基于视觉特征和领域本体的Web信息抽取

张 鑫,陈 梅,王翰虎,王嫣然
(贵州大学 计算机科学与信息学院,贵州 贵阳 550025)

摘 要:为了解决网页信息的自动抽取,该文提出了一种基于视觉特征和领域本体的Web信息抽取算法。该算法以基于领域本体的信息抽取为基础,根据网页的视觉特征来准确划定信息抽取区域,然后结合DOM树技术和抽取路径的启发式学习,获得Web页面中信息项的抽取路径。通过信息项的抽取路径自动生成信息项的领域本体,通过信息项的领域本体解析出信息项的抽取规则。使用本算法来进行Web信息的抽取,具有查全率与查准率高、时间复杂度低、用户负担较轻和自动化程度高的特点。

关键词:视觉特征;领域本体;Web信息抽取;路径学习;启发式学习
中图分类号:TP391.4 **文献标识码**:A **文章编号**:1673-629X(2011)02-0058-04

Visual Features and Domain Ontology-Based Web Information Extraction

ZHANG Xin, CHEN Mei, WANG Han-hu, WANG Yan-ran
(College of Computer Science and Information, Guizhou University, Guiyang 550025, China)

Abstract: Put forward a Web information extraction algorithm based on visual features and domain ontology in order to solve the problem of Web information automatic extraction. This algorithm is on base of domain ontology-based Web page information extraction, according to the visual characteristics of the sample Web page to accurately delineated the area of information extraction, and get the Web page information item extraction path by combining DOM tree technology and extraction path heuristic learning. Through the domain ontology which is automatically generated by the extraction path, get the extraction rules of the information items. Using this algorithm for Web information extraction has many advantages, such as higher recall and precision rate, lower time complexity, lighter user burden and higher degree of automation.

Key words: visual features; domain ontology; Web information extraction; path learning; discovery learning

0 引 言

随着计算机技术的迅速发展,互联网上的信息正在以指数形式增长,面对如此巨大的信息资源,人们迫切需要一些自动化的工具从海量的信息中迅速找到自己感兴趣的内容,以便对这些信息进行查询和再利用[1]。Web信息抽取(Information Extraction)正是在这种背景下应运而生。目前存在的信息自动抽取方法中,基于网页结构特征分析的信息抽取方法的查全率较高,但是这种抽取方法具有一定的盲目性,会抽取出大量的冗余信息,查准率不高;基于本体的信息抽取方法是以构建的领域本体为基础生成抽取规则,然后根据抽取规则来进行信息抽取,这种方法有较高的查全率和查准率,但是在领域本体的构建过程中需要有领域专家的参与,过程复杂,周期较长。基于路径学习的信息自动抽取方法是将网页结构分析与归纳学习相结合,通过学习待抽取信息在网页分析树中的路径来实现信息的自动抽取,该方法具有抽取速度快、用户负担较轻的特点,但是查全率和查准率不高[2-4]。而其余的信息自动抽取方法如基于知识库的信息抽取方法,基于文法分析的信息抽取方法和基于隐式马尔科夫模型的信息抽取方法,也都难以同时满足网页信息自动抽取中查全率与查准率高、抽取速度快和用户负担轻的要求。

文中提出了一种基于视觉特征和领域本体的Web信息自动抽取的方法,在该方法中首先通过网页的视觉特征来指导网页信息的区域划分,经过抽取路径的学习后来构建领域本体,通过构建的领域本体来解析出信息项的抽取规则,来实现信息的自动抽取。

收稿日期:2010-07-10;**修回日期**:2010-10-15
基金项目:贵州省2008年省级信息化专项基金项目(0830);贵州省科技计划工业攻关基金项目(黔科合GY字[2008]3035)
作者简介:张 鑫(1985-),男,山西怀仁人,硕士研究生,研究方向为信息安全;陈 梅,副教授,硕士生导师,研究方向为数据库技术与软件工程;王翰虎,教授,硕士生导师,研究方向为数据库系统、分布式系统、面向对象方法。

并采用该方法编制了一个科技文献信息自动抽取系统,并通过该系统对万方数据源中的科技文献进行抽取来进行测试,测试结果表明,该方法能有效地抽取出网页中的信息。

1　基于领域本体的 Web 信息抽取算法

1.1　基于领域本体的 Web 信息抽取算法概述

基于领域本体的 Web 信息抽取的基本方法是首先通过分析样本页面的结构特征和要抽取信息项的概念、类型和值,来构建用户所需信息项的本体描述。通过构建好的本体来生成信息抽取规则,然后通过信息抽取规则来对 Web 页面进行信息抽取。其抽取过程如图 1 所示。

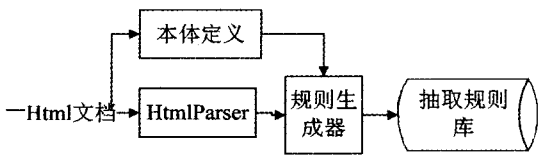

图 1　基于领域本体的 Web 信息抽取算法

1.2　基于领域本体的 Web 信息抽取算法存在的问题

通过对该算法的分析发现其存在的问题主要有以下几项:

(1)在领域本体的构建中,需要有专门的领域专家参加,而且构建过程复杂、耗时、耗力,自动化程度不高。

(2)在实际的 Web 页面中除了含有要抽取的信息,还含有不需要抽取的信息,在对这些网页进行信息抽取时,首先要做的就是如何区分有用和无用的信息区域。在基于领域本体的 Web 信息抽取算法中,首先通过待抽取信息在页面中出现的顺序来划分抽取区域,然后将待抽取信息位置相连的信息项划分为一个区域。而在实际的 Web 页面中,待抽取信息区域和无用信息区域并不是按照固定的页面信息项出现的顺序排列的。因此,通过这种方法来进行区域划分构建出的抽取规则就不准确,信息抽取的准确率也会受到影响。

2　基于视觉特征和领域本体的 Web 信息抽取算法

Web 页面中有很大一部分是数据导向型页面,在这类页面中,通常是将后台数据库中的内容直接导入到已经设计好的模板中,然后通过 HTML 生成树上某个或某几个具有特定视觉特征的嵌套容器来呈现给用户。在这些页面中,用户感兴趣的信息,也存在于这些信息块集合当中。在这些嵌套的容器中具有这样的特征:相同结构的信息块其数据容器基本相同;相同结构的信息块在 DOM 树中处于同一层。因此根据这些特征可以首先通过数据容器的视觉特征来进行数据区域的划分来得到其数据区域树,然后通过启发式学习来从这些结构相似的数据区域树中得到信息项的抽取路径。通过得到的抽取路径来自动构建其领域本体,从而通过对领域本体的解析便得到了信息项的抽取规则。所以基于视觉特征和领域本体的 Web 信息抽取算法主要包括三个部分:抽取区域选择,信息项抽取路径的学习和信息项领域本体和抽取规则的构建。

2.1　Web 页面信息领域本体的定义

基于领域本体的 Web 信息抽取的基本方法是首先通过分析样本页面的结构特征和要抽取信息项的概念、类型和值,来构建用户所需信息项的本体描述,这样就完成了信息项的本体描述[5,6]。在整个信息抽取过程中,本体的构建是整个抽取过程的基础和核心。在文中 Web 页面中待抽取的信息项 A 的领域本体的定义如下所示:

```
Concept ContentA
Super: {PreA};
Type: String;
End Content
```

其中 PreaA 表示待抽取信息项的前导符,ContentA_Value 表示待抽取的信息项 ContentA 的值。

2.2　基于视觉特征的抽取区域选择算法

以万方系统中查询结果显示页面为例,通过分析发现网页中待抽取信息主要有以下几个的视觉特征[7,8]:

(1)网页中待抽取的信息都是通过一个<ul style="visibility:visible;" class="list_ul">
标签来呈现给用户的。

(2)文献的标题是通过一个<li class="title_li">
标签来呈现给用户的。

(3)文献的收录情况是通过一个<li class="green-color">
标签来呈现给用户的。

(4)文献的摘要是通过<li class="zi">
标签来呈现给用户的。

通过这些视觉特征便可以快速地确定待抽取区域和去掉无用信息,其抽取区域选择算法如下所示:

输入:样本页面的 DOM 解析树

输出:待抽取区域的 DOM 子树集 CDOM

{ IBPATH = null;

CDOM = null;

通过先序的方法遍历 DOM,将得到的路径记入到 IBPATH 中;

Number = IBPATH 中的路径总数;

for i = 1 to Number

{ 获取 IBPATH 中的第 i 条路径;

从该路径的第一个标签开始和视觉特征 1 进行比较;

if(标签和视觉特征 1 相符合)

{将以该标签为树根的子树添加到 CDOM 中}

}

return CDOM;}

2.3 信息项抽取路径的学习算法

本算法使用启发式搜索,生成从待抽取区域的 DOM 子树的树根到信息项节点的路径,简称信息项抽取路径。信息项路径由若干个路径节点组成,每一个路径节点代表 DOM 子树中的一个节点,每条路径表示了一个信息项的抽取路径,每个 DOM 子树的叶子节点表示要抽取的信息项[9,10]。

输入:待抽取区域的 DOM 子树集 CDOM,要抽取的信息项(标题,收录情况,摘要);

输出:信息项(标题,收录情况,摘要)抽取路径

{EXIBPATH= null;

计算 CDOM 中树的总个数,并计入到 treecount 中;

for i=1 to treecount

{获取 CDOM 中的第 i 个子树;

Path=null;

首先通过先序遍历的方法遍历 CDOM,并将得到的路径表达式添加到 Path 中;

count=Path 中的路径总数;

for j=1 to count

{将 Path 的值赋值给 treePath[i][j];}

}

计算 treePath 中的行数和列数,分别计入到 RPathCount 和 CPathCount 中;

for j=1 to CPathCount

{

获取 treePath 中的第 i 列,//即获得 CDOM 中全部子树先序遍历时的第 j 条路径;

for i= 1 to RPathCount

{获取 treePath 的第 i 行、第 j 列,//即获得 CDOM 中第 i 个子树,经过先序遍历的所有路径集合;

for k=i+1 to RPathCount

{从树根开始比较 treePath[i][j]和 treePath[k][j];

if (2 条路径节点的节点标签和节点索引值相同)

{将该路径写入 IBPATH; }

}}}

计算 EXIBPATH 中的路径总数,计入到 IBPATHCount 中;

for i=1 to IBPATHCount

{

遍历第 i 条路径,将路径中的标签节点和视觉特征 2,3,4 相比较,

if(如果该路径中存在和视觉特征一样的标签)

{该路径保存到 EXIBPATH 中}

else if(该路径不存在和视觉特征一样的标签)

{该路径不是信息项的抽取路径,继续比较其他路径}}

return EXIBPATH;}

2.4 抽取规则的构建算法

在 Web 页面信息项领域本体定义中描述了待抽取信息项的名称、实例值、前导符及其后导符,并且这些描述是按照它们在 Web 页面中出现的顺序给出的;通过这种描述,Web 页面的信息被解析成为"伪"本体,根据 Web 页面领域本体定义中的前导符、后导符和数据类型来构建待抽取信息的抽取规则,其中前导符为正则表达式的开始符号,后导符为正则表达式的结束符号。例如通过对领域本体 ContentA 的解析可以构建出信息项 ContentA 的正则表达式为(PreA)(.*?)(/Prea),Web 页面抽取器便能通过该正则表达式从 Web 页面中抽取出需要的信息项 ContentA[9]。该算法首先通过对信息项抽取路径的学习得到了 EXIBPATH(信息项抽取路径),在 EXIBPATH 中顺序记录着在网页中要抽取的信息项的抽取路径,算法将这些抽取路径转化为待抽取信息项的本体描述,然后根据本体描述来构建信息项的抽取规则[11,12]。其算法的主要过程如下所示:

输入:抽取路径集合 EXIBPATH

输出:抽取规则集 EXRegex

{//信息项领域本体的构建

Ontology[i][4]=null;//本体集初始化,其中 i 表示要抽取的信息项的个数,j 表示本体定义中的 Concept,super,type,value 和 follow 这些保留字,其中 j=1 表示为 Concept 的保留字,j=2 表示为 super 的保留字,j=3 表示 type 的保留字,j=4 表示抽取路径中最后一个标签节点的名称。

计算 EXIBPATH 中所有的信息项抽取路径总数,计入到 count 中;

for i=1 to count

{遍历 EXIBPATH 中的第 i 条路径;

Ontology[i][1]=第 i 条路径所要抽取的信息项;

Ontology[i][2]=第 i 条路径中所有的标签节点;

Ontology[i][3]=第 i 条路径中所抽取信息项的值类型;

Ontology[i][4]=第 i 条路径中最后一个标签节

点。}

//抽取规则的构建,其中 EXRegex[i][1]用来表示第 i 条规则要抽取的信息项的名称,EXRegex[i][2]表示第 i 条规则要抽取的信息项的抽取规则。

计算领域本体 Ontology 中的本体个数,并记入到 count 中;

```
for i=1 to count
{ EXRegex[i][1]= Ontology[i][1];
EXRegex[i][2]=( Ontology[i][2])(.*?)(/Ontology[i][4]);}
return EXRegex;
}
```

3　实验结果

选用万方数据源的网页实例作为测试的信息源,以"信息安全"为查询条件查询出的所有文献信息数据为实验对象。实验过程只抽取标题、收录情况和摘要信息这三个信息项。在不对信息抽取区域划分改进的情况下建立页面信息项本体,构建抽取学习规则,进行实验。然后通过本算法对信息抽取区域进行划分的情况下,对每个抽取区域进行路径分析,获得信息项的抽取路径,从而构建信息项的抽取规则,来进行 Web 信息抽取。然后通过计算两个算法的抽准率和抽全率来进行对比分析。

3.1　抽准率的比较

由图2可以看出随着抽取信息条数的增加,基于领域本体的信息抽取算法和本算法的抽准增量开始缓步减少,抽准率逐步向稳定值逼进。由于本算法首先对抽取区域进行了划分,所以整个抽取过程中的抽准率要比原算法提高很多。

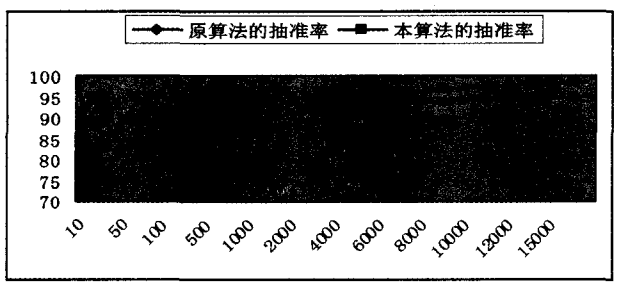

图2　抽准率的比较图

3.2　抽全率的比较

由图3可以看出随着抽取信息量的递增,两个算法的抽全率都在递减,当数据量达到20000条以上时,抽全率逐步达到一个稳定值,两者的差别不大,从整体来看本算法的抽全率要略高于原算法。

从以上的分析可以发现,和基于本体的 Web 信息抽取算法相比,本算法具有更高的抽准率和抽全率,并且在抽取的过程中本算法的自动化程度较高,并不需要领域专家的参与,大大提高了信息抽取的效率。

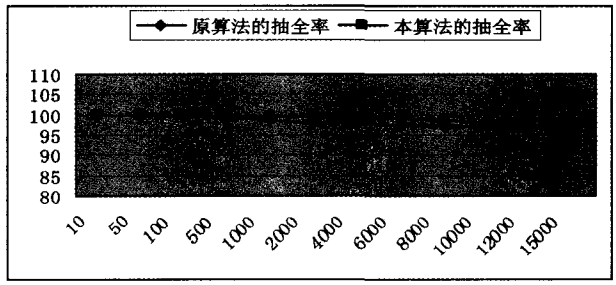

图3　抽全率的比较图

4　结束语

文中提出了一个基于视觉特征和领域本体的 Web 信息抽取算法,该算法是在基于本体的 Web 页面信息抽取算法的基础上,首先根据网页的视觉特征,对样本页面中待抽取信息进行区域划分,然后结合 DOM 树技术和抽取路径的启发式学习得到了 Web 页面的抽取规则。

使用本算法来进行 Web 信息的抽取,具有查全率与查准率高、时间复杂度低、用户负担较轻等特点,能够很好地满足网页信息自动抽取的要求。但是本算法还是存在着一些不足之处,例如该方法适用于页面结构变化不是很大的 Web 页面,对于页面变化较大的情况其信息抽取的准确率还有待提高。

参考文献:

[1] 邓志鸿,唐世渭,张　铭,等. Ontology 研究综述[J]. 北京大学学报:自然科学版,2002,38(5):730-738.

[2] Berners L T. The Semantic Web[J]. Scientific American, 2001,284(5):34-43.

[3] Church K W, Mercer R L. Introduction to the Special Issue on Computational Linguistics Using Large Corpora[J]. Computational Linguistics,1999,19(1):12-24.

[4] Li Chaoguang, Zhang Ming, Deng Zhihong, et al. Automatic metadata extraction for scientific documents [J]. Computer Engineering and Application,2002,38(21):189-191.

[5] 刘　耀, 穗志程. 领域 Ontology 概念描述体系构建方法探析[J]. 大学图书馆学报, 2006, 24(5):28-33.

[6] 周明健, 高　济,李　飞. 基于本体论的 Web 信息抽取[J]. 计算机辅助设计与图形学学报, 2004,16(4):535-541.

[7] 王　放,顾　宁,吴国文. 基于本体的 Web 表格信息抽取[J]. 小型微型计算机,2003,24(12):2142-2146.

[8] 荆　涛,左万利. 基于可视布局信息的网页噪音去除算法[J]. 华南理工大学学报,2004,32(S1):84-87.

[9] 于　琨,蔡　智,糜仲春,等. 基于路径学习的信息自动抽取方法[J]. 小型微型计算机系统, 2003, 24(12):2147-2149.

(下转第65页)

表 2　标准数据的聚类表

算法	原算法(图 3)	改进算法(图 4)
迭代 100 次聚类结果	加号 13 圆形 84 五角星 24	加号 31 圆形 33 五角星 50
CPU 用时(秒)	至少迭代 3 次 0.0310	迭代 1 次 0.0160

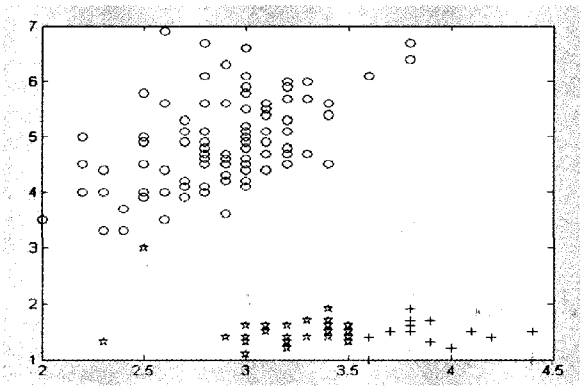

图 3　原算法 121 个数据聚类图

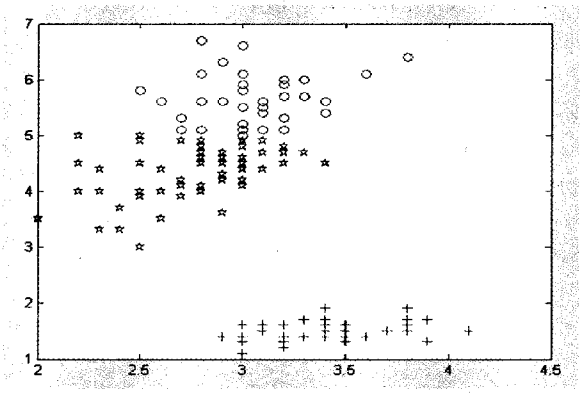

图 4　改进算法排除 7 个孤立点后聚类图

从聚类图和聚类表可以看出,对于标准数据来说,由于分布比较均匀,改进算法迭代次数明显减少,执行时间明显减少。

4　结束语

K-Means 算法作为一种常用的聚类算法,对球状分布的数据具有很好的效果,但是算法对初始聚类中心敏感,容易受到孤立点的影响。文中在聚类之前排除了孤立点的影响,提出了一种新的选取初始聚类中心的方法。

实验结果表明,改进算法更接近实际数据分布。虽然需要查找少量孤立点,会增加时间消耗,但是改进算法准确度较高,聚类效果较好。

参考文献:

[1] 周卫星,廖　欢. 基于 K 均值聚类和概率松弛法的图像区域分割[J]. 计算机技术与发展,2010,20(2):68-70.

[2] Mac Q J. Some methods for classification and analysis of multivariate observations [C]// In: Proc. 5th Berkeley Symposium in Mathematics. Berkeley, USA: Univ of California, 1967.

[3] GUHA S, RASTOGI R, SHIM K. CURE: An efficient clustering algorithm for large databases [C]// Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 1998: 73-84.

[4] Ester, Martin, Hans Peter Kriegel, et al. A Density Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise[C]//Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96). Ortland, Oregon: [s. n.], 1996.

[5] Wang W, Yang J, Muntz R. STING: A Statistical Information Grid Approach to Spatial Data Mining[C]//Proc. of 1997 Intl. Conf. on Very Large Databases. Athens, Greece: [s. n.], 1997:186-195.

[6] Kohonen T. Self — organized Formation of Topologically Correct Feature Maps[J]. Biological Cybernetics, 1982, 43(1): 59-69.

[7] 朱　明. 数据挖掘[M]. 合肥:中国科学技术大学出版社, 2002.

[8] 李业丽,秦　臻. 一种改进的 k-means 算法[J]. 北京印刷学院学报,2007,15(2):63-65.

[9] 苏锦旗,薛惠锋,詹海亮. 基于划分的 K-均值初始聚类中心优化算法[J]. 微电子学与计算机,2009,26(1):8-11.

[10] 步媛媛,关忠仁. 基于 K-means 聚类算法的研究[J]. 西南民族大学学报:自然科学版,2009,35(1):198-200.

[11] 连凤娜,吴锦林,唐　琦. 一种改进的 K-means 聚类算法[J]. 电脑与信息技术,2008,16(1):38-40.

[12] 陆声链,林士敏. 基于距离的孤立点检测研究[J]. 计算机工程与应用,2004(33):73-75.

(上接第 61 页)

[10] 张　程,陈自郁,古　平,等. 基于 DOM 树结构的 Blog 网页自动识别[J]. 计算机应用研究,2008(5):1489-1491.

[11] 王洪伟,吴家春,蒋　馥. 基于描述逻辑的本体模型研究[J]. 系统工程,2003,21(3):101-107.

[12] 郑冬冬,赵朋朋. Deep Web 爬虫研究与设计[J]. 清华大学学报(自然科学版),2005,45(S1):1896-1902.