文章编号: 1003- 0077(2004) 05- 0017- 06

基于统计的网页正文信息抽取方法的研究①

孙承杰, 关 毅

(哈尔滨工业大学 计算机学院 黑龙江 哈尔滨 150001)

摘要: 为了把自然语言处理技术有效的运用到网页文档中, 本文提出了一种依靠统计信息, 从中文新闻类网页中抽取正文内容的方法。该方法先根据网页中的 HTML 标记把网页表示成一棵树, 然后利用树中每个结点包含的中文字符数从中选择包含正文信息的结点。该方法克服了传统的网页内容抽取方法需要针对不同的数据源构造不同的包装器的缺点, 具有简单、准确的特点, 试验表明该方法的抽取准确率可以达到 95% 以上。采用该方法实现的网页文本抽取工具目前为一个面向旅游领域的问答系统提供语料支持, 很好的满足了问答系统的需求。

关键词: 计算机应用; 中文信息处理; 网页数据抽取; 包装器

中图分类号: TP391 文献标识码: A

A Statistical Approach for Content Extraction from Web Page

SUN Cheng-jie, GUAN Yi

(Dept. of Computer Science and Technology, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China)

Abstract: This paper proposes a statistical approach for extracting text content from Chinese news web pages in order to effectively apply natural language processing technologies to web page documents. The method uses a tree to represent a web page according to HTML tags, and then chooses the node which contains text content by using the number of the Chinese characters in each node of the tree. In comparison with traditional methods, the method needn't construct different wrappers for different data sources. It is simple, accurate and easy to be implemented. Experimental results show that the extraction precision is higher than 95%. The method has been adopted to provide web text data for a question answering system of traveling domain.

Key words: computer application; Chinese information processing; web data extraction; wrapper

1 引言

互联网的飞速发展给自然语言处理的研究带来新的机遇和挑战。把自然语言处理技术应用到网页处理中, 对网络中的信息进行深层次的加工处理, 有效地从浩瀚的信息海洋中挖掘可以为人所用的各种知识, 提取人们所需的信息, 已成为很多研究人员的研究目标。根据用途不同, 一个网页中的内容可以分为两类, 一类是提供给浏览器用的标记信息, 另一类是提供给用户阅读的信息, 自然语言处理技术针对的是后者。因此, 要对网页中的内容运用自然语言处理技术, 必须先把网页中的标记信息去掉。从内容上分, 一个网页一般是由导航信息、网页正文、广告信息、版权信息、相关链接等部分组成的。自然语言处理技术适用的部分是网页正文。所

① 收稿日期: 2004- 04- 22
基金项目: 国家 863 计划资助项目(2002AA117010- 09)
作者简介: 孙承杰(1980—) , 男, 硕士生, 主要研究方向自然语言处理、信息抽取.

以,如何提取网页中的正文,并把它转换成纯文本文件的技术是连接自然语言处理技术和网页信息的桥梁(纯文本文件是指不包含网页标记信息的文本文件)。它们的关系如图 1 所示。

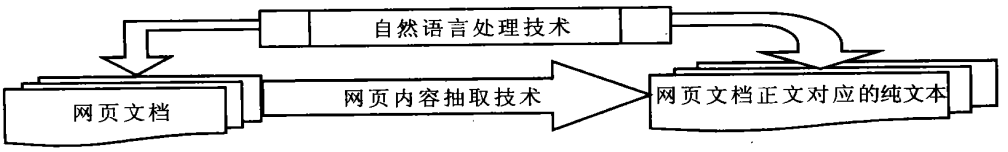

图 1　网页内容抽取技术的作用

这种技术集成到文摘系统中,可以对网页进行文摘;集成到文本分类系统中,可以对网页进行自动分类。这样,对网页的处理就像对纯文本的处理一样,扩大了原有技术的适用范围。所以,研究网页内容抽取技术对于将自然语言技术的适用范围扩展到网页处理有重要的意义。

目前国内外研究较多的是从网上抽取一些具有格式的信息,如会议论文信息[1],商品信息[2,3],图书信息[4]。这些研究的主要目的是把网页中的无结构化数据或半结构化数据变成结构化数据,而不是为了提取网页中的正文内容。文献[5]中提出了一种利用 HTML 标记来对网页中的内容进行分块的思想,但它的主要目的是排除网页中影响搜索引擎检索精度的干扰信息,也不是为了提取网页中的正文信息。因为正文信息也可以看作是网页中包含的一种数据,所以下文中使用了网页数据抽取这个术语来表述更一般的情况。

2　网页数据抽取的方法介绍

虽然 XML 语言近几年发展很迅速,但是目前互联网上的网页绝大部分是用 HTML 语言编写的。HTML 语言提供的标记主要是用来控制网页内容的显示格式的。如 table, tr, td, th 是用来绘制表格的。li, ol, ul 是用来表示列表的。这些标记的使用没有什么规律,网页设计人员可以随便设计。但是不同种类的数据一般是放在不同的显示单元中的。我们要提取的新闻类网页中的正文信息绝大部分就存在于某个 table 中。

传统的网页数据抽取方法,是使用包装器(wrapper)来抽取网页中感兴趣的数据。包装器是一个程序,它根据一定的信息模式识别知识从特定的信息源中抽取相关内容,并以特定形式加以表示。由于包装器所需的信息模式识别知识的获取是一个费时费力且需要较高智能的工作,因此目前网页数据抽取研究工作的重点之一就是探索如何能够较为容易的获得构造一个包装器所需规则的有效方法[6]。

文献[7]中介绍的TSIMMIS 工具中的包装器需要人工来书写抽取规则。规则被放在专门的文件中,规则的形式是[variables, source, pattern]。其中variables 保存抽取结果,source 保存输入,pattern 保存了数据在 source 中的模式信息。variable 可以用作后面的规则的 source。文件中最后一个规则执行结束后,variable 中保存了最后的抽取结果。这种需要人工书写规则的方法,费时、费力,而且容易出错,不易维护。

文献[8]介绍的 XWRAP 系统中的包装器采用了半自动化的方法来获取规则。它提供了友好的人机交互界面,用户可以根据系统的引导来完成规则的编写。最终,系统生成一个针对特定数据源的用 java 语言编写的包装器。在进行抽取之前,XWRAP 系统会对网页进行检查,修正其中的不符合规范的语法错误和标记,并把网页解析成一棵树,在我们的方法中也进行了类似的预处理。

文献[9]介绍的 RoadRunner 工具是一个完全自动化的包装器自动生成工具,它甚至不需

要用户提供待抽取的数据的样本和目标模式。它通过比较来自同一数据源的两个(或多个) 样本网页的结构来为包含在网页中的数据生成一定的模式。该方法假设目标网页都是从某个数据源自动生成的, 那么它就可以利用网页的标记结构重新得到网页中包含的数据的模式, 所以其适用范围有一定的局限性。

上面介绍的几种方法生成的包装器都是按一定的规则或模式来抽取数据。由于网页结构的复杂性及不规范性, 一个包装器的实现一般只能针对一个信息源。从文献[10] 中, 我们可以看到, 目前的网页数据抽取工具, 都需要针对特定的数据源来编写对应的包装器或抽取规则。所以, 如果信息是来自很多信息源, 就需要很多包装器, 这样包装器的生成及维护就成了一种复杂的工作。对于网络上大量存在的新闻类网页的正文信息抽取这样的任务来说, 使用针对特定信息源的包装器的方法来完成显然是不合算的, 我们需要的是一个普遍适用的包装器。

3 基于统计的中文新闻类网页正文信息抽取方法

本文提出了一种基于统计的方法来实现中文新闻类网页中正文信息的抽取。这种方法利用了中文网页的特性, 实现简单, 通用性好, 可以克服包装器方法需要针对特定数据源的缺点。本方法的适用范围: 适合于一个网页中所有的正文信息都放在一个 table 中的情况。新闻类网页中的正文都是放在 table 中的, 这是我们对取自不同网站的 5000 篇新闻类网页的分析结果。所以本方法非常适用于中文新闻类网页正文信息的抽取。

我们方法可以分为以下两步:

1. 根据网页中的 HTML 标记, 把网页表示成一棵树。

因为我们已经知道要抽取的正文是放在 table 中的, 文献[11] 告诉我们, 这类问题应该采用基于树结构的解决方案。所以, 我们需要先把网页表示成一棵树。由于网页结构的复杂性。在把网页表示成一棵树之前, 必须先对网页进行预处理, 使其变为规范的网页。规范网页的要求如下:

1)“〈”和“〉”只能用来包含网页标记(tag) , 当在其它地方出现这两个符号时应该用 & lt; 和 & gt; 代替。

2) 所有的标记必须匹配。即每个开始标记都对应一个结束标记。

3) 所有标记的属性值都必须放在引号中。如〈a href= “www. w3c. org”〉。

4) 所有的标记必须是正确嵌套的。如〈a〉…〈b〉…〈/ a〉…〈/ b〉是不正确的嵌套。正确的嵌套形式应该是〈a〉…〈b〉…〈/ b〉…〈/ a〉[3]。

经过规范的网页可以很容易的根据其中 HTML 标记把它表示成一棵树, 树中的每个结点包含了一对标记间的所有字符, 结点的名字为对应的标记的名字。

2. 从第一步得到的树中选择包含正文信息的节点。

我们已经知道正文信息存在于 table 中, 所以我们只关心 table 结点。选择的方法如下:

1) 找到 HTML 文档树中包含的所有的 table 结点。

2) 对 1) 中找到的 table 结点, 我们按照下面的方法进行进一步的处理。对每一个 table 结点, 去掉其中的 HTML 标记, 得到不含任何 HTML 标记的字符串。如果得到的字符串中所含有的中文字符的数量大于我们预先设定的阈值 P, 我们就把该 table 结点作为候选。这样, 我们可以去掉一些不包含数据, 只为了调整显示格式的 table 结点。

3) 对每个 table 结点, 按照由它得到的字符串的长度进行降序排序。这里需要说明的一点是: 因为大多数网站中, 每一个新闻网页都含有很多的相关链接的信息, 为了排除这种信息对

正文提取的干扰,我们在去掉 HTML 标记时,把超链中包含的字符也都去掉了。还有就是我们在统计字符串的长度时,我们统计的是字符串中文字符的个数。

去掉HTML 标记时,我们用的是顺序遍历的方法。因为〈script〉〈/script〉,〈form〉〈/form〉,〈style〉〈/style〉这些标记中不会含有我们的所需要的内容,还会对后面的处理有负面影响,所以我们去掉了这些标记之间所包含的全部内容。

4)经过上面的处理,排在前面的 table 结点基本上就包含了我们需要的正文信息。但由于 table 是可以嵌套的,所以有可能排在最前面的 table 结点除了正文,还包含了其他一些信息,比如版权信息。所以,为了更加准确的提取正文信息,我们还进行了下一步的处理。除了排在最前面的 table 结点,对队列中剩下的每一个 table 结点,依次考察它是不是它前面的 table 的后代结点,如果找到一个这样的结点,它不是它前面任意一个结点的后代结点或者它是前面某结点的后代结点并且它所含的信息量在该结点所含的信息量中占有的比率大于 T。这个结点就是我们的最终选择。

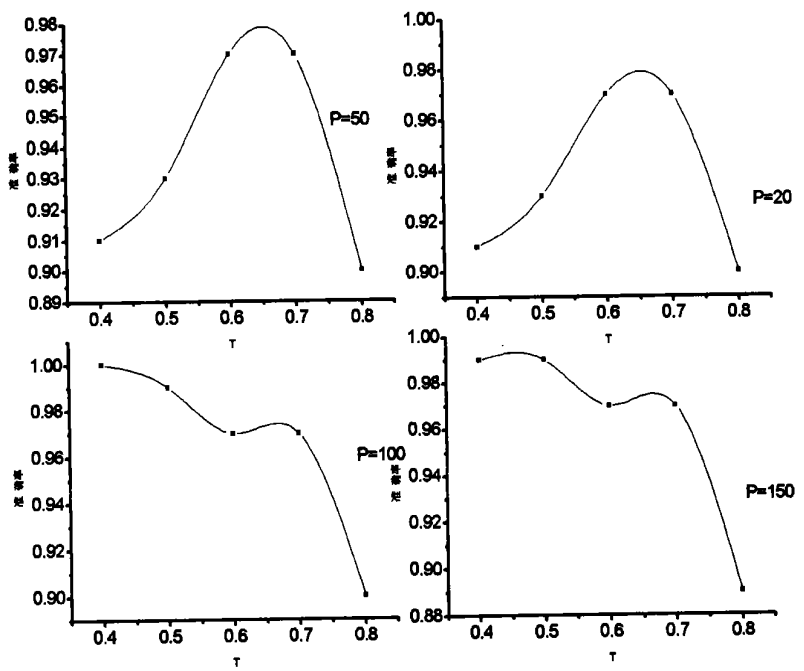

图 2　参数选择的实验结果

在我们的方法中用到了两个阈值 P 和 T。为了选取适合的值,我们进行了以下的实验。我们从 10 个不同的网站选取了 100 篇网页,然后对 P 和 T 分别取不同的数值,计算每一组 P 和 T 对应的准确率。实验结果如图 2 所示。

从试验结果我们可以看出,不管 P 取值如何,当 T 的值超过 0.7 时,方法的准确率都会迅速下降。这是因为包含正文的 table 大多是嵌套在一个大的table 中的。而大的 table 中通常会含有版权信息、广告信息等其他信息,所以包含正文的 table 在其中所占的信息量的比率不会太大。因此,如果 T 的取值过大,包含正文的 table 就不会被取出来。我们还可以看到 T 在 0.6– 0.7之间时,方法的准确率受 P 的取值的影响很小,而且准确率很高。

根据上面的实验结果,我们取 P= 50, T = 0.65。

下面用一个实际的例子来说明我们的方法。该例子所用的网页为: http://sports.sohu.com/2004/02/13/89/news219068957.shtml。

从该网页对应的HTML树中我们共得到68个table结点。在过滤掉含有中文字符较少的table后,我们得到9个table。每个table中的中文字符的个数及其所占的百分比如表1。

经过最后的处理后,我们得到编号2的table所含的内容应该是正文信息。因为编号2的table是嵌套在编号1的table中,且791/928> T= 0.65,符合我们的要求,所以我们选择编号2的table。这个结果经过考察是正确的。

从表中可看出,各个table中所含中文字符的比例都很接近,我们在筛选table时,没有采用相对数量,而采用绝对数量的原因。从表中还可看出,全部中文字符的个数和去掉HTML标记后表中中文字符的个数相比有很大的区别,因此我们在进行选择时去掉HTML的标记。

表1 对一个实际网页的分析结果

表的序号	1	2	3	4	5	6	7	8	9
表中去掉HTML标记后中文字符的个数	928	792	138	132	116	70	60	60	60
表中全部中文字符的个数	2540	912	1438	150	116	402	64	64	160
表中全部中文字符所占的比例	0.377	0.393	0.378	0.372	0.433	0.371	0.381	0.381	0.367

4 试验结果及结果分析

为了考察方法的实际效果,我们进行了下面的实验。

试验数据: 从表2所列的网站中随机选择了一些网页,为了满足方法假设的前提,我们尽量挑选含有正文信息的网页,但不排除例外。试验中,方法表现出了较好的鲁棒性。

试验结果如表2所示。在这个试验中, $P=50$, $T=0.65$。

表2 试验结果

数据来源	网页总数(个)	正确的结果(个)	错误的结果(个)	准确率(%)
www.sina.com.cn	60	56	4	93
www.sohu.com	40	38	2	95
www.beijingok.net	50	50	0	100
www.cctv.com	50	50	0	100
www.ctn.com.cn	50	46	4	92
www.aroundsuzhou.com	50	48	2	96
www.ctnews.com.cn	100	99	1	99
travel.zaobao.com	50	45	5	90
www.chinadaily.com.cn	100	96	4	96
news.china.com	100	90	10	90
总计	650	618	32	95

通过对出错的网页的考察,我们发现,大部分出错的网页有一个共同特征,它们包含的正文信息都很短,或者只有简单的一句话,或者是包含少量文字说明的图片新闻。这样的网页中,因为其所含的信息量较少,所以包含正文的table常常会被过滤掉,以致得不到正确的结果。还有的是因为网页中的正文信息没有放到单独的一个table中,而是作为一个大的table中

的一个单元出现,这样,虽然我们选出了包含正文的 table,但是这其中包含的非正文信息也会被我们提取出来,这种错误在 travel. zaobao. com 和 www. aroundsuzhou. com 上的网页比较常见。还有的网页根本就没有正文信息,如一些大网站的索引,我们把这样的网页也算作错误的网页,这也是 news. china. com 上的网页出错的主要原因,但这种错误不是方法本身的问题。

从上面的实验结果可以看出,该方法在具有通用性的同时,保持了较高的准确性。如果,网页是比较规范的新闻类网页,其准确率可以达到 100% 。这说明这种方法具有实用性。在实际工作中,我们对来自 130 个网站的网页进行了抽取,抽样统计的准确率在 90% 以上。

5 未来的工作

该方法目前主要是为面向旅游领域的问答系统提供语料加工服务的,因此试验中两个阈值的设定也是针对这个系统的。两个阈值对结果的影响还应该做进一步的探索,已得到适合不同应用场合的数据。另外,还可以把这种方法与现有的中文信息处理技术相结合,把中文信息处理技术的应用扩展到网页处理。如,在网络爬虫中结合我们的抽取方法与文本分类方法,就可以实现网页下载过程中的自动分类。本文采用的方法适用于多数网站,但也有部分网站不适合;另外也没有考虑网页内容自身带有表格信息的情况,这些均是本方法的局限性,以后的工作中还需要在方法的通用性方面继续加强研究。

参 考 文 献:

[1] 张绍华,徐林昊,等. 基于样本实例的 WEB 信息抽取[J]. 河北大学学报(自然科学版),2001,(12):431 - 437.

[2] 高军,王腾蛟,等. 基于 Ontology 的 Web 内容二阶段半自动提取方法[J]. 计算机学报,2004,27(3):310 - 317.

[3] David Buttler, Ling liu, et al. A Fully Automated Object Extraction System for the World Wide Web[A]. Proceedings of the 2001 International Conference on Distributed Computing Systems[C]. 2001: 361 - 370.

[4] Sergey Brin, Lawrence Page. The Anatomy of a Large-scale Hypertextual Web Search Engine[A]. Proceedings of the Seventh International Conference on World Wide Web[C]. 1998: 107 - 117.

[5] 宋睿华,马少平,等. 一种提高中文搜索引擎检索质量的 HTML 解析方法[J]. 中文信息学报[J],2003,17(4): 19 - 26.

[6] Line Eikvil. Information Extraction from World Wide Web A Survey. 1999.

[7] Hammer J., McHugh J. Semi-structured Data: The TSIMMIS Experience[A]. In: proceeding of the First East-European Symposium on Advance in Databases and Information Systems[C]. 1997: 1 - 8.

[8] Liu, L., Pu, C. et al. XWRAP: An XML-enable Wrapper Construction System for the Web Information Source[C]. In: proceedings of the 16th IEEE International Conference on Data Engineering, 2000: 611 - 620.

[9] Valter Crescenzi, Giansalvatore Mecca. RoadRunner: Towards Automatic Data Extraction from Large Web Site[A]. In: proceeding of the 26th International Conference on very Large Database Systems[C], 2001: 109 - 118.

[10] Alberto H. F. Laender, Berthier A. Ribeiro-Neto. A Brief Survey of Web Data Extraction Tools[J]. SIGMOD Record. 2002, 31(2): 84 - 93.

[11] Daisuke Ikeda, Yasuhiro Yamada. Expressive Power of Tree and String Based Wrapper[A]. In: on-line proceedings of IJCAI'03 workshop on Information Integration on the Web[C]. 2003.