实验二 近似连接

计 24 2012011335 柯均洁

一、 实验目的

使用 joinED, joinJaccard 两种算法，找出两个文档中相似度满足阈值的字符串对。要求运行时间尽可能快，使用内存尽可能小，在保证正确的情况下。

二、 设计思路

1. JoinED

Partition-based 算法：

- 基本思想：

  将被查询的每一个串分成 threshold+1 份，若被查询串与查询串的编辑距离小于 threshold，根据鸽巢原理，被查询串必然有一段与查询串的某段子串相同。

- 算法流程：

  a) readDataED：

  读入文档 1，文档 2

  b) buildED：

  将文档 1 中的字符串分成 thershold+1 份。其中 length%(threshold+1)份长为 ceil(length/(threshold+1))。剩下的长为 floor(length/(threshold+1))。以字符串的长度以及分段的编号为 index 建立一个倒排列表

  c) filterED：

  对于file2 中长度为len2 的字符串 str2，找file1中长度为len1的字符串 str1 为候选者，len1在[len2-threshold, len2+threshold]的区间内。设pi为第i段的起点，i表示第i段(0 ≤ i ≤ threshold),则str2中的第i段可能存在的区间：

  li=len(str2)/(threshold+1)

  pi= len(str2)*i/(threshold+1)

  left = max(0, pi - i, pi + (len(str1)-len(str2)) - threshold + i)

right = min(len(str1) - li, pi + i, pi +（len(str1)-len(str2)）+ threshold - i)

枚举 len2 的字符串在str1可能的区间是否有相同的子字符串，只要str2中存在一个分段在 str1 的可能区间有相同的子字符串，则将 str2 加入候选集 rawresultED 中。

d) 验证阶段计算候选者和这行字符串的 ED 距离，如果小于等于阈值，则将相关的文档编号和ED距离加入result。

2. JoinJaccard

i. readDataJac

对输入文件进行分词并分别保存在 words1, words2 中，并统计词频

ii. 对 words1 和 words2 按照词频从低到高进行排序（词频相同的则按字母序排列）

iii. buildJac：

将每个词条中的前 prefix 个单词取出，建立倒排列表。其中 prefix = floor((1 - threshold) * length + 1)。

iv. filterJac：

对 file2 中词条的前 prefix 个单词，取出对应的倒排列表，若 Jaccard 距离的阈值为δ，则取

$overlap\_threshold = O(x,y) \geq ceil(\frac{\delta}{1-\delta}(|x|+|y|))$。

1. 首先计算 prefix position filter 后的 ubound=prefix_overlap + min(suffixlen1, suffixlen2)
2. 当 ubound >= overlap_threshold 时，不断地对 suffix 进行二分，直到 ubound 小于 overlap_threshold 或者全部区间已经只有一个元素（此时即可计算出 suffix 的交集，从而计算出 jaccard）
   - 在对 suffix 进行二分的过程中，维护了一个 priority_queue，按照待二分的区间长度进行排序，首先将最长的区间取出进行二分，将划分后的左右区间压入 priority_queue，并更新阈值 ubound 和交集 intersec。