小实验二 <近似链接> 实验报告

最终效果

ID	Homework	Upload Timestamp	Status	Memory(GB)	Time(s)	Comment	Ops
5824(Marked)	exp2-final	2017/05/23 13:55:09	Correct.	3.828	33.395		
5702(Marked)	exp2	2017/05/22 18:16:41	Correct.	2.87	4.178		

编辑距离<近似链接>

算法思想

考虑对 R 和 S 两个集合中，对编辑距离小于等于 τ 的字符串进行链接。

使用 `Partition-based` 算法，将 S 中的每个字符串按照张度平均拆分成 $\tau+1$ 份，对于某 R 中的字符串 $r \in R$ 和 S 中的字符串 $s \in S$，如果两者的编辑距离不超过 τ，则 s 的 $\tau+1$ 份字符串必定至少存在一份是 r 的字串。

为此，我们可以将 S 中的每个字符串按照分成的 $\tau+1$ 个子串建立反向列表，然后对于 R 中的每个字符串，在反向列表中查询 S 中有哪些字符串的 $\tau+1$ 子串出现在了待查询的字符串中，然后再对这些字符串进行过滤即可。

为了减小待过滤的字符串集合，可以增加一些限制：

1. `Length-based` ，由于只有长度差不超过 τ 的字符串才可能满足要求，所以可以只用检查 S 中与 r 长度差不超过 τ 的字符串即可
2. `Shift-based` ，若 r 和 s 中的某段匹配了，则要求匹配位置之前和之后的字符串最小可能的编辑距离之和不超过 τ

实现细节

对于 S 中的字符串，我们可以对他们按照长度进行分组，相同长度的字符串划分成 $\tau+1$ 份字符串的方式是相同的，然后再对 $\tau+1$ 个不同位置的字符串分别用 `Hash` 建立反向列表。

具体来说，$pos_{l,i}$ 表示长度为 l 的字符串的第 i 个子串的起始位置，$len_{l,i}$ 表示长度为 l 的字符串第 i 个子串的长度，$index_{l,i}$ 表示 S 中所有长度为 l 的字符串的第 i 段子串构成的反向列表。

对于一个查询的字符串 r，只需在 $index_{[|s|-\tau,|s|+\tau],*}$ 中查询即可，对于 S 中长度为 l 的字符串反向列表, 也只用枚举起点在 $[pos_{l,i}-\tau, pos_{l,i}+\tau]$ 中的子串即可。

对于最后的过滤，使用一个动态规划即可。

Jaccard<近似链接>

考虑对 R 和 S 两个集合中，对Jaccard距离不小于 τ 的字符串进行链接。

Jaccard定义如下：

$$\frac{|r \bigcap s|}{|r \bigcup s|} \geq \tau$$

其中 r 是 R 中某行字符串的单词集合, s 类似;将上面的公式进行变换：

$$|r \bigcap s| \geq |r \bigcup s| \times \tau \geq \lceil max(|r|, |s|) \times \tau \rceil$$

此公式表明，若要 r 和 s 之间Jaccard距离满足要求，则 r 的任意 $|r| - \lceil |r| \times \tau \rceil + 1$ 个单词和 s 的任意 $|s| - \lceil |s| \times \tau \rceil + 1$ 个单词必定存在一个完全相同的单词。

由此，我们先统计出每个单词在 R 和 S 出现的次数，然后对于 S 中的每个字符串 $s \in S$，将其出现次数最少的 $|s| - \lceil |s| \times \tau \rceil + 1$ 个单词放入反向列表，然后对于 R 中的每个字符串 r，将其出现次数最少的 $|r| - \lceil |r| \times \tau \rceil + 1$ 个单词在反向列表中查询，对于查找到的结果在做一次过滤即可。