# mt2.0基于编辑距离算法增量更新介绍

waynelu(t.qq.com/waynelu)

2014-06

# about me

卢勇福（waynelu）

微博:

http://t.qq.com/waynelu

http://www.weibo.com/u/1849616271

github:

https://github.com/luyongfugx

https://github.com/mtjs/mt

# 提纲

- Mt1.0 chunk算法的问题
- 什么是编辑距离计算
- 编辑距离计算算法具体实现
- 在mt里面编辑距离计算算法

# Mt1.0 chunk算法的问题

mt1.0基于分块chunk算法来做增量更新，节省资源的量取决于块的大小和碎片的大小，无法做到字符级别的增量更新

# 什么是编辑距离计算

**概念：**

Levenshtein Distance (编辑距离)，编辑距离即从一个字符串变换到另一个字符串所需要的最少变化操作步骤

作者：

俄罗斯科学家Vladimir Levenshtein在1965年提出这个概念

# 编辑距离计算算法具体实现

编辑代价定义：

| | |
|---|---|
| 1.不变 | 0 |
| 2. 替换 | 1 |
| 3.插入 | 1 |
| 4.删除 | 1 |

删除，替换，插入这几种操纵的代价是1，即修改一个字符，不变则是0，表示没有修改，即操作代价是0

# 编辑距离计算算法具体实现

## 编辑距离计算公式:

edit(i, j)：第一个字符串的长度为i的子串到第二个字符串的长度为j的子串的编辑距离

## 通过动态规划法（dp）得到：

- if i == 0 且 j == 0，edit(i, j) = 0
- if i == 0 且 j > 0，edit(i, j) = j
- if i > 0 且j == 0，edit(i, j) = i
- if i ≥ 1 且 j ≥ 1 edit(i, j) = min{ edit(i-1, j) + 1, edit(i, j-1) + 1, edit(i-1, j-1) + f(i, j) }，当第一个字符串的第i个字符不等于第二个字符串的第j个字符时，f(i, j) = 1；否则，f(i, j) = 0

# 编辑距离计算算法具体实现

以batyu 修改为beauty， 编辑距离为3（右下数字）：

| | | b | e | a | u | t | y |
|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| b | 1 | 0 | 1 | 2 | 3 | 4 | 5 |
| a | 2 | 1 | 1 | 1 | 2 | 3 | 4 |
| t | 3 | 2 | 2 | 2 | 2 | 2 | 3 |
| y | 4 | 3 | 3 | 3 | 3 | 3 | 2 |
| u | 5 | 4 | 4 | 4 | 3 | 4 | 3 |

# 在mt里面编辑距离计算算法

我们记录每一个编辑步骤（红字）：

0：未修改, 1：替换,2：删除，3：插入

| | | b | e | a | u | t | y |
|---|---|---|---|---|---|---|---|
| | 0 | 3 | 3 | 3 | 3 | 3 | 3 |
| b | 2 | 0 | 3 | 3 | 3 | 3 | 3 |
| a | 2 | 2 | 1 | 0 | 3 | 3 | 3 |
| t | 2 | 2 | 1 | 1 | 1 | 0 | 3 |
| y | 2 | 2 | 1 | 1 | 1 | 1 | 0 |
| u | 2 | 2 | 1 | 1 | 0 | 1 | 2 |

# 在mt里面编辑距离计算算法

从右下脚开始往左上脚遍历：

0：未修改, 1：替换,2：删除，3：插入

删除：y-1. 替换,相等:x-1,y-1 插入:x-1

| | | b | e | a | u | t | y |
|---|---|---|---|---|---|---|---|
| | 0 | 3 | 3 | 3 | 3 | 3 | 3 |
| b | 2 | 0 | 3 | 3 | 3 | 3 | 3 |
| a | 2 | 2 | 1 | 0 | 3 | 3 | 3 |
| t | 2 | 2 | 1 | 1 | 1 | 0 | 3 |
| y | 2 | 2 | 1 | 1 | 1 | 1 | 0 |
| u | 2 | 2 | 1 | 1 | 0 | 1 | 2 |

# 在mt里面编辑距离计算算法

得到编辑代价最小的编辑步骤：

0-3-0-3-0-0-2（0：未修改, 1：替换,2：删除，3：插入）

根据操作步骤和新字符串beauty，我们可以得到如下数组：

[ [ 1, 0 ], ‘e’, [ 2, 0 ], ‘u’, [ 3, 0 ], [ 4, 0 ] ]

**其中数组表示没有修改，我们合并一下连续没有修改的字符得到：**

[ [ 1, 1 ], 'e', [ 2, 1 ], 'u', [ 3, 2 ] ]

**这就是增量文件内容**

# 在mt里面编辑距离计算算法

合并算法为：

增量文件：[ [ 1, 1 ], 'e', [ 2, 1 ], 'u', [ 3, 2 ] ]

旧版字符串：batyu

所以新版字符串：

- beauty
  =”batyu”.substr(1-1,1)+’e’+”batyu”.substr(2-1,1)+’u’+”batyu”.substr(3-1,2);

END

http://mt.tencent.com

无更新不下载！！！

Q&A