**算法设计与分析课程**

**论文阅读报告**

**《Efficient Merging and Filtering Algorithms for**

**Approximate String Searches》**

姓名：卢兑玧

学号：L170300901

班级：1703009

评分表：（由老师填写）

|  |  |
| --- | --- |
| 最终得分： | |
| 论文题目： | |
| 对计算问题的理解： | |
| 对算法的理解： | |
| 对实验过程和结果理解： | |
| 阅读过程的收获： | |
| 特 色： | 1 |
| 2 |
| 3 |

**一、对计算问题的概述**

相似字符串查找在现实生活中的应用非常广泛,例如相似网页检测、数据清洗、电商网站的推荐功能、蛋白质功能预测等。相似字符串查找多是用一个给定的相似性函数来判断两个字符串是否相似。相似性函数有多种,例如汉明距离、编辑距离、余弦度量以及杰卡德度量等,其中使用较为广泛的为编辑距离。目前,基于相似性函数的相似字符串查找的研究可分为三类,即基于阈值的相似字符串查找、top-k相似字符串查找和相似字符串连接。

对于相似字符串查询的应用程序需要具备高时效性来回答每个查询，特别是那些基于web的服务模型。例如，考虑了拼写检查程序，如Gmail、Hotmail或雅虎邮箱，因为有上百万的用户在使用这项服务，它需要在每秒钟被调用很多次，每一个拼写检查请求都需要尽快解决。从每个用户的角度来看，20ms的处理时间和2ms处理时间差别不大，但从服务器的角度来看，前者表示每秒50个查询（QPS），后者表示每秒500个查询。显然，后者的查询时间可以使得服务器每秒处理更多的用户请求。因此，开发相应的算法去尽可能高效的执行这样的查询是十分必要的。

**二、对算法的理解**

对于上述问题，本文作者提出了两种另外方法。

首先对原已存在的算法进行总结。对于此问题已有多种算法提出，这些技术假定有给定的相似函数来量化两个字符串之间的亲密度，比如edit距离，Jaccard相似度，余弦相似度等等。

在本文中，作者做出了两个主要贡献：

1.贡献一为：

提出了三种有效的算法来回答近似字符串搜索查询，称为ScanCount，MergeSkip，DivideSkip。

1. ScanCount

ScanCount算法采用了扫描倒排表的简单思想计算候选字符串。尽管事实十分天真，但当与各种过滤技术相结合时，这个算法任然可以实现高性能。

2）MergeSkip

MergeSkip算法利用了倒立函数之间的值差列表和阙值上的公克数类似的字符串可以跳过列表中许多不相关的候选项。

3）DivideSkip

DivideSkip算法结合了 MergeSkip算法以及MergeOpt算法的思想，将列表分为两组。其中一组是针对于长列表，而其他组则是用于其余列表。通过运行MergeSkip算法来合并不同的短列表阙值，并使用长列表来验证候选项。

2.贡献二为：

对合并和过滤算法进行了研究。

**三、对实验过程和实验结果的理解**

首先对 ScanCount，MergeSkip，DivideSkip三种算法给出了准备条件。

现有的一些算法都假设一个倒排列表索引用于回答集合S中字符串的克数，索引中对s的近似字符串查询，每克g对于S中的字符串，我们都有一个lg的id列表包括这个克，可能包括相应位置字符串中的克的信息。基于多个字符串相似函数的搜索问题可以通过求解一下问题得到解决:

T-occurrence问题：设Q为一个问题，并且G(Q,q)是a对应的Q-G的集合常量q。查找出现在至少T次在克的倒排表上G(Q,q)，其中T为常数。

**3.1** **ScanCount**

**1. 实验设置**

**实验环境**：

该算法对堆算法进行了改进堆数据结构及其上的相应操作堆。相反，我们只维护count的数组在s中的所有字符id，扫描N个倒排的列表一个。对于每个列表中的每个字符串id，增加计数对应于字符串乘以1.并报告字符串id在列表中出现至少T次。算法形式如下所示。

**数据**：

Input:set of RID lists and a threshold T;

Output:record ids that appear at least T times an the lists.

1. Initialize the array C of /S/ counters to 0’s;
2. Initialize a result set R to be empty;
3. FOR(each record id r on each given list){
4. Increment the value of C[r] by 1;
5. IF (C[r]==T)
6. Add r to R;
7. }
8. RETURN R ;

**2. 实验结果**

实验结果表明，如果他们能缩短倒排列表合并，那么尽管算法简单，但仍然可以实现与各种滤波结合，性能良好。

**3.2MergeSkip**

**1. 实验设置**

**实验环境**：

它的主要方法是跳过列表中不能包含的记录id查询的答案，通过使用阙值T。类似的对于堆算法，我们还维护了一个堆的边界列表。一个关键的区别在于，在每次迭代中，我们从堆中取出具有相同值的记录堆上的顶部记录。

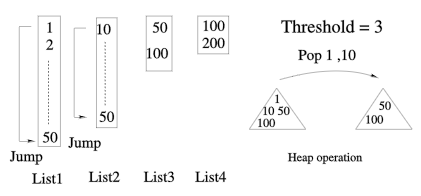
**数据**：

Input:set of RID lists and a threshold T;

Output:record ids that appear at least T times an the lists.

1. Insert the frontier records of the lists to a heap H;
2. Initialize a result set R to be empty;
3. WHILE (H is not empty){
4. Let t be the top record on the heap;
5. Pop from H those records equal to t;
6. Let n be the number of popped records;
7. IF(n≥T){
8. Add t to R;
9. Push next record(if any)on eac hpoppe dlist to H;
10. }
11. ELSE{
12. PopT−1−n smallest records from H;
13. Let t’be the current top record on H;
14. FOR (each of the T-1 popped lists){
15. Locate its smallest record r≥t’(if any);
16. Push this record to H;
17. }
18. }
19. }
20. RETURN R;

**2. 实验结果**



**3.3DivideSkip**

**1. 实验设置**

**实验环境**：

它的主要思想是合并MergeSkip和MergaOpt，两者都试图跳过列表中不相关的记录，但使用不同的指导方式。MergeSkip利用了价值列表之间的大小差异。我们的新算法DivideSkip使用这两种差异来进一步改进搜索性能。

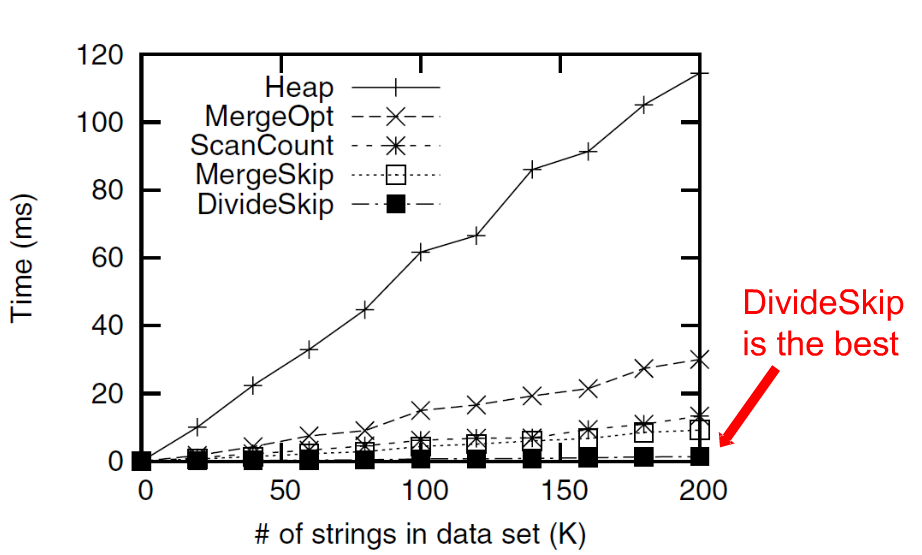
**数据**：

Input:set of RID lists and a threshold T;

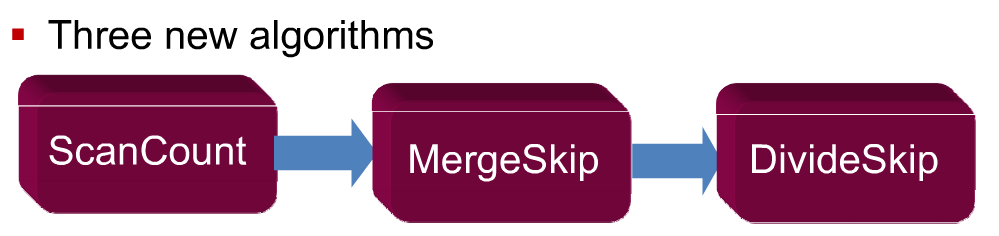
Output:record ids that appear at least T time an the lists.

1. Initialize a result set R to be empty;
2. Let Llong be the set of L longest lists among the lists;
3. Let Llong be the remaining short lists;
4. Use MergeSkip on Llong to find ids that appear at least T-L times;
5. FOR (each record r found){
6. FOR(each list in Llong)
7. Check if r appears on the list;
8. IF (r appears≥T times among all lists)
9. Add r to R;
10. }
11. RETURN R

**2. 实验结果**



**3.4结论**



**四、论文阅读的心得体会**

近似字符串匹配问题广泛的存在于文本检索、模式识别、信号处理以及生物信息学等诸多领域。编辑距离是最常用的度量字符串相似性的方法。为了解决近似字符串匹配问题,许多方法已经被提出。然而随着近些年网络数据、日志数据以及生物基因序列数据的快速增长,计算机系统的快速发展以及新问题的不断提出,现有的近似字符串匹配方法面临巨大的挑战。研究高效的近似字符串匹配方法具有重要的理论价值和实际意义。

本文研究了如何有效的求出a与给定字符串相似的字符串的集合。作者做了两个贡献。首先，开发了新的算法，大大提高了现有算法的性能。其次，研究了如何集成现有的过滤技术，以及合并技术，并说明看如何明智的将他们结合在一起。

由本文我知道了实现集成对性能有很大的影响，并且计算机领域推陈出新，变化迅速，应该不拘于现状，更多的研发出新的算法，不断突破现状，取得更先进的发展。

**五、对算法课的意见和建议**

1.可以多增加实验课，以加强我们的动手能力，并且通过自我操作可以更好的理解所学内容，并得到更多的自我认识和拓展。

2.类似于这次的阅读作业，我认为可以多加进行，通过自己的阅读以及查找相关资料，并在这个过程中锻炼了英文能力，一举多得。