Exp1实验报告

PB19000046 曹奕阳

PB19030827 崔晨宇

使用的算法

- bool查询

  参考https://github.com/pskrunner14/info-retrieval，分析查询词项时用到了Shunting Yard算法，以处理含有多个逻辑词与括号的查询。

- 语义查询

  python中sklearn库自带的相关算法。

运行示例与分析

- bool查询

  以原始数据集中两个文件夹(2018_01与2018_02)的文件作为bool查询的数据集，第一次运行结果如下：

```
 The running time of building the posting list: 3533.9615 secs
 Input your query: (AND, OR, NOT, '(' and ')' are allowed in the query.)
 market AND technology And oil
 ERROR: Invalid Query. Please check query syntax. #And不符合查询格式
 Continue? (y/n) y
 Input your query: (AND, OR, NOT, '(' and ')' are allowed in the query.)
 market AND technology AND oil
 Searching time: 0.0170 secs
 Doc IDs:  [76, 132, 228, 409, 549, 1023, 1191, 1259, 1304, 1688, 1813,
1846, 1977, 2047, 2131, 2174, 2213, 2370, 2433, 2598, 2681, 2702, 2839,
2974, 3070, 3079, 3216, 3260, 3281, 3287, 3351, 3411, 3476, 3517, 3551,
3760, 3779, 3787, 4094, 4689, 4715, 4794, 4863, 4939, 5012, 5828, 6123,
6180, 6249, 6410, 6509, 6619, 6665, 6708, 6744, 6872, 7048, 7092, 7196,
7346, 7379, 7920, 7963, 8153, 8423, 8495, 8521, 8672, 8736, 8894, 8921,
9084, 9204, 9675, 9730, 9923, 10032, 10384, 10585, 10672, 10709, 10913,
10920, 10930, 11103, ..., 121069, 121082, 121202, 121396, 121567, 121723,
121743, 121820, 121970, 122010, 122172, 122317] #篇幅所限，省略部分输出结果
 Continue? (y/n) y
 Input your query: (AND, OR, NOT, '(' and ')' are allowed in the query.)
 operation AND (NOT risk OR (chief AND NOT security))
 Searching time: 62.7422 secs
 Doc IDs:  [10, 28, 67, 72, 75, 83, 114, 119, 132, 157, 171, 174, 182, 185,
189, 200, 238, 252, 259, 273, 275, 291, 295, 304, 329, 333, 338, 348, 361,
384, 389, 403, 408, 414, 416, 427, 433, 452, 466, 469, 476, 478, 480, 484,
487, 492, 497, 522, 526, 534, 535, 541, 546, 553, 560, 586, 604, 628, 633,
640, 660, 665, 671, 677, 679, 683, 684, 685, 702, 706, 711, 712, 714, 719,
722, 725, 741, 743, 746, 762, 783, 784, 788, 798, 806, 810, 815, 829, 832,
834, 837, 844, 855, 860, 864, 877, 885, 910, 922, 924, 943, 953, 972, 979,
989, 996, 1004, 1013, 1014, 1024, 1025, 1029, 1038, 1039, ..., 17223, 17235,
17247, 17249, 17254, 17256, 17263, 17266, 17277, 17283, 17308, 17314, 17342,
17349, 17350, 17378, 17380, 17381, 17388, 17396, 17406, 17419, 17422, 17423,
17428, 17436, 17451, 17457, 17460, 17472, 17485, 17495, 17502, 17517, 17546,
17547, 17553, ... #第一处省略号是人为添加的，第二处是terminal大小限制，未输出全部
结果
  Continue? (y/n) n
```

第二次运行的输出结果如下：

```
  The running time of building the posting list: 2508.7581 secs
  Input your query: (AND, OR, NOT, '(' and ')' are allowed in the query.)
  market AND technology AND OIL
  Searching time: 0.0280 secs
  Doc IDs:  [76, 132, 228, 409, 549, 1023, 1191, 1259, 1304, 1688, 1813,
1846, 1977, 2047, 2131, 2174, 2213, 2370, 2433, 2598, 2681, 2702, 2839,
2974, 3070, 3079, 3216, 3260, 3281, 3287, 3351, 3411, 3476, 3517, 3551,
3760, 3779, 3787, 4094, 4689, 4715, 4794, 4863, 4939, 5012, 5828, 6123,
6180, 6249, 6410, 6509, 6619, 6665, 6708, 6744, 6872, 7048, 7092, 7196,
7346, 7379, 7920, 7963, 8153, 8423, 8495, 8521, 8672, 8736, 8894, 8921,
9084, 9204, 9675, 9730, 9923, 10032, 10384, 10585, 10672, 10709, 10913,
10920, 10930, 11103, ..., 121069, 121082, 121202, 121396, 121567, 121723,
121743, 121820, 121970, 122010, 122172, 122317]
```

几点思考：

- 两次的查询结果相同，但建立倒排表的时间差距较大。由于两次运行的程序相同，猜测运行时间差异与系统的缓存情况与实时性能相关。
- 编程过程中也发现，用全部五个文件夹来创建倒排表的耗时，远远超过用两个文件夹来创建倒排表的耗时。主要原因是倒排表的规模会越来越大，越往后处理，需要检索的元素越多，耗时会超过线性增长。
- 输出的倒排表若以xlsx格式存储，超过了65536行（或列），excel无法正常打开。所以输出采用了dat格式，可用写字板打开，虽然耗时较长。

- 语义查询（利用tf-idf)

选取了一个文件夹作为数据集，利用`TfidfVectorizer`相关函数实现求解。

```
  Input the semantic query: capital growth earnings back operation management growth
  [[0. 0. 0. ... 0. 0. 0.]
  [0. 0. 0. ... 0. 0. 0.]
  [0. 0. 0. ... 0. 0. 0.]
  ...
  [0. 0. 0. ... 0. 0. 0.]
  [0. 0. 0. ... 0. 0. 0.]
  [0. 0. 0. ... 0. 0. 0.]]
```

- 对数据进行再次处理（单词之间加空格），目的是匹配函数的参数类型，使得参数的每一个列表元素为一篇*处理过的单词*连接而成的文章。
- 不能直接将`TfidfVectorizer.fit_transform`的结果作为矩阵传给Excel表格，会由于存储空间不足而失败。采取的措施有：限制最大元素数量为1000，将结果先转为稀疏矩阵再进行输出。
- 查询的tf-idf值均为0，结果如上。原因可能是概率过小，或者非零项没有出现在被显示的区域。