信息检索项目总结报告

1. 项目设计

图示

描述已自动生成

图表 1

系统架构和信息模型如图1所示。

对于倒排索引，通过分词工具，对读取的文档进行分析，从而完成构建。

在查询内容上，对于布尔表达式，支持扩展布尔模型查询；对于自然语言，支持向量查询、语言模型查询、概率模型查询，并且支持不同模型查询结果的比较。不同模型有不同的得分计算方式。对于扩展布尔模型，使用tf-idf权重与zone score，为了让短文档被优先考虑，因此得分还要除以文档长度；对于向量模型，使用归一化权重与idf计分，同时使用近似查询；对于语言模型，使用uni-MLE计分；对于概率模型，为了反映查询term项在给定文档中出现情况，采用BM25计分。

对于查询结果，按得分从高到低排序，取前20条。

1. 项目实现

1. 扩展布尔模型

在扩展布尔模型中，对于每个域采用tf-idf计分。对于不同的域，采用zone score，得分加权求和，其中body的权为0.7，title的权为0.3。这涉及不同zone查询结果的合并，其中tf-idf得分加权叠加，代码实现如下所示。

|  |
| --- |
| vector<ListNode> InvertedIndex::zoneScore(**const** vector<ListNode>**&** vBody,  **const** vector<ListNode>**&** vTitle,  **bool** weighted) {  **int** n = vBody.size(), m = vTitle.size();  **int** i = 0, j = 0;      vector<ListNode> ans;  **while** (i < n || j < m) {  **if** (j == m) {  // 仅在body 中出现              ans.push\_back(ListNode(vBody[i].fileId,                                     gBody \* (weighted ? vBody[i].tf\_idf : 1)));              i++;          } **else** **if** (i == n) {  // 仅在title 中出现              ans.push\_back(ListNode(vTitle[j].fileId,                                     gTitle \* (weighted ? vTitle[j].tf\_idf : 1)));              j++;          } **else** **if** (vBody[i].fileId ==                     vTitle[j].fileId) {  // title + body 中出现              ans.push\_back(                  ListNode(vBody[i].fileId,                           gBody \* (weighted ? vBody[i].tf\_idf : 1) +                               gTitle \* (weighted ? vTitle[j].tf\_idf : 1)));              i++;              j++;          } **else** **if** (vBody[i].fileId < vBody[j].fileId) {  // 仅在body 中出现              ans.push\_back(ListNode(vBody[i].fileId,                                     gBody \* (weighted ? vBody[i].tf\_idf : 1)));              i++;          } **else** {  // 仅在title 中出现              ans.push\_back(ListNode(vTitle[j].fileId,                                     gTitle \* (weighted ? vTitle[j].tf\_idf : 1)));              j++;          }      }  **return** ans;  } |

2. 语言模型实现

采用uni-MLE计算得分，具体公式如下：

文本

描述已自动生成

图片包含 文本

描述已自动生成

其中 取值为 0.9。选取0.9的原因是更希望关注term项在指定文档中的出现情况，而不是在所有文档中出现的情况。当查询涉及的多个term项在一篇文档都出现时，这篇文档的得分就会更高。代码实现如下：

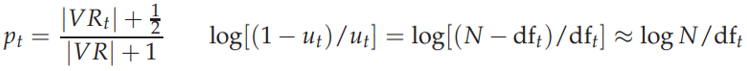
|  |
| --- |
| vector<pair<**double**, **int**>> InvertedIndex::languageModel(string s) {  **const** **static** **double** lambda = 0.9;      // 计算权重      vector<**double**> score(n + 1);  **for** (**int** d = 1; d <= n; d++) {  **double** tmp = 1;  **int** tag = 0;  **for** (**int** i = 0; i < s.size(); i += 3) {              string cur = s.substr(i, 3);  **if** (cur == "\n" || cur == " ") **continue**;  **if** (!dictBody[cur]) **continue**;              tag = 1;              assert(contentLength[d] != 0);              assert(cs != 0);              tmp \*= lambda \* dictBody[cur]->getFreqOfDoc(d) / contentLength[d] +                     (1.0 - lambda) \* dictBody[cur]->Mc;          }          score[d] = tmp \* tag;      }  **return** getTopK(score);  } |

3. 概率模型实现

采用BM25计算得分，公式如下：

图片包含 图示

描述已自动生成



其中 取0.75， 取1。采用BM25的原因是，该方法考虑了term项在文档中的出现频率，且根据文档长度做了平滑，查询结果会更加准确。

4. 遇到的问题与解决方案

a. 在计算uni-MLE的Mc时，最初是每次都做按照公式算一遍，这样效率比较低。其实cf和cs都可以提前计算好，对于每个postingList这个值是固定的，从而可以提高计算效率。

b. 在计算uni-MLE权重时，最初没有考虑没有出现在文档中的term项，导致索引指向了空指针，在使用空指针时出现了错误。对没有出现在文档中的term项进行特判并跳过后解决了这个问题。

1. 项目检索结果
2. 自然语言查询--日月

图形用户界面, 表格

描述已自动生成图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

表格

描述已自动生成

向量模型、语言模型、概率模型的检索结果有一定的相似性，前两条结果是一致的，其中概率模型和语言模型结果相似度极高。文档17、77、96的正文中都出现了日月，且文档长度均较短，因此分数较高。对于向量模型，文档2的得分较高，而文档2的内容是’ 白日依山尽，黄河入海流。欲穷千里目，更上一层楼。’，可能的原因是出现‘日‘的文档较少，且该文档长度较短，因此得分比同时出现’日月‘但长度大的文档高。另外，由于向量模型采用了近似方法，因此得分和精确计算的得分有一定偏差。

1. 自然语言查询—幻

图形用户界面

描述已自动生成

‘幻’没有在任何文档中出现，因此没有匹配的文档，所有文档得分都是0。这个查询的目的是测试模型的健壮性，对于没有出现的term项，应要么返回全0分，要么不返回任何结果。

1. 自然语言查询—虚幻黄昏

图形用户界面, 表格

中度可信度描述已自动生成表格

描述已自动生成表格

描述已自动生成

向量模型、语言模型、概率模型的检索结果前2条是一致的，两个文档都含有黄昏，且是短文档，因此得分最高。其它文档要么有‘黄昏’但长度过长，要么只有‘虚‘、’黄‘、’昏‘中的一个，因此得分较低。对于文档184，其在语言模型和概率模型中得分较高，但在向量模型中得分较低。可能的原因是在向量模型中，如果term项不在文档中是不计分的，但是在概率模型和语言模型中会进行计分。

1. 布尔表达式查询--and花落

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

在body中查询同时有‘花’、‘落’的文档。文档1、18、49、52等均含有这两个关键字，但由于文档49较长，因此得分稍低。对于查询项，其在文档1、18中出现情况是一样的，也就是说计算得分时涉及到的idf和tf是一样的，因此最终得分也是一样的，符合预期。

1. 布尔表达式查询--and花落--body+title

图形用户界面, 表格

描述已自动生成 图形用户界面, 文本, 应用程序

描述已自动生成

将只在body中查询的结果和同时在body和title中的查询结果对比可以发现，在两个域中查询时，文档108的得分得到明显提高，从10名开外提升到了第7名，可见zonescore确实将多个域的得分进行了加权叠加。