信 息 内 容 安 全 实 验 报 告

实验项目名称： 话题检测与分析

班级： SC011701

姓名： 程瑞淇（2017302241）、李向娟（2017302235）、王燕颉（2017302234）、曹至杰（2017302233）

成员分工： 程瑞淇：分词和词频统计代码编写、整理

李向娟:理论资料查找、KMeans 聚类部分代码编写

王燕颉:报告撰写、理论资料查找收集

曹至杰：TF-IDF 部分代码编写，资料查找

指导教师： 杨黎斌

实验时间： 2020.3.3

目录

话题检测和分析

1、实验目的

①掌握话题检测和分析的方法；

③能够设计合理的指标对性能进行衡量。

2、理论依据

2.1jieba 分词

Jiaba 分词主要有三步：构建词典；构建有向无环图；计算最大概率路径。

1.构建前缀词典：

基于统计词典构造前缀词典，统计词典有三列，第一列是词，第二列是词频，第三列是词性

2.构建有向无环图：

根据前缀词典对输入文本进行切分，比如“北”，有北、北京、北京大学三种划分方式。因此，对于每个字，可以构建一个以位置为 key，相应划分的末尾位置构成的列表为 value 的映射。

3.最大概率路径计算

在得到所有可能的切分方式构成的有向无环图后，我们发现从起点到终点存在多条路径，多条路径也就意味着存在多种分词结果。需要计算最大概率路径。

计算最大概率路径时，jieba 采用从后往前的方式，采用动态规划计算最大概率路径。每到达一个节点，它前面的节点到终点的最大路径概率就已经计算出来。

2.2TF-IDF 计算

1）.计算词频 TF

TF = 某个词在文章中出现的次数

考虑到文章有长短之分，为了便于不同文章的比较，进行词频标准化。

$$TF = \frac{某个词在文章中出现的次数}{文章的总词数}$$

2）.计算逆文档频率 IDF

$$IDF = \log\left(\frac{语料库的文档总数}{包含该词的文档数+1}\right)$$

3）.计算 TF-IDF

TF-IDF = TF × IDF

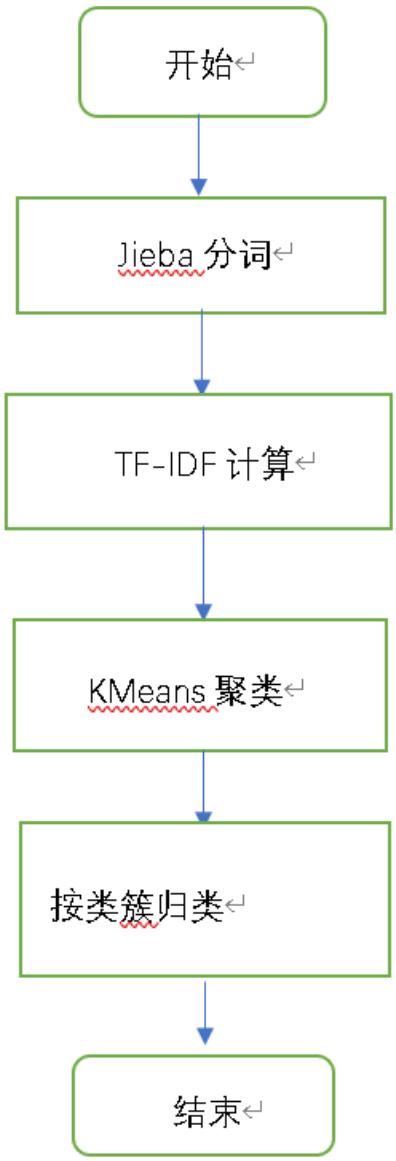
2.3KMeans 聚类

k 均值聚类算法（k-means clustering algorithm）是一种迭代求解的聚类分析算法，其步骤是，预将数据分为 K 组，则随机选取 K 个对象作为初始的聚类中心，然后计算每个对象与各个种子聚类中心之间的距离，把每个对象分配给距离它最近的聚类中心。聚类中心以及分配给它们的对象就代表一个聚类。每分配一个样本，聚类的聚类中心会根据聚类中现有的对象被重新计算。这个过程将不断重复直到满足某个终止条件。终止条件可以是没有（或最小数目）对象被重新分配

给不同的聚类，没有（或最小数目）聚类中心再发生变化，误差平方和局部最小。

3、代码实现过程

3.1 系统设计流程图



1、 使用 jieba 分词对文本进行中文分词，同时插入字典关于关键词；

2、 scikit-learn 对文本内容进行 TF-IDF 计算并构造 N*M 矩阵(N 个文档 M 个特征词)；

3、 使用 K-means 进行文本聚类(省略特征词过来降维过程)；

4、 对聚类的结果进行简单的文本处理，按类簇归类，也可以计算 P/R/F 特征值。

3.2 使用 jieba 中文分词

中文分词主要使用的是 Python+Jieba 分词工具，同时导入自定义词典 dict_baidu.txt。

jieba 中文分词涉及到的算法包括：

(1) 基于 Trie 树结构实现高效的词图扫描，生成句子中汉字所有可能成词情况所构成的有向无环图 (DAG)；

(2) 采用了动态规划查找最大概率路径，找出基于词频的最大切分组合；

(3) 对于未登录词，采用了基于汉字成词能力的 HMM 模型，使用了 Viterbi 算法。

jieba 中文分词支持的三种分词模式包括：

(1) 精确模式：试图将句子最精确地切开，适合文本分析；

(2) 全模式：把句子中所有的可以成词的词语都扫描出来，速度非常快，但是不能解决歧义问题；

(3) 搜索引擎模式：在精确模式的基础上，对长词再次切分，提高召回率，适合用于搜索引擎分词。

在本实验中，我们利用 jieba 精确模式进行分词,将每个文件夹下文档分词结果存储在同一个 txt 文件中。

3.3 词频统计和 TF-IDF 计算

sklearn 里面的 TF-IDF 主要用到了两个函数：CountVectorizer()和 TfidfTransformer()。

CountVectorizer 是通过 fit_transform 函数将文本中的词语转换为词频矩阵。

通过 get_feature_names()可看到所有文本的关键字，通过 toarray()可看到词频矩阵的结果。

利用 TfidfTransformer()直接计算每个词语的 tf-idf 权值并转化为词频矩阵，元素 w[i][j]表示 j 词在 i 类文本中的 tf-idf 权重，将计算结果存储在（Tfidf_result）文件中。

3.4Kmeans 聚类

主要通过调用 sklearn.cluster.KMeans 这个类完成。

初始化分类器，根据不同的算法，需要给出不同的参数。

（1）对于 K 均值聚类，我们需要给定类别的个数 n_cluster，默认值为 8；

（2）max_iter 为迭代的次数，这里设置最大迭代次数为 300；

（3）n_init 设为 10 意味着进行 10 次随机初始化，选择效果最好的一种来作为模型；

（4）init='k-means++' 会由程序自动寻找合适的 n_clusters；

（5）tol：float 形，默认值= 1e-4，与 inertia 结合来确定收敛条件；

（6）n_jobs：指定计算所用的进程数；

（7）verbose 参数设定打印求解过程的程度，值越大，细节打印越多；

（8）copy_x：布尔型，默认值=True。当我们 precomputing distances 时，将数据中心化会得到更准确的结果。如果把此参数值设为 True，则原始数据不会被改变。如果是 False，则会直接在原始数据

上做修改并在函数返回值时将其还原。但是在计算过程中由于有对数据均值的加减运算，所以数据返回后，原始数据和计算前可能会有细小差别。

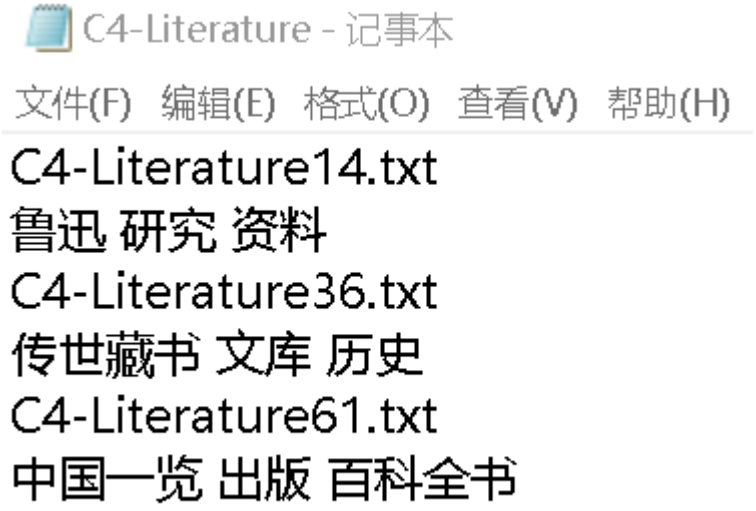
3.5 结果处理

合并实体名称和类簇：读取包含实体名的 txt 文件和聚类结果，用 lable 存储类标和类，content 存储实体名称，合并结果保存至新的

txt 文件中；合并类簇其中实体类名和类标一一对应，并定义定长字符串数组，对应类簇，统计同一类标下的实体名，并输出每个簇。

4、实验结果及分析

聚类结果如图：



C7-History - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

C7-History031.txt
改革开放 经济 中国改革开放与展望
C7-History092.txt
小说 话题 年代
C7-History161.txt
曾国藩 历史 小说

C17-Communication - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

C17-Communication22.txt
信息 邮电 韩国
C17-Communication32.txt
中韩 光缆 通信
C17-Communication48.txt
通信 外资 四川

C34-Economy - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

C34-Economy0026.txt
经济 资金 费用
C34-Economy0107.txt
经济 公司 美元
C34-Economy0248.txt
经济 日本 失业


C39-Sports - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
C39-Sports0027.txt
管理 体育 问题
C39-Sports0298.txt
健身健美 研究 训练