基于局部语义的网页净化算法

Reducing Web Noise Algorithm Based on Partly Semantic

谢 华　刘卫国　（中南大学信息科学与工程学院　湖南长沙 410083）

摘要：网页净化算法的目的是除去影响搜索引擎获取网页主题的噪音。本文提出一种基于局部语义的网页净化算法。算法遍历转化成 DOM 树后的网页，通过计算相邻节点的相似度，确定局部语义节点范围，然后提取局部语义信息，建立局部语义树模型，最后除去与网页主体的相关性低于预定阈值的局部语义节点，达到网页净化的目的。实验表明算法是有效的。

关键词：局部语义　网页净化　本地噪音

1　引言

随着 Internet 的发展与普及，以网页为主要信息载体的万维网已经成为一个信息共享的巨大平台。但是，网页上浏览者关注的信息经常与广告链接、站点导航信息、版权提示信息等网页噪音混合在一起。浏览者很容易区分网页中的有用信息与噪音。但是，对于 Web 信息处理来说，网页噪音会降低系统准确率。Lan Yi 等人把网页噪音分为两大类：全局噪音与本地噪音[1]。全局噪音指内容相同或相似的网页以及过时的网页。本地噪音指前面提到的广告链接、站点导航、版权提示等信息。消除本地噪音的过程叫做网页净化。

2　相关研究

国内外研究者针对大规模网页数据的网页净化算法进行了许多研究工作，相关算法也较成熟。针对大量数据的网页净化一般需要预先准备一个人工选择的包含大量网页的数据库，通过在数据库中的统计计算，得到网页噪音的判断规则。在处理数据库中网页的时候，需要初步判断哪些内容属于网页噪音，因此需要一种不依赖训练集的网页净化算法，另外网络服务个性化的相关研究也需要不依赖训练集的净化算法，但是这方面的研究并不多。以下是目前两个净化结果较理想、不依赖训练集的净化算法。

张志刚等人提出了一种针对单个网页的净化算法[2]。该算法将网页中常用的 HTML 标签＜table＞、＜tr＞、＜td＞、＜p＞、＜div＞等包含的内容看成内容块，从中提取文本特征，然后计算内容块与网页正文的相似程度，最后通过与给定阈值相比较，判断该内容块是否属于网页噪音。但是，该算法的基本单位分割过细，容易丢失与网页主题相关的内容块。

王琦等人同样把 HTML 标签＜table＞、＜td＞包含的内容视为内容块，计算块内超级链接的个数和非超级链接字符的长度[3]。该算法通过计算内容块相关程度判断网页噪音。但是，这种语义属性没有考虑网页的文本特征，无法判断不含超级链接的网页噪音，造成算法净化效果不理想。

针对以上两种方法的不足，本文提出一种改进的网页净化算法。

3　网页净化算法

3.1　算法基础

3.1.1　向量空间

Web 文本的表示方法有多种，一般的方法是先对 Web 文档中的信息进行预处理，提取出 HTML 标记中的文本信息，然后作为普通文本表示。近年来研究者提出了文本表示的向量空间模型（Vector Space Model，VSM）[4]，这是目前应用最多且效果较好的模型。

在 VSM 中，从文本中提取能代表文本含义的词语作为特征词，组成特征向量，并计算特征词的权重。例如文档可以表示为（t_1，t_2，…，t_N），其中 t_i（$1 \leqslant i \leqslant N$）是特征词。根据特征词重要程度，可以赋予不同的权重 wi 进行量化，这样文档也可表示为（w_1，w_2，…，

w_N),其中每一项 w_i 与相应的特征词 t_i($1 \leqslant i \leqslant N$)相对应。在 VSM 中,把 N 个特征词看成一个 N 维坐标系,则相应的权重 wi 为文档在坐标系中的坐标值,那么一个文档就可以被表示成一个 N 维空间中的向量。这样就把文档以向量的形式定义到实数域中,使得机器学习或其他领域中的计算方法得到应用,大大提高了文档的可计算性和可操作性。

3.1.2 DOM 树

HTML 是一种标识语言,它定义了一套标签来刻画网页显示时的页面布局。因此,HTML 网页结构最常用的表示方法是构造网页的标签树。现有的标签树构造工具很多。DOM(Document Object Model) 是一个常用的标签树构造工具,它可以将网页中的标签按照嵌套关系整理成树状结构,即将网页标签看成树内的节点,嵌套的网页标签看成该节点下的子节点。这样利用 DOM,网页被转化为由 HTML 节点构成的标签树,也叫做 DOM 树。

图 1 给出了一个简单的 HTML 网页文件的部分代码以及根据这段代码生成的 DOM 树。树的根节点 Body 对应 HTML 标签<BODY>。由于<Body>标签内嵌套着三个<Table>标签,所以根据 DOM 工具规则,在 Body 节点下添加三个 Table 节点。按照这个规则继续进行,图 1 上方的 HTML 代码就转化为图 1 下方的 DOM 树。

3.1.3 局部语义树

目前,网页净化算法需要预先人工制定信息提取范围,这种方法缺乏灵活性。当预定义范围偏小时,提取的特征词可能会偏少,不足以表达范围内的内容;当预定义范围偏大时,所取的特征词则可能会包含错误的信息。我们在遍历 DOM 树过程中引入链接密度的概念,通过计算相邻节点的相似程度,净化算法自动决定信息提取范围。

链接密度 ρ 是节点内超链接字符长度与节点内容长度的比值,代表了节点中超链接的疏密程度。经过对大量网页的分析发现,页面中相似的邻近节点通常具有近似的链接密度,因此可以利用链接密度计算相邻节点的相关度。给定两个节点 a 和 b,计算 a 与 b 相似度的公式如下:

$$Similar(T_a, T_b) = \begin{cases} \rho_a / \rho_b & \rho_a \leqslant \rho_b \\ \rho_b / \rho_a & \rho_a > \rho_b \end{cases} \quad \text{公式(1)}$$

其中 ρ_a 与 ρ_b 分别是节点 a 和 b 的链接密度。

```
<BODY>
  <TABLE>
      <TR>...</TR> <TR>...</TR> <TR>...</TR>
  </TABLE>
  <TABLE>
      <TR>
        <TABLE>
          <TR>...</TR> <TR>...</TR>
          <TR> <P>...</P> </TR></TABLE>
        <TABLE>
          <TR>...</TR> <TR>...</TR>
          <TR> <P>...</P> </TR></TABLE>
      </TR>
      <TR>
        <TABLE>
          <TR>...</TR> <TR>...</TR>
        </TABLE>
      </TR>
  </TABLE>
  <TABLE>
      <TR>...</TR> <TR>...</TR>
  </TABLE>
</BODY>
```

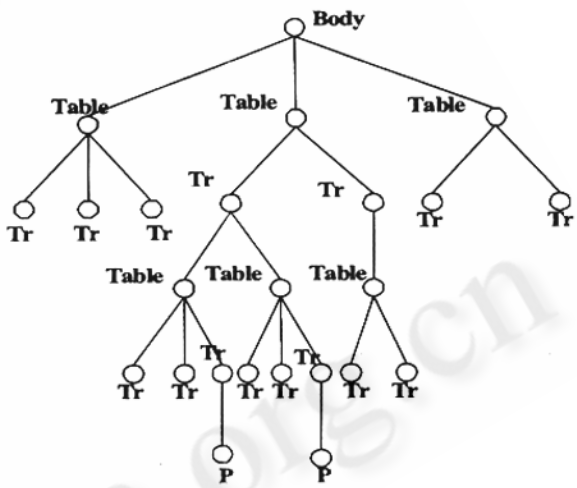

图 1　HTML 代码示例及对应的 DOM 树

如图 2 所示,通过计算相邻节点的相似度,在 DOM 树上划分出四个包含的节点都是彼此相关的范围 PS_1、PS_2、PS_3、PS_4。我们将四个范围看成四个内容块。接下来,需要计算各内容块对应的特征向量。

在量化方法上,对权重的计算,比较常用的是 TF－IDF 方法[5],其计算公式为:

$w_i = tf_i / Log(N/N_i)$　　公式(2)

其中 tf_i 是第 i 个特征词在文档 D 中的出现次数,N 是文档集中的文档总数,Ni 是文本所在文档集中含有第 i 个特征词的文档数。从公式中可以知道,计算 TF－IDF 值需要在文档集中进行相关统计计算,但是本文算法不依赖文档集,所以在计算分量权重 Wi 的时候需要将计算范围缩小到当前页面中。我们对 TF－IDF 函数进行了修改。修改后的 TF－IDF 函数如

下：

$$w_i = [Log(n_i/N)\sum_{j=1}^{N} tf_{ij}] / \sqrt{\sum_{i=1}^{n}[\sum_{j=1}^{N} tf_{ij}]} \quad \text{公式(3)}$$

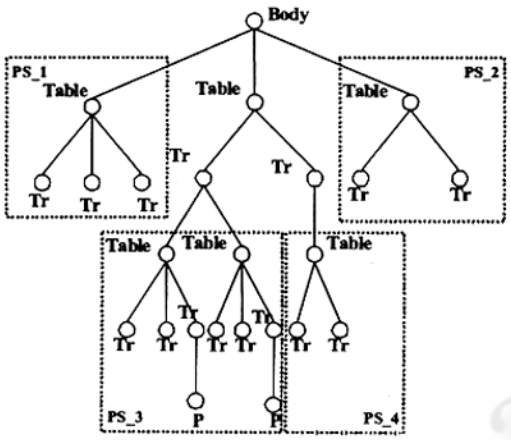

图 2　划分范围后的 DOM 树

其中 N 为内容块的总数，n 为特征词的总数，n_i 为出现了特征词 w_i 的节点数，tf_{ij} 为特征词 i 在节点 j 内出现的次数。与传统的 TF－IDF 函数相比，新的函数将 N，n_i 的涵义缩小到单个页面内，克服了对文档集的依赖。

在本文中，我们定义语义信息为特征向量和链接密度。计算出所有的内容块的特征向量后，内容块转变为包含语义信息的节点，我们称之为局部语义节点。DOM 树转变成由局部语义节点构成的局部语义树。

3.2　网页净化

如图 3 所示，网页净化算法分 4 个部分：DOM 构造工具、语义处理转化器、剪枝器、语义分析器。DOM 构造工具采用 HTMLParser [6]。语义处理转化器计算 DOM 树上节点的相似度，确定信息提取范围，提取语义信息。剪枝器利用语义分析器对转化后的节点进行分析计算，去除网页中的本地噪音。

下面介绍网页净化中的两个关键算法：网页分割算法和剪枝算法。

3.2.1　网页分割算法

网页分割的过程是将 DOM 树转化为局部语义树的过程。语义处理转换器从 DOM 树的叶子节点出发，向上逐级计算邻近节点的相似度。当一个节点下的所有子节点彼此相似时，便可将这个节点下的所有节点视为当前信息提取范围，然后继续向上计算，直到节点下的子节点彼此不相似为止。下面给出具体的网页分割算法的描述。

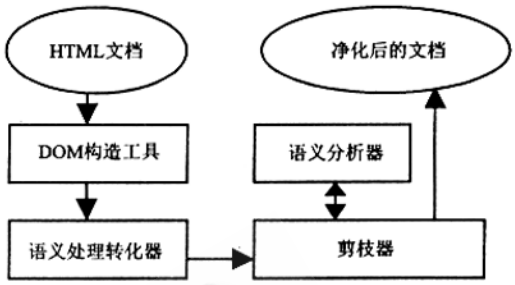

图 3　网页净化过程

```
HTMLSegment(Root) //Root 为根节点
if Tag 是叶子节点 then //Tag 为 DOM 树的节点
    计算 Tag 密度
    更新父节点的网页密度并返回
else
    for each Tag∈Root.Children do
      HTMLSegment(Tag)
    end if
    用公式(1)计算 Root 子节点的相关度
    将相关节点加入到同一信息提取范围
end for
```

3.2.2　剪枝算法

网页分割完成后，接下来对各局部语义块进行语义分析，计算相关块的特征向量，判断网页内容块是否为本地噪音。

首先，算法利用语言处理工具对局部语义块进行处理，得到特征词集合。然后对集合内的特征词进行选择，去除对表达块的内容贡献不大的特征词。接下来利用公式(3) 计算出特征词的权重后，得到块特征向量。最后，计算局部语义节点特征向量与网页正文节点特征向量的余弦值。如果计算结果高于相似度阈值，说明该节点与网页正文的相关程度较低，可以作为本地噪音去除。具体算法如下：

```
CleanNoise(Root)
for each PSi∈Root.Children do
    用公式(3)计算 PSi 的块特征向量
    if PSi 的特征词数 is max then
    标记网页正文块 PS_m 为 PSi
    end if
end for
for each PSi∈Root.Children do
```

```
    计算 PS_m 与 PS_i 的余弦值 s
    if s > λ then //λ为相似度阈值
      从局部语义树中删除 PS_i
    end if
end for
```

4 实验分析

算法用 Java 语言实现，语言处理部分采用中科院计算所汉语词法分析系统 ICTCLAS[7]。实验分为两个部分：有效性实验和适应性实验。

4.1 有效性实验

不依赖训练集的网页净化算法首先要求能准确的识别单个网页的网页噪音。因此我们选择了一个主题型网页。主题型网页通常在网页主要部分包含大量文字，在网页的右侧或者右侧包含了大量链接，在网页的其余部分则是一些导航信息、广告信息、版权信息等本地噪音。主题型网页在 Web 上比较常见，具有代表性，因此实验可以反映算法对网页噪音的识别能力。实验结果表明，本文的算法能够准确地判别该类网页的主体信息和网页噪音，并能完整地保留与网页主体信息相关的链接，能保留更多的有用的链接信息。

4.2 适应性实验

不依赖训练集的网页净化算法要有很好的适应能力，即不依赖于具体的网页模版。这里的网页模版是指一种网页布局结构，比如导航栏的位置、广告的位置、内容主体的位置、版权信息的位置等等。在 Web 中，同一个网站的不同栏目以及不同网站的页面经常会使用不同的模版，因此可以用不同网站不同栏目的网页可以验证算法的适应能力。因此我们从几大门户网站的不同栏目，例如人文与艺术、商业与经济、娱乐与休闲、计算机与因特网、教育、区域、自然科学、政府与政治、社会科学、医疗与健康、社会与文化等，选取了部分网页。同时我们还用搜索引擎以不同关键字随机从网上获取部分网页作为实验数据。表 1 显示了测试结果。

在表 1 中，完整性是保存的主题内容及相关链接长度占来源网页主题内容及相关链接长度的百分比。压缩比是结果网页文件大小占来源网页文件大小的百分比。在这里平均压缩比为算法的测试程序自动计算的结果，完整性是通过人工对比分析的结果。

表 1　净化实验结果

来源网站	类别数	网页数	平均完整性	平均压缩比
新浪	28	56	88.70%	14.30%
搜狐	27	54	78.50%	13.02%
赛迪	19	38	94.01%	12.01%
人民网	36	72	81.40%	11.00%
天极网	16	32	82.00%	11.50%
太平洋网	11	22	84.02%	13.30%
CSDN	24	48	91.02%	16.01%
其他	30	60	74.30%	11.00%

由表 1 的实验结果可以看出，本算法在保存网页的主要信息、消除本地噪音方面具有较高的准确性，而且不依赖网页模版。

5 结语

本文针对个性化网络服务中网页净化对不依赖训练集的需要，提出了一种基于局部语义的网页净化算法。算法将 HTML 文件转化为 DOM 树，计算相邻节点的相关度，确定信息提取范围，逐步构造局部语义树，最后通过剪枝算法除去与页面主题不相关的节点，达到清除网页中本地噪音的目的。

参考文献

1 Lan Yi, Bing Liu, Xiaoli Li. Eliminating Noisy Information in Web Pages for Data Mining[R]. Washington, DC, USA: SIGKDD '03, 2003.

2 张志刚、陈静、李晓明，一种网页净化方法[J]. 情报学报，2004，23(4)：387－393.

3 王琦、唐世渭、杨冬青、王腾蛟，基于 DOM 的网页主体信息自动提取[J]，计算机研究与发展，2004，42(10)：1786－1792.

4 Salton G, Wang A, Yang C S. A vector space model for automatic indexing [J]. Communication of ACM, 1975, 18(11): 613－620.

5 Lewis D. D., et al. Training algorithms for linear text classifiers. In: Proceedings of the Nineteenth International ACM SIGIR Conference on Research and Development in Information Retrieval, 1996. 298～306.

6 http://htmlparser.sourceforge.net/.

7 http://www.nlp.org.cn/.