lab6

1. 目录结构

```
lab6/
│  functions.py              // 主调用程序
│  main.py                   // 可运行程序
│
├─article                    // 存放文档集
│
├─dict                       // 存放词典
│      hit_stopwords.txt     // 停用词词典
│
└─result                     // 存放实验中产生的结果文件
        champion_list.json   // 胜者表
        low_list.json        // 低端表
```

2. 实现细节

2.1. task1

- `read_file(path)`：**读取文档**

  对于文档集中的 10000 个文档逐个读取，逐行读取文本，去除换行符和空行。

- `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)`：**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、标点，数字和单字的处理。

- `build_inverted_index(segmentation)`：**建立倒排索引并统计词项频率**

  每个词都使用一个字典按照“文档 id – 词项频率”的格式建立倒排索引表。

- `build_champion_list(inverted_index, r)`：**建立胜者表和低端表**

对每个词，使用优先队列按词项频率选取最高的 r 篇文档建立胜者表，其余组成低端表。胜者表和低端表中均额外保存有该词的文档频率。

- `save_json_file(data, path)`：**保存 json 文件**

  将胜者表和低端表分别保存为 json 文件。

2.2. task2

- `load_json_file(path)`：**加载 json 文件**

  加载 task1 保存的胜者表和低端表。

- `champion_list_retrieval(keywords, k, champion_list)`：**胜者表检索**

  读取所有查询词对应的胜者表，取其交集。如果交集中正好有 k 个文档 id，直接返回；如果交集中多于 k 个文档 id，需要按照 TF–IDF 值选取最高的 k 个文档 id；如果交集中少于 k 个文档 id，需要继续进行低端表检索，并将两次检索结果合并。

  对于查询词集合 $T = [t_1, t_2, \ldots, t_n]$ 和文档 d ，TF–IDF 值的计算公式如下：

  $$\text{TF-IDF}_{T,d} = \sum_{i=1}^{n} (1 + \log\ tf_{t_i,d}) \cdot \log \frac{N}{df_{t_i}}$$

  其中， N 为文档集中的文档总数。

- `low_list_retrieval(keywords, k)`：**低端表检索**

  读取所有查询词对应的低端表，取其交集，再从交集中随机选取 k 个文档 id。选择随机抽取结果的方式，是因为低端表中的文档数量多且 TF–IDF 值一般均较低，先排序后选取或优先队列选取的代价太大。

- `respond(docIDs)`：**返回检索结果**

  如果检索到的 docID 数量为 0，则打印“抱歉，没有与此相关的检索结果”；否则，将检索到的 docID 依次打印出来，同时打印检索到的 docID 的总数。

- `if __name__ == '__main__':`：**main 函数**

  使用一个 while (true) 循环，首先接收用户输入的查询词集合，然后接收用户输入的 Top K 的 K 值。接下来记录检索开始时间，并调用 `champion_list_retrieval` 函数进行检索，检索完成后返回检索结果，记录检索结束时间，计算检索时间，打印本次检索用时，最后询问用户是否想要继续检索（是/否），如果用户输入“否”，则跳出循环，否则重复该流程。

3. 创新点

- 结合了高端表、低端表（索引分层）思想和胜者表算法，同时在低端表检索中采用随机抽取结果的方式，进行了查询优化，大幅降低了检索时间。
- 使用 TF-IDF 值作为最终选择 Top-K 文档的指标依据，突出罕见词项的的查询作用。

```
############################################################

robot：您好，请输入关键词
user：中国 上海 北京
robot：好的，请设置 K 的值
user：2
robot：好的，接下来将为您检索非精确的 Top 2 文档
robot：共 2 个检索结果： 3312.txt 2120.txt
robot：本次检索用时 0.0 秒

robot：请问您是否想要继续检索（是/否）
user：是

############################################################

robot：您好，请输入关键词
user：中国 上海 北京
robot：好的，请设置 K 的值
user：3
robot：好的，接下来将为您检索非精确的 Top 3 文档
robot：共 3 个检索结果： 3312.txt 2120.txt 9084.txt
robot：本次检索用时 0.07201623916625977 秒

robot：请问您是否想要继续检索（是/否）
user：否
```