

归纳学习 XPATH Web 信息提取规则

郭太飞, 何洁月

(东南大学 计算机科学与工程学院, 江苏 南京 210096)

摘要: XPATH 在 Web 信息提取中起重要作用, 但是这些 XPATH 规则通常要人工生成。文中讨论了在 XPATH 与基于文本上下文规则的信息提取方法结合的系统中如何归纳学习 XPATH 规则。生成的 XPATH 规则结构简单, 可以为基于文本上下文的信息提取系统提供较为准确的信息定位。

关键词: 信息提取系统; XPATH; 归纳

中图分类号: TP311.5

文献标识码: A

文章编号: 1673-629X(2007)03-0098-04

Inductively Learn XPATH Web Information Extraction Rules

GUO Tai-fei, HE Jie-yue

(Computer Science & Engineering Institute, Southeast University, Nanjing 210096, China)

Abstract: XPATH plays an important role in Web information extraction, but these XPATH rules usually generated by hand. Discusses about how to inductively learn XPATH rules used in an XPATH and text-context-based rules combined information extraction system. The generated rules have simple structure, and they can support as an accurate locator for text-context-based information extraction system.

Key words: information extraction systems; XPATH; induction

0 引言

随着 Internet 的发展, 信息的获取已经非常容易; 但是, 人工从大量的 Web 页面中提取信息并不是一件轻松且有趣的事。应该使用自动化信息提取系统来完成 Web 信息的提取。

W4F^[1], XWRAP^[2], Lixto^[3]等主要根据文档结构, 这方面的例子还有开源的 HTMLRipper (<http://www.fisheroft.ca/>)。该类系统使用的是人工定义的页面结构规则, 精确度高, 但规则定义繁琐, 且鲁棒性不好。SRV^[4], RAPIER^[5], WHISK^[6]等主要依赖文本上下文提取信息, 这类系统具有一定的学习能力, 封装了规则生成的逻辑复杂性, 但忽略了文档结构的作用, 常常面临搜索空间过大的问题, 学习一类页面常常非常耗时。

XPATH 是一种能够在 XML 文档中定位和寻找信息的语言。利用 XPATH 的导航能力可以直接定位

到包含信息的节点, 这可以大大减小基于文本的信息提取系统的搜索空间。

文中主要研究了如何从 Web 页面和标注好信息的文档中学习用于信息提取的 XPATH 规则。

1 系统结构简述

整个系统由两个模块组成, 即规则学习模块和信息提取模块。

系统使用了两种规则, 即 XPATH 规则和文本规则。XPATH 规则基于 XPATH 表达式, 具体形式见下节“XPATH 表达式和规则”, 文本规则是形如 startOf('age'): = equals(neighbour(-2), "a") \wedge isOneOf(neighbour(1), {"year, month, day"}) 的 Horn 子句。上面规则的意思是: 前面第 2 个单词是 a, 且其后第一个单词是 year, month, day 之一的单词, 是信息 age 的起始词。

规则学习模块从 Web 页面和标注好的文档中学习 XPATH 规则和文本规则(见图 1)。

信息提取模块先用 XPATH 规则提取出可能包含信息的节点, 再利用文本规则从这些节点中提取信息(见图 2)。

文中主要研究如何归纳学习 XPATH 规则。

收稿日期: 2006-05-28

基金项目: 江苏省高技术研究计划(G2004034)

作者简介: 郭太飞(1981-), 男, 江苏扬中人, 硕士研究生, 研究方向为语义网络、信息集成; 何洁月, 副教授, 硕士生导师, 研究方向为网络计算、生物信息学。

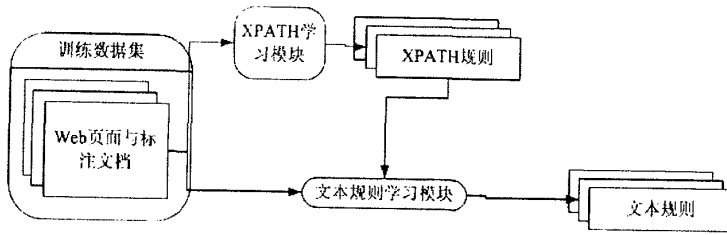


图1 规则学习模块

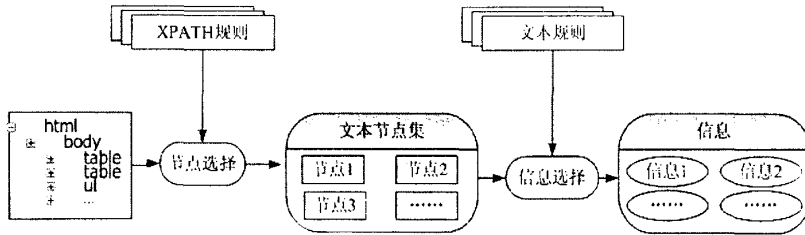


图2 信息提取模块

2 XPATH 表达式和规则

XPATH主要用于定位满足特定条件的节点(集),也可以完成相应的聚集计算。文中只使用XPATH定位节点的功能。完全的XPATH表达式是很不直观的,可以使用简写形式。XPATH规范(v1.0)可以参见 <http://www.w3.org/TR/xpath>,.NET开发人员还可以参见MSDN里的详细介绍。

文中使用XPATH表达式作为XPATH规则的规则体,使用简写形式。如 `ContainsInfo(node, 'age'):- Satisfy(node, '/html/body/ol[position()>2]/li[1]/text()')` 的形式。

●这里,施加了一些限制:

- * XPATH表达式总是使用绝对路径表达式。
- * XPATH路径上的每个轴总是 child。
- * 除了 `position()` 之外,不使用其它的函数。

●由于HTML文档的特点,还有如下限制:

- * XPATH表达式总是以 `/html/body` 开头。
- * 不使用命名空间,因为HTML文档命名空间总是固定的;不使用QName测试,只用NCName测试,这是因为没有使用命名空间。

* XPATH规则总是以 `text()` 测试结束,因为信息通常包含在文本中。

* 谓词依赖于HTML属性和 `position()` 函数。

●信息提取系统对XPATH规则的要求是:

* 必须覆盖所有包含信息的节点。这是可以保证的,因为平凡规则 `/html/body/text()` 总是满足覆盖性。

* 应该尽可能排除不包含信息的节点。

系统假设对于每一项信息仅采用一条XPATH规则进行节点选择,这不是必需的。可以采用序列覆盖策略,用多条XPATH规则覆盖一项信息的所有可能节点。

3 XPATH 规则的归纳学习

使用归纳的方法学习XPATH规则。训练案例是Web页面和标注好信息的文档,只需要输入几个典型的训练案例即可,不需要人工标注太多的文档。

归纳的对象是HTML元素,HTML规范决定了几乎所有信息都是放在文本节点(`text()`)中的,故系统要求所有XPATH规则以 `text()` 结尾。归纳的主要依据是元素的属性和位置。基于属性的归纳主要是属性的存在性和属性值的字面特征。仅使用一些简单特征,如 `=` 或 `like` “`prefix *`” 或 `like` “`* suffix`”,也还可以自动生成正则表达式,但这不在文中讨论范围之内。

元素位置特征依赖于 `position()` 函数,主要使用了上下限和同余等价类。

XPATH规则的学习算法以伪代码表示如下:

```
XPATHRule learnXPATHRule(Information inf){
```

将HTML文档转化为XHTML文档;

[可选步骤]滤去所有的 `script`, `style`, `comment`, 因为这些节点很可能不包含信息;

从训练案例集(DOM森林)中找到所有包含 `inf` 的 `text` 节点(`T[i]`),并确定每个 `text` 节点的绝对路径(`P[i]`): `/html/body/...`;

确定所有绝对路径的最大公共子前缀,并生成相应的XPATH规则 `Pm`;

对 `Pm` 上每个节点测试 `Xu` (平凡的 `html` 和 `body` 除外),取对应的包含信息的DOM节点集 `S[j]`;

生成谓词集 `A = A1 ∪ A2`, `A1 = { @attr = "val" | (∀ Sj) Sj.attr = "val" }`, `A2 = { @attr | Sj.attr 存在 }`

求 `S[j]` 的 `position()` 的值域,选择合适的如 `position() > val1` 和 `position() < val2` 加入 `A` 集合

求所有 `S[j]` 的 `position()` 函数的同余等价类,并选择合适的同余类,添加谓词 `position() mod n = r` 到 `A` 集合

求 `S[j]` 每一属性的值,若未生成 `@attr = "val"`,则计算属性所满足的最大前缀和后缀,若非空,则将 `@attr like prefix + "*" + @attr like "*" + suffix` 加入 `A`

将 `A` 中所有的谓词附加到 `Xu` 上

[可选步骤]对 `Pm` 上的每一个谓词,若删去后不会扩大节点选择的范围,则删去

对 `Pm` 添加后缀 `/text()`,表示选择元素节点的文本而非元素

节点本身;

返回 Pm;

!

* 滤去 script:删除所有<script>子树和所有@on...属性。滤去 style:删除所有<style>、<link>子树和所有@style 属性。滤去 comment:删除所有<!-- -->。

* 文献[7,8]介绍了 XPATH 解析技术,文献[8]介绍了 XPATH 语法解析器的构建,以此可以方便地对 XPATH 进行一些扩展,如支持 like 测试和正则表达式测试。

* /html/body 是一切节点的公共子前缀。

* 若属性值与某个字符串相等,则不需要再考

虑前缀/后缀。等价同余类可以强力测试,用从 2 到 N 的自然数整除 $\text{position}()$,取同余的最大的整数 n 和对应的余数 r ,得 $\text{position()} \bmod n = r$ 。施加于同一个 $X[u]$ 的所有谓词应采用 and 连接(如 $li[@name = 'Lily' \text{ and } \text{position()} \bmod 5 = 1]$),而不是串联(如 $li[@name = 'Lily'][\text{position()} \bmod 5 = 1]$)。对命题逻辑, and 与串联等价,但是, $\text{position}()$ 函数超出了命题的范围。可以证明,不包含 $\text{position}()$ 的谓词可用 and 或串联任意连接;但包含 $\text{position}()$ 的谓词只能采用 and 连接,或总是置于第一个[]内。若将 $\text{position}()$ 条件放在后面,则 $\text{position}()$ 的值会随着前面谓词的变化而改变。

* 最后,删除冗余谓词使用了不可回溯的删除法,可以使用其它方法求谓词集的最小有效子集;计算最小有效子集是 NP 难的,这里使用的是贪婪算法的近似解法。

4 实验结果与分析

使用生物数据源,Transfac 基因数据库:

<http://www.gene-regulation.com/cgi-bin/pub/databases/transfac/getTF.cgi?AC=id>

其中, $\text{id} ::= G\{d,6\}$

以 G000120,G000121,G000122,G000123 代入 id,获取 HTML 文档。

其中有一块数据形如图 3 所示。



图 3 Transfac 页面

标记 AC,学习到 XPATH 规则: $\text{/html/body/pre/a}[1]/\text{text}()$

这里 $[1](\text{position}() = 1 \text{ 的缩写})$ 可以作为 a 的选择条件

标记所有 BS,则规则为:

$\text{/html/body/pre/a}[@\text{href} = \text{"pub/databases/transfac/doc/gene1.html\#BS"}]/\text{text}()$

标记 BS 内以 T 开头的链接,则生成规则:

$\text{/html/body/pre/a}[@\text{href} = \text{"pub/databases/transfac/doc/gene1.html\#BS"}]/\text{a}[2]/\text{text}()$

这些 XPATH 规则有较好的信息提取能力,规则也很简单,用于信息提取基本是无错的。

标记 AC 的值,如 G000123,则生成规则:

$\text{/html/body/pre}/\text{text}()$

这条 XPATH 规则性能较差,基本上只能作为文本规则信息提取的“预提取”步骤,而不能单独用于直接提取信息。但若在系统中允许使用 following-sibling 轴,则应该可以生成性能很好的规则: $\text{/html/body/pre/a}[1]/\text{following-sibling::text}()[1]/\text{text}()$

5 结论与展望

使用归纳法可以生成用于 Web 信息提取的 XPATH 规则。这些规则的信息提取能力依赖于页面结构,对基因调控数据库 Transfac 这样的 Web 页面,几乎可以直接用包含 following - sibling 轴的 XPATH 规则提取信息;但 XPATH 规则并不总是如此有效。但无论如何,XPATH 规则总是有用的,因为它可以大大减小文本规则的搜索空间。

这里,基于属性值的测试仅仅使用了简单的字符串相等、前缀、后缀三种简单谓词,也可以考虑正则表达式,若属性符合正则文法,则必定可以构造生成该文法的正则表达式。虽然理论上根据一些语句自动构造可能的正则表达式只是一个数学化的过程,但当前,这类工具通过 Internet 也是很难找到的。然而,可以预置一些常用的正则表达式模板(<http://www.william-long.info/archives/433.html>),如 email 地址、自然数、整数、实数、电话、人名、身份证号、计量单位等,这些是很有用的。JDK 中预置了 regex 包(java.util.regex)用于正则表达式操作,eclipse 插件 regex tester(<http://brosinski.com/regex/>)可以用于可视化的正则表达式匹配。

Web 页面上实际信息之间是有关联的,尤其是位置(position())关联常常非常明显,但在文中为了归纳方法的简单起见,并未考虑 preceding - sibling 和 following - sibling 轴。可以考虑兄弟关系的归纳,尤其是 following - sibling。关于横向连接的解析和优化技术可以参见文献[9]。当然,仅包括两个只向后的轴(即 child 和 following - sibling)的 XPATH 规则可以转化为有穷自动机(DFA 或 NFA)以提高运行效率。可以参见文献[10]。

(上接第 97 页)

消息进行分解,即可取得相应的信息进行应用相关的处理。

4 总结

文中讨论了一个通用整合功能模板的思想。通过对多通道系统语义表示的分析,将一个比较高效的整合模型和算法封装成模板供多通道系统开发者调用,有利于把工作的重点放到与用户界面和应用相关的功能设计上;同时,利用模块化的思想,在其他多通道技术的研究成果基础上对多通道整合模型的改进工作可以只反映到模板上,对使用它的多通道系统则无需改动。此外,这个思想实现为多通道系统中的信息的表示方式和描述方法,以及以此为基础的整合策略提供

参考文献:

- [1] Sahuguet A, Azavant F. Building Light - Weight Wrappers for Legacy Web Data - Sources Using W4F[C]// Proceedings of the 25th International Conference on Very Large Data Bases VLDB '99. [s.l.]: Morgan Kaufmann Publishers Inc, 1999: 738 - 741.
- [2] Liu Ling, Pu Calton, Han Wei. XWRAP: An XML - enabled Wrapper Construction System for WEB Information Source [C]// Data Engineering, 2000. Proceedings. 16th International Conference. [s.l.]: [s.n.], 2000: 611 - 621.
- [3] Baumgartner R, Flehrs S, Gottlob G. Visual Web information Extraction with Lixto[C]// Proceedings of the 27th International Conference on Very Large Data Bases VLDB'01. [s.l.]: Morgan Kaufmann Publishers Inc, 2001: 119 - 128.
- [4] Freitag D. Machine Learning for information extraction in informal domains[J]. Machine Learning, 2000, 39(2 - 3): 169 - 202.
- [5] Califf M E, Mooney R J. Relational Learning of Pattern - Match Rules for Information Extraction[C]// In: Proc. of the Sixteenth National Conf, on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence. Orlando, Florida: [s.n.], 1999: 328 - 334.
- [6] Soderlan S. Learning Information Extraction Rules for Semi - Structured and Free Text[J]. Machine Learning, 1999, 34(1 - 3): 233 - 272.
- [7] 俞 巍. XPath 的两种解析技术[J]. 计算机时代, 2006 (1): 51 - 53.
- [8] 张 昱, 付 雄. 含 XPath 的表达式解析与应用[J]. 小型微型计算机系统, 2004(3): 122 - 126.
- [9] 王 钊, 耿 蓉, 王国仁. XPath 的轴连接查询技术研究[J]. 小型微型计算机系统, 2005(11): 72 - 77.
- [10] 王 强, 武港山. 对 XPath 模式定位能力的扩充[J]. 计算机研究与发展, 2001(6): 35 - 39.

了一个良好的研究背景。

参考文献:

- [1] Coutz J, Nigay L, Salber D. Agent - Based Architecture Modelling for Interactive Engineering Systems[M]. London: Critical Issues in User Interface, 1995: 191 - 209.
- [2] 俸 文. 多通道人机交互技术的研究[D]. 南京: 南京理工大学, 2004.
- [3] 普建涛, 董士海. 任务制导的多通道分层整合模型及其算法[J]. 计算机研究与发展, 2001, 38(8): 966 - 967.
- [4] 张宏超, 俸 文, 周 方, 等. 多通道整合的相关问题及算法[J]. 计算机工程, 2004, 30(13): 67 - 69.
- [5] 董士海. 人机交互和多通道用户界面[M]. 北京: 科学出版社, 1999.