文章编号:1001 - 9081(2006)03 - 0655 - 04

基于包装器模型的文本信息抽取

王敬普,林亚平,周顺先,岳 文
(湖南大学 计算机与通信学院, 湖南 长沙 410082)
(wangjingpu@hotmail.com)

摘 要: 在分析基于标志和基于文本模式两类算法的基础上,提出了一种新的包装器归纳学习算法。新算法综合上述两类算法的优点,不但能利用页面的标志信息进行信息定位,而且能利用文本的模式信息来进行信息抽取和对抽取结果进行必要的过滤。实验结果表明,新算法具有较高的信息抽取精度与信息表达能力。

关键词: 信息抽取;包装器;标志;文本模式;归纳学习
中图分类号: TP181 **文献标识码:** A

Text information extraction based on wrapper model

WANG Jing-pu, LIN Ya-ping, ZHOU Shun-xian, YUE Wen
(*College of Computer and Communication, Hunan University, Changsha Hunan* 410082, *China*)

Abstract: A new wrapper induction algorithm was proposed for text information extraction after analyzing two types of algorithms based on landmark and text pattern. The new algorithm can take the advantage of above-mentioned two algorithms. It can locate the information based on the landmark information of Web pages, and can use the text pattern to extract and filter large quantity of Web text. Experiment results show that the new method achieves higher accuracy and expressiveness of information extraction.

Key words: information extraction; wrapper; landmark; text pattern; induction

0 引言

自动文本信息抽取是文本信息处理的一个重要环节[1,2]。信息抽取是指从文本中自动抽取相关的或特定类型的信息。目前信息抽取模型主要有三种:基于词典的抽取模型[3,4]、基于隐马尔可夫模型(Hidden Markov Model,HMM)的抽取模型[5~7]和基于规则的抽取模型[9~13]。

基于词典的文本信息抽取模型需要首先构造抽取模式词典,然后使用该模式词典从未标记文本中抽取所需信息。文献[3]提出了一种从训练示例中学习的方法来自动构建模式词典;文献[4]应用多级自举算法生成语义和抽取模式词典。上述基于词典的模型需要大量的手工操作与很强的专业知识背景,因此不适宜海量 Web 文本信息的处理。为了克服手工操作和知识背景的缺陷,隐马尔可夫模型(HMM)被应用于信息抽取。文献[5]利用学习到的 HMM 来抽取计算机科学研究论文的标题、作者和摘要等头部信息;文献[6]结合 HMM 和最大熵原理,提出了一种最大熵隐马尔可夫模型;文献[7]利用文本排版格式、分隔符等信息对文本进行分块,在分块的基础上建立隐马尔可夫模型来进行文本信息抽取。上述基于 HMM 的模型由于要考虑整个文本,因此不适合含有较多无关标记(Token)的 Web 文本的处理,因为大量无关 Token 将造成 HMM 节点过多,使训练开销增大,HMM 建模的有效性降低。

包装器是一种基于规则的文本信息抽取模型,是信息引擎[8]的重要组件,能从各种页面中抽取相关的信息。包装器的规则集易于建立,抽取精度高,因此适合于含有较多半结构化信息的 Web 页面处理。文献[9]将归纳学习方法引入包装器的自动生成,并基于归纳学习方法给出了六个包装器类。但因其只考虑了与待抽取数据紧相邻的分隔符,因此不能包装某些属性值缺失或信息项次序不固定的资源。文献[10]基于非确定有限状态机提出了两类抽取器:单通道和多通道抽取器。其规则语言允许使用语义类和析取项,所以能够包装属性值缺失或信息项次序多变的信息。但其主要不足是无法使用未紧随抽取项之后或之前的分隔符,因而抽取精度不高。文献[11]对文献[9,10]进行了改进,它首先将页面的层次结构表示成一个内嵌目录树,并为树中的每个叶子节点生成一条规则;然后再为每个内部列表节点生成一条额外的迭代规则,因此能够包装具有任意层嵌套结构的信息源。由于它在规则产生时不但考虑了与抽取信息相邻的分隔符,而且还考虑了与抽取信息不相邻但具有明显标志的分隔符,因此其表达能力高于文献[9,10]中的算法。上述几种归纳学习算法均基于页面的标志信息,因此对标志不明显或者标志缺失的信息,均无法正常处理。文献[12]从另一个角度出发,通过学习数据的自身结构来归纳数据的文本模式信息。这些模式信息不但能进行信息抽取,而且能实现包装器的平衡。因其不考虑页面的标志信息,因此不受页面布局的影响。但该算法的缺点是对于页面上的信息难于定位,模式过于抽象时抽取精度较低,模式过于具体时抽取的召回率较低。

为了改善上述基于包装器模型的信息抽取的精度与召回率,并提高其表达能力,本文提出了一种新的包装器归纳学习算法。该算法综合利用页面的标志信息及文本模式信息的优点,首先基于页面的标志信息进行信息定位,然后利用学习到的模式信息进行 Web 文本信息的抽取与过滤。实验结果表

收稿日期:2005 - 09 - 17 **修订日期:**2005 - 12 - 02 **基金项目:**国家自然科学基金资助项目(60272051)

作者简介:王敬普(1979 -),男,河南驻马店人,硕士研究生,主要研究方向:机器学习; 林亚平(1955 -),男,湖南邵阳人,教授,博士生导师,主要研究方向:计算机网络、机器学习; 周顺先(1971 -),男,湖南邵阳人,博士研究生,主要研究方向:机器学习; 岳文(1980 -),女,湖南邵阳人,硕士研究生,主要研究方向:机器学习.

明,新包装器模型具有较高的抽取精度与信息表达能力。

1 包装器模型

包装器是一种软件构件,负责将数据和查询请求由一种模式转换成另一种模式。因此,一个包装器实际上可看作是一类页面到该页面所含元组集合的函数。在 WWW 信息应用中,包装器是一个软件过程,应用已经定义好的信息抽取规则,将展现在输入 Web 页面中的信息数据抽取出来,转换成用特定的格式描述的信息,提供给其他信息系统作进一步的处理。包装器一般包括三个部分:规则库、规则执行模块和信息转换模块。应用包装器的抽取过程如图 1 所示。

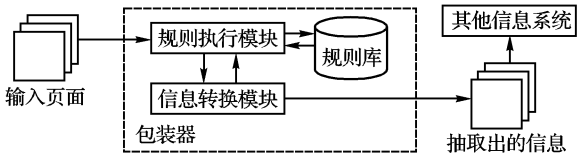


图 1 应用包装器的抽取模型

在图 1 所示的抽取过程中,包装器根据输入页面的类型从规则库中选择对应的抽取规则集并提供给规则执行模块。规则执行模块将此抽取规则应用到输入页面上,抽取出页面所包含的信息,并把该信息输入到信息转换模块。信息转换模块将被抽取出来的信息转换成特定的、能够被其他信息系统所识别的格式。

信息抽取的规则在包装器中占有重要地位,包装器依靠抽取规则从输入页面中提取信息。我们的归纳学习算法旨在生成高精度的抽取规则。

2 预备知识

2.1 分级树

Web 页面的优势在于其可读性强,在构造页面结构的时候有一些符合人们阅读习惯的常识。例如页面上的信息往往显示出了一种分级结构;而且,半结构化信息往往用元组的列表形式来描述,并以简单的分隔符来区分元组之间的数据项。为了能够适应页面结构的复杂性(例如具有多层嵌套结构),本文提出分级树(Hierarchical Tree,HT)的概念。

在 HT 中,一个页面被描述为树状结构,其叶子节点表示用户感兴趣的内容,其内部节点代表 k 元组的列表,k 元组的每一个数据项或者是一个叶节点或者是另一个 k 元组的列表 L(L 被称作嵌套表)。

一个文档可以看作 Token 的序列(词、数字、HTML 标记等)。在 HT 树中根节点的内容是整个文档的 Token 序列,任一节点 x 的内容是它父节点 p 的内容的子序列。因此,利用 HT 可以把抽取规则简化为只需考虑从父节点抽取子节点信息的简单任务。

利用分级树结构,可以把一个复杂的抽取任务分解成几个相对简单的任务,因此能够包装有任意多层嵌套的数据,而这些嵌套结构的数据是很多算法不能正确处理的。由于每一个节点独立于其兄弟节点被抽取,因此不要求这些节点有固定的次序。因此,采用本文提出的分级树结构,能处理某些信息点缺失或者信息点以不同次序出现的 Web 文档。

2.2 文本模式

文本的模式用来描述所要抽取信息的自身结构。例如,当抽取电话号码的时候,电话号码能够通过一个简单的模式来描述:"(Number) Number- Number";当抽取 URLs 的时候,我们能利用 URLs 的自身结构,即 URLs 大多都以"http://www."开始,而以".html"结束。考虑到 Web 文档信息的特点,本文基于如下特性来描述所抽取信息自身结构:

1)训练集中标记信息的长度范围(用 Token 的个数来表示)。

2)训练集中出现的 Token 类型。该特性由具体的一些通配符组成,这些通配符与标记好的训练集中的筹码相匹配。图 2 给出了自定义的通配符分级语义树,可以根据实际的需要扩展。

3)开始模式。用来描述一个信息项的开始,例如"http://www."表示一个 URL 的开始。

4)结束模式。用来描述一个信息项的结束,例如".html"表示一个 URL 的结束。

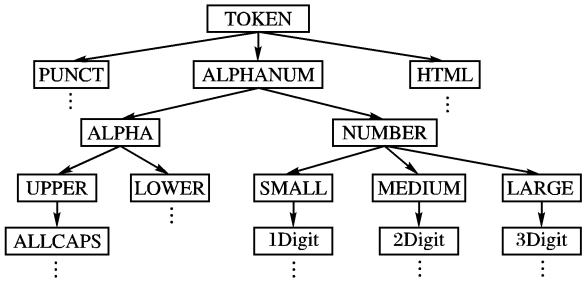


图 2 通配符分级语义树(虚线表示可以继续扩展)

上述模式信息既可用作抽取器又可用作鉴别器。当页面的标志不明显时,可以把这些模式信息作为抽取器来提高召回率;标志明显时这些模式信息可用作鉴别器,过滤掉模式不同而标志相同的信息,从而提高抽取的精度。

3 LPWI 算法

基于前面的描述,本节给出一种基于归纳学习的自动文本信息抽取算法(Landmark and text Pattern based Wrapper Induction,LPWI)。该算法综合考虑页面的标志信息以及文本的模式信息,利用 HT 来简化复杂的抽取过程。其生成的抽取规则可用有限状态机描述如下。

3.1 有限状态机

定义 1(标志) 标志为一个或多个连续的 Token,可以用来定位所要抽取的信息,通常是页面布局的一部分。

定义 2(标志自动机) 一组标志以固定的次序来应用时,就构成了一个标志自动机(Landmark Automata,LA)。

标志自动机是一种非确定的有限状态机,当在状态 S_i 输入一个标志 $l_{i,j}$,转换到状态 S_j,则可表示为:$S_i \xrightarrow{l_{i,j}} S_j$。

为了简化描述的复杂度,本文考虑一种特殊的标志:线性标志。线性标志被描述为标记和通配符的序列。每一个通配符可以描述一类标记。

定义 3(线性标志自动机) 线性标志自动机是具有下列特性的一类 LA:

1)每个 LA 只有一个接受状态;

2)在每个非接受的状态,只有两个可能的状态转换:循环转向它自己和转换到另一个状态;

3)每个非循环的状态变换均用一个标志标记;

4)循环状态表示:跳过所有的 Token 直到遇到导向下一个状态的标志。

算法根据输入的数据集来产生抽取规则,每个抽取规则都是 LA 的一个分支。每个分支由固定次序的 SkipTo( ) 与 SkipUntil( ) 规则组成。SkipTo(X) 函数表示从某位置开始,跳过所有的标记直到遇到标记 X,并跳过 X。SkipUntil(X) 表示跳过所有的标记,直到遇到标记 X,但不跳过 X。所有的这些分支就构成了简单标志语法(Simple Landmark Grammar,SLG)。

3.2 LPWI 算法

归纳学习算法 LPWI 用来产生 SLG,产生的 SLG 可以用来确定子节点在它父节点中的开始和结束位置。LPWI 是一个顺序覆盖算法,它首先产生一个规则去覆盖尽可能多的集合中的正例,然后从集合中删除被覆盖的正例,再在剩余元素的基础上产生另一个规则去覆盖尽可能多的集合中的正例,如此循环直至所有的元素被覆盖。最后算法返回规则的析取作为最后的提取规则:

```
算法:LPWI
输入:带标记的训练集(Examples)
输出:标志与模式规则集(SLGs)
{ RuleSets = ∅                    //初始时 RuleSets 为一个空的 SLG
  While( Examples ≠ ∅)
  {
    LandmarkRules = LearnDisjunct( Example)
    //学习标志规则
    PatternRules = LearnPattern( Example)
    //学习模式规则
    IF ( PatternRules 规则比 LandmarkRules 规则有意义)
       Rulesets = Rulesets + PatternRules
    Else
      Rulesets = Rulesets + LandmarkRules
    Examples = Examples – Covered( Rulesets)
    //删除被规则覆盖的训练集
  }
  Return Rulesets
}
```

函数 LearnDisjunct( ) 用来产生理想的标志规则析取支。它首先以标记好的训练集为基础来产生一个候选项的集合,每一个候选项是一个有两个状态的标志自动机。函数随后不断地选出和提炼候选项,直到找到最完美的候选项。当选出的候选项不能进行正确的抽取的时候,函数就对候选项进行提炼。提炼包括两个方面:标志提炼和拓扑结构的提炼。标志提炼是在候选项中加入新的筹码,使标志变得更明确;而拓扑结构提炼是在状态机上加入新的状态。直到得到完美的析取支为止。

函数 LearnPattern( ) 用来对所要抽取的信息进行文本模式学习,它力图找到最有意义的模式。两种模式的意义比较是根据模式能匹配的训练集的个数来决定的。如果一种模式比另一种模式匹配的训练集多,则称该模式更有意义。模式信息的形成采用自下而上的方法,这样可以找到最明确的文本模式。语义树的最底层就是最明确的信息,当下层的模式信息不能正确匹配时,算法转到上层来继续寻找模式信息。

当得到的模式规则比标志规则更有意义时,模式规则被加入 SLG,反之标志规则被加入 SLG。当两种规则的意义一样时,我们选择标志规则,因为标志规则可以对信息进行定位,而模式规则如果过于抽象,则可能产生很多干扰项,这样就降低了抽取的精度。

算法首先将抽取规则集置空,然后根据输入的标记好的训练集来产生抽取规则。算法首先学习得到所要抽取信息的标志知识,然后学习所要抽取信息的模式知识。当模式知识比标志知识有意义时,模式知识被加入规则集来进行抽取。最后利用模式知识对抽取结果进行过滤。可见,我们的算法不但能利用页面上有用的标志知识,而且能利用所要抽取信息的模式知识,这样能综合利用这两种方法的优势。实验结果也显示,我们的算法不但能提高抽取精度而且能提高召回率。

3.3 相关表达能力

评估一个包装器的一个重要指标是包装器的表达能力,一个包装器的表达能力表示这个包装器能处理的信息资源的能力。说包装器类 W_1 较包装器类 W_2 的表达能力强是指类 W_1 能正确抽取的信息资源包含类 W_2 能正确抽取的信息资源。设 Π 表示所有 Web 页的集合,若 W 表示一包装器类,设 $\Pi(W)$ 表示 Π 中包装器 W 能正确抽取的 Web 页子集。

定义 4　设 W_1 和 W_2 是两个不同的包装器类,若 $\Pi(W_1) \supset \Pi(W_2)$ 则称包装器类 W_1 比包装器类 W_2 的表达能力强。

性质 1　LPWI 类包装器比 STALKER 类包装器表达能力强。即 Π (LPWI) $\supseteq \Pi$ (STALKER)。

证明:1)包装器类 STALKER 利用页面的标志信息,而由包装器类 LPWI 的构造过程可以看出 LPWI 同样能充分利用页面的标志信息,并且 LPWI 能利用文本的模式信息,可以说是对 STALKER 类包装器的扩展,因此 Π (LPWI) $\supseteq \Pi$ (STALKER)。

2)存在一个页面 P 属于 Π,LPWI 能正确包装而 STALKER 不能包装。例如:对于下面的页面代码:

```
<p> Name: <b> Yala </b> <p> Cuisine: Thai <p> <i>
4000 Colfax, Phoenix, AZ 85258 (602) 508-1570
</i> <br> <i>
523 Vernon, Las Vegas, NV 89104 (702) 578-2293
</i> <br> <i>
403 Pico, LA, CA 90007 (213) 798-0008
</i> <BLOCKQUOTE>
```

当需要抽取邮政编码信息时,页面没有明显的标志信息来定位它,因此 STLAKER 算法不能正确抽取。但邮政编码有很规则的模式:5Digit,因此 LPWI 能正确进行抽取。

综合 1),2)可知,LPWI 类包装器比 STALKER 类包装器表达能力强。

4 实验结果及分析

4.1 评价标准

信息抽取技术采用召回率(Recall,R)、精度(Precision,P)来作为评价标准。总精确度(General Precision,GP)用来描述含有多个槽(slot)的一个信息源的总体精确度。我们用 ce,te 和 fe 来表示所有抽取出的正确信息个数、没有抽取出的正确信息个数和抽取出的错误信息个数。其计算公式为:

$$GP = \frac{\sum_{slot} ce}{\sum_{slot} (ce + te)} \times 100\%$$

$$R = \frac{ce}{ce + te} \times 100\%,\ P = \frac{ce}{ce + fe} \times 100\%$$

P 和 R 取值在 0 和 1 之间,通常存在反比的关系,即 P 增大会导致 R 减小,反之亦然。因此评价一个系统时,应同时考虑 P 和 R,比较常用的评价指标为 F 值评价法:$F = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$,其中 β 是一个预设值,决定对 P 侧重还是对 R 侧重,通常设定为 1。这样用 F 值就可评价出系统的性能。

4.2 实验数据源

RISE 信息网站[13]是美国加利福尼亚大学信息科学机构建立的,是一个“在线信息资源网,用来对学习算法的性能进行实验分析”。网站中包含广泛合理的信息源,这些信息常被信息抽取和包装器归纳学习算法用来进行实验分析比较。[9~11]

4.3 算法性能比较

在文献[10]的实验基础上,基于 RISE 网站的如下两类信息作为数据源来进行实验:一类是 STALKER 和 LPWI 都能包装的;另一类数据源是 STALKER 不能包装而 LPWI 能包装的。为了能够充分比较这两个算法的性能,我们将实验环境和参数设置成一样。在这些数据源中选择最难包装的信息源

S3 来测试两个算法的总精确度,结果见图 3。

表 1 两种算法的性能比较(其中“-”表示算法不能处理)

信息点	STALKER			LPWI		
	P(%)	R(%)	F(%)	P(%)	R(%)	F(%)
S11-Name	94	100	97	96	100	98
S11-Email	98	100	99	100	100	100
S11-Update	66	95	78	100	100	100
S11-Organiz.	48	97	64	96	97	96
S11-Alt. Name	100	100	100	100	100	100
S11-Provider	100	100	100	100	100	100
S24-Language	100	100	100	100	100	100
S24-URL	91	100	95	100	100	100
S24-Image	100	100	100	100	100	100
S24-Translat	89	95	92	89	95	92
S24-ListExtr	100	100	100	100	100	100
S26-House	100	100	100	100	100	100
S26-Number	100	100	100	100	100	100
S26-Price	97	100	98	100	100	100
S26-ListExtr	100	100	100	100	100	100
S21-Name	-	-	-	100	100	100
S21-Loc.	-	-	-	100	100	100
S21-snippet	-	-	-	100	86	92
S21-URL	-	-	-	100	100	100

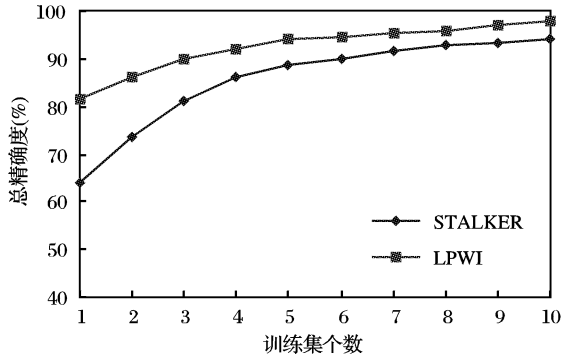
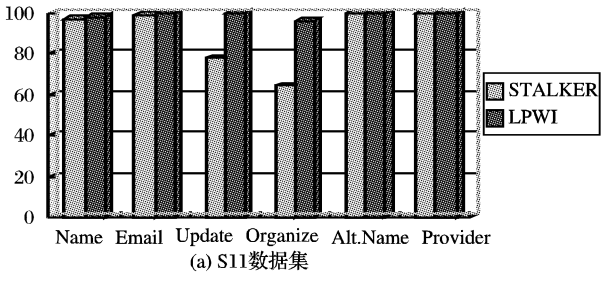


图 3 在 S3 信息源中总精确度比较



(a) S11数据集

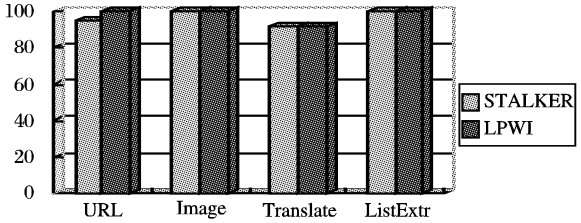


图 4 S11 和 S24 两个数据集上的 F 值比较($\beta=1$)

从图 3 可以看出,LPWI 算法比 STALKER 算法的总精确度要高。这主要是我们的算法能基于页面的分层结构,将复杂的抽取工作分解成几个相对简单的任务;并且算法能利用学习到的模式知识来进行抽取和过滤。在多个信息点具有相同的标志的情况下,学习到的模式知识可以对抽取结果进行过滤,这样可以提高抽取的精确度。

选取两组具有代表性的数据源 S11 和 S24 来比较 STALKER 和 LPWI 的 F 值,实验结果见图 4。随后在每个数据源中随机选出 10 个数据进行标记,形成训练集,作为 LPWI 算法的输入值,归纳学习抽取规则;并用其余未标记数据做测试集,用 LPWI 进行信息抽取,具体测试结果见表 1。从图 4 可知,对于 STALKER 和 LPWI 算法都能处理的信息源,我们算法的 F 值比 STALKER 的要高。这主要是算法不但利用了页面的标记信息而且利用文本的模式信息。例如在数据源 S11 中抽取 Update 时,由于 Update 和其他的信息点用到同样的标志信息,因此对于基于标志的 STALKER 算法,其精度不高。而 Update 有规则的模式信息,对于能充分利用文本模式信息的 LPWI 算法,其 F 值可以达到 100%。当抽取 Organize 时,虽然模式信息不明显,但是仍可以利用模式信息对抽取结果进行必要的过滤,提高抽取的精度。而对于 S21,STALKER 不能包装的原因是,页面上的数据是一个异构的表格,每一个元素用到的布局都不相同,因此迭代规则很难形成。而 LPWI 算法能利用其中的文本模式信息,因此能处理这样的信息。从以上实验结果比较可知,LPWI 算法具有比 STALKER 算法更好的抽取精度和更强的信息表达能力。

参考文献:

[1] MCCALLUM A, NIGAM K, RENNIE J, *et al*. A machine learning approach to building domain-specific search engines[A]. Proceedings of IJCAI-99 [C], 1999.622-667.

[2] SODERLAND S. Learning Information Extraction Rules for Semi-structured and Free Text[J]. Machine Learning, 1999, 34(13): 233-272.

[3] RILOFF E. Automatically Constructing a Dictionary for Information Extraction Task[A]. Proceeding for the Eleventh National Conference on Artificial Intelligence [C], 1993. 811-816.

[4] RILOFF E, JONES R. Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping[A]. Proceedings of the Sixteenth National Conference on Artificial Intelligence [C], 1999. 811-816.

[5] SEYMORE K, MACALLUM A, ROSENFEL R. Learning Hidden Markov Model Structure for Information Extract[A]. AAAI'99 Workshop on Machine Learning for Information Extraction [C], 1999. 37-42.

[6] FREITAG D, MCCALLUM A, PEREIRA F. Maximum Entropy Markov Models for Information Extraction and Segmentation[A]. Proceedings of ICML[C], 2000.591-598.

[7] 刘云中,林亚平,陈治平. 基于隐马尔可夫模型的文本信息抽取[J]. 系统仿真学报, 2003,16(3):507-509.

[8] KNOBLOCK CA, MINTON S, AMBITE J-L, *et al*. The Ariadne approach to Web-based Information Integration[J]. International Journal of Cooperative Information Sources, 2001,(10):145-169.

[9] KUSHMERICK N. Wrapper induction for information extraction[D]. Department of Computer Science, University of Washington, 1997.

[10] HSU C-N, CHANG C-C. Finite-state transducers for semi-structured text mining[A]. Proceedings of IJCAI-99 Workshop on Text Mining: Foundations, Techniques and Applications[C], 1999.

[11] MUSLEA I, MINTON S, KNOBLOCK CA. Hierarchical wrapper induction for semistructured information sources[J]. Autonomous Agents and Multi-Agent Systems, 2001,4(1/2).

[12] LERMAN K, MINTON S, KNOBLOCK CA. Wrapper Maintenance: A Machine Learning Approach[J]. Journal of Artificial Intelligence Research, 2003,18: 149-181.

[13] RISE, A repository of online information sources used in information extraction tasks[EB/OL]. http://www.isi.edu/info-agents/RISE/index.htm, 1999.