实验三　倒排索引和布尔检索模型

1. 在 tweets 数据集上构建 inverted index

- 读取 tweets 数据集，将每个 tweets 信息作为列表中的一个字符串元素

```
tweets = []
path = 'C:\\Users\\hp\\Desktop\\IR 实验\\tweets.txt'
with open(path, 'rb') as file:
    lines = file.readlines()
    for line in lines:
        line = line.decode("ascii")
        tweets.append(line)
```

- 提取每个 tweets 的正文（第九个引号到第十个引号之间的部分）

```
def substring(str1):
    count = 0
    str = ""
    for s in str1:
        if s == '"':
            count = count + 1
        elif count == 9:
            str = str + s
    return str


tweets_text = []    #tweets_text 中的元素为原每个 tweet 的正文
for tweet in tweets:
    tweets_text.append(substring(tweet))
```

- 对每个 tweets 正文进行同实验一中的 tokenization，stemming，去 stopwords 等预处理
- 对上一步 tweets 文档中预处理后的单词建立字典，每个单词的 value 为其所在的所有 tweets 的标签构成的集合，tweets 标签为之前 tweets 列表的索引值

```
dict1 = {}
label = 0
for pretext in textword:
    for word in pretext:
        if word not in dict1:
            dict1[word] = set()
        dict1[word].add(label)
    label = label + 1
```

2. 实现 Boolean Retrieval Model

- 这里使用集合间的运算实现检索。在检索开始前，创建 answer_set 集合用以盛放满足检索条件的 tweets 标签，在输出时，将 answer_set 转变为 list 并将元素排序后输出。

```
def print_answer(answer_set):                          #Output search results
    empty_set = set()
    if empty_set == answer_set:
        print('None')
    else:
        answer_list = list(answer_set)
        answer_list.sort()
        print(', '.join(map(str, answer_list)))
```

- 对于 OR 检索，需要输入以”OR:”开头的字符串，检索内容从第 3 个字符开始读取，在对 query 字符串进行相同预处理后，对于 query 中出现的每个单词，求它在字典中的 value 与 answer_set 的并集。

```
if 'OR:' in query:
    query_word = filter(query[3:])  #注意 query 也需要进行预处理
    print(query_word)
    for word in query_word:
        if (word != '')and(word in dict1):
            answer_set = dict1[word] | answer_set
```

- 对于 AND 检索，answer_set 需要初始化为含有所有文档的标签，query 从第 4 个字符开始读取，经过同样预处理后，对于 query 中出现的每个单词，求它在字典中的 value 与 answer_set 的交集。对于 NOT 检索，过程类似，最后求差集即可。

```
if 'AND:' in query:
    query_word = filter(query[4:])
    print(query_word)
    if query_word != []:      #第一次要跟所有文档标签做与操作，因此这里集合内容需初始化
        for i in range(30548):
            answer_set.add(i)
        for word in query_word:
            if word != '':
                if word in dict1:
                    answer_set = dict1[word] & answer_set
                else:
                    answer_set = set()
    else:
        print('None')
if 'NOT:' in query:
    query_word = filter(query[4:])
    print(query_word)
    for i in range(30548):
        answer_set.add(i)
    for word in query_word:
        if word != '':
            if word in dict1:
                print(list(dict1[word]))  #输出含有该单词的文档
                for i in dict1[word]:
                    answer_set.discard(i)
```

3. 测试

- OR:

  INPUT:　OR:House Improve

  OUTPUT:

  ['hous', 'improv']

  0, 2, 100, 239, 291, 327, 522, 523, 537, 779, 803, 1018, 1069, 1316, ……

- AND:

  INPUT:　AND:Ron Weasley birthday

  OUTPUT:

  ['ron', 'weasley', 'birthday']

10314, 16613, 16745, 16787, 16792, 16794, 16809, 16817, 16820, ……

- NOT:

  INPUT: NOT:recall Skill says

  OUTPUT：

  ['recal', 'skill', 'say']

  3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 18, 19, 20, 22, 23, 24, 25,……

经小范围验证，输出结果均正确。