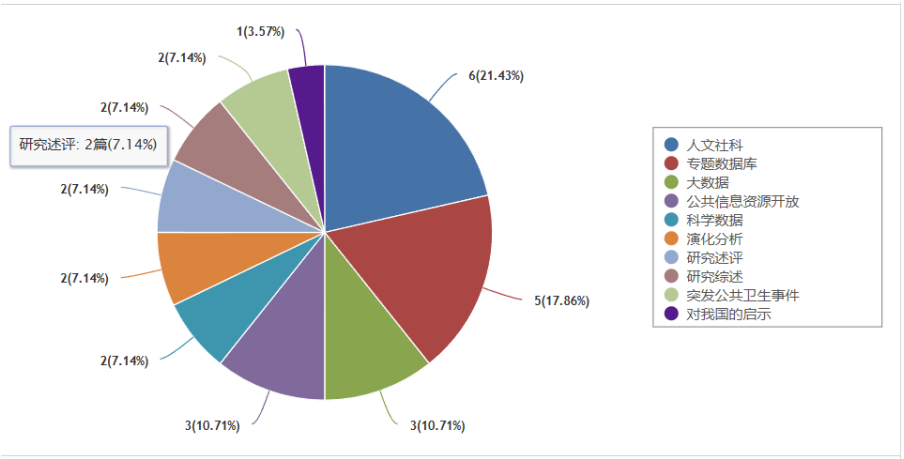


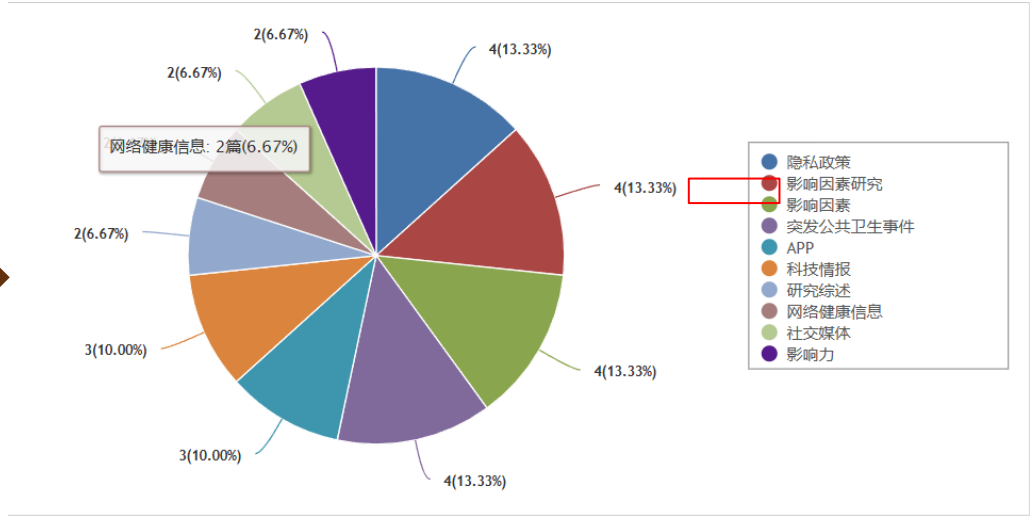
第二讲 词频统计

2018年-2021年



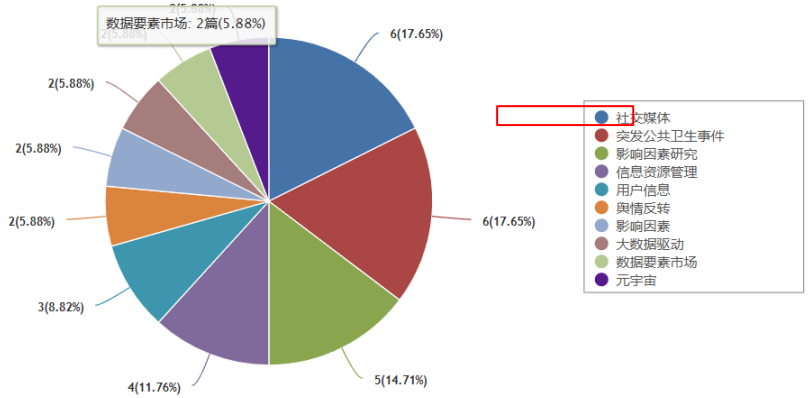


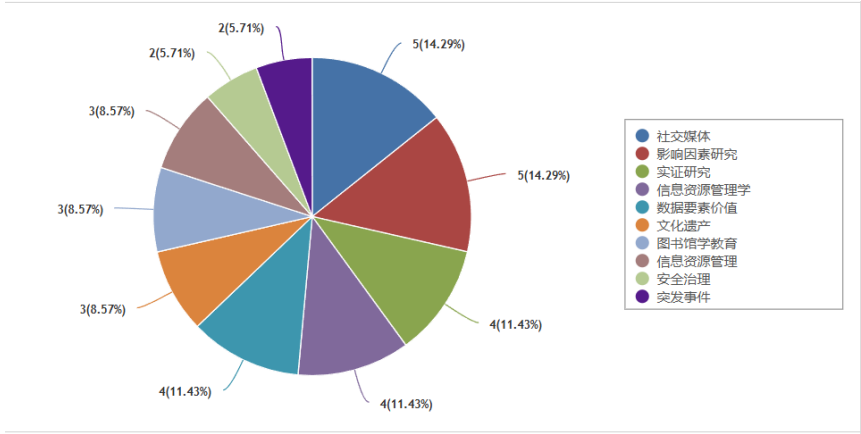


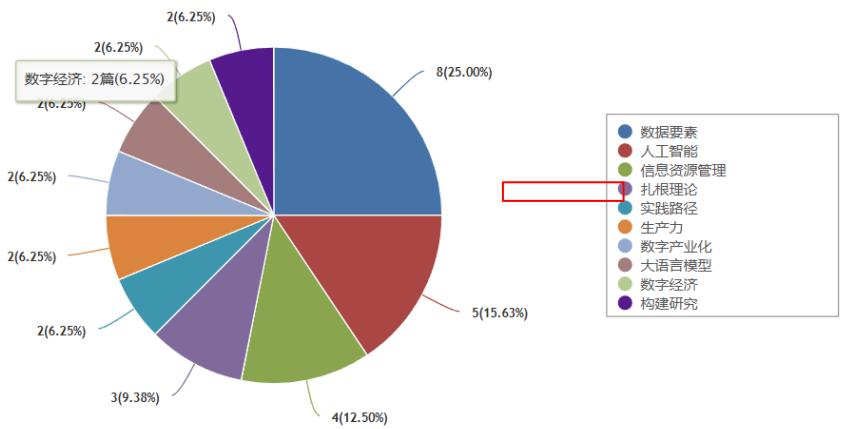


第二讲 词频统计

2022年-2024年







从 2014-2024 年 CNKI 信息资源管理领域的发文主题变化来看，核心趋势可概括为：

主题聚焦方向调整：早期分散于“长期保存”“可视化分析”“知识图谱”等多元主题，后期逐步向“实证研究”“社会化媒体”“人工智能”等方向集中，热门主题的占比显著提升。

技术与应用结合深化：从基础的“数字资源”“信息存储”类主题，拓展到“云计算”“大数据”“电子病历”等技术落地场景，同时新增“用户行为”“健康信息行为”等应用研究维度。

热点主题迭代明显：“社会化媒体”“人工智能”等新兴主题后期成为核心方向，而早期的“世界记忆工程”“虚拟社区”等主题占比逐渐降低，体现领域研究随行业发展的动态调整。

第二讲 词频统计

2.

[28]:
```
# 输出词频的前N个
for i in range(20):
    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)
('校尉', 21)
```

[21]:
```
# 定义  画图 函数
```

[23]:
```
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
```

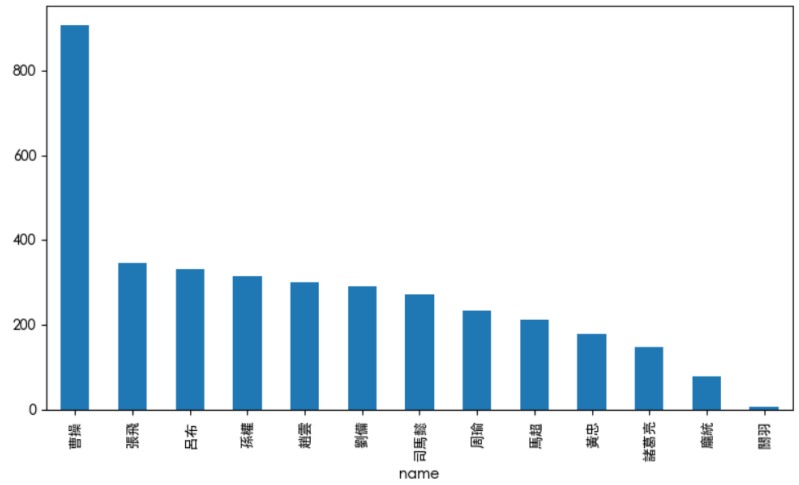
[24]:
```
# 定义 画图 函数
```

[25]:
```
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()
```

[26]:
```
# %pylab inline
```

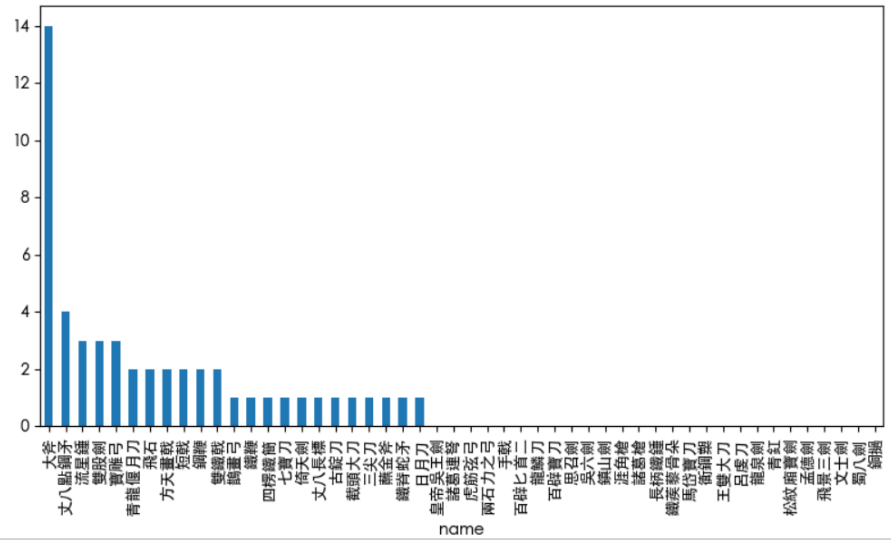
[27]:
```
%matplotlib inline
```

[28]:
```
draw_dict(name_dict)
```




```
'鐵鞭': 1,
'鋼鞭': 2,
'四楞鐵簡': 1,
'雙鐵戟': 2,
'諸葛連弩': 0,
'寶雕弓': 3,
'鵰畫弓': 1,
'虎筋弦弓': 0,
'兩石力之弓': 0,
'手戟': 0,
'短戟': 2,
'飛石': 2,
'流星錘': 3,
'銅撾': 0}
```

[39]:
```
draw_dict(weapon_dict)
```



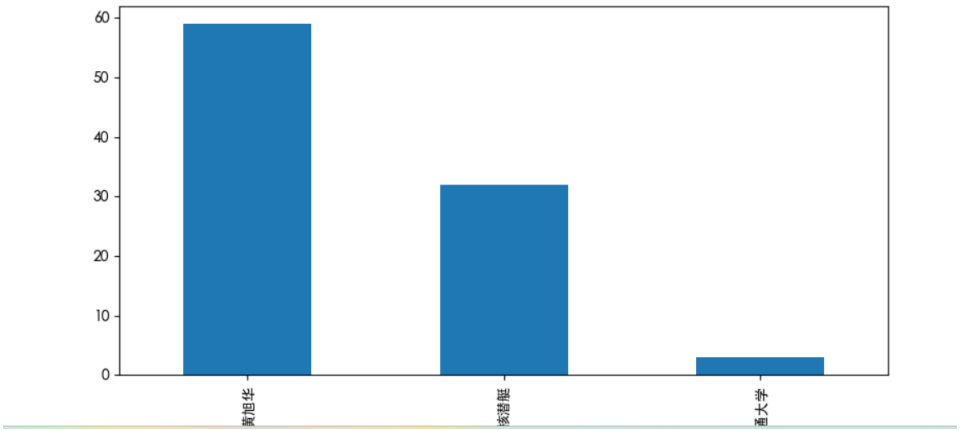
第二讲 词频统计

3.

```
[17]: # 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)
('技术', 12)
('研制', 12)
('我国', 12)
('工程', 11)
('访谈', 10)
('第一代', 8)
('介绍', 8)
('主要', 8)
('科学', 8)
('思想', 7)
('人生', 7)
('及其', 7)
('历史', 7)
('传记', 7)
('过程', 6)
('按照', 6)
('要求', 6)
```

```
[11]: for term in terms:
          terms_dict[term]=data_txt.count(term)

[12]: terms_dict

[12]: {'黄旭华': 59, '核潜艇': 32, '国立交通大学': 3}

[23]: # 定义  画图 函数

[13]: 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

[14]: # 定义 画图 函数

[15]: 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()

[16]: # %pylab inline

[17]: %matplotlib inline

[18]: draw_dict(terms_dict)
```



第二讲 词频统计

4. 关于《Long live the scientists: Tracking the scientific fame of great minds in physics》的阅读总结

《Long live the scientists: Tracking the scientific fame of great minds in physics》一文，依托谷歌图书（含 3600 余万本全球数字化图书）与谷歌学术（索引 9100 余万条学术成果）的海量资源，聚焦物理学领域顶尖科学家的科学声誉展开系统性研究。

研究以牛顿和爱因斯坦为核心案例（二者均为史上最具影响力的物理学家，且具备学科、姓名检索可比性），并纳入 234 位知名科学家样本，通过姓名提及频率统计、共现分析等方法展开探究。核心发现如下：其一，伟大科学家的声誉具有极强持久性，即便逝世数百年仍被广泛铭记，1980 年代后牛顿、爱因斯坦的提及度均显著回升；其二，时间维度上呈现明显阶段性，牛顿在 1680-1880 年声誉鼎盛，1948 年成为关键分水岭，此后爱因斯坦的全球声誉全面超越牛顿；其三，存在显著的 “群体偏好效应”，英式英语书籍中牛顿更受青睐，而美式英语、德语载体中爱因斯坦提及度更高，这与 2005 年英国皇家学会民调结果形成呼应与补充。

共现分析明确了二者声誉的核心支撑：牛顿的声誉主要与万有引力定律、微积分、运动定律相关，爱因斯坦则以相对论（占比 28.1%）和量子理论（占比 16.9%）为声誉核心。此外，研究验证了谷歌图书及 Ngram Viewer 作为替代计量工具的可行性，突破传统引文计量局限，能更全面衡量学者超越学术界的社会影响力。

研究同时指出局限，如谷歌图书存在非英语书籍覆盖偏差、姓名歧义可能导致计量误差等。未来可将研究拓展至其他学科，细化书籍分类（如教科书、通俗读物），进一步完善科学影响力的多元评价体系。

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

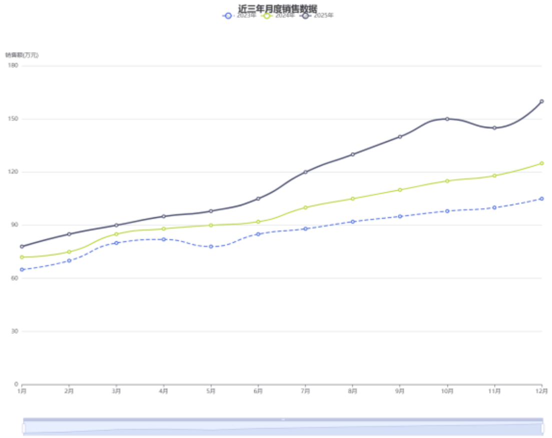
1.

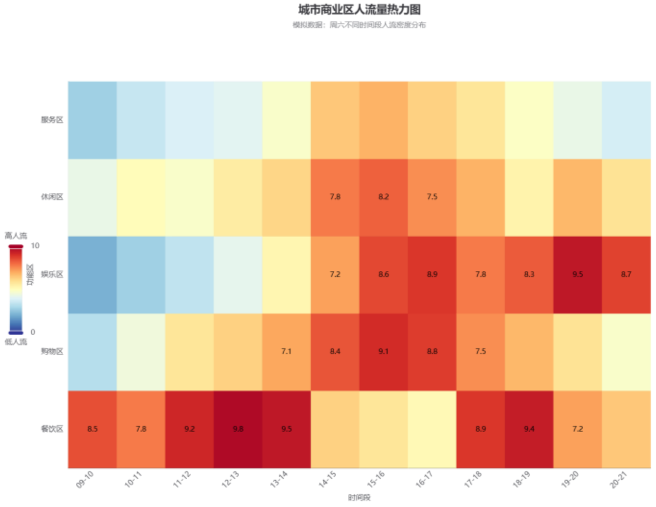
查找单词	字体	词频	颜色
选手	默认	30	auto
荒野	默认	30	auto
求生	默认	23	auto
赛事	默认	21	auto
比赛	默认	18	auto
七星山	默认	16	auto
流量	默认	14	auto
景区	默认	12	auto
王昌	默认	11	auto
直播	默认	10	auto
奖金	默认	10	auto
挑战赛	默认	8	auto
冷美人	默认	8	auto
身体	默认	8	auto
记者	默认	7	auto
主办方	默认	7	auto
参赛	默认	7	auto
报名	默认	7	auto
节目	默认	6	auto
媒体	默认	6	auto
节目组	默认	6	auto
网友	默认	6	auto

这张词云以“荒野求生选手”为轮廓，核心词“荒野”“赛事”“选手”锚定生存挑战主题，“流量”“直播”凸显其爆火的传播属性，“七星山”“张家界”点明地点，“冷美人”“王昌繁”等则带出话题选手，直观呈现这场“生存游戏 + 流量盛宴”的双重底色。

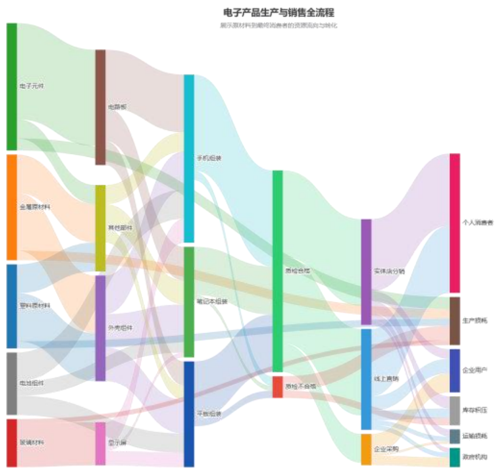
第三讲 词云与可视化

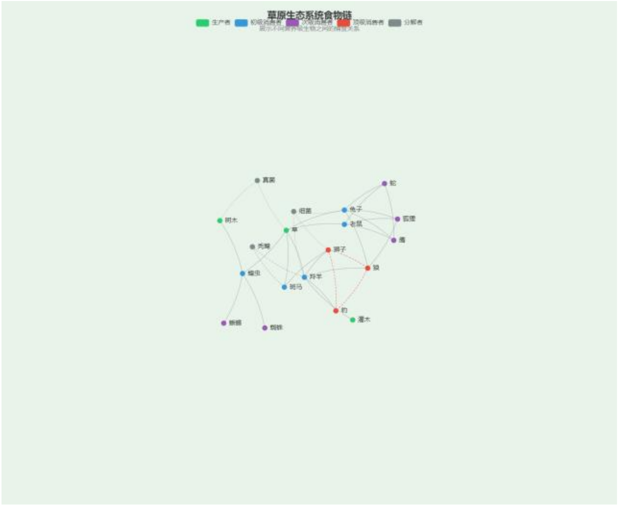
2.


近三年月度销售数据


城市商业区人流量热力图


互联网产品生态系统

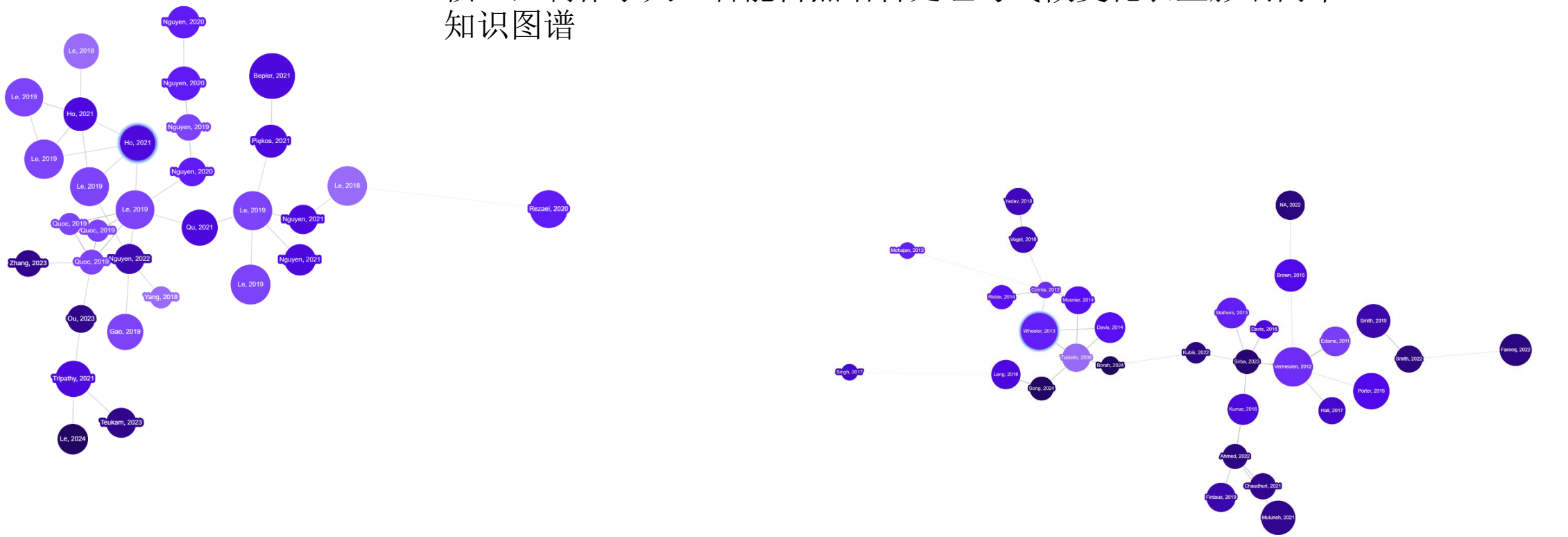

电子产品生产与销售全流程


草原生态系统食物链

第三讲 词云与可视化

3.

使用Inciteful.xyz工具，分别以BERT论文和全球变暖影响研究为核心，制作了人工智能自然语言处理与气候变化农业影响两个知识图谱



第三讲 词云与可视化

4.

```
[22]: # 重新生成词云
from wordcloud import WordCloud
wordcloud_cn = WordCloud(font_path="simsun.ttf").generate(final)
%pylab inline
import matplotlib.pyplot as plt
plt.imshow(wordcloud_cn, interpolation='bilinear')
plt.axis('off')
```

```
%pylab is deprecated, use %matplotlib inline and import the required libraries.
Populating the interactive namespace from numpy and matplotlib
```

```
[22]: (-0.5, 399.5, 199.5, -0.5)
```

好了！是不是感觉自己很不错呢

我们还是换一个形状，换一个黄院士的头像吧，我已经替你下载好啦，在文件夹里面

给词云加上特定的形状

```
[23]: big_pic = imread('huangxuhua.jpg')
```

```
[24]: wordcloud_cn_x = WordCloud(mask=big_pic,width=500,height=300, background_color= 'white',scale=
plt.imshow(wordcloud_cn_x)
plt.axis('off')
```

```
[24]: (-0.5, 1374.5, 1924.5, -0.5)
```

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.

情感分析

执行情感分析：

```
text = '''“这是一部男人必看的电影。”人人都这么说。但单纯从性别区分，就会让这电影变狭隘。
《肖申克的救赎》突破了男人电影的局限，通篇几乎充满令人难以置信的温馨基调，而电影里最伟大的主题是“希望”。
当我们无奈地遇到了如同肖申克一般囚禁了心灵自由的那种囹圄，我们是无奈的老布鲁克，灰心的瑞德，还是智慧的安迪？
运用智慧，信任希望，并且勇敢面对恐惧心理，去打败它？
经典的电影之所以经典，因为他们都在做同一件事——让你从不同的角度来欣赏希望的美好。'''
HanLP.sentiment_analysis(text)
```

运行　可编辑 & 一键云端运行　READY

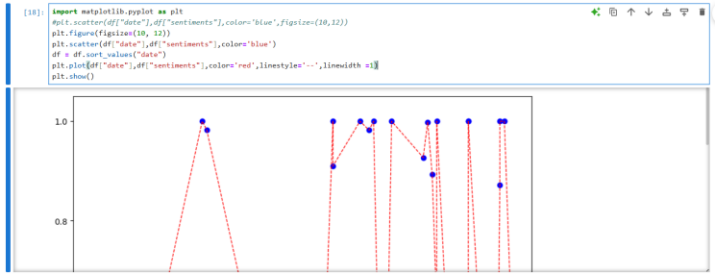
```
0.9505730271339417
```

返回值为文档的情感极性，表示为 `[-1, +1]` 之间的数值。

2.

```
[15]: text = "I am happy today. I feel sad today."

[17]: from textblob import TextBlob
      blob = TextBlob(text)

[19]: blob

[19]: TextBlob("I am happy today. I feel sad today.")

[25]: # 原封不动的打印出来了？
      # 实际上已经把文本分成了句子了，看一看
      blob.sentences

[25]: [Sentence("I am happy today."), Sentence("I feel sad today.")]

[27]: blob.sentences[0].sentiment

[27]: Sentiment(polarity=0.8, subjectivity=1.0)

[29]: # 上面的结果什么意思呢？
      # 情感极性0.8，主观性1.0。说明一下，情感极性的变化范围是[-1, 1]，-1代表完全负面，1代表完全正面。
      # 我表达的是我很高兴，那么这个结果是对的

[31]: blob.sentences[1].sentiment

[31]: Sentiment(polarity=-0.5, subjectivity=1.0)

[33]: # 整段文本的情感呢？
      blob.sentiment

[33]: Sentiment(polarity=0.15000000000000002, subjectivity=1.0)

[35]: # 你可能会觉得没有道理，怎么一句“高兴”，一句“沮丧”，合并起来最后会得到正向结果呢？
      # 首先不同极性的词，在数值上是有区别的，我们应该可以找到比“沮丧”更为负面的词汇，而且这也符合逻辑，谁会这么“天上一脚，地下一脚”矛盾地描述自己此时的心情呢？
```

```
[38]: text_cn = u"我今天很快乐。我今天很愤怒。"

[40]: #注意在引号前面我们加了一个字母u，它很重要。因为它提示Python，“这一段我们输入的文本编码格式是Unicode，别搞错了哦”。至于文本编码格式的细节，有机会我们再详细聊。

[42]: from snownlp import SnowNLP

[44]: senti_cn = SnowNLP(text_cn)

[46]: # 看看snownlp包的分句能力
      for sentence in senti_cn.sentences:
          print(sentence)

      我今天很快乐
      我今天很愤怒

[48]: senti_cn_1 = SnowNLP(senti_cn.sentences[0])

[50]: # 一个细节上的问题，英文是x.sentiment,中文是x.sentiments，多了一个s
      # 另外，在句法上和英文的也略有不同，比如直接用语句：senti_cn.sentences[0].sentiments是会报错的
      senti_cn_1.sentiments

[50]: 0.971889316039116

[52]: senti_cn_2 = SnowNLP(senti_cn.sentences[1])

[54]: senti_cn_2.sentiments

[54]: 0.07763913772213482
```




```
[16]: plt.savefig('timeline.png') # 看不到？点一次？

      <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前5，可以这样执行：

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

[19]:
```

	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，而是感觉哪哪人都多，所以特意错开，这次实在是馋a解厅了，所以挺在正日子也出来了，从下午四点多的时候我雷报号就排到一百多了，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报着号了，毫不多我们是六点半到的，到那的时候我雷号码前面还有才三十多号，我思着肯定没问题了，等一会就能吃上的，没想到是都了，就认我们到那坐到等位区开始，大约是十分二十分一叫号，中途多次告都想走了，喻哈，吧，等到最后早上九点才吃上的，服务员感觉也应以前清闲时间到了，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理解，论让服务员给再大多，就扩课让服务员下的，然后对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解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解厅，就是比一般的要人性化，不过还次就是选错日子了，以前还是得提前预约，要不就到赶节日去，大火爆了！	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，而是感觉哪哪人都多，所以特意错开，这次实在是馋a解厅了，所以挺在正日子也出来了，从下午四点多的时候我雷报号就排到一百多了，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报着号了，毫不多我们是六点半到的，到那的时候我雷号码前面还有才三十多号，我思着肯定没问题了，等一会就能吃上的，没想到是都了，就认我们到那坐到等位区开始，大约是十分二十分一叫号，中途多次告都想走了，喻哈，吧，等到最后早上九点才吃上的，服务员感觉也应以前清闲时间到了，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理解，论让服务员给再大多，就扩课让服务员下的，然后对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解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解厅，就是比一般的要人性化，不过还次就是选错日子了，以前还是得提前预约，要不就到赶节日去，大火爆了！

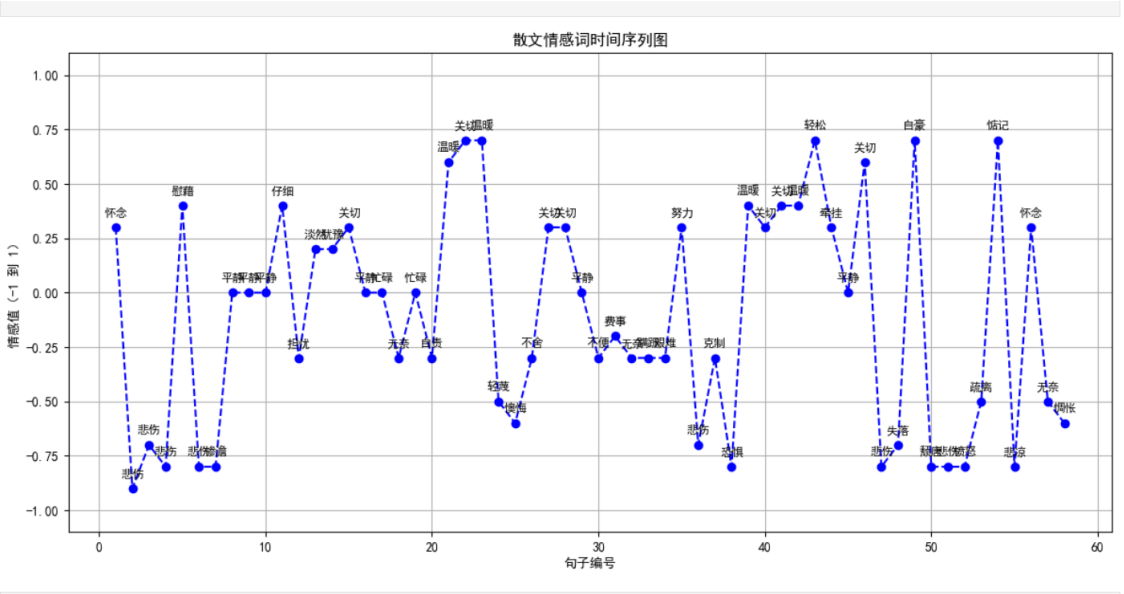
第四讲 情感分析

2.

```
try:
    # 发送 POST 请求
    response = requests.post(url, headers=headers, data=json.dumps(data))

    # 检查响应状态码
    if response.status_code == 200:
        # 解析 JSON 响应
        result = response.json()
        # 提取模型生成的内容
        generated_text = result['choices'][0]['message']['content']
        print("细粒度情感实体抽取结果:")
        print(generated_text)
    else:
        # 处理错误响应
        print(f"请求失败，状态码: {response.status_code}")
        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"
    },
```



这些代码主要展示了中文情感分析技术，包括使用SnowNLP进行基础情感分析（处理"我今天很快乐。我今天很愤怒。"文本，获得0.97和0.08的情感得分），以及通过API调用进行细粒度情感实体抽取（识别出头部、全身、皮肤、心脏等部位的对应症状和情感描述）。代码包含完善的错误处理机制，适用于社交媒体分析、用户反馈分析等场景，体现了从基础情感分析到高级细粒度分析的技术流程。

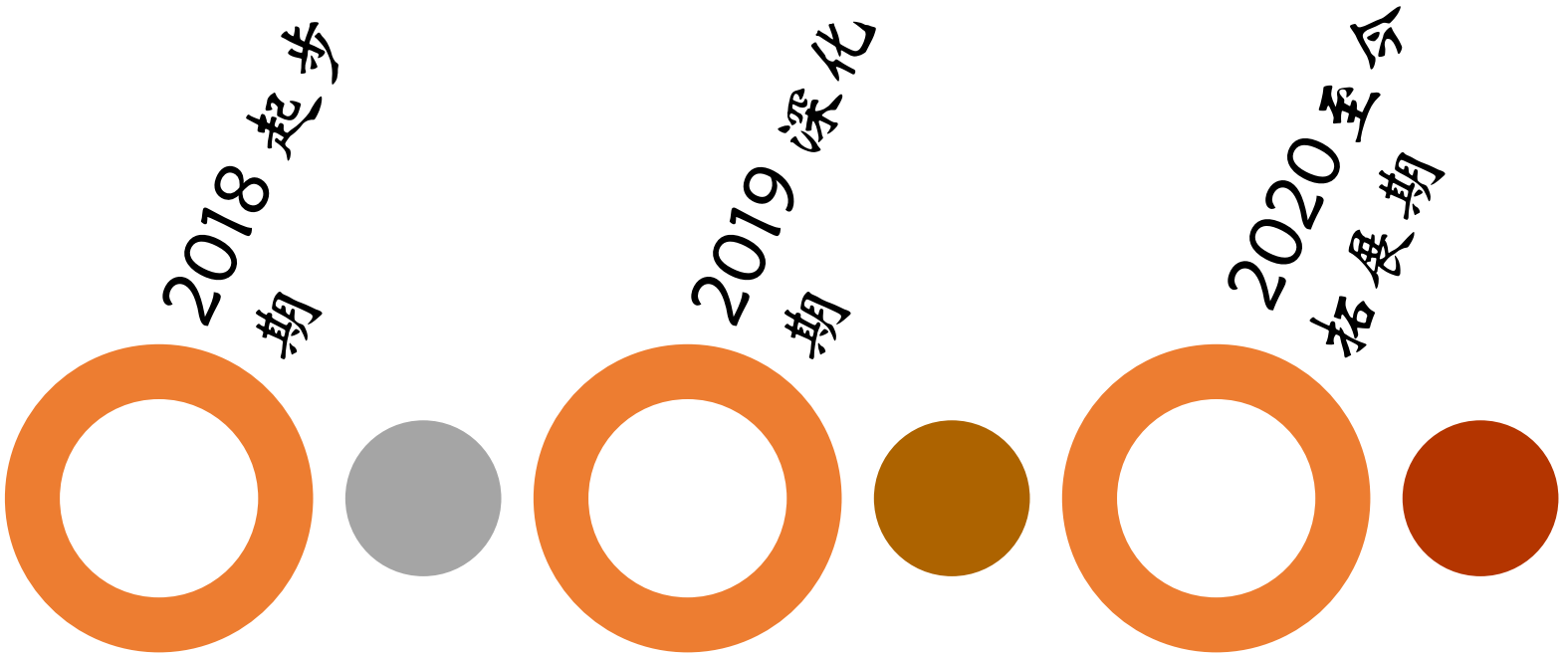
第五讲 新媒体数据分析

1. 略了...

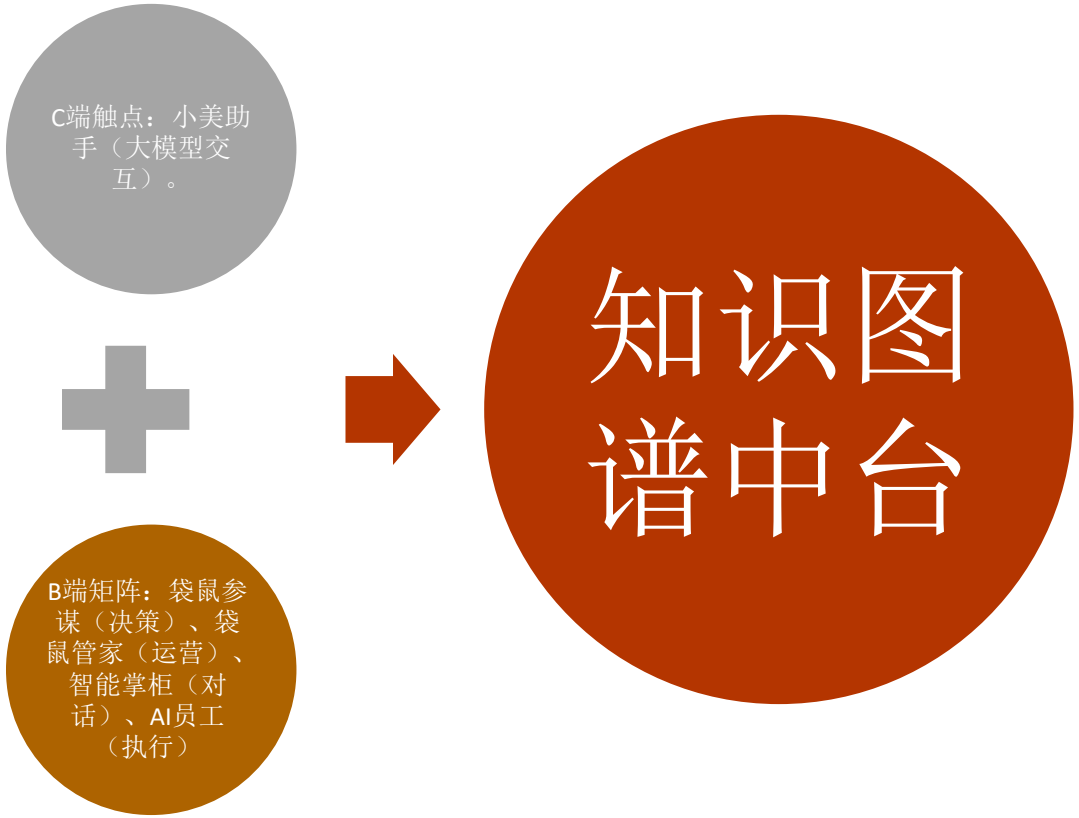
第六讲 知识图谱理念

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）

美团大脑演进路线：从数据到生态



生态构建：双向赋能体系



助手名称	主要定位	核心功能	赋能的知识图谱能力
袋鼠参谋	AI决策工具（独立App）	解决“开什么店、开在哪”等重决策问题，提供赛道分析、选址建议-5。	融合地理位置、商圈客流、竞争格局、品类热度等多维知识进行推理与预测。
袋鼠管家	数字运营经理（嵌入后台）	自动化处理菜品上下架、活动排期、评价维护等高频率运营事务-5	基于商品知识图谱与用户反馈标签，自动化优化门店线上资产与营销动作。
智能掌柜	到店管理助手（嵌入后台）	以对话方式提供经营诊断、日报生成、差评分析，甚至AI接听预定电话-5。	关联交易、评价、客群画像数据，实现经营状况的智能解读与交互。
AI数字员工	全链路经营助手	涵盖客服、排班、运营等角色的数字员工，为每家门店提供全场景提效-2-8。	是知识图谱与自动化任务（RPA）的结合，将结构化知识转化为具体操作。

案例分析：知识图谱如何驱动“智能选址”

以“袋鼠参谋”协助茶饮店选址为例，剖析知识图谱在其中的作用。

需求理解：

商家输入意向（如“想在A城开一家中端水果茶店”）

知识调用与多因子分析：

- 空间知识：调用A城各商圈地理信息、人流热力。
- 市场知识：分析各商圈现有茶饮门店（竞对）的密度、品牌分布、价位区间。
- 消费知识：结合历史消费数据，判断不同商圈对“水果茶”品类的偏好度、消费能力。
- 商户知识：评估各商圈平均租金水平、合规要求。

推理与建议生成：

- 模型综合上述多维知识进行推演，输出建议选址列表，并解释关键因素（如：“B商圈年轻客群集中，竞对以咖啡为主，水果茶存在缺口，但租金偏高”）。

洞察评价：价值与挑战

战略价值

从“工具”到“生态”：美团大脑正从为内部业务提效的“技术工具”，演化为一个连接供需两端、赋能整个本地生活产业的AI原生生态底座。

构建深度壁垒：知识图谱依赖长期、高质量的数据沉淀与领域标注。美团在生活服务领域的深耕，使其构建的图谱具有独特性和排他性，成为核心竞争壁垒-5。

瞄准行业痛点：服务零售线上化率仅9%-2-8，大量中小商家缺乏数字化运营能力。美团通过AI助手降低使用门槛，正是为了撬动这片蓝海市场，与平台自身增长形成飞轮。

主要挑战

知识更新与动态性：生活服务信息（如价格、营业状态）变化极快，如何实现知识图谱的低延迟、自动化更新是持续挑战。

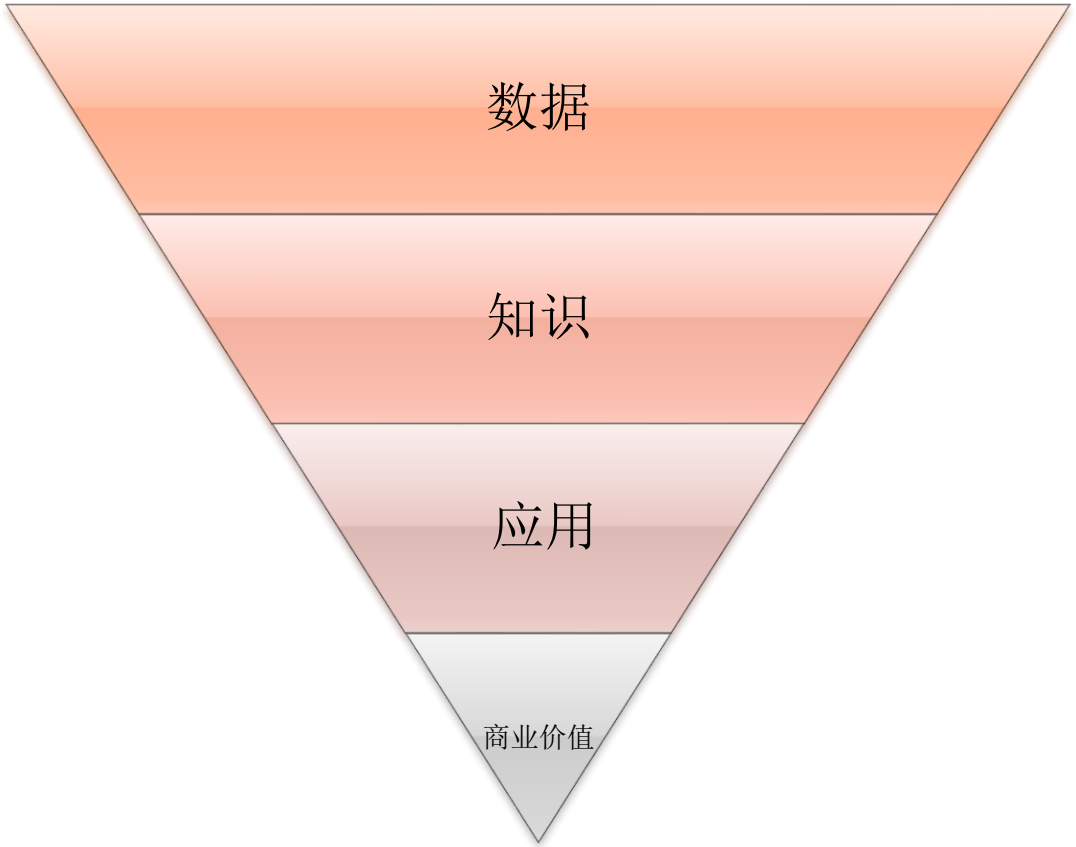
复杂推理的可靠性：“袋鼠参谋”类重决策建议的准确性直接关系到商家重大投入，如何确保复杂推理的稳定与可信，并建立有效的效果复盘机制-5，至关重要。

生态协同的复杂性：C端与B端多个智能体间的协作流程、责任界定、利益分配机制非常复杂，顺畅运转需要精密的系统设计与运营。

未来展望：从理解到改造

总结：美团大脑的进化，是从“理解世界”到“改造商业”的范式跃迁。

展望：未来竞争在于知识深度与智能体协作效率。

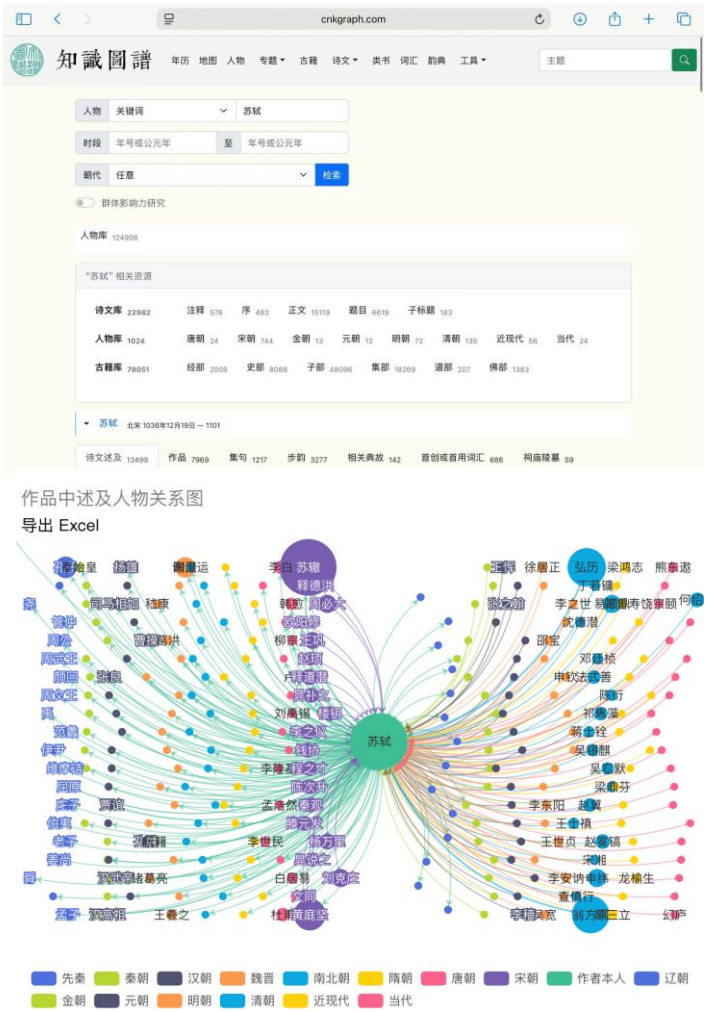


第六讲（2） 知识图谱工具

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）。

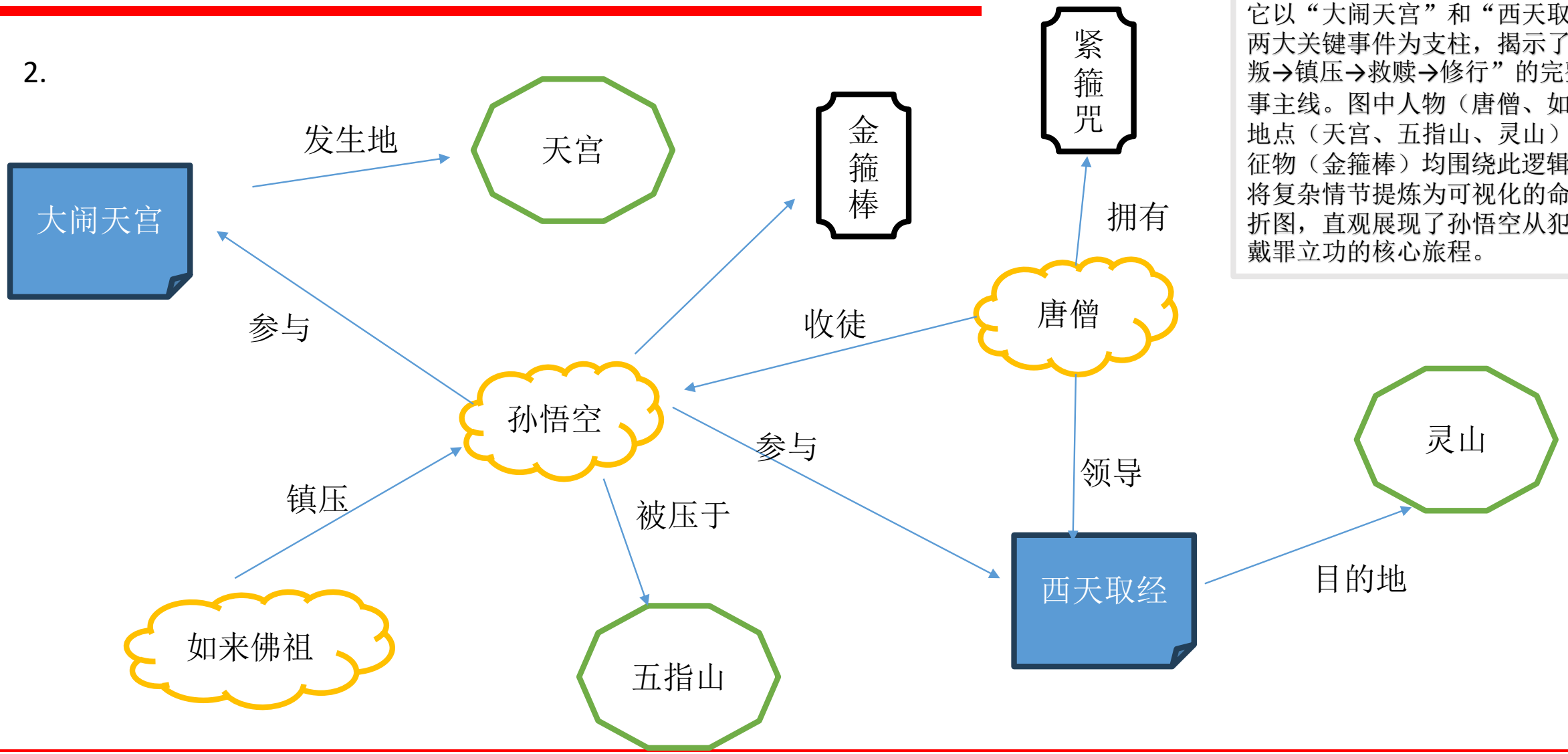
第六讲（2） 知识图谱工具

1.



第六讲（2） 知识图谱工具

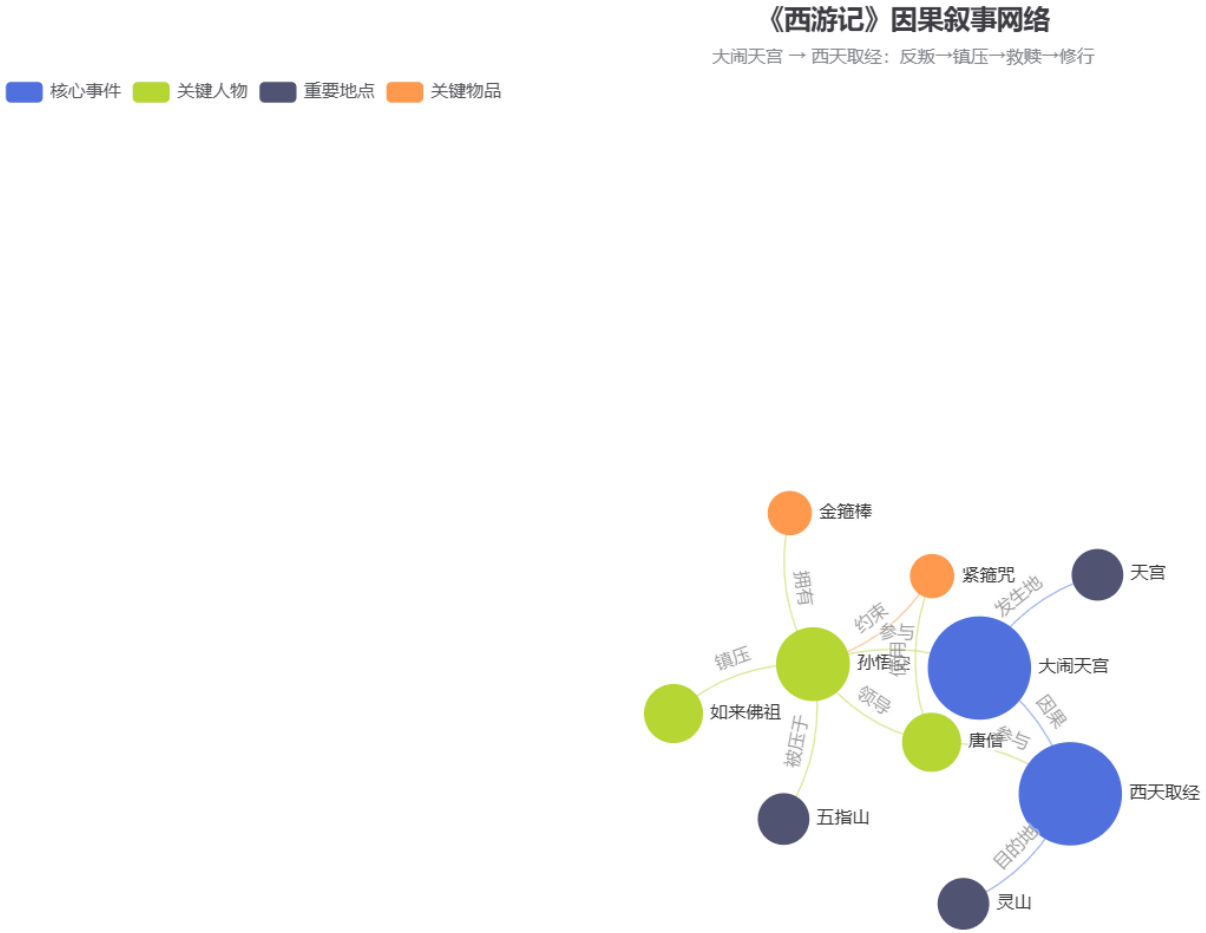
2.



该图谱以孙悟空为核心，构建了一个清晰的《西游记》因果叙事网络。它以“大闹天宫”和“西天取经”两大关键事件为支柱，揭示了“反叛→镇压→救赎→修行”的完整故事主线。图中人物（唐僧、如来）、地点（天宫、五指山、灵山）与象征物（金箍棒）均围绕此逻辑展开，将复杂情节提炼为可视化的命运转折图，直观展现了孙悟空从犯错到戴罪立功的核心旅程。

第六讲（2） 知识图谱工具

3.



第六讲（2） 知识图谱工具

4.