ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.20, No.5, May 2009, pp.1282−1291
doi: 10.3724/SP.J.1001.2009.03283

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于网页格式信息量的博客文章和评论抽取模型*

曹冬林[1,2,3+], 廖祥文[1,2], 许洪波[1], 白 硕[1]

[1](中国科学院 计算技术研究所 网络科学与技术研究部,北京 100190)

[2](中国科学院 研究生院,北京 100049)

[3](厦门大学 智能科学系,福建 厦门 361005)

Extraction Model Based on Web Format Information Quantity in Blog Post and Comment Extraction

CAO Dong-Lin[1,2,3+], LIAO Xiang-Wen[1,2], XU Hong-Bo[1], BAI Shuo[1]

[1](Department of Network Science and Technology, Institute of Computing Technology, The Chinese Academy of Sciences, Beijing 100190, China)

[2](Graduate University, The Chinese Academy of Sciences, Beijing 100049, China)

[3](Department of Cognitive Science, Xiamen University, Xiamen 361005, China)

+ Corresponding author: E-mail: another@xmu.edu.cn

Cao DL, Liao XW, Xu HB, Bai S. Extraction model based on web format information quantity in blog post and comment extraction. *Journal of Software*, 2009,20(5):1282−1291. http://www.jos.org.cn/1000-9825/3283.htm

Abstract: Based on the information theory, this paper presents a model based on Web format information quantity in blog information extraction. First, the vision information in blog Web page and the effective text information are combined to locate the main text which represents the theme of the blog Web page. Second, the format information of blog Web page is used to calculate the information quantity of each block and the minimal separating information quantity of separate position is used to detect the boundary of posts and comments in the main text. This model is language insensitive and can be used in a lot of blogs which are written in different natural languages. Experimental results show that this method achieves high precision in locating main text and separating the post and comment.

Key words: blog information extraction; minimal main text subtree; effective information ratio; Web format information; vision information; information quantity of separate position

摘 要: 从信息论的角度出发,提出了一个基于网页格式信息量的博客文章和评论抽取模型.首先,结合网页视觉上的位置信息和文本的有效信息来定位网页正文.其次,利用博客网页中的格式信息作为信息单元并计算每个信息块所包含的格式信息量,通过计算最小切分位置信息量来切分正文中的文章和评论.该模型具有与语言无关的特点,因此具有一定的通用性.实验结果表明,该模型在博客正文定位和正文切分方面达到了较高的精确率.

* Supported by the National Basic Research Program of China under Grant Nos.2004CB318109, 2007CB311100 (国家重点基础研究发展计划(973)); the National High-Tech Research and Development Plan of China under Grant No.2007AA01Z441 (国家高技术研究发展计划(863))

Received 2007-07-03; Accepted 2008-02-27

关键词: 博客信息抽取;最小正文子树;有效信息率;网页格式信息;视觉信息;切分位置信息量
中图法分类号: TP181 文献标识码: A

随着Internet的发展,博客作为一种新的大众化的信息发布媒体越来越受到网民和企业界的重视.与传统的BBS和讨论群不同,博客更注重传达个人的意见以及情感.博文作为博客本人的意见以及情感的体现,在一定程度上具有很强的研究价值.博文是一个巨大的知识库,与通常意义上的知识库不同,博客有大量的知识添加人员,其知识覆盖面大、实时性强.如何从博客中获取知识,成为目前信息检索领域一个新的研究课题.在博客搜索中,需要精确地对博文内容进行检索.为了提高检索的精确率,精确抽取博文内容成为在搜索引擎建索引之前的一个重要环节.

对网页的信息抽取,国内外都进行了大量的研究.在去除广告链接方面,链接/文本比率取得了不错的效果.该方法通过计算网页格式标签块中的链接/文本的比率,取链接/文本比率较小的网页格式标签块作为网页的有效文本.因此,该方法可以去除网页中大量的广告链接信息.但该方法对博客中诸如“关于作者”、“版权”等文字噪音没有作用.其主要原因是,这些文字信息与正文信息有着相近的链接/文本比率[1].基于规则的方法在抽取特殊信息方面也有很高的精确率.其代表的方法有基于自然语言处理的方法、基于分装器的方法[2−8]和基于Html-aware的方法[9].这些方法可以精确地从特定网页中抽取时间、网页点击数等信息.如:文献[3]使用MDR(mining data region)的方法自动找到网页中存在的数据记录,该方法能够找到网页中存在的非连续数据记录;文献[8]总结了现有的信息抽取方法,并提出了一种基于局部树对齐的方法来实现数据记录的抽取;文献[9]提出了一种IEPAD的方法,从网页格式标签中发现一些特定模式,并通过发现的模式来实现抽取数据记录的目的.这些方法的主要问题在于需要编制一些精炼的规则,并且这些规则往往是与语言相关的,对于不同版式和语言的网页,必须重新编制相应的规则.虽然可以通过机器学习的方法从语料库中学习到规则[10],但是学习到的规则取决于语料库的覆盖面以及语料库标注的准确程度.要建立一个大规模的精确语料库需要花费大量的人力和物力.

新闻网页和博客网页有一定的相似之处,页面中都存在由文字组成的文章.对于这些网页,我们需要去除噪音信息,从中取出能够代表网页主题信息的文章,我们称其为正文.但与新闻网页不同的是,新闻网页的正文仅仅包含文章,而博客网页的正文分成文章和评论两部分.因此,在抽取博客正文信息时需要进行正文定位和正文切分两部分工作.正文切分的工作有些类似于主题划分问题.在主题划分方面也已经有了大量的相关研究:Reynar在其博士论文中对大量的主题划分算法进行了综合的描述和评价[11].Qi提出的基于曲线的主题划分方法在文本流的边界判别上也取得了不错的效果[12].但是,博客文章和评论的主题在大部分情况下是相同的,而主题划分一般都是区分不同主题,因此,主题划分的方法在博文的正文切分中很难达到较高的切分精确率.

由于博客的特殊性,现有的一些正文抽取算法和主题划分算法都很难对其进行精确的定位和切分.因此,在博客信息抽取方面需要研究新的定位和切分算法,以适应博客检索的需求.本文围绕博客抽取问题,研究了正文定位和正文切分问题.

1 基于网页格式信息量的博客文章和评论抽取模型

在对网页格式信息的分析中,我们使用了DOM(document object model)树的方法将网页解析成树型结构.依据该结构,我们定义相应的正文定位任务如下:

设W为我们需要定位的正文,T为博客网页S的DOM树结构.我们的正文定位目标为找到T中的一个包含正文W的最小子树T_m.由于T_m是包含正文的最小子树,因此对任意子树T_i且$T_i \subset T_m$,T_i都不可能包含完整的正文W,否则,T_m就不是包含正文的最小子树.我们称T_m为最小正文子树.

同样,我们的正文切分任务如下:

设W_p是正文W中的文章,W_c是正文W中的评论.我们的正文切分目标为,在最小正文子树T_m中找到一个切分位置S_p,使得S_p能够将子树T_m切分成两部分T_l和T_r.其中,T_l和T_r满足条件$W_p \in T_l$并且$W_c \in T_r$.

1.1 正文定位

在正文定位中需要过滤所有非正文的噪音.我们把博客中的噪音分成 3 类:第 1 类是广告,这些链接和图片对检索系统没有任何帮助;第 2 类是有效链接,这些链接包括本博客链接到其他博客的链接信息.这些链接的作用反映在链接分析中,在正文定位中我们也视其为噪音;第 3 类是一些无效文本,如版权信息、关于作者的信息等等.链接/文本比率的方法虽然在新闻网页中可以取得不错的效果,但是应用到博客中还存在一些问题:首先,无法去除无效文本信息,如网页中的版权信息;其次,博客正文中可能发布大量的链接信息,这也使得该方法有一定的局限性;最后,链接/文本比率并不是一种精确定位正文的方法,该方法是通过去除多余的链接来减少网页中的噪音.另外一种基于文本量的方法在正文文本量较大(即正文的文字数较多)的情况下能够取得很好的效果,该方法计算每个网页格式标签块的非链接的文本量,将非链接文本量最大的部分作为正文.但是该方法同样不能去除无效文本信息,同时,由于不考虑链接/文本的比率,该方法在定位时会将一些链接噪音加入正文.如版权信息非链接文本量多于正文非链接文本量时,该方法会错误地将版权信息作为正文.综合上面两种方法,在正文定位中我们结合了基于链接/文本比率和基于文本量的方法,将两者以类似于信息论中的信息量计算的方式进行结合.

由于将链接信息作为噪音,因此我们定义一个有效信息率,计算公式为

$$EI = \log_2\left(1+\frac{NW_e}{NW_a}\right) \tag{1}$$

其中,NW_e 为文本中非链接文本所占的字节数,NW_a 为文本所占的字节数.该公式反映的是有效文本在全体文本中的比率,加 1 是为了确保比率值大于 0.通过有效信息率,我们可以进一步计算该文本的有效信息总量,计算公式为

$$I_e=EI\times NW_e \tag{2}$$

有效信息总量在一定程度上反映了网页格式标签块中的有效文本信息,能够很好地去除第 1 类和第 2 类的噪音.在大多数情况下,由于博客正文的文本多于第 3 类噪音的文本,通过有效信息总量,就可以很好地去除.但当博客正文文本较少,与第 3 类噪音文本相当时,不易去除第 3 类噪音.在该情况下,我们注意到博客正文的文本在网页中的视觉位置是大于第 3 类噪音文本,因此,我们在正文定位算法中加入了网页中的视觉位置信息,通过文本的视觉位置信息和有效信息总量来综合判断.这里的视觉位置信息是指网页的宽度、高度等视觉信息,但在我们的算法中仅仅考虑网页格式标签块的宽度信息.我们的正文定位算法如下:

正文定位算法.

1. 分析博客网页得到 DOM 树;
2. 计算 DOM 树中每一个节点的有效信息总量;
3. 分析网页得到每一个节点的网页视觉宽度信息;
4. 从 DOM 树的根节点开始完成以下步骤:
   - 4.1. 对于当前的根节点,找到含有最大视觉宽度并且含有的文本数超过指定阈值的直接孩子节点.将该孩子节点作为根节点;
   - 4.2. 若所有的直接孩子节点具有相同的视觉宽度,则找到含有最大有效信息总量的直接孩子节点,并将其作为根节点;
   - 4.3. 若信息损失率超过指定阈值,则转至步骤 5;否则转至步骤 4.1 继续执行.信息损失率的计算公式如下:

$$LossRatio = \frac{I_e(P)}{I_e(C)} \tag{3}$$

   其中,$I_e(P)$是父节点的有效信息率,$I_e(C)$是孩子节点的有效信息率;
5. 将选中的孩子节点的文本范围作为正文的抽取范围.

算法中的 *LossRatio* 阈值不是人工设定的,因为对于不同的网页,其阈值很可能不同.如:对于一个链接噪音

较多的网页,其阈值就比较小;而对于一个几乎没有噪音或噪音较少的网页,其阈值就比较大.为了动态设定信息损失率的阈值,我们首先使用正文定位算法遍历子树直至叶子节点,并记录下树中每层次的 *LossRatio*,然后计算所有 *LossRatio* 的平均值,将平均值作为阈值,最后寻找从根开始第 1 个超过阈值的节点作为终止节点.

1.2 正文切分

在博客网站建立的时候,网页设计者在网页设计上就通过相应的格式信息来达到视觉上文章和评论间有所区别.因此,我们完全可以从网页格式信息的角度出发来找到相应的切分位置.与基于 DOM 树的子树比较的方法不同,我们不去比较子树之间的相似度来找到相应的切分位置.因为这种方法依赖于如何定义子树的相似度,并在一定程度上具有经验性质.

计算子树的相似度,实际上就是计算子树的冗余度[13].在数据压缩上,信息论就显示了其突出的冗余信息检测能力.从信息论的角度出发[14,15],如果存在 3 个字符串 B_1,B_2 和 B_3,则将字符串 B_3 加入到 B_1 或 B_2 的末尾,都将引起 B_1 或 B_2 信息量的增加.若字符串 B_3 与 B_1 更相近,那么把 B_3 加入到 B_1 所引起的信息量增量将比把 B_3 加入到 B_2 所引起的信息量增量要少.因此,如果我们将切分位置两边的网页格式信息映射为两个字符串,则可以利用这一特点来进行切分位置的选择.

如图 1 所示,由于任意多个切分位置都可以转化为两个切分位置的比较,因此,假设有两个可能的切分位置(S_1 和 S_2),于是整个网页被划分为 3 部分(M_1,M_2 和 M_3).如果我们将 M_3 中的格式信息作为一个整体 A,那么格式 A 在 M_1 中的概率为 P_1,格式 A 在 M_2 中的概率为 P_2.信息量的计算公式为 $H=-n\times\log_2(P_i)$,即每个字符信息熵的总和,其中,n 是字符个数,P_i 是字符出现的概率.因为 M_3 中的格式信息作为一个整体 A,所以在该情况下字符个数为 1.若将 M_3 加入 M_2,并且不考虑概率变化,则有信息增量 $\Delta H_1=-1\times\log_2(P_2)$.同理,若将 M_3 加入 M_1 并且不考虑概率变化,则有信息增量 $\Delta H_2=-1\times\log_2(P_1)$.若 $P_1>P_2$,则 $-\log_2(P_1)<-\log_2(P_2)$,因此有 $\Delta H_1>\Delta H_2$.这说明 M_3 加入 M_1 中引起的信息增量少于 M_3 加入 M_2 中引起的信息增量,格式 A 更多地出现在 M_1 中.因此,我们应该将切分位置 S_2 作为一个更好的切分选择.同理,若 $P_1<P_2$,则我们应该将切分位置 S_1 作为一个更好的切分选择.

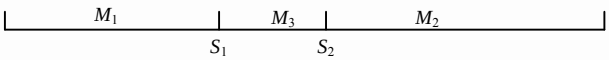


Fig.1 Status of two separate positions

图 1 两个切分位置时的状态

由于每个可能的切分位置都将网页切分成两部分,我们通过计算两部分的信息量的总和来求得该切分位置的信息量,因此,我们定义一元模型的切分位置信息量如下:

$$IS=\sum_{k=1}^{2}\left(\sum_{i=1}^{m}((-P_k(l_i)\log_2 P_k(l_i))\times Num_k)\right) \tag{4}$$

其中,m 是网页中格式标签的种类个数,l_i 是网页中的第 i 种格式标签,k 是被切分位置切分的第 k 块,$P_k(l_i)$是第 i 种格式标签在第 k 块中的概率,Num_k 是第 k 块中格式标签的个数,而 $Num_k\times P_k(l_i)$就是第 k 块中第 i 种格式标签出现的次数.若记 $Num_k\times P_k(l_i)$为 n_{ki},则公式(4)可以改写为

$$IS=\sum_{k=1}^{2}\left(\sum_{i=1}^{m}(-n_{ki}\log_2 P_k(l_i))\right) \tag{5}$$

在上面对图 1 的分析中,我们是将 M_3 中的格式信息作为一个整体来考虑的.下面我们考虑每个网页格式标签作为一个独立的个体并且相互之间独立的一元模型下的切分情况.对于图 1 中两种可能的切分位置的情况,假设网页共有 m 种不同的格式标签.若第 i 种格式标签出现在 M_j 中,其概率记为 $P_j(l_i)$,出现次数记为 n_{ji},那么块 M_1 和 M_2 的信息量为 $\sum_{i=1}^{m}(-n_{1i}\log_2 P_1(l_i)-n_{2i}\log_2 P_2(l_i))$.如果选择 S_1 作为切分选择,那么信息量将在 M_1 和 M_2 的基础上增加 ΔH_1.在一元模型中,我们需要考虑由 M_3 加入到 M_2 所引起的第 i 种格式标签概率的变化.这是因为在 M_2 中第 i 种格式标签的概率为 $P_2(l_i)$,而将 M_3 加入到 M_2 后,引起第 i 种格式标签的概率变化为 $P_2(l_i)'$.由于将 M_3

和 M_2 合并不会引起 M_1 中概率分布的变化,因此信息增量 $\Delta H_1=\sum_{i=1}^{m}(-(n_{2i}+n_{3i})\log_2 P_2(l_i)'+n_{2i}\log_2 P_2(l_i))$.同样,如果选择 S_2 作为切分选择,信息增量为ΔH_2,且 $\Delta H_2=\sum_{i=1}^{m}(-(n_{1i}+n_{3i})\log_2 P_1(l_i)'+n_{1i}\log_2 P_1(l_i))$.下面,我们来比较上述两个信息增量.

$$\Delta H_1-\Delta H_2=\sum_{i=1}^{m}(-(n_{2i}+n_{3i})\log_2 P_2(l_i)'-n_{1i}\log_2 P_1(l_i))-\sum_{i=1}^{m}(-(n_{1i}+n_{3i})\log_2 P_1(l_i)'-n_{2i}\log_2 P_2(l_i)),$$

其中,$P_1(l_i)'$是第 i 种格式标签在块 M_1 和 M_3 中的概率,$P_2(l_i)'$是第 i 种格式标签在块 M_2 和 M_3 中的概率.根据公式(5), $\sum_{i=1}^{m}(-(n_{2i}+n_{3i})\log_2 P_2(l_i)'-n_{1i}\log_2 P_1(l_i))$ 实际上就是以 S_1 作为切分选择得到的切分位置信息量.因此,我们可以得出下面的公式:

$$\Delta H_1-\Delta H_2=IS_1-IS_2 \tag{6}$$

其中,IS_i 是切分位置 S_i 的切分位置信息量.上述公式表明,比较信息增量实际上就是比较切分位置的信息量,并且产生最小信息增量的切分为具有最小切分位置信息量的切分.因此,我们只需要计算每个可能的切分位置的信息量,找到具有最小切分位置信息量的切分位置.

考虑到网页中格式标签之间并不是相互之间独立,我们进一步考虑各个格式标签之间的前后依赖关系.对应的二元模型的切分位置信息量如下:

$$IS=\sum_{k=1}^{2}\sum_{i=1}^{m}\sum_{j=1}^{m}((-P_k(l_j\mid l_i)\log_2 P_k(l_j\mid l_i))\times Num_k(l_i)) \tag{7}$$

其中,$P_k(l_j|l_i)$是在第 k 块中第 i 种格式标签出现的情况下第 j 种格式标签出现的概率,$Num_k(l_i)$是在第 k 块中第 i 种格式标签出现的次数.

依据上述公式(4)和公式(7),我们的正文切分算法如下:

正文切分算法.

1. 分析正文部分的网页得到其 DOM 树;
2. 去除 DOM 树中的所有非网页格式标签节点;
3. 对根节点的每个直接孩子节点完成如下步骤:
   - 3.1. 依据孩子节点将树分成两部分:第 1 部分包含该孩子子树以及孩子节点的左部所有兄弟子树;第 2 部分包含孩子节点右部的所有兄弟子树;
   - 3.2. 使用遍历方法将两部分转化为线性结构;
   - 3.3. 计算该切分位置的信息量;
   - 3.4. 如果所有直接孩子都被遍历,则转至步骤 4;
4. 选择具有最小切分位置信息量的切分.

在将树形结构转化为线性结构时,我们考虑了两种遍历方式:一种是前序遍历,这种方式保留了各个网页格式标签在网页文本中的前后次序;另外一种是层次遍历,这种方式则考虑了树形结构中的父子关系,使得 $P_k(l_j|l_i)$ 表示第 j 种格式标签作为第 i 种格式标签的子节点出现的概率.在一元模型中,由于前序遍历和层次遍历对模型没有影响,因此,一元模型我们仅仅考虑前序遍历方式,而二元模型考虑了两种不同遍历方式对结果的影响.

在切分算法中,我们只要遍历最小正文子树根节点的直接孩子即可,无须遍历所有子孙节点.这主要是因为:(1) 若定位算法找到最小正文子树,则通过直接孩子节点就可以正确切分文章和评论.(2) 节省计算量.我们以图 2 为例来说明这一点.首先,我们不考虑所有正文定位过程中定位到的子树中不包含完整正文的情况,因为在这种情况下,该子树中不包含完整的正文或没有正文,因此不可能产生正确的切分.排除上述因素后,在正文切分之前存在图 2 中所显示的 6 种子树状况.其中,图 2(a)~图 2(d)中的文章和评论位于不同的直接孩子中,这种情况下,我们可以使用节点 3 的开始位置作为文章和评论的切分位置.在图 2(e)中,文章和评论位于相同的直接孩子节点 2 中,因为我们定义的最小正文子树为包含正文(文章和评论)的最小子树,因此节点 1 并不是最小正文

子树的根节点,节点 2 才是最小正文子树的根节点.在该情况下,最小正文子树定位错误,因此无法通过直接孩子节点来切分文章和评论.图 2(f)是图 2(e)的一种特殊情况,虽然正文定位并没有定位到最小正文子树,但因为根节点只有 1 个直接孩子,因此在切分处理中,我们可以进一步定位到节点 2,将其作为根节点,然后图 2(f)就可以转化为图 2(a)得到正确的切分.从上面 6 种情况我们可以看到,在正文定位错误(没有找到最小正文子树)的情况下,我们的切分算法只有在图 2(f)的情况下才可能切分正确.但由于状态图 2(f)出现的次数很少,所以在正文定位错误的情况下,算法将产生错误的切分.在定位正确的情况下,通过直接孩子可以找到一个正确的文章和评论的切分位置.

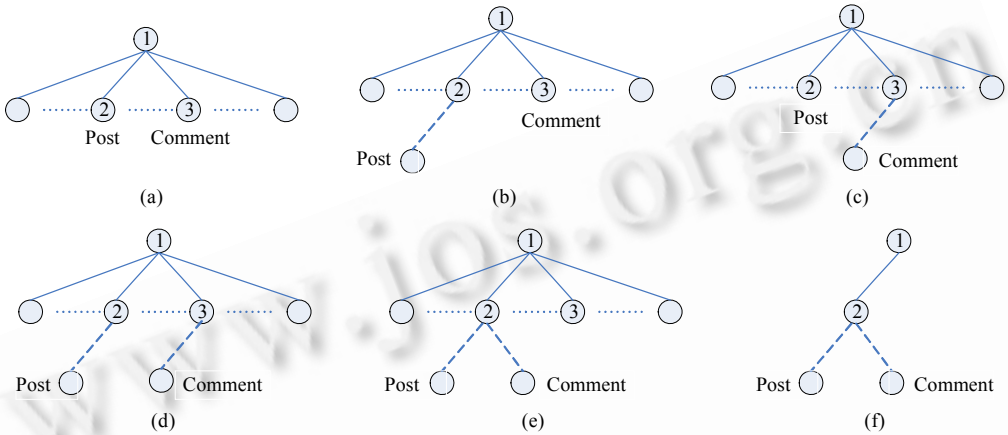


Fig.2 Six situations of separating

图 2 6 种切分情况

2 实验及分析

2.1 测试语料预处理

实验中我们采用 TREC 2006 博客任务中的语料.该语料是从一些著名的国际博客网站上采集得到 r .内容包含博客主页、博文和 feeds 等,语料依据采集时间来组织.我们从该语料中抽取 2005 年 12 月 7 日采集的语料并 wt 了 3 步处理:第 1 步,依据每个博文网页的域名归类,取网页数目排名前 100 的博客域名内的网页.这种博客域名内的网页具有代表性.第 2 步,对每个网页下载其必要的 CSS(cascading style sheets)格式信息.这些格式信息用于获取网页的视觉宽度信息.最终得到 25 910 个博客网页.第 3 步,依据格式信息对网页进行归类,对每一类网页进行标注,标注博客网页正文信息和切分位置信息.正文位置是以最小正文子树的根节点作为标记,切分位置是以评论子树的根节点作为标记.虽然有些博客网页中不存在评论,但其评论子树的根节点的网页格式标签仍然存在,因此我们可以使用评论子树的根节点作为切分位置.

语料库中 25 910 个网页的分布见表 1.由于有的域名同属于一个网站,因此 25 910 个网页涵盖了表 1 中列出的 9 个博客网站.

Table 1 Corpus distribution

表 1 语料库分布

Site	# Pages
www.livejournal.com	16 944
blogspot.com	7 401
nospeedbumps.com	382
www.plogress.com	340
ipunkrock.com	318
www.blogespierre.com	215
weblogs.java.net	110
redjar.org	100
www.sff.net	100

2.2 实验评价

在实验之前,我们定义了以下 4 个精确率计算公式:

$$Precision(MainText)=\frac{NL}{NCorpus},\ Precision(PostComment)=\frac{NS}{NCorpus},$$

$$Precision(PostComment\mid MainText)=\frac{NSL}{NL},\ Precision(MainText+PostComment)=\frac{NSL}{NCorpus},$$

其中,*NCorpus* 是语料库中的博客页面数,*NL* 是在正文定位中正确的博客页面数,*NS* 是在正文切分中正确的博客页面数,*NSL* 是在正文定位和正文切分中都正确的博客页面数.

这些公式中,最重要的是 *Precision*(*MainText*)和 *Precision*(*PostComment*|*MainText*).*Precision*(*MainText*)反映了正文定位的精确率,而 *Precision*(*PostComment*|*MainText*)反映了正文切分的精确率.

2.3 实验结果

实验主要分成 3 部分:首先,我们测试了正文定位和正文切分中的一元模型和二元模型;然后,对正文定位和正文切分分开考虑,比较正文定位算法和其他算法的定位效果,并比较正文切分算法中一元模型和二元模型在不同文章和评论比率中的效果;最后,测试了算法的语言敏感度,测试算法在不同语言的博客网页中的效果.

2.3.1 算法总体效果测试

对语料中的 25 910 个网页进行测试,得到相关模型的 4 个精确率,见表 2.

Table 2 Algorithm precision

表 2 算法精确率

	Precision (unigram) (%)	Precision (bigram+preorder) (%)	Precision (bigram+level order) (%)
MainText	87.33	87.33	87.33
PostComment	81.08	65.35	72.24
PostComment|*MainText*	92.82	74.81	82.69
MainText+PostComment	81.07	65.34	72.22

最终的精确率显示,一元模型优于二元模型.其主要原因是,在经过正文定位后,正文内的格式标签个数较少,在二元模型中容易产生数据稀疏的情况.在二元模型中,使用层次遍历的效果要好于前序遍历,这是因为层次遍历反映了 DOM 树的层次关系更贴近树的结构形式,而前序遍历更多地反映的是网页格式标签的先后顺序关系.我们可以看到:在正文定位上,算法的精确率达到了 87.33%;在正文切分上,一元模型和二元模型的精确率分别达到了 92.82%和 82.69%.算法在各个站点上的效果见表 3.

表 3 的数据结果反映出,我们的定位算法在 8 个站点上取得了 80%以上的正文定位精确率,切分算法在 7 个站点上取得了 80%以上的切分精确率.算法总体效果是在 6 个站点上取得了 80%以上的精确率.算法总体效果在www.blogespierre.com上最差,为 55.81%.其主要原因是该网站几乎没有广告信息,在正文定位时,算法经常定位到最小正文子树根节点的父节点,并且博客的文章和评论使用相同的网页排版格式使得切分算法难以区分.而效果较好的 6 个博客站点的网页中文章和评论之间的格式具有较大的不同.实际上,我们的算法效果要好于表 3 中给出的精确率,这是因为我们在评测时使用最小正文子树的根节点作为正确的正文位置,以评论子树根节点作为正确的切分位置,但实际上,最小正文子树的父节点虽然带有一定的噪音,但如果相对于正文噪音很小,则可以基本上等价于正文,并且文章和评论之间不仅仅有 1 个正确的切分位置.

在表 2 和表 3 中,*Precision*(*PostComment*)和 *Precision*(*MainText+PostComment*)是几乎相等的,这是因为在正文切分中我们仅仅考虑最小正文子树根节点的直接孩子节点,并没有递归遍历孙子节点.在前面对图 2 的 6 种正文切分情况的分析中,我们有"在正文定位错误的情况下,算法将产生错误的切分"的结论.其逆否命题就是,当算法产生正确的切分时(*Precision*(*PostComment*)),正文定位也将是正确的(*Precision*(*MainText+PostComment*)).因此,两个精确率指标几乎是相等的.

Table 3 Algorithm precision on each blog site (unigram) (%)

表 3 各个站点的算法效果(一元模型) (%)

Site	*MainText*	*PostComment*	*PostComment*|*MainText*	*MainText*+*PostComment*
www.livejournal.com	84.54	76.16	90.06	76.13
blogspot.com	92.03	90.64	98.47	90.62
nospeedbumps.com	99.48	93.98	94.47	93.98
www.plogress.com	100.00	100.00	100.00	100.00
ipunkrock.com	97.17	96.86	99.68	96.86
www.blogespierre.com	74.42	55.81	75.00	55.81
weblogs.java.net	96.36	69.10	71.70	69.10
redjar.org	100.00	96.00	96.00	96.00
www.sff.net	98.00	98.00	100.00	98.00

2.3.2 正文定位和正文切分效果测试

为了测试噪音对正文定位算法的影响,我们考虑在不同正文与网页文本比率情况下的定位精度.当正文与网页文本比率小时,说明网页中存在大量的噪音.我们比较了 4 种定位算法,其中两种为我们的正文定位算法(一个带视觉宽度信息,一个不带视觉宽度信息),另外两种是链接/文本比率方法和文本量的方法.测试结果如图 3 所示.图中横坐标表示正文与网页文本的比率,纵坐标是正文定位的精确率.*i*/10 表示正文与网页文本的比率在[(*i*−1)/10,*i*/10)的区间内.

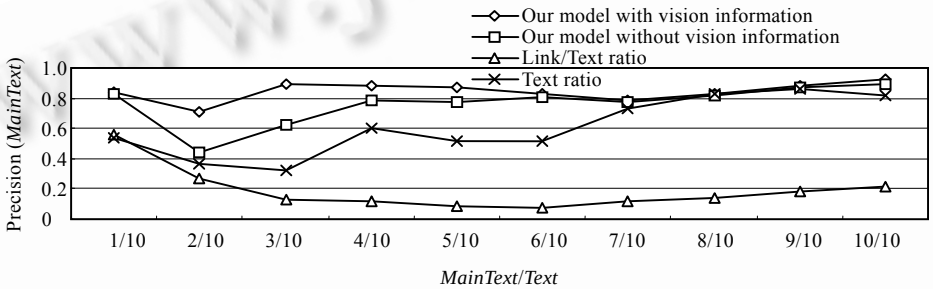


Fig.3 Precision of locating main text on different MainText/Text ratio

图 3 不同正文与网页文本比率情况下的正文定位精度

从图 3 中我们可以看出,网页中的噪音(文本和链接)对我们的正文定位算法(带视觉信息)影响不大,我们的算法基本上保持一个平稳的状态,最低点出现在正文与网页文本比率在[1/10,2/10]区间的时候,但仍然保持 0.7 以上的精确率.当比率逐步升高时,噪音减小,我们的定位算法精确率也逐步提升.链接/文本比率无法精确定位正文,这从图中的曲线可以看出.文本量模型在噪音较少的情况下定位效果接近我们的定位算法.

由于不同的博客网页具有不同的文章和评论情况,有的评论较少或没有评论,有的评论较多,因此,我们需要考虑不同文章和评论的文本比率情况下的切分精度.我们测试了一元模型、二元模型前序遍历和二元模型层次遍历,结果如图 4 所示.图中横坐标为文章与评论的文本比率,纵坐标表示切分的精确率.

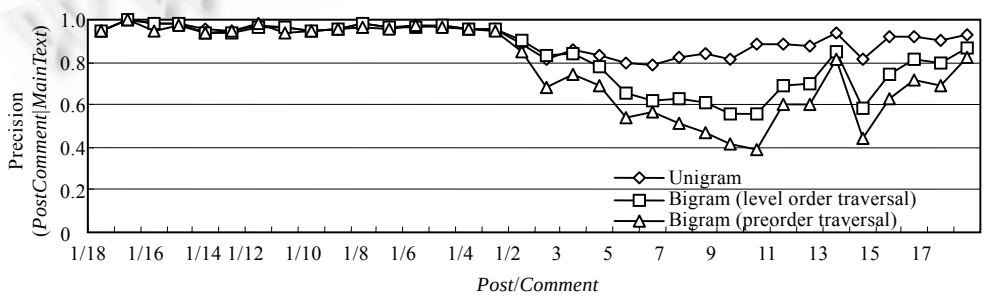


Fig.4 Effect of *Post*/*Comment* ratio on main text separating

图 4 文章与评论比率对正文切分的影响

从图 4 中我们可以看出,当评论文本数大于文章文本数时,无论是一元模型还是二元模型都达到了很高的精确率;但是当文章文本数大于评论文本数时,模型精确率都有不同程度的下降,其中二元模型变化比一元模型要大,而二元模型前序遍历对此更为敏感.我们的切分算法本质上是检测格式信息的冗余,使得相同格式的网页格式标签划到同一类别中.评论的格式相对于文章格式更有规律,当评论文本数多的时候,正文中出现多条评论,每条评论的格式基本上是简单的重复,因此,当评论文本数多时,精确率都非常高.当文章文本数多时,评论格式上的冗余减少,使得切分算法在检测切分边界的精度上有所下降,同时,由于正文中的格式信息稀疏,二元模型比一元模型敏感,这可以从一元模型和二元模型的变化曲线上看出来.

2.3.3 语言敏感性测试

由于 TREC 数据中还有非英文博客,因此将英文网页和非英文网页区分开,并且使用正文定位和正文切分(一元模型)的方法来测试算法的语言敏感性.我们将上面的 25 910 个网页分成两个数据集:5 873 个非英文博客网页和 20 037 个英文博客网页.对这两个数据集,我们重复上述一元模型的实验,得到的结果见表 4.

Table 4 Precision of language sensitive test (unigram)

表 4 语言敏感性测试的精确率(一元模型)

	Non-English pages (%)	English pages (%)
MainText	87.93	87.16
PostComment	83.19	80.47
PostComment|*MainText*	94.56	92.31
MainText+*PostComment*	83.14	80.46

表 4 的数据结果表明,英文博客网页和非英文博客网页的精确率相差不大,我们的模型具有与语言无关的特性.

3 结 论

从博客网页中精确抽取文章和评论对于提高博客检索的质量具有重要的意义.我们通过对博客网页中的格式信息、视觉位置信息和有效文本信息进行分析,提出了基于网页格式信息量的博客文章和评论抽取模型.该模型计算网页中每个格式信息单元所提供的信息量来定位博客正文以及对博客文章和评论进行切分.该模型与博客所使用的语言无关,具有很好的通用性.在实验中,我们使用 TREC 2006 博客任务中的语料数据作为实验数据,综合比较了定位算法和切分算法的一元模型和二元模型.实验结果表明,我们的正文定位达到 87.33%的精确率.由于二元模型具有较强的敏感性,而一元模型相对来说更适合于博客正文的切分,一元模型的切分精确率达到了 92.82%.同时,我们通过实验进一步说明了算法的语言无关性.在下一步的研究工作中,我们将研究正文定位和正文切分之间的协作关系,使得正文定位和正文切分之间可以相互进行修正,提高总体的算法精确率.

References:

[1] Gupta S, Kaiser GE, Grimm P, Chiang MF, Starren J. Automating content extraction of html documents. World Wide Web, 2005, 8(2):179−224.

[2] Irmak U, Suel T. Interactive wrapper generation with minimal user effort. In: Proc. of the 15th Int'l Conf. on World Wide Web. New York: ACM, 2006. 553−563.

[3] Liu B, Grossman R, Zhai YH. Mining data records in Web pages. In: Proc. of the 9th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM, 2003. 601−606.

[4] Crescenzi V, Mecca G, Merialdo P. RoadRunner: Towards automatic data extraction from large Web sites. In: Apers PMG, Atzeni P, Ceri S, Paraboschi S, Ramamohanarao K, Snodgrass RT, eds. Proc. of the 27th Int'l Conf. on Very Large Data Bases. San Francisco: Morgan Kaufmann Publishers, 2001. 109−118.

[5] Lu YY, Meng WY, Zhang WJ, Liu KL, Yu C. Automatic extraction of publication time from news search results. In: Proc. of the 22nd Int'l Conf. on Data Engineering Workshops. Washington: IEEE Computer Society, 2006. 50−59.

[6] Zhao HK, Meng WY, Yu C. Automatic extraction of dynamic record sections from search engine result pages. In: Dayal U, Whang KY, Lomet D, Alonso G, Lohman G, Kersten M, Cha SK, Kim YK, eds. Proc. of the 32nd Int'l Conf. on Very Large Data Bases. 2006. 989−1000.

[7] Ling YY, Meng XF, Meng WY. Automated extraction of hit numbers from search result pages. In: Yu JX, Kitsuregawa M, Leong HV, eds. Proc. of the 7th Int'l Conf. WAIM 2006. LNCS 4016, Heidelberg: Springer-Verlag, 2006. 73−84.

[8] Zhai YH, Liu B. Structured data extraction from the Web based on partial tree alignment. IEEE Trans. on Knowledge and Data Engineering, 2006,18(12):1614−1628.

[9] Chang CH, Lui SC. IEPAD: Information extraction based on pattern discovery. In: Proc. of the 10th Int'l Conf. on World Wide Web. New York: ACM, 2001. 681−688.

[10] Zheng JH, Wang XY, Li F. Research on automatic generation of extraction patterns. Journal of Chinese Information Processing, 2004,18(1):48−54 (in Chinese with English abstract).

[11] Reynar JC. Topic segmentation: Algorithms and applications [Ph.D. Thesis]. University of Pennsylvania, 1998.

[12] Qi Y, Candan KS. CUTS: CUrvature-Based development pattern analysis and segmentation for blogs and other text streams. In: Proc. of the 17th Conf. on Hypertext and Hypermedia. New York: ACM, 2006. 1−10.

[13] Reis DC, Golgher PB, Silva AS, Laender AF. Automatic Web news extraction using tree edit distance. In: Proc. of the 13th Int'l Conf. on World Wide Web. New York: ACM, 2004. 502−511.

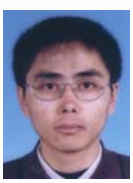
[14] Cover TM, Thomas JA. Elements of Information Theory. 2nd ed., New York: Wiley-Interscience, 2006.

[15] Shannon CE. A mathematical theory of communication. The Bell System Technical Journal, 1948,27:379−423, 623−656.

附中文参考文献:

[10] 郑家恒,王兴义,李飞.信息抽取模式自动生成方法的研究.中文信息学报,2004,18(1):48−54.

曹冬林(1977－),男,江西都昌人,博士,助教,主要研究领域为Web挖掘,Web信息检索,自然语言处理.



许洪波(1975－),男,博士,副研究员,主要研究领域为文本挖掘和信息分析,包括文本分类与聚类,信息过滤,问答系统,话题识别与跟踪.

廖祥文(1980－),男,博士,主要研究领域为Web挖掘,Web信息检索.

白硕(1956－),男,博士,研究员,博士生导师,CCF高级会员,主要研究领域为大规模内容处理.