第十一周工作报告

计 92 丘骏鹏 2009011282

2013-05-10

1 本周工作

本周主要做的是检测重复记录（Data Record）工作

1.1 具体实现

采用后缀树，实现了 Ukkonen 在 1995 年提出的一个$O(n)$ 的一个后缀树构建算法。（论文：Ukkonen, Esko. "On-line construction of suffix trees." Algorithmica 14.3 (1995): 249-260.）。具体的描述挺复杂，参考了Stackoverflow 上面的一个解答 及其补充进行实现。（之前有搜索过一些 Github，pastebin 上面的开源代码，但是人工构造一些测例后发现我找到的一些开源实现有 bug，此外我们还需要根据本项目的需要修改代码实现，所以还是自己写了一个）。

举一个 html 文件为例：

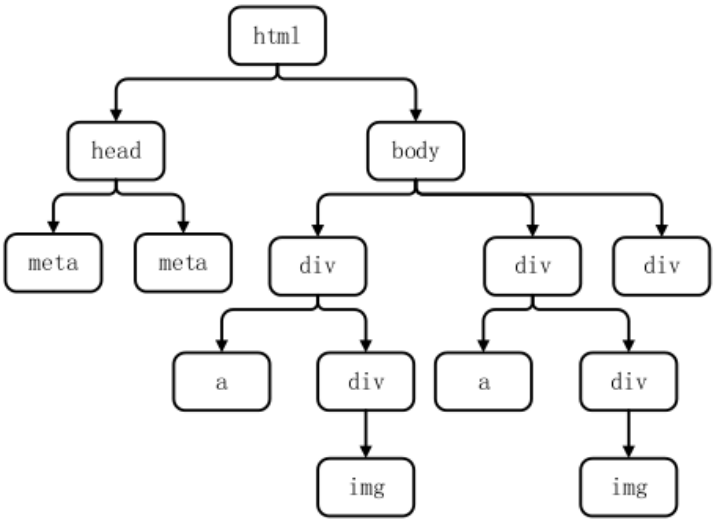

Figure 1: DOM Tree

采用先序遍历，然后将 `<tag><tag/>` 标签对换成 `<tag+depth>` 标签加上深度的表示方式，得到以下序列

```
html1 head2 meta3 meta3 body2 div3 a4 div4 img5
div3 a4 div4 img5 div3 a4 div4 img5 div3
```

构造出一个后缀树为：

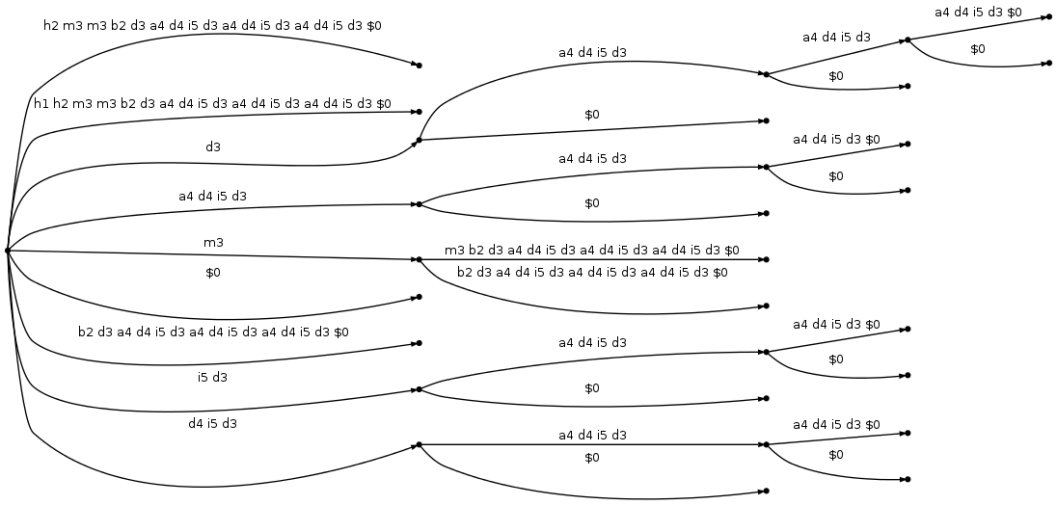

Figure 2: Suffix Tree

注：为了能够在图片中显示完整，标签名做了简化

根据构造出的后缀树可以马上得到所有的重复序列及其在原序列中出现的次数，但是有以下问题：

1. 重复序列的区间可能会相交
2. 重复序列的区间可能会有包含关系

因为第一个问题的存在，由后缀树得到的公共序列不能直接使用。我们采用的是先序遍历的方式，因此可以保证每个子树的所有标签序列位于一段连续的区间内。因此，可以在遍历后缀树得到公共序列的时候人为“切断”一些公共序列，“切口”位于后一个标签的深度比前一个标签的深度更大的地方（即两个子树的分隔点）。以上面的序列为例，遍历后缀树可以得到所有的重复子序列为：

```
a4 div4 img5 div3
a4 div4 img5 div3 a4 div4 img5 div3
div4 img5 div3
div4 img5 div3 a4 div4 img5 div3
img5 div3
img5 div3 a4 div4 img5 div3
div3
div3 a4 div4 img5 div3
meta3
```

改进以后公共的序列为：

```
a4 div4 img5
div4 img5
img5
div3
div3 a4 div4 img5
meta3
```

这样就可以保证我们得到每个序列都是一个子树的一部分，这样就不会出现各个序列的区间相交的情况。

为了解决子序列互相包含的情况（如"a4 div4 img5" 和"div4 img5"），需要首先遍历后缀树，得到所有的序列的区间，然后对所有的区间进行合并。普通的区间合并的做法会比较复杂，需要借助其他数据结构，但是得到的这些公共序列有一些特点：不相交，且互相包含的区间结束索引是一致的（即如果一个序列 A 包含序列 B，序列 B 一定是 A 的某个后缀）。这样就可以只需要扫描一遍所有的序列，利用 Hash 进行去重，然后合并所有的区间。

最后得到的公共的序列为：

```
div3 a4 div4 img5
meta3
```

根据以上结果，对于每个重复出现的序列，去掉除了第一次出现的序列以外的所有的重复的序列，原来的序列：

```
html1 head2 meta3 meta3 body2 div3 a4 div4 img5
div3 a4 div4 img5 div3 a4 div4 img5 div3
```

可以简化为：

```
html1 head2 meta3 body2 div3 a4 div4 img5 div3
```

从结果来看，基本完成了我们的目标。

2 下一步工作

在网页数据上，目前只在小量的数据集上做过测试，还未完全利用到原有的框架中。从时间上看，每个网页处理的时间需要 0.2s 左右，但是这些都只要离线处理，这个时间是可以接受的。

这个周末开始准备做一些大一些规模的测试和模板提取的工作。