基于标签路径聚类的文本信息抽取算法

2010年12月30日 · 1 min read

1、 网页预处理

可以通过以下3 个预处理规则来过滤网页中的不可见噪声和部分可见噪声：

1. 仅删除标签本身；
2. 删除标签本身及其相应的起始与结束标签包含的HTML 文本；
3. 对HTML 标签进行修正和配对，删除源码中的乱码。

2、区域噪音的处理

为了实现网页的导航，显示用户阅读的相关信息，并帮助用户实现快速跳转到其他页面，网页中一般要设计列表信息，把提供指向权威页面链接集合的一个或多个Web 页面称为HUB 页面，如图1 所示。

图 1　典型的列表信息模块(HUB 页面)

在处理此类信息时，本文设计了2 个噪音识别参数。Length=Length(content)为<tag>...</tag>标签内纯文本信息的长度，设定字符的ASCII code>255?length+2:length+1。

$$C_n = \frac{N_{string}}{N_{link} + N_{string}} \times \frac{NODE_{nohref}}{NODE_{href} + 1} \times 100\%$$

其中， Cn 为列表噪音判定系数；Nstring 是块中非链接字符的字数；Nlink 是块中链接字符的字数；NODEhref 是块中有href属性的节点数；NODEnohref 是块中没有href 属性的节点数。

3、基于标签路径聚类的网页分割

网页分割算法基于启发式规则，算法分为2 步：(1)Xpath聚类；(2)对聚类的Xpath 进行分割。本文约定DOM 树的叶节点按照其在原始HTML 文件中出现的先后顺序编号。

(1)Xpath 聚类。对具有最大相似度的叶节点进行聚类。节点取得最大相似度时2 个节点Xpath 完全相同。本文用向量,1 ,2 , $\boldsymbol{X}_i = \{x_{i,1}, x_{i,2}, \cdots, x_{i,n}\}$ }表示第i 个Xpath 的聚类。其中， xi, j 表示第i 个Xpath 聚类中的第j 个叶节点。

定义节点间距为1 个Xpath 聚类中2 个节点编号之间的间隔。

$$\Delta Span_{i,j,k} = \left| x_{i,j} - x_{i,k} \right|$$

式(2)表示第i 个Xpath 聚类的第j 个与第k 个节点之间的编号间隔。

定义平均周期为一个Xpath 聚类中相邻节点间距的均值。

$$\Delta T_i = \frac{1}{n-1} \sum_{j=1}^{n-1} \Delta Span_{i,j,j+1}$$

定义间距方差为考察一个聚类中各个节点离散程度的量。

$$\sigma^2(\Delta T_i)_j = \frac{1}{n-1} \sum_{j=1}^{n-1} (\Delta Span_{i,j,j+1} - \Delta T_i)^2$$

(2)分割点，将一个聚类中的不连续点称为分割点。为了反映分割点的具体位置定义了一个变量θ ，它是前后2 个间隔之间的比值。

$$\theta = \frac{\Delta Span_{i,(j+2),(j+1)}}{\Delta Span_{i,(j+1),j}} = \frac{x_{i,(j+2)} - x_{i,(j+1)}}{x_{i,(j+1)} - x_{i,j}}$$

为了增强分割鲁棒性，为θ 设定一个阈值范围。实验表明当θ ∈[0.85,2] 时可以得到较好的分割效果。

算法采用如下启发式规则：

(1)如果$\theta \notin [0.85,2]$，则将向量i X 在分割点处分割开。

(2)如果一个向量的平均周期$\Delta T > PreSpan$，且没有进行分割，节点数目大于预定义值，则认为已经到达网页内嵌块聚类的边界。

4 、算法描述

4.1 Xpath 聚类算法

将一个目标页面表示为DOM 树结构，采用深度优先遍历策略，提取DOM 树中的每个叶节点。对于每次遍历的叶节点，通过比较其Xpath，将其序号添加到具有最大相似度的Xpath 聚类中。具体算法描述如下：

```
Input DOMTree
Output XpathCluster
Cluster(DOM Tree)
{ XpathCluster = ∅ ;
For each xpath of leaf node
{
if (XpathCluster.xpath.Find(xpath))
{XpathClusfer.xpath.Insert(node);}
Else
{XpathCluster.Insert(xpath);
XpathCluster.xpath.Insert(node);
}
}
Return XpathCluster;
}
```

由于在聚类过程中，可能将非正文信息聚类到正文信息类中，因此先分析其方差。若一个聚类中的方差很大，则定位到分割点，将目标正文信息块与其周围的分隔噪音块分割开。另外，利用文本信息块的聚类平均周期、信息
长度和HUB 判别等统计参数帮助定位分割信息条。当第1 个满足全部启发式规则和统计信息的聚类出现时，可以认为已经找到了正文信息块，完成分割任务。

分割算法描述如下：

```
Input XpathCluster //Xapth 聚类
Output SegBoundary //分割边界
Variables: Integer: Length_Threshold; //正文长度的最小阈值
Float：Cn_ Threshold;// n C 列表噪音判定系数的阈值
WebPageSeg
{ SegBoundary = ∅ ;
Count=0;
While(Count!=XpathCluster.size())
{
If(XpathCluster.at(count).var0 is within threshold)
{If(xpathCluster.at(count).size()>MAXSIZE&&xpathCluster.at(cou
nt).length> Length_Threshold
&& xpathCluster.at(count). Cn > Cn_Threshold && ΔT >
Pr eSpan ) //check
{SegBoundary.insert(each node within XpathCluster. at(count))
Break;
}
Else Count++;
}
}Else{//利用启发式规则(1)进行分割
Detect segment point use(2.3.4)
Sort(new cluser);
Count++;
}
}
Return SegBoundary;
}
```

4.2 节点集合内的文本抽取算法

节点集合内的文本抽取算法描述如下：

```
Input SegBoundary[]；//分割出来的符合条件的文本块
Output TextHashMap<tagpath,table textchunk,document frequency>
//基于HashMap 的文本块模板映射
Variables Integer: Frequency_Threshold; //table/div 嵌套次数的
//
```

```
    阈值
StringBuffer: textChunk; //文本块
For each chunk p in SegBoundary[]
While p has more HTML nodes
nNode=p.nextnode;
If nNode is not table/div Tag
textChunk= textChunk+extracted text from nNode; //抽取nNode
//间的文本信息
else if nNode is table/div Tag
{
if TextHashMap.contains(tagpath)==true
{ documentfrequency++;}
Else{
Documentfrequency=1;
}
TextHashMap.put(tagpath,textChunk, documentfrequency);
}
While TextHashMap has more {tagpath,textChunk, document
frequency}
h is TextHashMap's item
If document frequency of h≥Frequency_Threshold
Print textChunk of item h
```

5、阀值的确定

在上述算法中， 需要设定3 个阈值参数：
Length_Threshold，Cn_Threshold，Frequency_Threshold，它们对算法的时间复杂度和抽取效果具有一定调节作用，处理网页结构
相似的网页时，可以通过训练样本自适应地算出相应的阈值。

对不同类型网页的阈值，3 个参数的数据分布有较大不同，Length、Cn 的数据分布绝大多数处于较小范围内，这些数据也是需要去掉的噪音数据，因此，使用K-means[4]对样本数据进行聚类处理，而frequency 数据相对前2 个参数没有明显的分布趋势，数据量不大，而且也处在{1-10}这样的一个较窄的局部区间中，实验表明，聚类分析效果不明显，因此，本文用算数平均值求解。

(1)单个样本网页的阈值训练

$$Length\_Threshold = Mid(Kmeans(Length[X], Clusternum)) \quad (6)$$

$$Frequency\_Threshold = \frac{\sum_{i=0}^{Y-1} documentfrequency[i]}{Y} \quad (7)$$

$$C_n\_Threshold = Mid(Kmeans(C_n[Z], Clusternum)) \quad (8)$$

(2)M 个同类样本的阈值训练

$$Length\_Threshold = Min(Length\_Threshold[M]) \quad (9)$$

$$Frequency\_Threshold = Min(Frequency\_Threshold[M]) \quad (10)$$

$$C_n\_Threshold = Min(C_n\_Threshold[M]) \quad (11)$$

其中，Kmeans(Array[],Cluseternum) 为聚类处理函数，Array[]为处理数据集合，Clusternum 为聚类数目；Min(Array[]) 获取集合最小值。

文章作者：山西工程职业技术学院网络电教中心 刘云峰

打赏作者

微信支付

支付宝

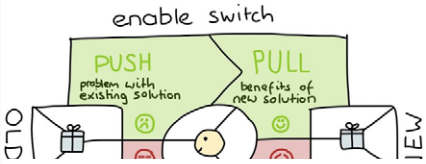
« 网站热力图工具：CrazyEgg

一种提高搜索引擎检索质量的网页解析法 »

0

Reply

0



什么情况下用户会发生“转移”

Jul 14, 2018 · 5 sec read

心理账户在产品营销的应用思考

什么是心理账户 心理账户是芝加哥大学行为科学教授理查德·萨勒(Richard Thaler)提出的概念。他认为，除了实际账户外，在人的头脑里还存在着另一种心理账户。人们会把在现实中客观等价的支出或收益在心理上划分到不同的账户中。比如，我们会把工资划归到 ...

Mar 5, 2018 · 5 sec read

基于人性弱点的营销

营销的核心是动机，抓住用户动机最好的方式是基于人性的弱点对产品进行设计。微信张小龙曾经说过，产品的终极目标是满足人性需求，贪嗔痴（欲望、嫉妒、执着）。类似负面情绪在产品设计中的作用一样。针对人性弱点除了在产品设计上使用外，还可以运用到营 ...

Feb 11, 2018 · 3 sec read

Leave a Reply

Name

E-mail address

Website Link

Post Comment