南京工業大學
NANJING TECH UNIVERSITY

用户数据采集与关联分析

（结课作业）

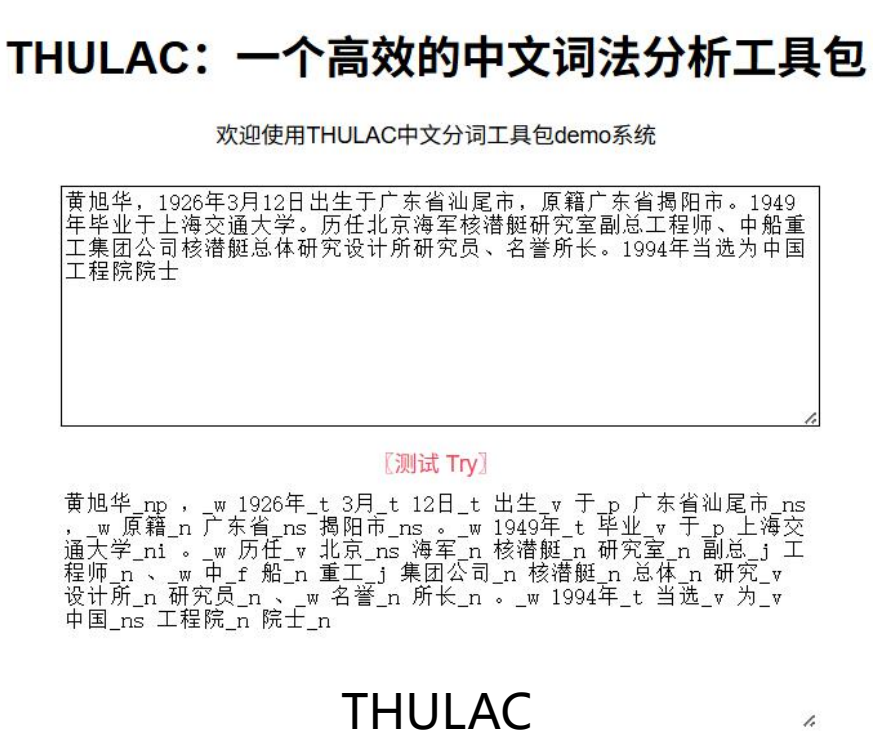
郑瀚程

13361063629

1210167366@qq.com

第一章 课程导言与分词

1、在线分词系统使用



THULAC

微词云

2、安装Anaconda

```
[1]: print("Hello world! Hello Zheng Hancheng")
```

```
Hello world! Hello Zheng Hancheng
```

第一章 课程导言与分词

3、课后作业001

1 基本操作

[1]:
```
!pip install jieba
```

```
Requirement already satisfied: jieba in /Users/wuzhixiang/anaconda3/lib/python3.11/site-packages (0.42.1)
```

[3]:
```
import jieba  # 在命令行里面安装分词软件包jieba # pip install jieba
```

[5]:
```
seg_list = jieba.cut("南京大学生都爱南京市长江大桥")
```

[7]:
```
print( '    '.join(seg_list))
```

```
Building prefix dict from the default dictionary ...
Dumping model to file cache C:\Users\Lenovo\AppData\Local\Temp\jieba.cache
Loading model cost 0.848 seconds.
Prefix dict has been built successfully.
```

南京　　大学生　　都　　爱　　南京市　　长江大桥

[9]:
```
seg_list = jieba.cut("南京大学生都爱南京市长江大桥")
```

[11]:
```
print('*'.join(seg_list))
```

南京*大学生*都*爱*南京市*长江大桥

2 加入用户词典

[13]:
```
seg_list1 = jieba.cut("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？")
```

[15]:
```
print('#'.join(seg_list1))
```

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#？

载入词典

[18]:
```
jieba.load_userdict('dict.txt')
```

[20]:
```
seg_list = jieba.cut("吴志祥是南京工业最好大学青年教师，经济与管理学院的，他对那种二次元小魔仙是无感的，这怎么行？python看上去还有点人性")
```

[22]:
```
print('%'.join(seg_list))
```

吴志祥%是%南京工业最好大学%青年教师%，%经济%与%管理%学院%的%，%他%对%那种%二次元小魔仙%是%无感%的%，%这%怎么%行%？%python%看上去%还%有点人性

第一章 课程导言与分词

3、课后作业001

3 载入停用词表 ¶

```
[24]: seg_list = jieba.cut("没有使用停用词表的分词结果，就会有很多没有用的词啊，虚词、感叹词什么的")
```

```
[26]: print('*'.join((seg_list)))
```

没有*使用*停用*词表*的*分词*结果*，*就*会*有*很多*没有*用*的*词*啊*，*虚词*、*感叹词*什么*的

载入停用词表

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

```
[31]: 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")
```

```
[33]: final = ''
```

```
[35]: for seg in seg_list:
          if seg not in stopwords:
              final += seg+'*'
```

```
[37]: print (final)   # 做实体抽取的时候，停用词表很管用
```

使用*词表*之后*效果*就*好看*很多*什么*之类*词*就*不见*

4 jieba分词与SnowNLP分词进行比较

```
[39]: from snownlp import SnowNLP
```

```
[40]: s = SnowNLP(u'质量不大好')
```

```
[43]: print(",".join(s.words))
```

质量,不大,好

```
[45]: ss = jieba.cut('质量不大好')
```

```
[47]: print(",".join(ss))
```

质量,不大好

```
[49]: 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？")
```

```
[51]: 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,？

功勋科学家-黄旭华-传记文本分词

现在，可以开启你的小组项目的第一个小小任务啦！就是对一小段有关"功勋科学家"的文本进行分词处理。

```
[1]: # 简单分词
```

```
[1]: import jieba
```

```
[3]: 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
```

```
[53]:
```

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

```
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\Lenovo\AppData\Local\Temp\jieba.cache
Loading model cost 0.807 seconds.
Prefix dict has been built successfully.
```

```
[57]: 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/、/总管/所长/，/1994/年/当选/为/中国工程院/院士/。
```

```
[59]:
```

```
[7]: # 加入用户词典
```

```
[61]: [9]: jieba.load_userdict('dict.txt')
```

```
[11]: 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
```

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

```
[64]: 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/、/总管/所长/，/1994/年/当选/为/中国工程院院士/。
```

```
[66]:
```

```
[15]: # 加入词典之后，新词汇被分出来了啦！
```

```
[68]:
```

```
[17]: # 使用停用词表
```

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

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

```
[71]:
```

```
[23]: stopwords
```

```
[73]: [23]: ['的', '了', '是', '啊', '、', '，', '。', '！', '得意']
```

```
[25]: 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
```

```
[75]:
```

```
[77]: [27]: final = ''
```

```
[29]: for seg in seg_list_huang:
          if seg not in stopwords:
              final+= seg+'/'
```

```
[79]:
```

```
[31]: 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/总管/所长/1994/年/当选/为/中国工程院院士/
```

第一章 课程导言与分词

3、 课后作业002

功勋科学家-黄旭华-传记文本分词

现在，可以开启你的小组项目的第一个小小任务啦！就是对一小段有关"功勋科学家"的文本进行分词处理。

```
[1]: # 简单分词

[1]: import jieba

[3]: 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，历任北京特种舰艇研究室副总工程师、中船重工集团(

[5]: print('/'.join(seg_list_huang))

Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\Lenovo\AppData\Local\Temp\jieba.cache
Loading model cost 0.807 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/特种/舰艇/研究室/副/总工程师/、/中/船/重
工/集团公司/舰艇/总体/研究/设计所/研究员/、/总管/所长/，/1994/年/当选/为/中国工程院/院士/。

[7]: # 加入用户词典

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

[11]: 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，历任北京特种舰艇研究室副总工程师、中船重工集团(

[13]: 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/特种/舰艇/研究室/副/总工程师/、/中船重工集
团公司/舰艇/总体/研究/设计所/研究员/、/总管/所长/，/1994/年/当选/为/中国工程院院士/。

[15]: # 加入词典之后，很多词汇被分出来了啦！

[17]: # 使用停用词表

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

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

[23]: stopwords

[23]: ['的', '了', '是', '啊', '、', '，', '。', '；', '停用']

[25]: 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，历任北京特种舰艇研究室副总工程师、中船重工集团(

[27]: final = ''

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

[31]: 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/特种/舰艇/研究室/副/总工程师/中船重工集团公司/舰艇/总
体/研究/设计所/研究员/总管/所长/1994/年/当选/为/中国工程院院士/
```

课后作业003（运行结果）

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

1.词频统计能初步反映文本的核心主题和高频词汇。

2.主要问题是未去除停用词，标点、换行符影响统计结果。

3.不能提现词语间关系，可通过网络分析知识图谱来改进。

第一章 课程导言与分词

3、课后作业004

```
[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-ac1f3b7750384ed095a91fab577905d2"  #直接复制过来
```

```
[6]: # 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",
    "免疫抑制信号通路",
    "PD-1/PD-L1",
    "TGF-β路径",
    "个体化免疫治疗"
  ]
}
```

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",
    "免疫抑制信号通路",
    "PD-1/PD-L1",
    "TGF-β路径",
    "个体化免疫治疗"
  ]
}
```

1.DeepSeek效率很高，响应速度快。

2.大语言模型在处理文本、信息整合等任务上表现很好，能快速完成基础工作。

第一章 课程导言与分词

4、论文阅读总结

1). 研究目的

学术论文的研究方法分类对学科评估与方法推荐具有重要意义。传统人工分类成本高、主观性强，自动分类可提升效率与一致性。

2). 研究方法与数据

采用图书情报领域820篇全文论文，专家标注16种研究方法，分为7类。使用多标签分类方法：问题转换法（BR/CC/RAKEL + NB/SVM）与算法自适应法（ML-KNN）。特征选择：N-Gram + 卡方检验；文本表示：TF-IDF + 向量标准化。

3). 研究结果

最佳模型：CC-NB（分类器链+朴素贝叶斯），F1值0.705。全文内容相比摘要显著提升分类性能。不同方法分类效果差异大："实验法"F1最高，"内容分析法"较低，与特征表征能力及样本量相关。

4). 主要结论

全文信息能有效提升研究方法自动分类效果。方法特征显著性与训练集规模共同影响分类性能。多标签分类中，朴素贝叶斯结合链式策略表现最优。

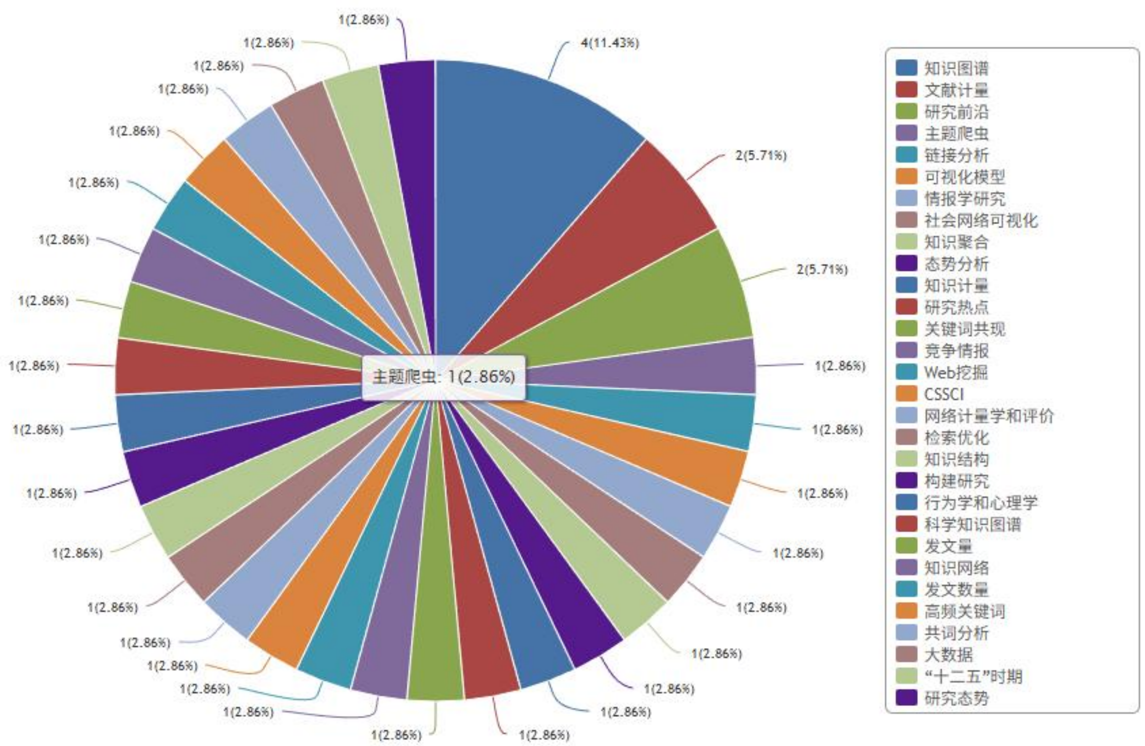
5). 展望

数据规模有限，小样本方法分类效果差。未来可结合深度学习、段落级分析，并扩展多学科数据。

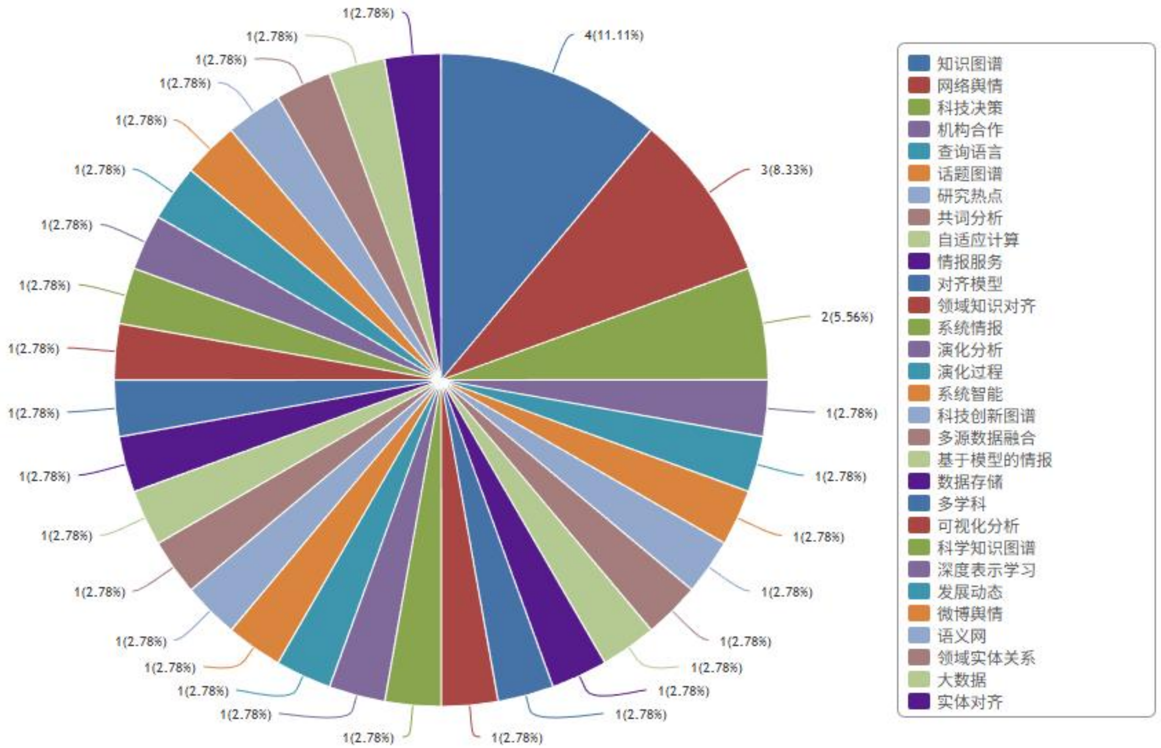
核心贡献：首次基于全文内容实现研究方法多标签自动分类，为学术文献方法论分析提供可行路径。

第二章 词频统计与分析

1、情报学报“知识图谱”主题变化趋势



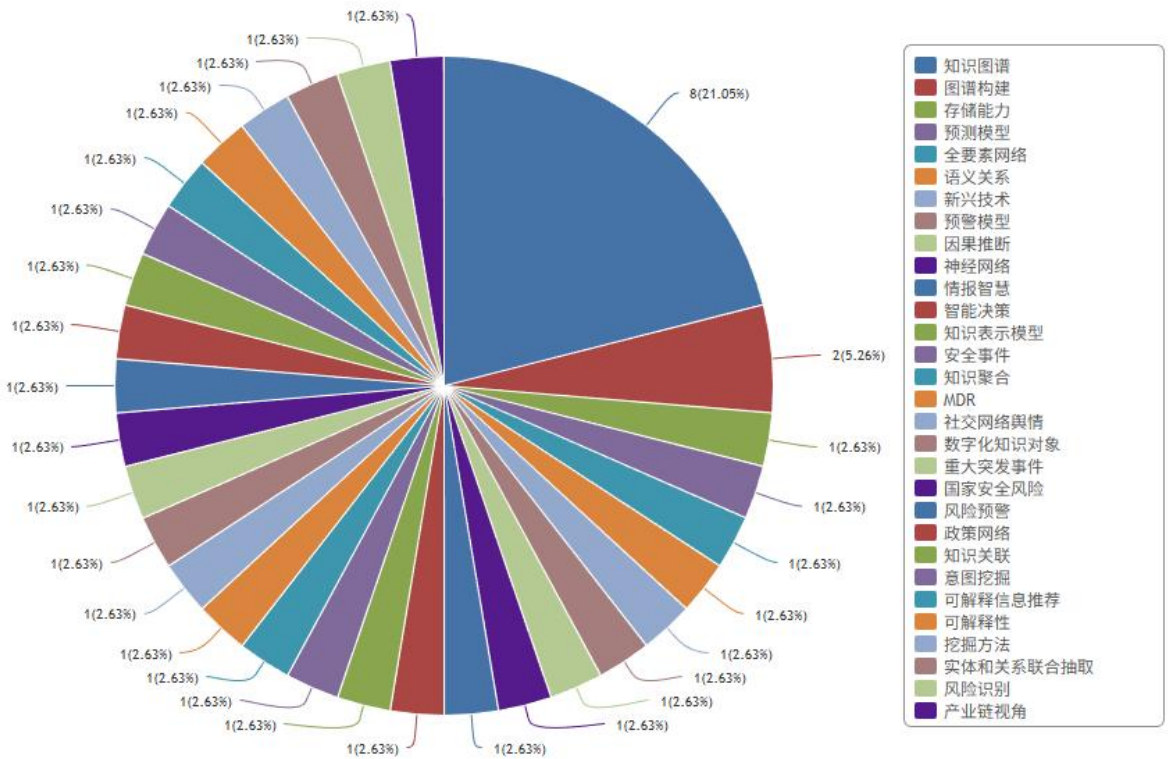
2015-2017



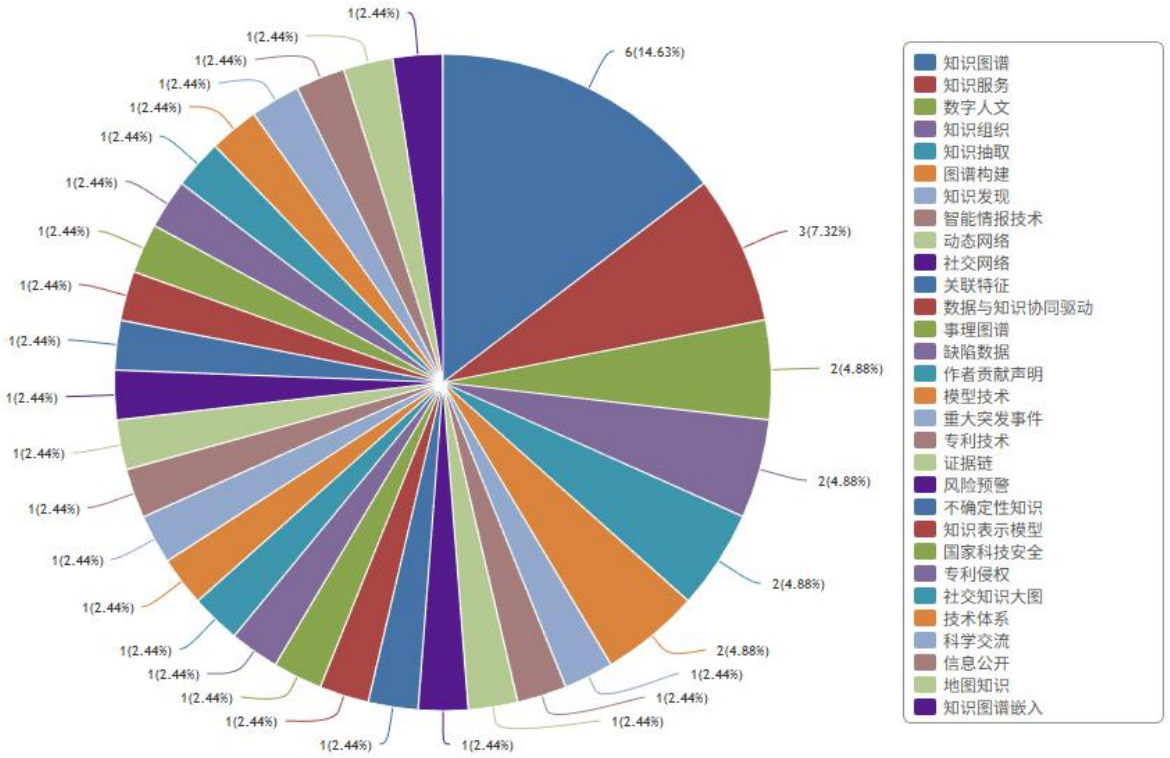
2018-2020

第二章 词频统计与分析

1、情报学报“知识图谱”主题变化趋势



2021-2023



2024-2025

第二章 词频统计与分析

2、全文词频统计

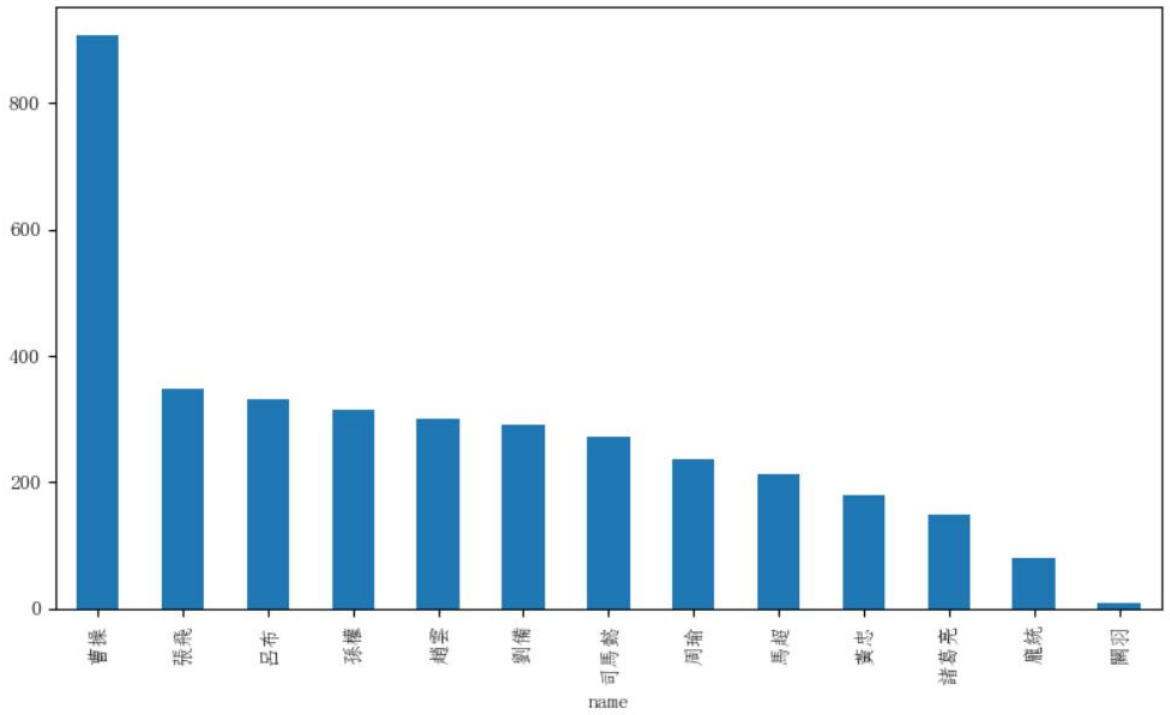
('董卓', 97)('吕布', 60)('曹操', 59)('袁紹', 57)('天下', 53)('玄德', 48)('貂蟬', 37)('太守', 36)('朝廷', 32)('孫堅', 31)('不可', 31)('次日', 26)('商議', 25)('引兵', 25)('李儒', 25)('天子', 24)('左右', 23)('玄德曰', 22)('太師', 22)('今日', 21)('大喜', 21)('軍士', 21)('校尉', 21)('太后', 20)('何進', 20)('王允', 20)('二人', 19)('司徒', 18)('郭汜', 18)('不能', 18)('公孫瓚', 18)('諸侯', 18)('盧植', 17)('如此', 17)('軍馬', 17)('何人', 17)('百姓', 17)('肅曰', 17)('朱雋', 17)('百官', 16)('李肅', 16)('張濟', 16)('張飛', 15)('皇甫嵩', 15)('留王', 15)('大事', 15)('如何', 15)('張角', 15)('文醜', 15)('何故', 15)('劉表', 14)('冀州', 14)('正是', 14)('刺史', 14)('袁術', 14)('因此', 14)('此人', 14)('張梁', 14)('只見', 14)('曹嵩', 13)('五千', 13)('不到', 13)('樊稠', 13)('是夜', 13)('英雄', 13)('五百', 13)('不得', 13)('將軍', 13)('性命', 12)('華雄', 12)('李蒙', 12)('人馬', 12)('張讓', 11)('相府', 11)('一日', 11)('趕來', 11)('一人', 11)('主公', 11)('社稷', 11)('大臣', 11)('三十', 11)('叔父', 11)('蔡瑁', 11)('丞相', 11)('星夜', 11)('不見', 10)('三人', 10)('相見', 10)('一面', 10)('不敢', 10)('忽見', 10)('不知', 10)('下文', 10)('問曰', 10)('大呼', 10)('分解', 10)('只得', 10)('何太后', 10)('乘勢', 10)('再拜', 10)

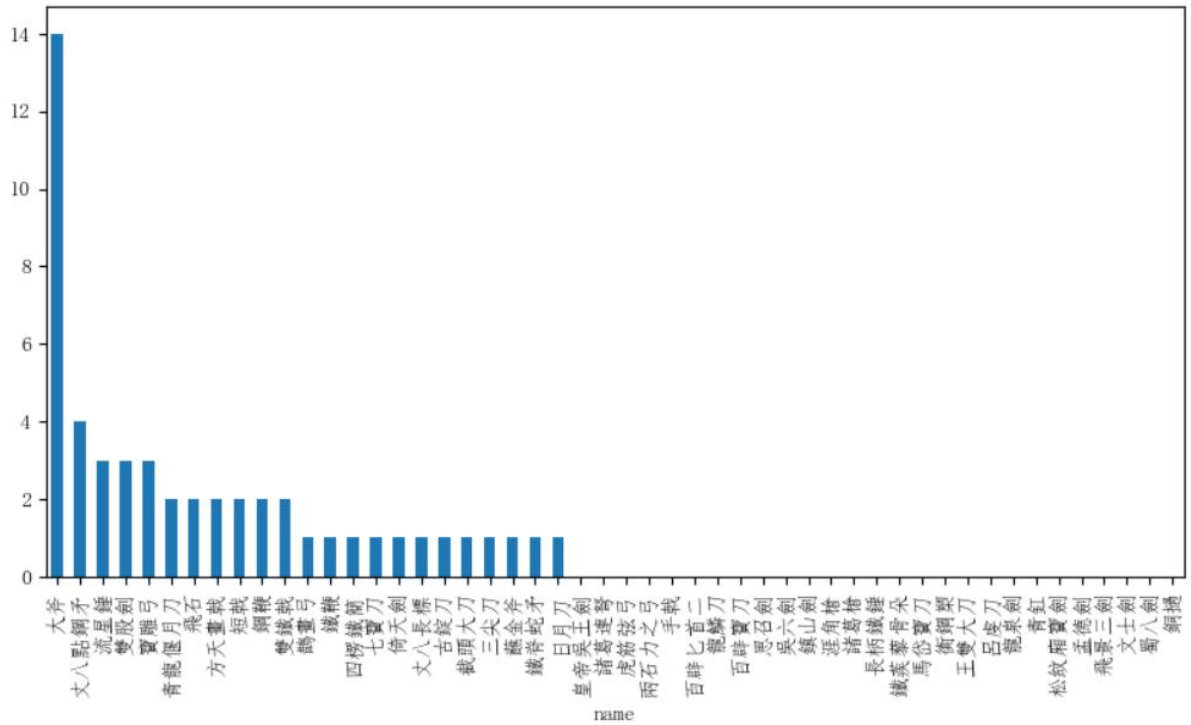
```
[28]: # 输出词频的前N个
for i in range(100):
    print(articlelist[i])
```

```
('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)
('孫堅', 31)
('不可', 31)
('次日', 26)
('商議', 25)
('引兵', 25)
('李儒', 25)
('天子', 24)
('左右', 23)
('玄德曰', 22)
('太師', 22)
('今日', 21)
('大喜', 21)
('軍士', 21)
('校尉', 21)
('太后', 20)
('何進', 20)
('王允', 20)
('二人', 19)
('司徒', 18)
('郭汜', 18)
('不能', 18)
('公孫瓚', 18)
('諸侯', 18)
('盧植', 17)
('如此', 17)
('軍馬', 17)
('何人', 17)
('百姓', 17)
('肅曰', 17)
('朱雋', 17)
('百官', 16)
('李肅', 16)
('張濟', 16)
('張飛', 15)
('皇甫嵩', 15)
('留王', 15)
('大事', 15)
('如何', 15)
('張角', 15)
('文醜', 15)
('何故', 15)
('劉表', 14)
('冀州', 14)
('正是', 14)
('刺史', 14)
('袁術', 14)
('因此', 14)
('此人', 14)
('張梁', 14)
('只見', 14)
('曹嵩', 13)
('五千', 13)
('不到', 13)
('樊稠', 13)
('是夜', 13)
('英雄', 13)
('五百', 13)
('不得', 13)
('將軍', 13)
('性命', 12)
('華雄', 12)
('李蒙', 12)
('人馬', 12)
('張讓', 11)
('相府', 11)
('一日', 11)
('趕來', 11)
('一人', 11)
('主公', 11)
('社稷', 11)
('大臣', 11)
('三十', 11)
('叔父', 11)
('蔡瑁', 11)
('丞相', 11)
('星夜', 11)
('不見', 10)
('三人', 10)
('相見', 10)
('一面', 10)
('不敢', 10)
('忽見', 10)
('不知', 10)
('下文', 10)
('問曰', 10)
('大呼', 10)
('分解', 10)
('只得', 10)
('何太后', 10)
('乘勢', 10)
```

第二章 词频统计与分析

2、指定类型词频统计

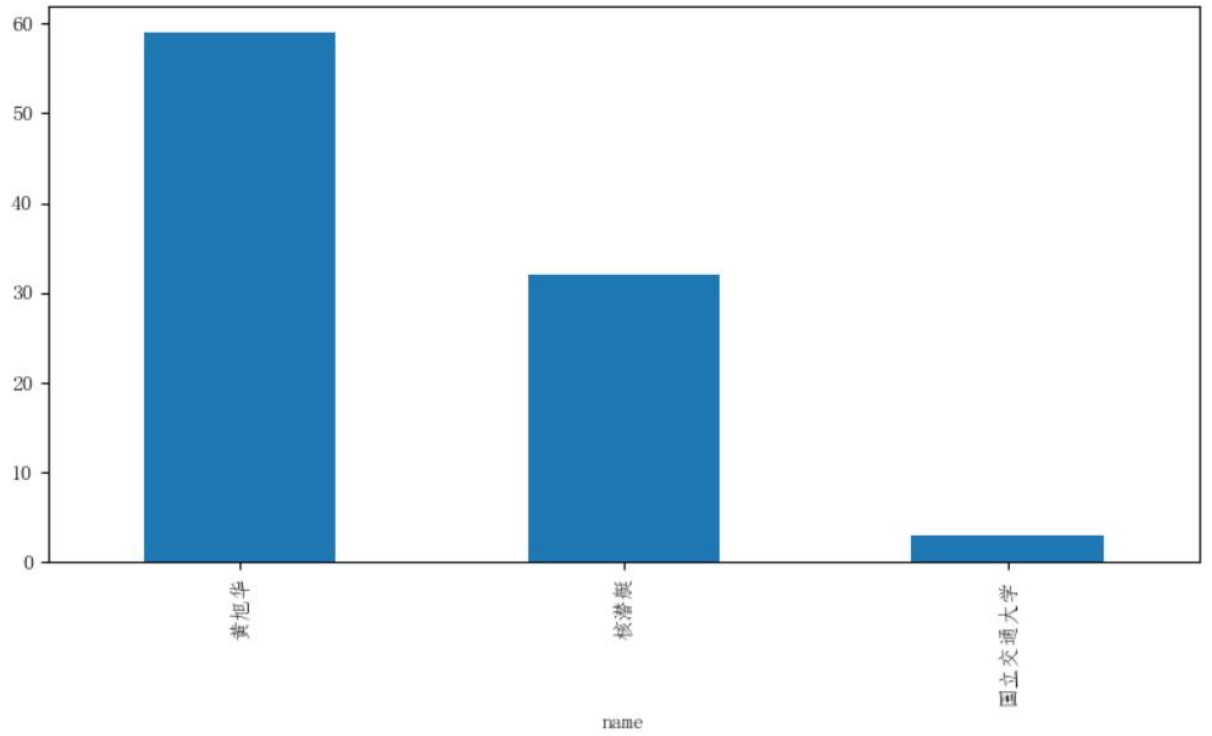




第二章 词频统计与分析

3、链接黄旭华

('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)('进行', 13)('院士', 13)('专业', 13)('技术', 12)('我国', 12)('研制', 12)('工程', 11)('访谈', 10)('介绍', 8)('科学', 8)('主要', 8)('第一代', 8)('人生', 7)('传记', 7)('历史', 7)('及其', 7)('思想', 7)('过程', 6)('一生', 6)('设计', 6)('成就', 6)('传主', 6)('要求', 6)('按照', 6)('研究', 6)('实物', 5)('精神', 5)('重点', 5)('求学', 5)('实现', 5)('先后', 5)('黄旭', 5)('轨迹', 4)('完整', 4)('船舶', 4)('依据', 4)('照片', 4)('系统', 4)('中国', 4)('其中', 4)('还原', 4)('重要', 4)('完成', 4)('事件', 4)('成熟', 4)('描述', 4)('任务', 4)('反映', 4)('保密', 4)('时间', 4)('学习', 4)('09', 4)('各种', 3)('大学', 3)('719', 3)('形成', 3)('客观', 3)('曲折', 3)('逐一', 3)('经历', 3)('国立交通大学', 3)('李世英', 3)('事业', 3)('提升', 3)('包括', 3)('本章', 3)('撰写', 3)('自己', 3)('代表', 3)('清单', 3)('突破', 3)('学生', 3)('一个', 3)('叙述', 3)('展示', 3)('参加', 3)('关于', 3)('处理', 3)('生活', 3)('同时', 3)('分析', 3)('负责', 3)('贡献', 3)('历程', 3)('计划', 3)('地下党', 3)('特点', 3)('直接', 3)('节点', 3)('国家', 3)('章节', 3)('回顾', 3)



第二章 词频统计与分析

4、论文阅读

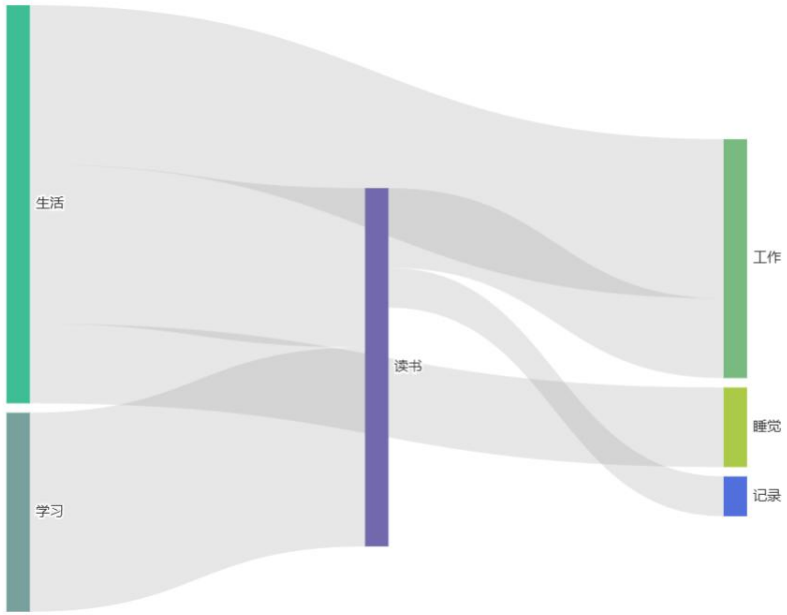
1).研究目的
探讨如何利用数字化图书追踪物理学巨匠的科学声誉演变，并检验是否存在“同群偏好”（科学家在本国或同语言群体中更受认可）。研究旨在超越传统引文分析，评估科学家在学术之外的文化影响力。
2).研究方法与数据
选取牛顿与爱因斯坦为主要案例，基于Google Books的3600万册数字化图书与Google Scholar的9100万学术项目数据，通过名字出现频次分析科学家的全球声誉。同时使用多语言语料与共现分析，探究声誉的地域差异及其主要贡献关联。
3).研究结果
全球范围内，爱因斯坦的学术声誉自20世纪中期起超过牛顿；但在英国英语语料中，牛顿始终更受关注，印证“同群偏好”。科学家的主要成就与其声誉高度相关。早期科学家在当代图书中仍被频繁提及，表明其科学影响的持久性。
4).主要结论
伟大科学家的思想影响力可跨越数世纪，其声誉在学术与公众文化中均长期存在。科学家的声誉受语言与国家背景调节，支持“同群偏好”假设。Google Books 与 Ngram Viewer 可作为补充性“替代计量”工具，用于衡量科学家在更广泛社会与文化中的影响。
5).展望
可扩展至其他学科，比较不同领域科学家的声誉模式。需区分学术声誉与公众声誉，并优化姓名消歧、语言偏差等问题。未来可结合更多元的数据源（如社交媒体、新闻）构建更立体的科学家影响力评估体系。核心贡献是首次结合大规模数字化图书与学术文献，量化分析了科学家长期声誉的演变及地域差异，为科学影响力研究提供了跨学科、跨文化的新视角。

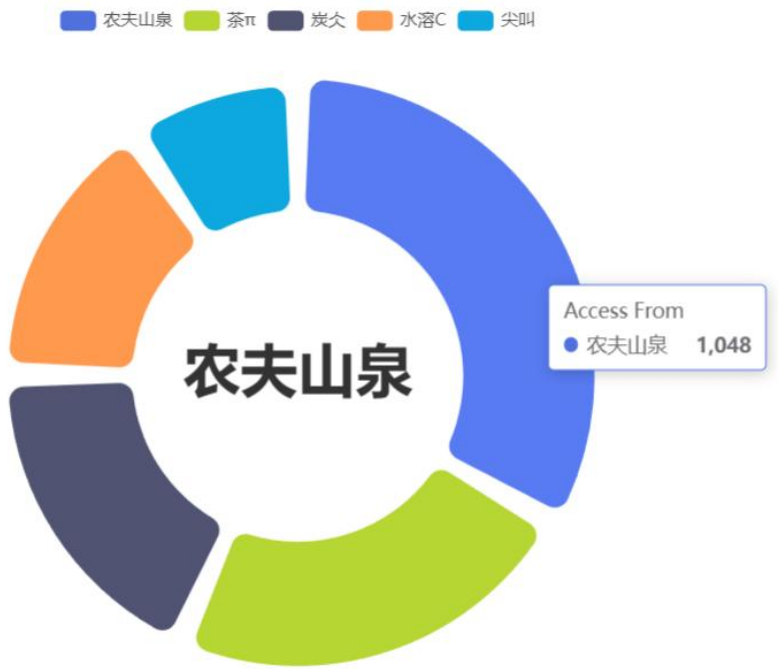
第三章 词云图

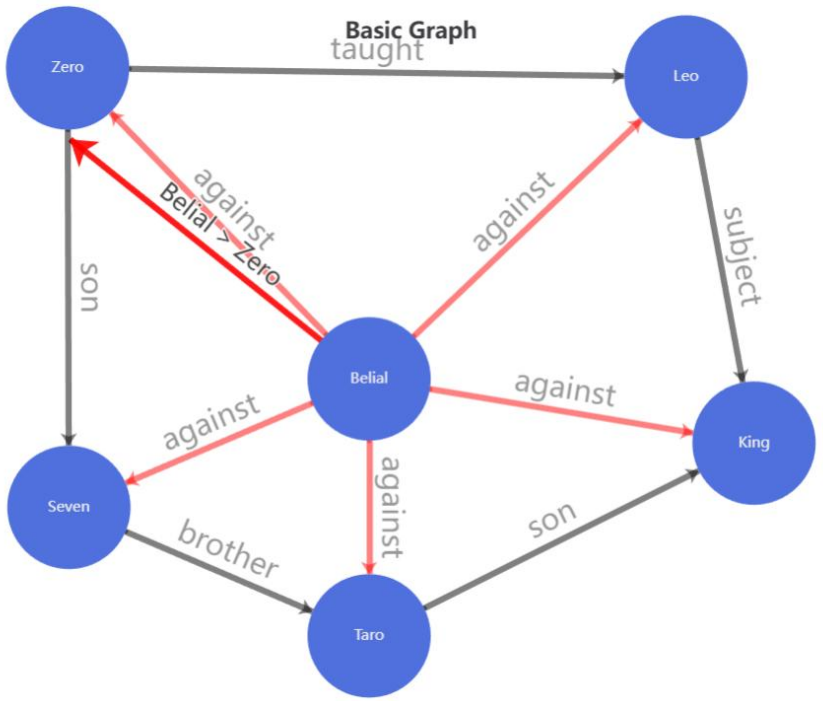
1、词云工具（微词云）

文本节选自京东平台农夫山泉产品下的评论，体现出买家对农夫山泉饮用水重点关注对象在使用方便、口感良好、水质优越等方便。

2、 Echarts应用

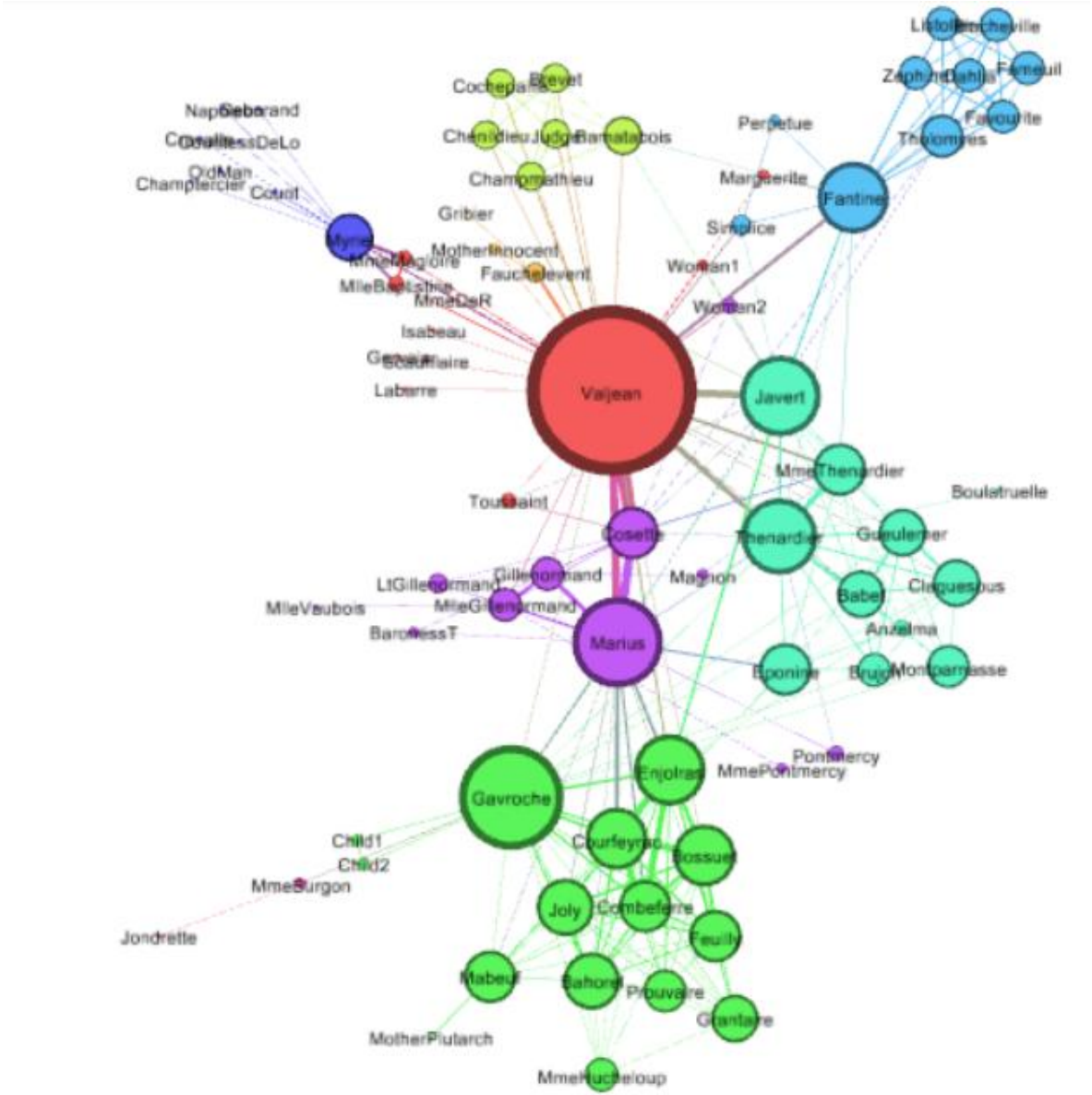






第三章 词云图

3、 Gephi利用已有数据集生成图谱



4、 功勋科学家黄旭华

第四章 情感分析

1、百度智能云情感分析

```
# 调用情感分析（添加延迟）
text = "苹果是一家伟大的公司"
result_senti = safe_sentiment_analysis(text, delay=1.0)

if result_senti:
    # 展示结果
    print("情感分析结果:")
    if 'items' in result_senti:
        item = result_senti['items'][0]
        print(f"情感极性: {item.get('sentiment', 'N/A')}")
        print(f"置信度: {item.get('confidence', 'N/A')}")
        print(f"正面概率: {item.get('positive_prob', 'N/A')}")
        print(f"负面概率: {item.get('negative_prob', 'N/A')}")
    else:
        print(f"API返回错误: {result_senti}")
```

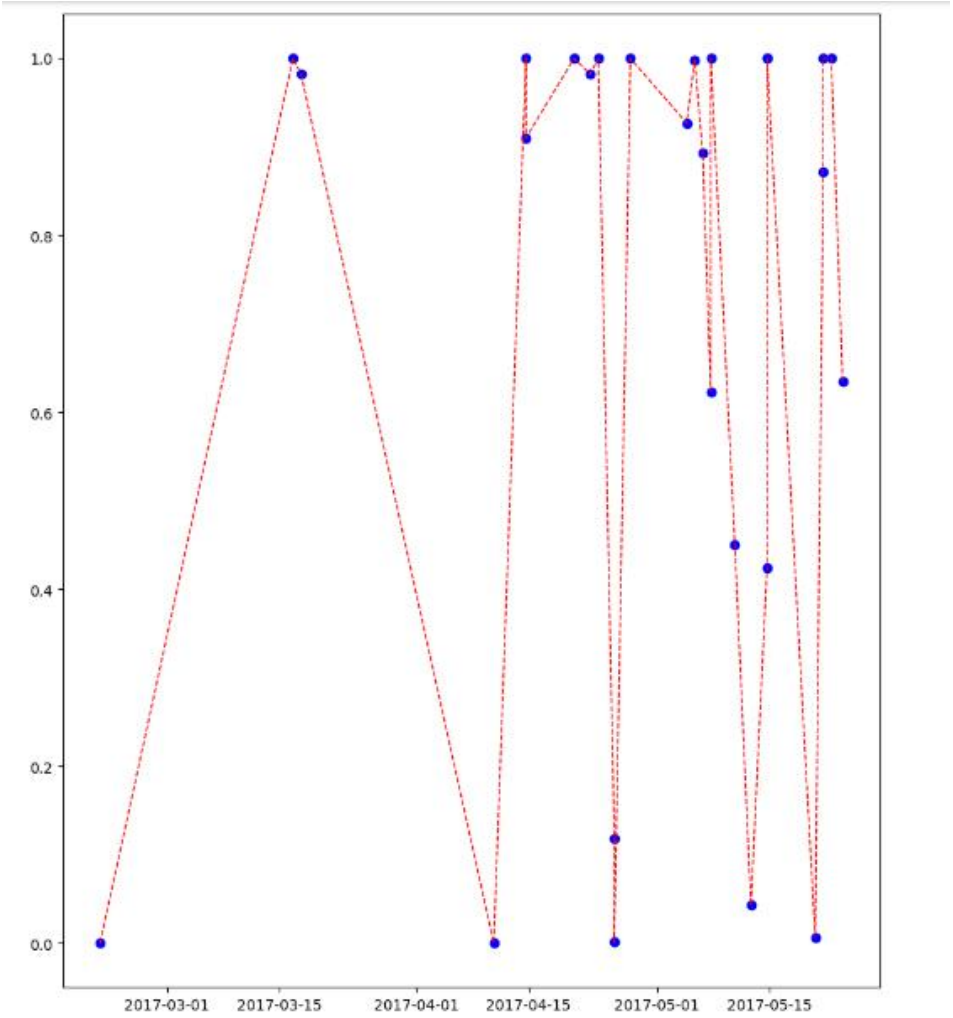
```
客户端创建成功
情感分析结果:
情感极性: 2
置信度: 0.997489
正面概率: 0.99887
负面概率: 0.00112987
```

修复代码

```
from aip import AipNlp
import time
APP_ID = '***'
API_KEY = '***'
SECRET_KEY = '***'
client = AipNlp(APP_ID, API_KEY, SECRET_KEY)
print("客户端创建成功")
# 设置请求延迟
def safe_sentiment_analysis(text, delay=1.0):
    """
    安全的API调用，避免QPS超限
    参数:
        text: 待分析文本
        delay: 延迟时间（秒），免费用户建议至少1秒
    """
    # 先等待指定的延迟时间
    time.sleep(delay)
    try:
        result = client.sentimentClassify(text)
        return result
    except Exception as e:
        print(f"API调用异常: {e}")
        return None
# 调用情感分析（添加延迟）
text = "苹果是一家伟大的公司"
result_senti = safe_sentiment_analysis(text, delay=1.0)
if result_senti:
    # 展示结果
    print("情感分析结果:")
    if 'items' in result_senti:
        item = result_senti['items'][0]
        print(f"情感极性: {item.get('sentiment', 'N/A')}")
        print(f"置信度: {item.get('confidence', 'N/A')}")
        print(f"正面概率: {item.get('positive_prob', 'N/A')}")
        print(f"负面概率: {item.get('negative_prob', 'N/A')}")
    else:
        print(f"API返回错误: {result_senti}")
```

第四章 情感分析

2、作业代码002



2、作业代码003

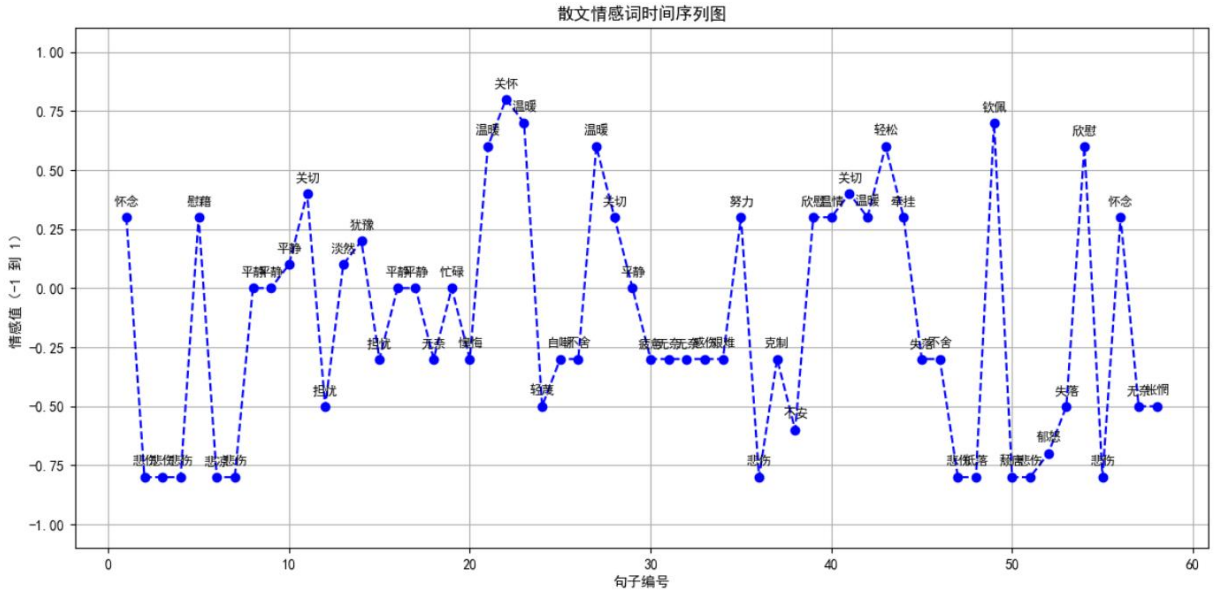
细粒度情感实体抽取结果:

{"实体": [{"部位": "全身","症状": "失眠""情感": "失去重心"},{"部位": "头部","症状": "头痛","情感": "疲乏无力"},{"部位": "皮肤","症状": "异常敏感，触碰如针扎疼痛","情感":"不适"},{"部位": "心脏/胸部","症状": "心慌、胸闷","情感": "惊恐"},{"部位": "背部","症状": "沉重如压石头","情感": "压抑"},{"部位": "感官系统","症状": "对光线和声音极度敏感","情感": "惊恐"},{"部位": "行为表现","症状": "不愿出门、不与人交流、关在屋里拉紧窗帘","情感": "生活无意义"}]}

第四章 情感分析

2、作业代码004

句子1：情感词：怀念，情感值：0.3
句子2：情感词：悲伤，情感值：-0.8
句子3：情感词：悲伤，情感值：-0.8
句子4：情感词：悲伤，情感值：-0.8
句子5：情感词：慰藉，情感值：0.3
句子6：情感词：悲凉，情感值：-0.8
句子7：情感词：悲伤，情感值：-0.8
句子8：情感词：平静，情感值：0.0
句子9：情感词：平静，情感值：0.0
句子10：情感词：平静，情感值：0.1
句子11：情感词：关切，情感值：0.4
句子12：情感词：担忧，情感值：-0.5
句子13：情感词：淡然，情感值：0.1
句子14：情感词：犹豫，情感值：0.2
句子15：情感词：担忧，情感值：-0.3
句子16：情感词：平静，情感值：0.0
句子17：情感词：平静，情感值：0.0
句子18：情感词：无奈，情感值：-0.3
句子19：情感词：忙碌，情感值：0.0
句子20：情感词：懊悔，情感值：-0.3
句子21：情感词：温暖，情感值：0.6
句子22：情感词：关怀，情感值：0.8
句子23：情感词：温暖，情感值：0.7
句子24：情感词：轻蔑，情感值：-0.5
句子25：情感词：自嘲，情感值：-0.3
句子26：情感词：不舍，情感值：-0.3
句子27：情感词：温暖，情感值：0.6
句子28：情感词：关切，情感值：0.3
句子29：情感词：平静，情感值：0.0
句子30：情感词：疲惫，情感值：-0.3
句子31：情感词：无奈，情感值：-0.3
句子32：情感词：无奈，情感值：-0.3
句子33：情感词：感伤，情感值：-0.3
句子34：情感词：艰难，情感值：-0.3
句子35：情感词：努力，情感值：0.3
句子36：情感词：悲伤，情感值：-0.8
句子37：情感词：克制，情感值：-0.3
句子38：情感词：不安，情感值：-0.6
句子39：情感词：欣慰，情感值：0.3
句子40：情感词：温情，情感值：0.3
句子41：情感词：关切，情感值：0.4
句子42：情感词：温暖，情感值：0.3
句子43：情感词：轻松，情感值：0.6
句子44：情感词：牵挂，情感值：0.3
句子45：情感词：失落，情感值：-0.3
句子46：情感词：不舍，情感值：-0.3
句子47：情感词：悲伤，情感值：-0.8
句子48：情感词：低落，情感值：-0.8
句子49：情感词：钦佩，情感值：0.7
句子50：情感词：颓唐，情感值：-0.8
句子51：情感词：悲伤，情感值：-0.8
句子52：情感词：郁怒，情感值：-0.7
句子53：情感词：失落，情感值：-0.5
句子54：情感词：欣慰，情感值：0.6
句子55：情感词：悲伤，情感值：-0.8
句子56：情感词：怀念，情感值：0.3
句子57：情感词：无奈，情感值：-0.5
句子58：情感词：怅惘，情感值：-0.5



第六章 关联数据与知识图谱

1、“美团大脑”分析

1). 整体架构与核心目标

定位：美团基于海量业务数据构建的本地生活领域超大规模知识图谱系统

数据源：

600万+合作商家信息；

用户搜索、下单、评价行为日志；

地理时空数据；

实时供需动态。

核心目标：

实现“人、货、场”的智能匹配；

提升搜索推荐精准度；

优化商家运营与平台治理效率。

2). 知识图谱构建方法论与技术特点

多模态知识融合：

结构化数据（商家信息、价格、SKU）

非结构化数据（用户评论、图片描述）

时空轨迹数据（配送路径、热力区域）

动态图谱更新机制：

实时捕捉商家状态变化（如营业时间、菜品上下架）

基于事件驱动的知识演化（如节日促销、天气影响）

关系推理与社区发现：

基于用户行为挖掘“相似商家”、“同好圈子”

构建“用户-偏好-场景”的深层关联网络

第六章 关联数据与知识图谱

1、“美团大脑”分析

3). **最新应用案例：**智能套餐设计与供应链优化

背景：2023年美团重点推进“套餐化运营”，提升客单价与用户满意度

实施方式：

利用知识图谱分析历史订单，识别高频组合菜品（如“披萨+可乐”“烤鱼+啤酒”）

结合商圈特征、时段偏好、消费能力，动态生成个性化套餐

为商家提供备货建议，降低损耗率

效果：

试点商家套餐购买率提升24%

平均客单价增长18%

供应链浪费降低约7%

第六章 关联数据与知识图谱

1、“美团大脑”分析

4). 评价与展望

优势：

从“连接信息”到“理解场景”，实现真正意义上的情境化服务

形成“数据驱动运营 - 运营反馈数据”的闭环迭代

在多业务线（外卖、到店、酒旅）中实现知识共享与交叉赋能

挑战：

数据实时性与一致性维护成本高

中小商家数字化程度不均，知识注入存在断层

用户隐私与数据安全边界需持续明晰

展望：

探索“图谱+大语言模型”融合，实现自然交互式生活助手

向行业开放部分图谱能力，构建本地生活数字生态

拓展至智慧城市、区域经济分析等社会级应用

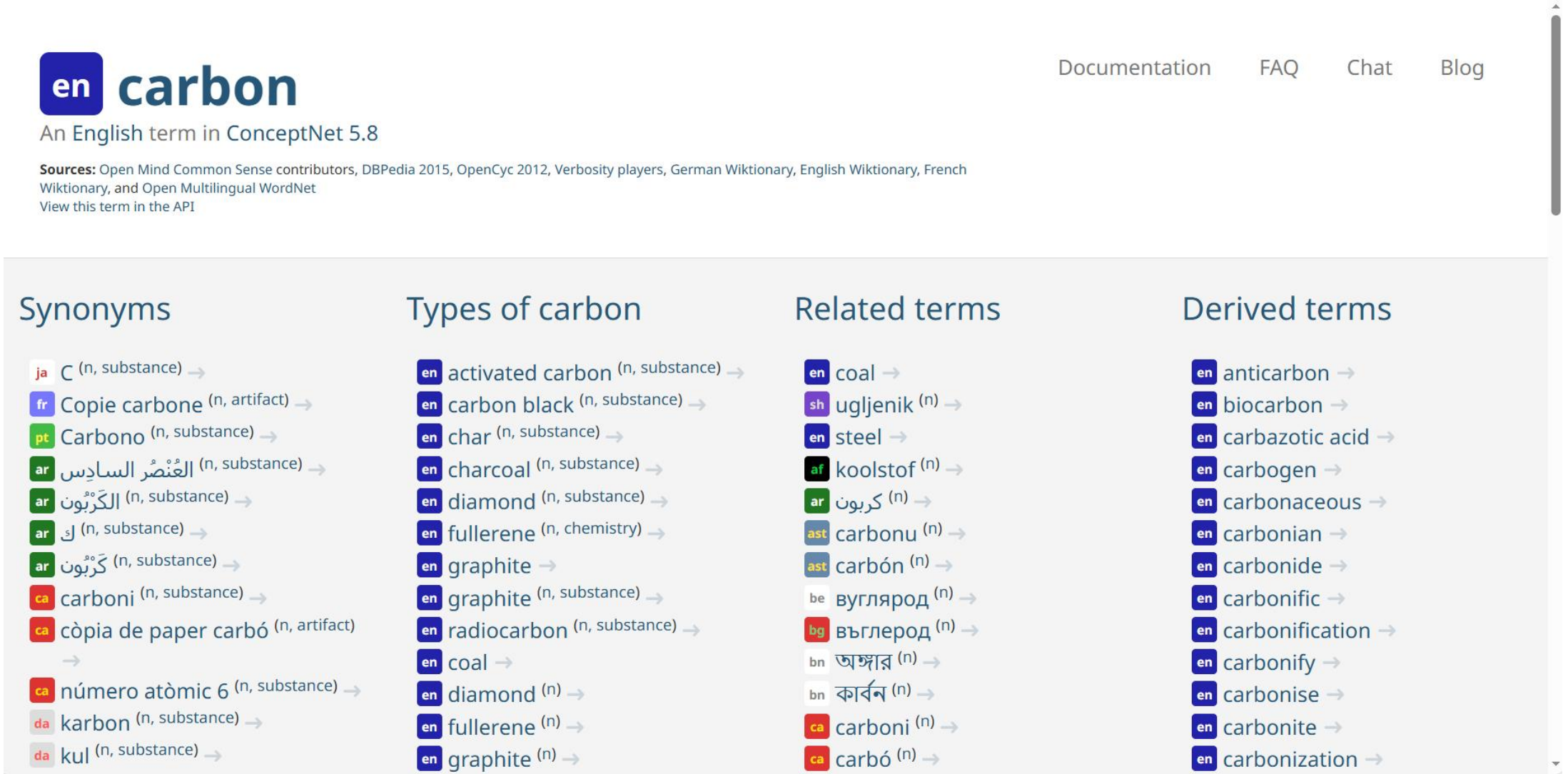
第六章 关联数据与知识图谱

1、“美团大脑”分析

5). 小结

美团大脑不仅是一个技术系统，更是驱动本地生活行业数字化转型的“核心知识引擎”。其以业务场景为导向、以动态图谱为底座、以智能应用为出口的构建路径，为垂直领域知识图谱落地提供了可借鉴的范式。未来，如何平衡平台治理与生态开放、数据价值与用户隐私，将是其持续进化的关键课题。

2、在线工具应用（concept.io搜索carbon）



en carbon

An English term in ConceptNet 5.8

Sources: Open Mind Common Sense contributors, DBPedia 2015, OpenCyc 2012, Verbosity players, German Wiktionary, English Wiktionary, French Wiktionary, and Open Multilingual WordNet
View this term in the API

Documentation FAQ Chat Blog

Synonyms

- ja C (n, substance) →
- fr Copie carbone (n, artifact) →
- pt Carbono (n, substance) →
- ar العُنْصُر السادِس (n, substance) →
- ar الكَرْبُون (n, substance) →
- ar ك (n, substance) →
- ar كَرْبُون (n, substance) →
- ca carboni (n, substance) →
- ca còpia de paper carbó (n, artifact) →
- ca número atòmic 6 (n, substance) →
- da karbon (n, substance) →
- da kul (n, substance) →

Types of carbon

- en activated carbon (n, substance) →
- en carbon black (n, substance) →
- en char (n, substance) →
- en charcoal (n, substance) →
- en diamond (n, substance) →
- en fullerene (n, chemistry) →
- en graphite →
- en graphite (n, substance) →
- en radiocarbon (n, substance) →
- en coal →
- en diamond (n) →
- en fullerene (n) →
- en graphite (n) →

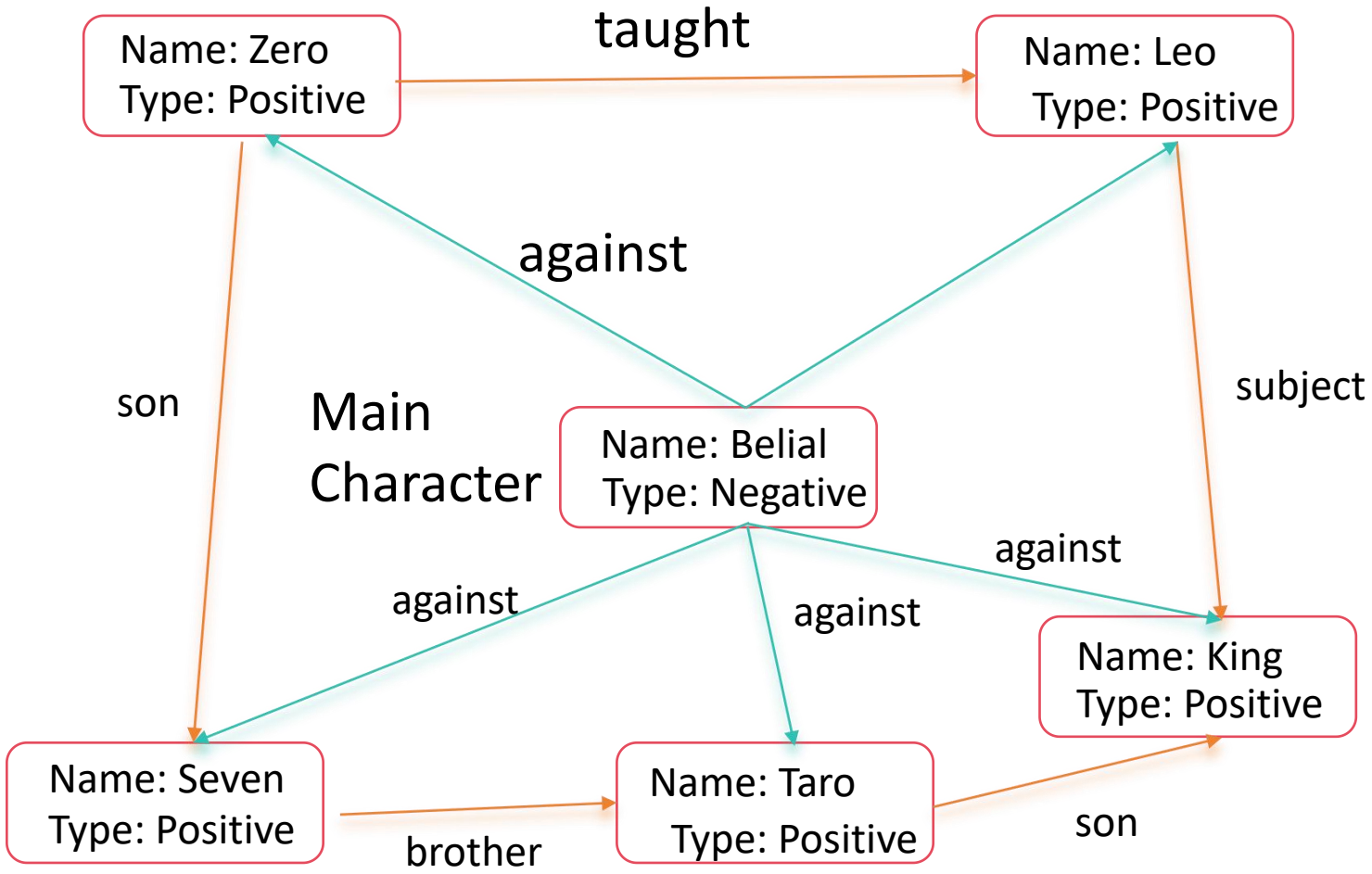
Related terms

- en coal →
- sh ugljenik (n) →
- en steel →
- af koolstof (n) →
- ar كربون (n) →
- ast carbonu (n) →
- ast carbón (n) →
- be вуглярод (n) →
- bg въглерод (n) →
- bn অঙ্গার (n) →
- bn কার্বন (n) →
- ca carboni (n) →
- ca carbó (n) →

Derived terms

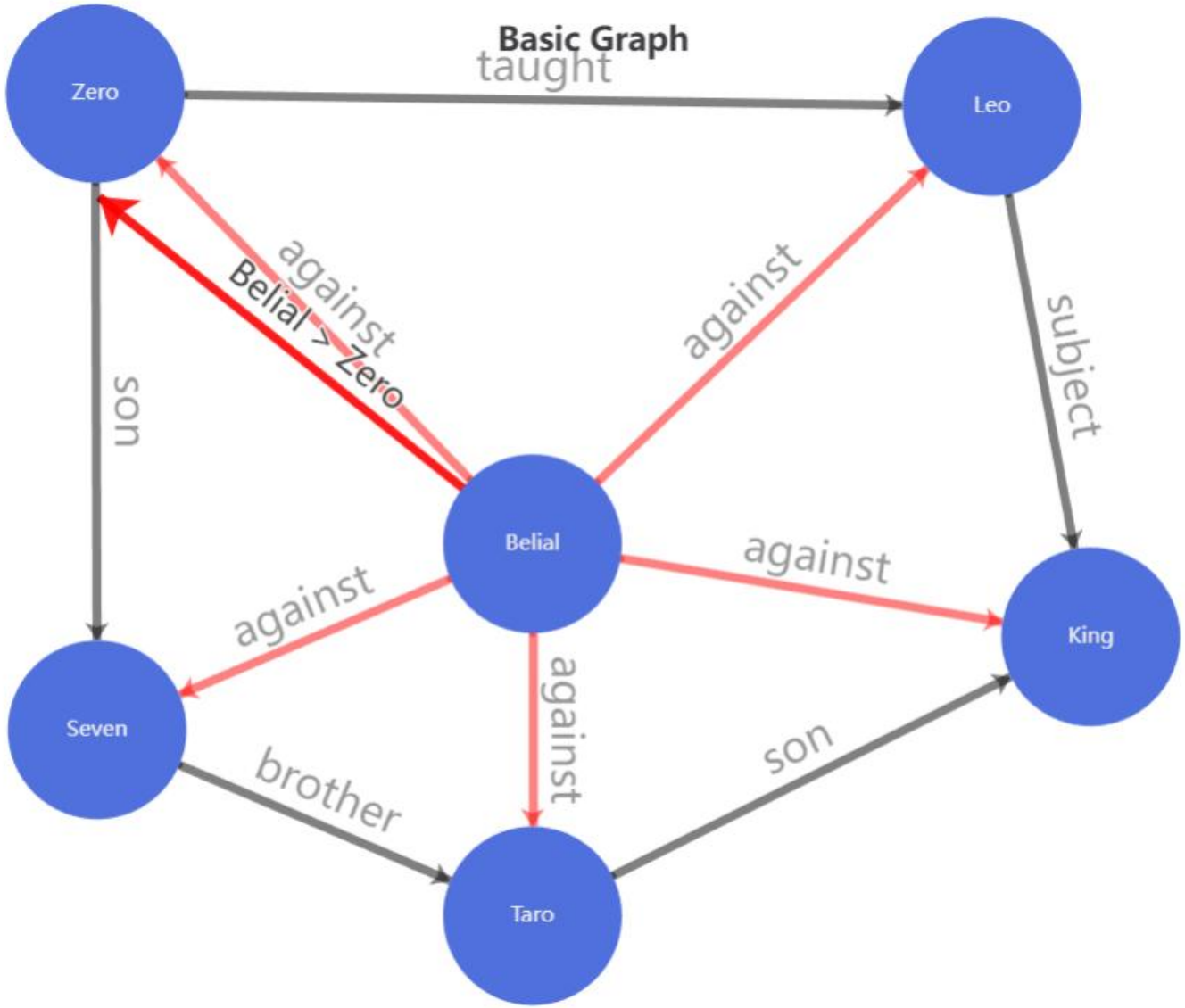
- en anticarbon →
- en biocarbon →
- en carbazotic acid →
- en carbogen →
- en carbonaceous →
- en carbonian →
- en carbonide →
- en carbonific →
- en carbonification →
- en carbonify →
- en carbonise →
- en carbonite →
- en carbonization →

3、白板建模


Name: Zero
Type: Positive
taught
Name: Leo
Type: Positive
against
son
Main
Character
Name: Belial
Type: Negative
subject
against
against
against
Name: King
Type: Positive
Name: Seven
Type: Positive
Name: Taro
Type: Positive
brother
son

第六章 关联数据与知识图谱

4、 Echarts建模



ECHARTS 首页 文档 下载 示例 资源 社区

代码编辑 完整代码 配置项

JS TS 运行

```
234          relation: {
235            name: "son",
236            id: "1"
237          },
238          lineStyle: {
239            color: "black",
240            width: 5,
241            curveness: 0.0
242          }
243        },
244        {
245          source: "Leo",
246          target: "King",
247          label: {
248            show: true,
249            position: "middle",
250            fontSize: 24,
251            formatter: function(params) {
252              return params.data.relation.name;
253            }
254          },
255          relation: {
256            name: "subject",
257            id: "1"
258          },
259          lineStyle: {
260            color: "black",
261            width: 5,
262            curveness: 0.0
263          }
264        }
265      ]  // links数组结束
266    }  // series数组的第一个元素结束
267  ]  // series数组结束
268};  // option对象结束
```

第六章 关联数据与知识图谱

5、neo4j使用