计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
47(9): 1567-1573, 2010

基于扩展领域模型的有名属性抽取

王 宇[1,2] 谭松波[1] 廖祥文[1,2] 曾依灵[1,2]
[1](中国科学院计算技术研究所 北京 100190)
[2](中国科学院研究生院 北京 100190)
(wangyu2005@software.ict.ac.cn)

Extended Domain Model Based Named Attribute Extraction

Wang Yu[1,2], Tan Songbo[1], Liao Xiangwen[1,2], and Zeng Yiling[1,2]
[1](*Institute of Computing Technology, Chinese Academy of Sciences, Bejing* 100190)
[2](*Graduate School of Chinese Academy of Sciences, Bejing* 100190)

Abstract Web information extraction is an important task of Web mining. Various applications could benefit from the advancement in this area. These applications include semantic Web, vertical search, sentiment analysis, etc. Current techniques require lots of human interaction which preclude the universal application of Web information extraction. To automate the extraction process, recent research works identify specific features of special domains and extract information by machine learning techniques. However, because of the dependence on specific features, it is very difficult to extend such methods to other domains. In this paper, the Web information extraction problem is analyzed and a subtask is proposed. This new subtask is called named attribute extraction task. Statistics results from multiple datasets prove that named attribute extraction task covers more than 60% attributes in these domains, which show the importance of this subtask. Named attributes are attributes of objects which are encoded in the name-value pair form. That is, the names and values of attributes are settled nearby in the Web pages. Therefore, once the names of attributes are located, the values can be extracted automatically. In this paper, an extended domain model is proposed to summarize attribute names of a domain. And an information extraction method based on this model is developed. Experiments show that the method can extract named attributes at the precision 80%, and at the recall higher than 90%.

Key words information extraction; attribute extraction; named attribute; extended domain model; visual Web page analysis

摘 要 网页信息抽取是互联网挖掘的重要课题.为了自动化抽取过程,最新的研究利用特定领域的特征,通过机器学习方法对信息抽取过程进行统一建模.但是,对领域特征的依赖使得这类方法难以推广到其他领域中去.因此,对信息抽取问题进行了分析,从中分离出一个可以完全自动化的信息抽取子任务,即有名属性抽取任务.在多个领域的数据集上进行的统计表明,这个子任务覆盖了60%以上的待抽取属性,因此它在整个信息抽取中占有重要地位.并给出了一种基于扩展领域模型的有名属性抽取方法,实验结果表明,这种方法的准确率接近或大于80%,召回率大于90%.

关键词 信息抽取;属性抽取;有名属性;扩展领域模型;网页视觉分析

中图法分类号 TP391.3

收稿日期:2009-06-12;**修回日期**:2010-01-15

基金项目:国家"九七三"重点基础研究发展计划基金项目(2004CB318109,2007CB311100)

0 引 言

网页信息抽取研究一直在向着更少人工干预的方向发展,这是因为互联网上存在着异常丰富的数据源,即便是某个特定的应用领域,往往也存在上千个网站.手工为每个网站撰写包装器是不可能的.

传统的信息抽取方法包括包装器(wrapper)语言[1-2]、包装器归纳[3]、半自动信息抽取[4-10],这些方法的自动化程度逐渐提高.不过,即使是半自动信息抽取方法,它仍然需要用户人工收集训练网页以及指定字段语义.所有这些方法都在不同程度上依赖用户操作.

为了进一步提高信息抽取的自动化程度,研究人员提出了一些解决方案[11-13],它们都是针对特定领域的部分字段,通过有监督机器学习的方法进行抽取.训练好抽取模型后,这些方法可以完全自动地抽取同领域的任何网页.但是,这类方法存在两个根本的缺陷:

1) 机器学习的方法依赖于特定领域的特征集合,针对每个新的领域,必须重新设计特征集合;

2) 即便确定了特征集合,在新的领域中,有监督机器学习方法仍然需要大量的手工标注.

这两个缺陷使得现有的全自动信息抽取方法难以推广到新的领域.考虑到现有方法的缺陷,我们尝试从另一个角度解决全自动信息抽取问题.在网页中,许多属性值附近都存在一些关键词用于说明属性值的含义,这类属性称为有名属性.有名属性的这个特点使得我们可以通过关键词来定位属性值.因此,本文首次提出了有名属性抽取任务,并提出了一种基于扩展领域模型的有名属性抽取方法.许多领域中都存在有名属性,因此这种方法可以方便地推广到各种领域,且无需进行太多的人工标注.这解决了推广困难的问题.同时,有名属性抽取任务覆盖了60%以上的待抽取属性,是整个信息抽取问题的一个有价值的子问题.实验结果表明,我们的方法能够以接近或高于80%的准确率和大于90%的召回率抽取有名属性.

1 扩展领域模型

信息抽取的目标是从许多数据源中获取数据,并将这些数据整合成统一的数据模式.因此,在针对一个领域进行信息抽取之前,必须首先规定一个公共的数据模式.

定义 1. 在一个领域 D 中,所有待抽取的属性构成了一个数据模式,称为领域模型.领域模型可以使用元组来表示,$DM(D)=\langle attr_1, attr_2, \cdots, attr_i, \cdots, attr_m\rangle$,其中每个 $attr_i$ 代表一个属性,m 代表数据模式中属性的个数.

但是,属于同一个领域的各个网站具有不同的数据模式,这首先体现在各个网站使用不同的关键词来描述同一个的属性.例如,描述书籍市场价格的关键词就包括"市场价"、"标价"和"零售价"等.因此,为了信息抽取和信息整合的需要,必须建立这些数据模式与领域模型之间的映射关系,这种映射关系具体表现为每个属性对应于一系列不同的关键词.

定义 2. 对于领域模型中的每个属性 $arrt_i$,不同的网站使用不同的关键词描述这个属性,所有这些关键词称为这个属性的关键词列表 $K(attr_i)$.

定义 3. 领域模型与各种具体数据模式到领域模型的映射关系,称为扩展领域模型(EDM).

扩展领域模型的例子如图 1 所示.每个属性都有一个关键词列表,图中仅画出了"市场价"属性的部分关键词列表.

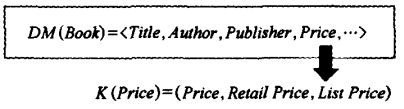

Fig. 1 The extended domain model of book domain.

图 1 书籍领域的扩展领域模型

2 属性名定位

有名属性抽取的第 1 步是定位属性值附近的关键词.具体来说,我们要定位关键词所在的文本节点,称为属性名节点.属性名定位可以分为两个步骤:首先在 DOM 树中找到所有包含关键词的文本节点;接下来识别其中真正的属性名节点.

2.1 寻找关键词

在文本节点中寻找任意已知关键词,从本质上来看是一个多串匹配问题.但是,我们的任务与传统算法解决的问题略有不同.我们要在词语序列中寻找不被任何其他关键词串完全覆盖的关键词串.例如,设关键词 A 是关键词 B 的一个子串,待匹配串也是 B,那么,我们只需要找到 B,而传统的算法会同时给出 A,B 两个匹配.我们以 Aho-Corasick 算

法[14]为基础提出了关键词匹配算法.算法的时间复杂度为$O(n+m)$,n是被搜索字符串的长度,m是在其中找到的关键词数目.具体算法见2.3节.

2.2 识别属性名节点

包含关键词的文本节点并不都是属性名节点.最常见的例外情况是关键词出现在一个句子中.例如,在"产品的市场价非常……"中,"市场价"只是句子的一个成分,它所在的文本节点不是属性名节点.但是,句子和属性名节点是有区别的,在句子中关键词与其他词语随意地混合在一起,而在属性名节点中关键词附近常常存在一些语义或语法上的线索,如图2所示,关键词用下划线标出.关键词和其他文本通常被文本节点的边界或是标点符号区分开.有时关键词与数据值混合在一起,如图2(c)所示.这些数据值通常是由标点符号、数值和名词构成的名词短语.

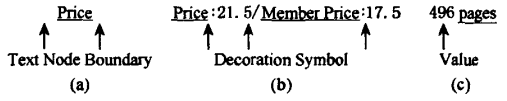

Fig. 2 Common attribute name nodes.

图2 常见属性名节点

因此,可以定义4个判定谓词来描述这种现象.即$LeftSep(k,n)$,$RightSep(k,n)$,$LeftValue(k,n)$和$RightValue(k,n)$,其中k是关键词,n是关键词所在的文本节点.$LeftSep/RightSep$为真当且仅当关键词k的左边/右边是文本节点的边界或是标点符号.$LeftValue/RightValue$为真当且仅当关键词k的左边/右边是一个合法的数据值.以这4个判定谓词为基础,识别真实属性名节点的规则是:

1) $LeftSep(k,n)$ and $RightSep(k,n)$;

2) $LeftSep(k,n)$ and $RightValue(k,n)$;

3) $LeftValue(k,n)$ and $RightSep(k,n)$.

满足以上任意一条规则,节点n即为属性名节点.

2.3 算法描述

属性名定位的完整算法如算法1所示.在抽取之前,所有的关键词首先被构建成后缀树.待抽取网页也被转换成DOM树.这两者将作为算法的输入.

算法1. 属性名定位算法.

输入:网页DOM树dom、关键词后缀树$suffixTree$;

输出:属性名节点列表$nameList=\{(n_0,k_0),(n_1,k_1),\cdots\}$,$n_i$为属性名节点,$k_i$为关键词.

1) 初始化n为dom中第1个文本节点.

2) 使用Aho-Corasick算法进行字符串匹配,但是,当且仅当发生后缀链接跳转或整个字符串匹配结束时才返回找到的匹配k,这保证了k不被任何其他匹配覆盖.

3) 对于每个关键词匹配k和它所在的文本节点n,如果它符合2.2节的任意一条规则,则将(n,k)加入$nameList$中.

4) 令n为dom中的下一个文本节点.转到第2)步.

3 属性值定位

3.1 基本想法

定位属性名节点是为了进一步找到属性值节点.根据属性名和属性值节点相对位置的不同,寻找属性值节点的方法也有所不同.如果属性名和属性值混合在同一个文本节点中,如图2(c)所示,则属性名节点同时又是属性值节点,无需进行进一步的属性值抽取.这类属性称为名值混合属性.

除了名值混合属性,其他属性的属性名和属性值分布在DOM树的不同节点中.代表属性名和代表属性值的节点通常出现在视觉上相近的位置,因此,在找到属性名之后,可以在它的附近寻找可能的属性值.文献[15]曾给出节点之间关系密切程度的度量,但是这种度量不能直接用于信息抽取.

为了精确地度量节点之间的关系,我们对属性进行了分类,对不同的属性采用不同的度量方法.对于某些属性,它的属性名和属性值位于HTML代码中相近的位置,网页渲染引擎自然地将它们排版在视觉相近的位置,这类属性称为结构对齐类属性.另一类属性用特殊HTML标签强制对齐了属性名和属性值,这类属性称为视觉对齐类属性.针对这两类属性,我们采用不同的方法来寻找属性值.最终,属性值定位的算法同时处理名值混合属性、结构对齐类属性和视觉对齐类属性,其基本框架如算法2所示.

算法2. 属性值定位算法.

输入:网页DOM树dom;属性名节点列表$nameList=\{(n_0,k_0),(n_1,k_1),\cdots\}$,$n_i$为属性名节点,$k_i$为关键词;

输出:属性值节点列表$valueList$.

1) 从$nameList$中获取一个元素(n,k).

2) 如果n除了包含关键词k外还包含其他文本,则它属于名值混合属性.属性值节点就是当前

节点n. 将节点 n 加入 $valueList$.

3) 如果 n 仅包含关键词 k,且在不包括更多文本的情况下,n 的最高祖先节点不是表格类节点,则它属于结构对齐类属性. 我们首先通过 $CandidateValues(n)$找到所有候选属性值节点,然后选择使 $SemDist(n,value)$最小的候选属性值节点 $value$ 作为属性值,将 $value$ 加入 $valueList$.

4) 如果 n 仅包含关键词 k,且在不包括更多文本的情况下,n 的最高祖先节点是表格类节点,则它属于视觉对齐类属性. 同样通过 $CandidateValues(n)$和 $SemDist(n,value)$找到属性值,它们的定义与结构对齐类属性的相关定义有所不同.

5) 从 $nameList$ 中获得下一个元素(n,k),转到第 2)步.

3.2 结构对齐类属性

结构对齐类属性的属性名和属性值位于 HTML 代码中相近的位置. 因此,我们以属性名节点为中心,向前和向后选择所有的文本节点作为候选数据节点.

定义 4. 假设网页中所有文本节点形成集合 T,每个文本元素按照它们在网页中的顺序被标记为 $t_1,t_2,\cdots,t_n$,属性名节点为 t_i,则候选属性值节点可定义为

$$CandidateValues(t_i)=\{t \mid \forall t,\ t\in T \wedge t\neq t_i\}.$$

为了定义属性名节点与属性值节点间的语义距离 $SemDist(name,value)$,我们分别从网页代码和 DOM 树的角度引入两个度量.

定义 5. 对于任意两个文本节点 t_i,t_j,i,j 为它们按照网页顺序编号的下标,则它们的代码距离定义为

$$CDist(t_i,t_j)=|j-i|.$$

这个定义说明,在 HTML 代码中越接近的两个节点它们的语义距离越小.

定义 6. 对于任意两个文本节点 t_i,t_j,t_i 为属性名节点,它们在 DOM 树中的最低公共子节点定义为 $lca(t_i,t_j)$,一个节点 t 的深度定义为 $depth(t)$,则 t_i,t_j 的树距离定义为

$$TDist(t_i,t_j)=depth(t_i)-depth(lca(t_i,t_j)).$$

这是因为 DOM 树代表了网页设计者设想的层次关系. $lca(t_i,t_j)$越靠近 t_i 则 t_i,t_j 的语义距离越小. 最终,属性名与属性值节点的语义距离是代码距离和树距离的组合.

定义 7. 对于任意两个文本节点 t_i,t_j,t_i 为属性名节点. 则语义距离定义为

$$SemDist(t_i,t_j)=\alpha CDist(t_i,t_j)+(1-\alpha)TDist(t_i,t_j).$$

在这个定义中,α 的取值并不重要. 这是因为对于属性名节点 t_n 和两个候选属性值节点 t_{v1},t_{v2},若 $CDist(t_n,t_{v1})<CDist(t_n,t_{v2})$,则根据 DOM 树的性质必然有 $TDist(t_n,t_{v1})\leqslant TDist(t_n,t_{v2})$,因此,也必然有 $SemDist(t_n,t_{v1})<SemDist(t_n,t_{v2})$.

3.3 视觉对齐类属性

抽取视觉对齐类属性的关键是要区分属性名和属性值的阅读方向:从左到右或从上到下. 通常,读者通过比较水平方向节点和垂直方向节点的语义和视觉特征来判断阅读方向. 我们也采用相似的方法,首先将与属性名对齐的表格节点分为水平对齐和垂直对齐两个集合,抽取这两个集合的特征,然后通过机器学习来模拟读者判断阅读方向的行为. 两个集合的特征如下.

1) 关键词数量:关键词通常出现在属性名中而不会出现在数据值中;

2) 以冒号结尾的文本数量:属性名通常以冒号结尾;

3) 以粗体显示的文本元素数量:网页设计者通常使用粗体使属性名变得醒目;

4) TH 节点的数量:TH 的语义就代表了表头节点,即属性名节点.

从两个集合中共可以得到 8 个特征,通过线性分类器可以判断表格的阅读方向. 训练数据是从数据集中随机选择的 30 个有表格的网页.

定义 8. 设 t_i 为属性名,属性名右边与之精确对齐的表格单元集合为 C_0,属性名下方与之精确对齐的表格单元集合为 C_1. 则候选属性值集合可定义为

$$CandidateValues(t_i)=\begin{cases}C_0, & \text{阅读方向为水平},\\ C_1, & \text{阅读方向为垂直}.\end{cases}$$

在所有候选属性值中,我们选择与属性名节点距离最近的表格节点作为属性值.

定义 9. 对于任意两个文本节点 t_i,t_j,t_i 为属性名节点. 则语义距离定义为

$$SemDist(t_i,t_j)=VDist(t_i,t_j),$$

$VDist(t_i,t_j)$为两个节点中心的距离.

4 实验评估

4.1 数据集介绍

有名属性抽取任务的目标是一次抽取一个领域

的数据,因此,测试数据集中的每个领域都必须包含足够数量的网站.目前信息抽取领域存在的标准数据集都不符合这个要求.例如,测试数据集 TBDW[16]总共只包含 51 个深度搜索引擎,这些引擎又来自多个不同的应用领域,每个领域只有几个网站.因此,我们从 Yahoo Directory 收集了"书籍"、"笔记本电脑"和"视频分享"这 3 个不同领域的数据,在每个领域按照顺序收集了前 50 个可访问的网站物品页面.然后,网页中的所有属性都通过人工标注确定它是否为有名属性.如果是有名属性还需要标定有名属性的属性名和属性值,并收集这个属性的关键词.

为了验证有名属性抽取任务的价值,我们计算有名属性占所有属性的比例.在书籍领域中,有名属性比例是 75.29%.笔记本电脑领域是 87.11%,视频分享领域是 68.42%.在 3 个领域中有名属性比例都接近或高于 70%,在笔记本领域中这一比例甚至接近 90%.这说明假设是成立的.

4.2 属性抽取

有名属性抽取方法包含属性名定位和属性值定位这两个步骤,为了更好地理解这两个步骤,我们首先分别评价它们.在评价属性名定位的效果时,基准方法是将所有包含关键词的文本节点当作属性名节点.

实验结果如表 1 所示.我们针对每个领域都进行了 4 组实验,这些实验分别代表基准方法、使用规则 1、使用规则 2 和规则 3、同时使用所有规则的情况.在同时使用所有规则的情况下,3 个领域的属性名定位的准确率都在 80%以上,召回率都在 95%以上.与基准方法相比,召回率略有下降,但准确率有了很大提高.这说明我们的属性名识别规则可以有效地区分真实属性名节点和其他包含关键词的节点.

Table 1 Results of Name Location Algorithm

表 1 属性名定位算法的准确率和召回率 %

Domain	Book		Laptop		Video Sharing	
	Precision	Recall	Precision	Recall	Precision	Recall
Base	33.53	100	29.89	100	63.67	100
Rule 1	87.86	71.51	89.83	83.25	97.39	94.30
Rule2,3	66.67	24.42	63.83	17.44	50.00	4.43
All Rule	**81.28**	**95.93**	**83.82**	**99.42**	**93.41**	**98.73**

从这 4 组实验中我们还可以发现,规则 2 和规则 3 的准确率比较低,在加入这两条规则后属性名定位的准确率有所下降.这是因为规则 2 和规则 3 的约束较弱,它们仅要求当前文本节点中的其他文本构成一个名词短语.但是,这两条规则在书籍和笔记本电脑领域的召回率都达到了 24.42%和 17.44%.因此,在通常情况下我们同时使用所有规则.如果优先考虑准确率则可以忽略规则 2 和规则 3.

为了单独评价属性值定位的效果,避免属性名定位算法的影响,我们以人工标注的属性名为基础,寻找每个属性名对应的属性值.评价准则同样是标准的准确率和召回率.基准方法与我们的算法基本相同,只是将衡量节点密切程度的度量方法由 $SemDist(name,value)$ 替换成了文献[15]中的定义.

从表 2 中可以看出,总的准确率和召回率始终相等.这是因为属性名和属性值是一一对应的关系,因此算法得到的结果集和人工标注结果集的大小一样.根据准确率和召回率的定义,它们自然保持相等.同时,在所有领域中,算法的准确率始终高于 90%,相对于基准方法有很大提高.这是由于文献[15]中的定义并不能准确反映两个节点之间的关系,而我们的方法克服了它的缺点.

Table 2 Results of Value Location Algorithm

表 2 属性值定位算法的准确率和召回率 %

Domain	Book		Laptop		Video Sharing	
	Precision	Recall	Precision	Recall	Precision	Recall
Base	70.35	70.35	74.21	74.21	94.47	94.47
Mixed	100.00	34.88	100.00	1.89	100.00	17.17
Structure	94.03	36.63	98.18	33.96	94.34	50.50
Visual	82.22	21.51	92.08	58.49	94.83	27.78
All	**93.02**	**93.02**	**94.34**	**94.34**	**95.45**	**95.45**

在 3 类属性中,名值混合属性的抽取准确率是 100%.这是因为属性名和属性值位于同一个文本节点中,准确定位了属性名之后,属性值也就完全确定了.结果对齐类属性的错误主要是因为属性名和属性值之间夹杂了一些说明信息,因此与属性名在结构上最接近的节点不是属性值.例如,紧跟着关键词"product reviews summary"之后的不是产品评价,而是说明信息"(Powered by PowerReviews. com)";视觉对齐类属性的错误主要是因为阅读方向判断出错.

最终,我们将两个步骤串联起来,计算有名属性抽取算法抽取的属性名-值对的整体准确率和召回率.结果如表 3 所示.3 个领域的准确率都接近或高于 80%,召回率都高于 90%.针对不同属性,现有的

全自动抽取算法[11-13]的准确率在66.3%～98.6%之间,召回率在60.4%～89.9%之间.有名属性抽取算法与现有的全自动抽取算法关注不同类型的属性,且我们的算法是领域无关的,而现有的算法针对特定领域进行优化,因此直接比较两者的结果是不公平的.但现有算法的结果充分说明了全自动信息抽取的困难程度.从总体上看,有名属性抽取算法在准确率方面还有进一步提高的余地,特别是属性名定位算法的准确率.我们将在这方面进行更多的研究.

Table 3 Overall Results of Named Attribute Extraction
表3 有名属性抽取算法的整体准确率和召回率 %

Domain	Precision	Recall
Book	75.61	90.11
Laptop	82.03	93.24
Video Sharing	92.41	94.34

5 结 论

本文在网页信息抽取自动化方面进行了有益的研究,主要贡献如下:

1) 为了弥补现有自动化抽取方法只能应用于受限领域的状况,首次提出了有名属性抽取任务,并在多个领域内证明了这个任务的价值;

2) 提出了扩展领域模型作为有名属性抽取的基础,并讨论了扩展领域模型的优势和局限;

3) 提出了一套完整的有名属性自动化抽取框架,提出了一种新的度量属性名和属性值之间关系密切程度的方法.实验结果表明,这些方法保证了较高的准确率和召回率.

参 考 文 献

[1] Liu L, Pu C, Han W. XWRAP: An XML-enabled wrapper construction system for Web information sources [C] //Proc of Int Conf on Data Engineering. 2000: 611-621

[2] Sahuguet A, Azavant F. Building intelligent Web applications using lightweight wrappers [J]. Data & Knowledge Engineering, 2001, 36(3): 283-316

[3] Muslea I, Minton S, Knoblock C. A hierarchical approach to wrapper induction [C] //Proc of the 3rd Annual Conf on Autonomous Agents. New York: ACM, 1999: 190-197

[4] Valter C, Giansalvatore M, Paolo M. RoadRunner: Towards automatic data extraction from large Web sites [C] //Proc of the 27th Int Conf on Very Large Data Bases. San Francisco: Morgan Kaufmann, 2001: 109-118

[5] Arvind A, Hector G-M, Stanford U. Extracting structured data from Web pages [C] //Proc of the 2003 ACM SIGMOD Int Conf on Management of Data. New York: ACM, 2003: 337-348

[6] Kai S, Georg L. ViPER: Augmenting automatic information extraction with visual perceptions [C] //Proc of the 14th ACM Int Conf on Information and Knowledge Management. New York: ACM, 2005

[7] Yang Shaohua, Lin Hailue, Han Yanbo. Automatic data extraction from template-generated Web pages [J]. Journal of Software, 2008, 19(2): 209-288 (in Chinese)
(杨少华, 林海略, 韩燕波. 针对模板生成网页的一种数据自动抽取方法[J]. 软件学报, 2008, 19(2): 209-288)

[8] Ma Anxiang, et al. Deep Web data extraction based on result pattern [J]. Journal of Computer Research and Development, 2009, 46(2): 280-288 (in Chinese)
(马安香, 等. 基于结果模式的 Deep Web 数据抽取[J]. 计算机研究与发展, 2009, 46(2): 280-288)

[9] Hu Dongdong, Meng Xiaofeng. Automatically extracting Web data using tree structure [J]. Journal of Computer Research and Development, 2004, 41(f0): 1607-1613 (in Chinese)
(胡东东, 孟小峰. 一种基于树结构的 Web 数据自动抽取方法[J]. 计算机研究与发展, 2004, 41(10): 1607-1613)

[10] Mei Xue, et al. Fully automatic wrapper generation for Web information extraction [J]. Journal of Chinese Information Processing, 2008, 22(1): 22-29 (in Chinese)
(梅雪, 等. 一种全自动生成网页信息抽取 Wrapper 的方法[J]. 中文信息学报, 2008, 22(1): 22-29)

[11] Jun Z, et al. 2D conditional random fields for Web information extraction [C] //Proc of the 22nd Int Conf on Machine Learning. New York: ACM, 2005: 1044-1051

[12] Jun Z, et al. Simultaneous record detection and attribute labeling in Web data extraction [C] //Proc of the 12th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2006: 494-503

[13] Jun Z, et al. Webpage understanding: An integrated approach [C] //Proc of the 13th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2007: 903-912

[14] Alfred V A, Margaret J C. Efficient string matching: An aid to bibliographic search [J]. Communication of ACM, 1975, 18(6): 333-340

[15] Luigi A, et al. Automatic annotation of data extracted from large Web sites [C] //Proc of WebDB. 2003: 7-12

[16] Yasuhiro Y, et al. Testbed for information extraction from deep Web [C] //Proc of the 13th Int World Wide Web Conf on Alternate Track. New York: ACM, 2004: 346-347

Wang Yu, born in 1984. PhD candidate. His main research interests include information extraction, Web search and Web mining.

王 宇,1984 年生,博士研究生,主要研究方向为信息抽取、互联网检索与挖掘.

Tan Songbo, born in 1978. PhD and associate professor. His main research interests include Web search and mining, text classification, and sentiment analysis.

谭松波,1978 年生,博士,副研究员,主要研究方向为互联网搜索与挖掘、文本分类与情感分析(tansongbo@software.ict.ac.cn).

Liao Xiangwen, born in 1980. PhD. His main research interests include natural language processing and information retrieval.

廖祥文,1980 年生,博士,主要研究方向为自然语言处理及信息检索.

Zeng Yiling, born in 1980. Received his BS degree from University of Science and Technology of China in 2004, and now is a PhD candidate in the Institute of Computing Technology, the Chinese Academy of Sciences. His main research interests include text mining, large scale text content processing and information retrieval.

曾依灵,1980 年生,博士研究生,主要研究方向为文本挖掘、大规模文本处理、信息检索等.

Research Background

Web information extraction has found application in a huge range of problem domains, such as vertical search, sentiment classification, and Web mining. More specifically, the main target of Web information extraction is the attributes of Web objects. Many researches focus on this area and try to automate the attribute extraction process; however, this problem has proved to be difficult. In this paper, instead of automating the whole attribute extraction process, a subtask is divided from the general problem and is solved separately. This new subtask is called named attribute extraction. Statistics results from multiple datasets prove that named attribute extraction task covers more than 60% attributes in these domains; therefore, addressing this subproblem is an important step towards the complete solution of the more general problem. An extended domain model based extraction method is proposed, which is very effective in extracting named attributes. This method relies on the co-occurrence of attribute names and attributes values. Our work is supported by 863 National Hi-Tech Research Development Program (2007AA01Z441 and 2007AA01Z438) and the 973 National Basic Research Key Programs (2004CB318109 and 2007CB311100).

基于扩展领域模型的有名属性抽取

作者： 王宇， 谭松波， 廖祥文， 曾依灵， Wang Yu， Tan Songbo， Liao Xiangwen， Zeng Yiling

作者单位： 王宇,廖祥文,曾依灵,Wang Yu,Liao Xiangwen,Zeng Yiling(中国科学院计算技术研究所,北京,100190;中国科学院研究生院,北京,100190)， 谭松波,Tan Songbo(中国科学院计算技术研究所,北京,100190)

刊名： 计算机研究与发展 ISTIC EI PKU

英文刊名： JOURNAL OF COMPUTER RESEARCH AND DEVELOPMENT

年，卷(期)： 2010,47(9)

被引用次数： 1次

参考文献(16条)

1. Liu L. Pu C. Han W XWRAP:An XML-enabled wrapper construction system for Web information sources 2000

2. Sahuguet A. Azavant F Building intelligent Web applications using lightweight wrappers[外文期刊] 2001(03)

3. Muslea I. Minton S. Knoblock C A hierarchical approach to wrapper induction 1999

4. Valter C. Giansalvatore M. Paolo M RoadRunner:Towards automatic data extraction from large Web sites 2001

5. Arvind A. Hector G -M. Stanford U Extracting structured data from Web pages[外文会议] 2003

6. Kai S. Georg L ViPER:Augmenting automatic information extraction with visual perceptions 2005

7. 杨少华. 林海略. 韩燕波 针对模板生成网页的一种数据自动抽取方法[期刊论文]-软件学报 2008(02)

8. 马安香 基于结果模式的Deep Web数据抽取[期刊论文]-计算机研究与发展 2009(02)

9. 胡东东. 孟小峰 一种基于树结构的Web数据自动抽取方法[期刊论文]-计算机研究与发展 2004(10)

10. 梅雪 一种全自动生成网页信息抽取Wrapper的方法[期刊论文]-中文信息学报 2008(01)

11. Jun Z 2D conditional random fields for Web information extraction[外文会议] 2005

12. Jun Z Simultaneous record detection and attribute labeling in Web data extraction 2006

13. Jun Z Webpage understanding:An integrated approach 2007

14. Alfred V A. Margaret J C Efficient string matching:An aid to bibliographic search 1975(06)

15. Luigi A Automatic annotation of data extracted from large Web sites 2003

16. Yasuhiro Y Testbed for information extraction from deep Web[外文会议] 2004

本文读者也读过(2条)

1. 叶正. 林鸿飞. 苏绥. 刘菁菁. Ye Zheng. Lin Hongfei. Su Sui. Liu Jingjing 基于支持向量机的人物属性抽取[期刊论文]-计算机研究与发展2007,44(z2)

2. 吴月萍. 陈玉泉. WU Yue-ping. CHEN Yu-quan 基于Web的概念属性抽取的研究[期刊论文]-中国管理信息化 2009(10)

引证文献(2条)

1. 崔智刚. 申新鹏. 魏向阳. 赖碧云 异构数据库应用架构研究[期刊论文]-价值工程 2012(17)

2. 张利军. 申新鹏. 丁振华. 谈少民 ERP动态领域模型[期刊论文]-价值工程 2012(19)