lab2

1. 目录结构

```
lab2/
|  WordCloud.py              // 可运行程序
|
├─article                    // 存放文档集
|      1.txt
|      2.txt
|      3.txt
|      4.txt
|      5.txt
|      6.txt
|      7.txt
|      8.txt
|      9.txt
|      10.txt
|      11.txt
|      12.txt
|      13.txt
|      14.txt
|      15.txt
|      16.txt
|      17.txt
|      18.txt
|      19.txt
|      20.txt
|
├─dict                       // 存放词典
|      hit_stopwords.txt     // 停用词词典
|
├─picture                    // 存放背景图片
|      china_map.jpg
|
└─result                     // 存放实验结果
        wordcloud.png        // 词云图
```

2. 实现细节

2.1. 分词

- `read_file(path)`：**读取文档集**

  对于文档集中的 20 个文档逐个读取，逐行读取文本，去除换行符和空行。注意在读取时需要设置 `encoding='utf-8'`，否则会出错。

- `read_stopwords(path)`：**读取停用词词典**

  使用哈工大停用词表，下载地址为 https://github.com/goto456/stopwords/blob/master/hit_stopwords.txt。

- `segment_word(articles, stopwords)`：**分词**

  调用 jieba 库，对于每个文档的每个句子进行分词，对分词结果进行去除停用词和单字的处理。

2.2. 词频统计

- `calculate_tf_idf(word, count, total, articles)`：**计算 TF-IDF 指标**

  对于一个词，分别计算其词频（TF）和逆文档频率（IDF），再计算 TF-IDF 指标值。

  词频（TF）表示在一个文档中出现的频率，即一个词在一个文档中出现的次数与文档中总词数的比例，计算公式如下：

$$TF(t,d) = \frac{\text{词 } t \text{ 在文档 } d \text{ 中出现的次数}}{\text{文档 } d \text{ 中的总词数}}$$

  逆文档频率（IDF）表示一个词对于语料库中所有文档的普遍重要性，即一个词在语料库中出现的文档数的倒数的对数，计算公式如下：

$$IDF(t,D) = \log\left(\frac{\text{语料库 } D \text{ 中的文档总数}}{1 + \text{语料库中包含词 } t \text{ 的文档数}}\right)$$

- `calculate_word_frequency(segmentation)`：**统计词频**

  调用 collections 库的 Counter 对词进行计数，再调用 `calculate_tf_idf` 函数获得每个词的 TF-IDF 指标值。

2.3. 绘制词云

- `plot_wordcloud(word_frequency, path)`：**绘制词云图**

  调用 wordcloud 库绘制词云图。首先将字典中存储的“词—TF-IDF”键值对转换成符合 wordcloud 库要求的格式。为了解决中文显示问题，需要设置 `font_path=r'C:\Windows\Fonts\msyh.t`

`tc'` （我选择了微软雅黑 常规字体，更换为其他字体需要修改为相应路径）。为了自定义背景图片，需要设置 `mask=plt.imread("picture/china_map.jpg")` （更换为其他背景图片需要修改为相应路径）。然后创建 WordCloud 对象并根据“词—TF-IDF”键值对生成词云图。如果要根据图片色设置字体颜色，需要在读取背景图片后调用 `ImageColorGenerator` 提取颜色，然后调用 `recolor` 函数对创建好的 WordCloud 对象进行字体颜色设置。

3. 创新点

- 绘制词云图时自定义中文字体，自定义背景图片，根据图片色设置字体颜色。