XXX组的信息检索项目总结报告 [四号字体]

小组成员的学号和姓名 [小四号字体]

[正文均为5号字体]

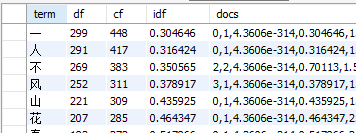
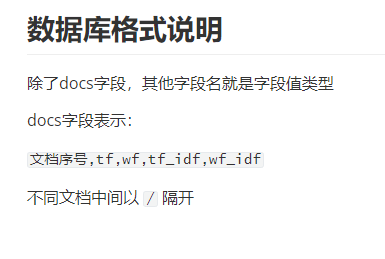
一、项目实现过程中遇到的问题及解决方案

1．基于布尔模型的检索实验与研讨

1.1 布尔模型的前端设计

布尔模型要求实现AND OR AND\_NOT三种算法，在这些算法的基础上，要进行一个tf-idf和wf-idf的计算，进行得分排序。并且布尔模型是唯一支持条件检索的模型，所以这里也要设计一个条件表达式。

解决问题1 AND OR AND\_NOT的计算。首先是要构造出每个term项的一个文档序列链表。即该term项在哪些文档中出现过。我们这里采用的是一个数据库的合作关系，进行前后端分离。Term项保存为一下格式



即docs中的内容为docsid,tf,wf,tf\_idf,wf\_idf/docsid,tf,wf,tf\_idf,wf\_idf/……

对于前端设计算法而言，只需要取docsid tf\_idf wf\_idf 三个字段，用一个数据结构保存为一个节点。然后term项作为头节点，构造出一个term项的相关链表。



对于所有的查询term项同上，构造出Term链表。在这样的基础上，便可以通过链表的交并否运算来获取结果链表，并返回。

解决问题2 计算得分，排序权重。因为链表的结构中带有tf\_idf 以及 wf\_idf，所以在进行链表的交并运算的时候，同时把docid相同的tf\_idf 以及 wf\_idf 累加起来，最后得到的结果链表，按照tf\_idf 或者 wf\_idf 排序获得 tf\_idf的排序结果或者 wf\_idf的排序结果。

解决问题3 条件表达式的兼容

在本系统中，用\* 代表 AND运算

+ 代表 OR 运算

* 代表 NOT 运算

并且支持( )的优先级运算

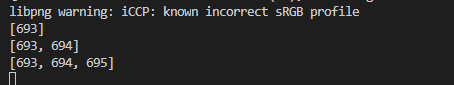
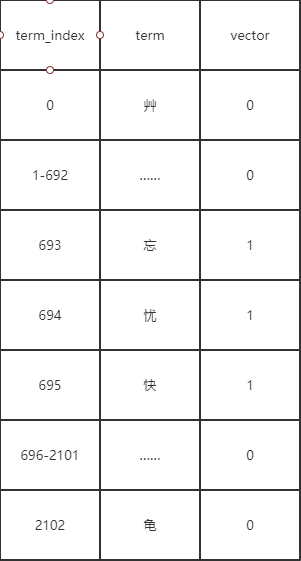
具体实现，首先要有AND OR NOT的自定义算法，然后将用户输入的表达式（中缀表达式）通过栈实现中缀表达式转换成后缀表达式，再通过后缀表达式和栈结合实现链表的计算，得到结果链表。

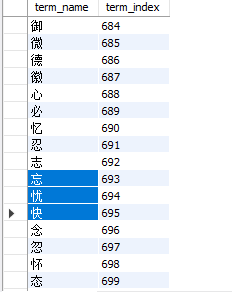
2．基于向量模型的检索实验与研讨

2.1 向量模型的前端设计

在向量模型中，前端有几个问题，要实现用户输入的term提取，构建查询向量。计算查询向量与文档向量的相似度获得每篇文档的得分。根据文档得分，进行排序，获取topk文档。

解决问题1 作为查询的term项向量，相较于庞大的数据库而言是一个稀疏向量，所以应该采用稀疏保存法。因为在向量模型中，查询term项时没有权值的，所以是一个均匀分布，那么只要保存term项在哪些维度出现过即可。作为这样的保存方式也方便后续进行的相似度得分计算。下列是某个查询向量的构造结果：

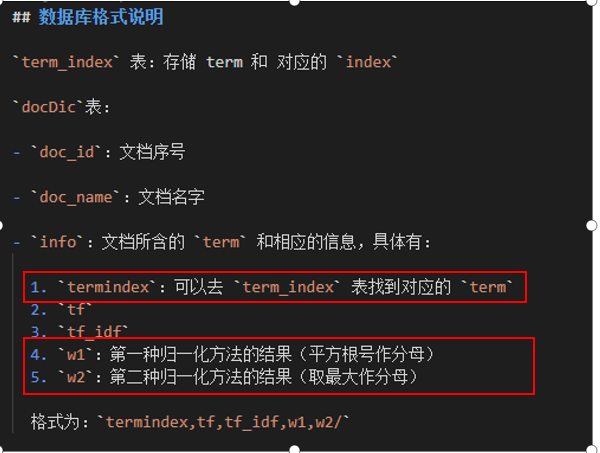


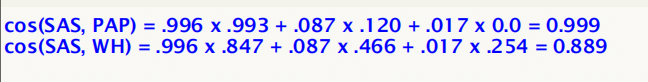


解决问题2 通过获取后端完成归一化的数据，进行COS余弦相似度计算。因为向量模型中的查询向量每个term项是均匀权重的。这里后端也要对每篇文档提供一个向量。

后端文档向量模型如下：



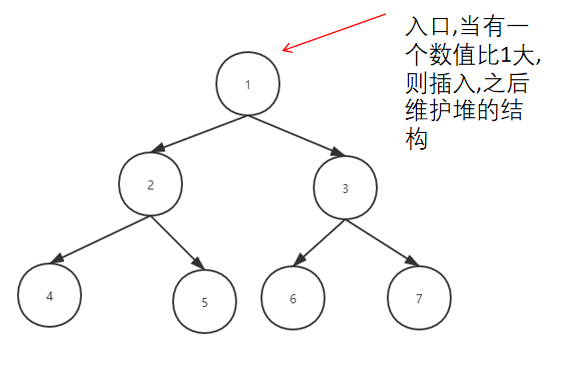




利用向量的标量积进行查询向量和文档向量相乘，得到一个数值。因为，文档向量和查询向量都进行了归一化，所以该值也是两个向量之间的夹角余弦值。

解决问题3 TOPK排序算法，因为全部数据有n+文档，数据量较大，而我们仅仅只要前20得分的文档，所以为了得到更高效率的算法，即占用空间小，速度快的算法。这里采用的是一个小顶堆的算法构建。将n+篇的文档全部在小顶堆中跑一遍，最后小顶堆剩下的元素就是得分前20的文档。

具体实现是维护一个空间大小为20的小顶堆，将每个元素与小顶堆的顶端元素进行比较，如果比小顶堆顶端元素小，则不进行进入到堆中，如果大于顶端元素，那么就将该元素替换小顶堆的顶端元素，并且向下维护堆，使其保证父节点的值始终小于子节点的值。

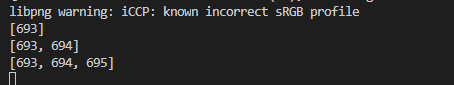


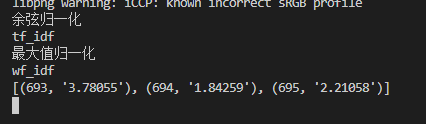
3．基于概率模型的检索实验与研讨

3.1 概率模型的前端实现

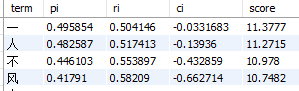
作为概率模型，这里需要解决的一个比较重要的问题是要进行有关和无关的划分，在这个基础上，还要进行与文档得分的计算，最后进行得分的topk排序。

解决问题 有关和无关，这个是要基于用户的查询的。所以这里采用向量模型的查询向量的构造基础，利用向量模型的查询向量的构造规则来构造概率模型的查询向量，这是一个带概率权重的查询向量。





上述两图分别是向量模型和概率模型的查询向量的构建。很明显，在第二张图片的概率模型中，带上了每个查询维度的权值设置。在这样的基础上，也就有更高的概率获得用户所需的查询结果。对于某个term项的权重来源，来自于后端计算的每个term项的得分结果，如下图所示，计算pi ri ci 最后归一化得到score



获得查询向量的权值之后，可以利用之前的向量模型的计算，以及之前构建的文档向量的基础，进行计算，最后获得一个所有文档的结果。在利用自定义的小顶堆计算，获得得分前20的文档，并返回给用户。

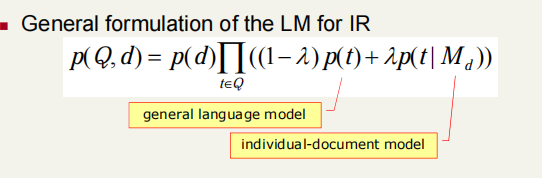
3.2 语言模型的前端实现

语言模型要解决的问题是计算每个term项在文档中的

P(term\_i| Mc) —— 在文档集词袋中（先验概率） term\_i 出现的概率

P(term\_i| Md) —— 在某篇文档词袋中（先验概率） term\_i 出现的概率后端返回这两个结果值之后，前端通过用户的查询结果来确定有效term项，遍历后端保存好的所有文档，并利用有效term逐个文档计算得分，最后仍然利用小顶堆进行topk计算，返回结果给用户。

对于每篇文档的计算，采用不同的λ带入公式，计算评价出效果最好的λ值，在本次实验中，最好的λ值约为0.5，所以在系统中便采用0.5计算。

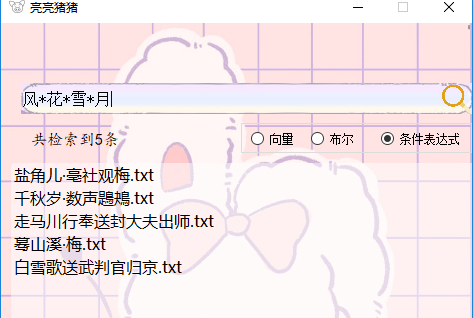


在这个模型中，还要处理一个零概率和小概率的问题，由于后端计算出来的P(term\_i| Mc)、P(term\_i| Md)的数都是小于1的值，而公式中采用的是累乘，所以在相关度越高的文档，反而得分越小，所以在前端进行计算之前，对每个P(term\_i| Mc)、P(term\_i| Md)进行线性+1，之后再进行累乘的话，相关度越高，得分便也越高。对于零概率term项，该模型中是进行忽略 ，不计算的，保证不会因为乘以0而让文档得分变为0.

二、项目检索结果对比分析

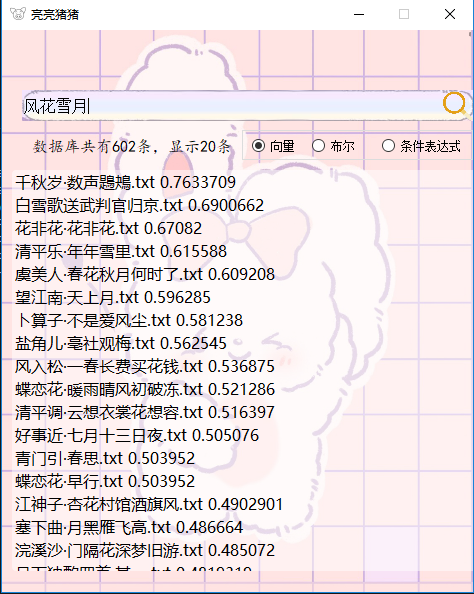
[针对小组的数据集特点，根据5条查询需求，结合小组的实验和研讨的内容与结果，对三种基本信息模型的4种查询功能所实现的检索结果进行对比分析。]

1. 向量模型与布尔模型的对比分析

布尔模型的AND计算

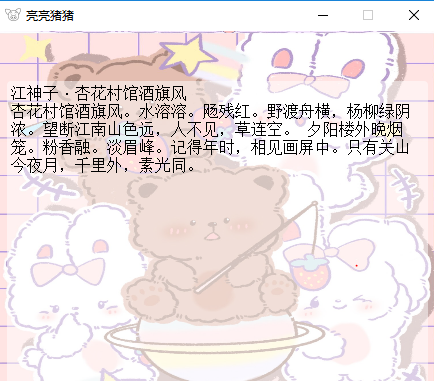


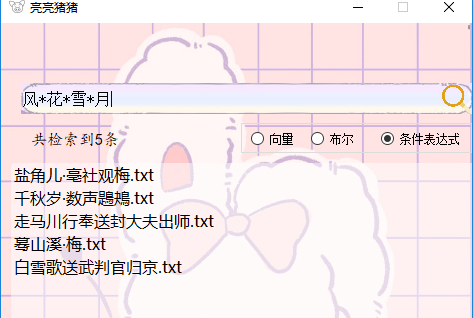
布尔模型的AND OR计算的结合



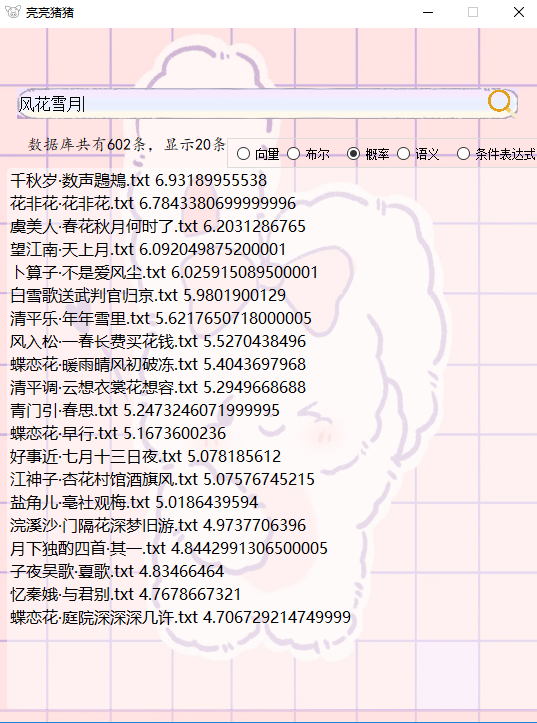
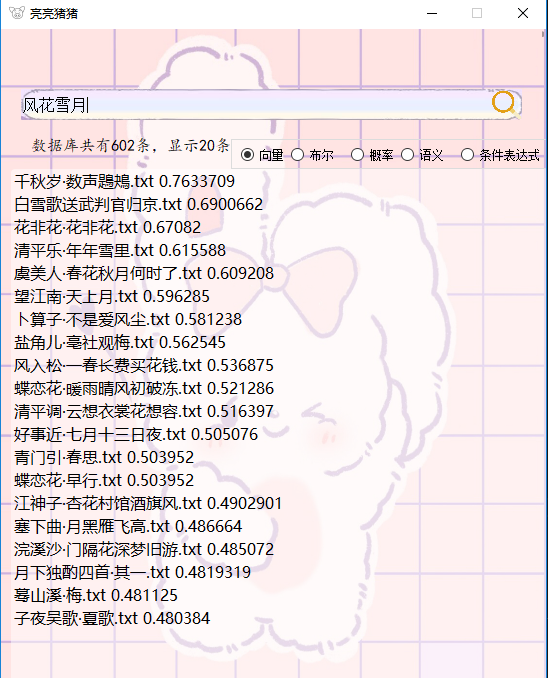
向量模型

对比分析：向量模型而布尔模型还是有一个比较大区别得，向量模型是一个相似度求解计算，而布尔模型是一个严格得表达式计算。在向量模型中，即便某篇文档中，不含有其中得某个term项，他也会被显示出结果来，举个例子,比如检索”风花雪月”向量模型中的<江神子·杏花村馆酒旗风>文档中并不含有雪这一项,但是却显示出来,并且拥有比较好的得分评价。



而布尔模型检索出来的结果就要严格AND操作,必须同时含有”风花雪月”,仅仅检索出5条结果而已

1. 概率模型与向量模型的对比分析



概率模型 向量模型

在这两个模型得对比中，是最相似得一组对比，概率模型是在向量模型得基础上实现的。不同的是，在概率模型的查询向量中，增加了查询向量的权值，所以它会针对，某些概率较大的term项返回结果，这样所得的结果也是用户更想要的。比如，”风花雪月“中的花，在古诗中出现的概率更高，所以带花的文档权重会偏高一些，返回结果自然也就带花的文档会获得更高的得分。

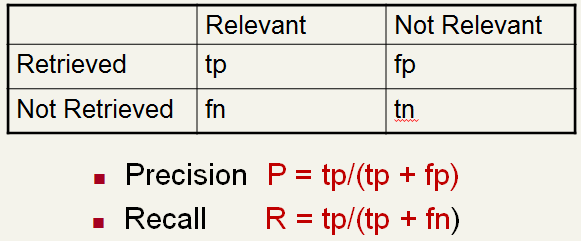
三、项目检索性能评价

[说明对小组的数据集有效的信息模型和有效检索方法。]

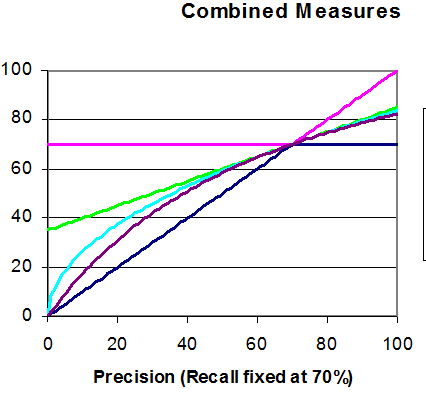
[在选定的信息模型和检索方法的基础上，进行系统评价。

具体方法：

根据5条查询需求，分别计算ptecision和recall以及Harmonic的F值（β=1,α=1/2），画出每条查询需求的R-P图和C-M图。

 ， 



仅画出的结果

]