Homework4 报告

霍瑜 201605130119

实验过程：

1）分词，利用实验三代码
2）构建 term，与实验三的区别在于不去重
3）记录推特推文长度以及计算平均长度
4）合并 term 并构建单词-文档-次数三元组的字典，线性扫描一遍即可
5）根据询问的单词以及 BM25 公式计算文档得分

$$f(q,d)=\sum_{w\in q\cap d} c(w,q)\frac{(k+1)c(w,d)}{c(w,d)+k(1-b+b\frac{|d|}{avdl})}\log\frac{M+1}{df(w)}$$

6）得分排序，输出
7）测试

实验结果：

k \ b	0.5	0.8
100	MAP = 0.4344 NDCG = 0.6144	MAP = 0.4400 NDCG = 0.6272
1000	MAP = 0.4335 NDCG = 0.6133	MAP = 0.4385 NDCG = 0.6260

由上看来减小 k 和增大 b 都会使结果更优，这是因为增大了词频的影响并减小了文档长度的影响。

以下开始逐步缩小 k，观察效果

k = 10 b = 1 : MAP = 0.44995467835882985 NDCG = 0.6395595785999254
k = 1 b = 1: MAP = 0.5002232241534921 NDCG = 0.6873228738066968

效果持续攀升
k = 0.1 b = 1: MAP = 0.5288893166165983 NDCG = 0.7107330862893652

当降到 TF-IDF 的标准形式时，效果稍微差了点
k = 0 b = 1: MAP = 0.5230980822314885 NDCG = 0.7072255244189543

由上可见 TF-IDF 形式虽然很简洁，但在效果上还是有很好的表现。

主要原因我觉得是此数据集的单个文档的单词太少，体现不出 BM25 权衡文档长度以及词频的优势，所以才会和 TF-IDF 效果差不多