信息处理技术大作业简介

文件介绍

- Final_project
  - **final_project.py**【项目源代码】
  - **lib**
    - First.txt【第一层分类字典】
    - Second.txt【第二层分类字典】
    - united(only used in test).txt
  - **out**
    - result.txt【输出结果txt文本未去重】
    - result.xlsx【输出结果xlsx已去重】
  - **scrapy**
    - ori_news.txt【爬虫结果txt文本经过正则处理】
    - scraber.py【爬虫源代码】

算法思路

选取字典

自定义词典并且按照类别的不同分成两层，二者求和作为最终特征值，如果同一层有多重特征值则求平均（尽管这会导致分类的不准确，但往往特征值会比较接近）：

- First.txt
  - 用于归类主要类别，包括思政、获奖与会议活动、科研、晋升
  - 每一类开头均为整百，其余空格用“/n”替代
  - 以字典中排序序号*1000000作为特征值

- Second.txt
  - 用于归类次要类别，包括院系、国内地区、国际地区、国际组织
  - 每一类开头均为整百，其余空格用“/n”替代
  - 以字典中排序序号*1000作为特征值

K-means算法的调整

当选择等距选取或随机选取，最终结果会导致部分类的不均匀，因此选择按照给定的list选取起始centroid，list共有10个元素，当选取k值小于等于10，则在list中选取；当选取k值大于10，超过10个的部分在整体中等距选取：

```
1.  list = [50000000, 150000000, 250000000, 350000000, 450000000, 50000, 15
    0000, 250000, 350000, 450000]
```

文本处理

- 源文本输入后使用正则匹配去除标点
- 字典输入后重新按照python内置sort函数排序，由于我们还需要反向查找某个特征词原本属于的特征词类，因此选择新建dictionary，同时包含特征词和原本序号（dictionary的下标序号表达现在的新序号，新序号便于对半查找）

结果分析

本次测试包含 `k = 5/8/10/15` 四种k值选择下的输出，最终选取了`k = 15`的结果作为分析样本

- result.xls
  - 共含有2980条数据
  - ori_test原有3394条数据，去重后只剩下了2980条
  - 分类结果总体很一般：
    - 类0 -- 召开会议
    - 类1 -- 成立某些机构、创立项目
    - 类2 -- 国际交流项目、学术研讨会
    - 类3 -- 竞赛佳绩、获奖
    - 类4 -- 学术科研成果
    - 类5 -- 院系新闻
    - 类6 -- 校区相关
    - 类7 -- 产学研结合、调研、成果转化、中标投产
    - 类8 -- 大人物访问考察
    - 类9 -- 党组织生活、座谈会、交流会、研讨会
    - 类10 -- 入选、晋升、院士相关、外事
    - 类11 -- 国际学术交流
    - 类12 -- 党中央精神学习
    - 类13 -- 院系新闻
    - 类14 -- 无法分类的新闻报道
- result.txt
  - 共含有3394条数据
  - 内容见上