一种基于 SVM 和 AdaBoost 的 Web 实体信息抽取方法

孙 明[1] 陆春生[2] 徐秀星[1] 李庆忠[1] 彭朝晖[1]
[1](山东大学计算机科学与技术学院 山东 济南 250101)
[2](中国人力资源和社会保障部信息中心 北京 100716)

摘 要 提出一种基于 SVM 和 AdaBoost 的 Web 实体信息抽取方法。首先提出一种基于 SVM 的 Web 页面主数据区域识别方法,基于 Web 实体实例在页面中的展示特征,有效地将 Web 页面进行数据区域分割,识别出 Web 实体实例所在的主数据区域;然后基于 Web 实体属性标签的特征,提出一种基于 AdaBoost 的集成学习方法,从页面的主数据区域自动地抽取 Web 实体信息。在两个真实数据集上进行实验,并与相关研究工作进行比较,实验结果说明该方法能够取得良好的抽取效果。

关键词 Web 信息抽取 页面分割 集成学习

中图分类号 TP311 **文献标识码** A DOI:10.3969/j.issn.1000-386x.2013.04.028

A WEB ENTITY INFORMATION EXTRACTION METHOD BASED ON SVM AND ADABOOST

Sun Ming[1] Lu Chunsheng[2] Xu Xiuxing[1] Li Qingzhong[1] Peng Zhaohui[1]
[1](*School of Computer Science and Technology, Shandong University, Jinan 250101, Shandong, China*)
[2](*Information Center, Ministry of Human Resources and Social Security of China, Beijing 100716, China*)

Abstract In this paper, a Web entity information extraction method based on SVM and AdaBoost is proposed. Firstly, an identification method for Web page's main data region based on SVM is proposed, which segments Web page data region effectively based on the display characteristics of Web entity instances in the page, identifies the main data area where the Web entity instances locates. Secondly, based on the characteristics of the Web entity attribute labels, a method based on AdaBoost ensemble learning is proposed, which automatically extracts the Web entities information from the main data area of the page. A variety of experiments are conducted on two real data sets, and the comparison is done with correlated research works as well, experimental results show that this method is able to achieve fairly good extraction effect.

Keywords Web information extraction Page segmentation Ensemble learning

0 引 言

随着 WWW 的快速发展,Web 上信息量以爆炸性的趋势增长,从而使 Web 成为一个巨大的、分布广泛的信息源。Web 网页中蕴含了各个领域的大量有价值的信息。通过 Web 信息抽取技术,对 Web 页面数据进行理解,将来自不同网站的 Web 信息进行有效地抽取与集成,能够进一步提高互联网上信息的利用率,提高数据的应用价值。

Web 实体是一类由多个数据元素及数据属性标签按照特定模式组织在一起的数据对象。Web 实体信息抽取是 Web 信息抽取的主要目标,Web 实体信息抽取为 Web 信息抽取与分析的其他环节,包括重复记录检测、Web 实体间联系发现等,奠定技术基础。因此,Web 实体信息抽取问题的研究具有重要的理论和应用价值。

很多研究学者和研究机构对 Web 实体抽取方法开展了大量的研究工作,但是现有的多数方法主要针对单个网站进行处理,对于大规模 Web 信息抽取来说,Web 实体信息自动化抽取的效果不够理想。研究表明,通过对目标网站上的训练页面进行语义标注,为识别出的每个数据元素分配一个有意义的标签来表示该数据元素的语义,可以方便地得到目标网站的训练样例,结合现有的包装器生成技术,可以自动地为目标网站生成包装器,完成实体信息抽取的任务[1]。

为了适应大规模 Web 实体信息抽取的需要,本文提出一种基于 SVM 和 AdaBoost 集成学习的 Web 实体信息抽取方法,首先提出一种基于 SVM 的 Web 页面主数据区域识别方法,基于 Web 实体实例在页面中的展示特征,有效地将半结构化及非结构化页面进行数据区域分割,识别出 Web 实体实例所在的主数据区域,然后基于 Web 实体属性标签的特征,提出一种基于 AdaBoost 的集成学习方法,从页面的主数据区域自动地抽取 Web 实体的属性信息。本文在两个真实数据集上进行了实验,并与相关研究工作进行了比较,实验结果说明该方法能够取得良好的抽取效果。

1 相关研究

Web 实体信息抽取方面已有较多的相关研究工作,根据抽取方法的自动化程度[2,3],Web 实体抽取可以分为手工构造、半

收稿日期:2012 - 04 - 18。国家科技支撑计划项目(2008BAH32B01)。**孙明**,副教授,主研领域:Web 数据集成与分析,SaaS 数据管理。**陆春生**,研究员。**徐秀星**,硕士。**李庆忠**,教授。**彭朝晖**,副教授。

自动抽取和全自动抽取等三类方法。

1.1 手工构造的抽取方法

手工构造的抽取方法需要为每个待抽取的数据源编写用于抽取实体数据的包装器,包装器是由一系列抽取规则及其应用程序组成[4]。因此使用这类方法需要具有较深的计算机专业背景,对一般人员来说此方法并不适用。典型的方法有 TSIMMIS[5], Web-OQL[6], W4F[7]。TSIMMIS 是最早提出手工构建包装器框架的方法之一,它的主要组件是一个包装器,其中有一个说明性文件,用于定义页面中所要抽取的 Web 数据在页面中的位置以及如何将这些数据"打包"成对象。Web-OQL 是一种 Web 数据查询语言模型,能够实现 Web 数据查询、半结构化数据查询及 Web 站点重构等功能。W4F 为用户提供向导,使用 HEL 语言编写抽取规则,使用内部定义的 NSL 格式存储抽取出的数据。

手工构造的 Web 数据抽取方法能够达到较高的准确率,但是抽取过程需要较多的人工参与,并且需要用户有较深的专业背景,通常不太适合于大规模 Web 实体抽取。

1.2 半自动的数据抽取方法

半自动的数据抽取方法,代表方法主要有 SRV[8], RAPIER[9], WIEN[10], LIXTO[11,12]等。SRV 是一种自上而下的关系型数据的抽取方法,将一组标记了待抽取实例的页面作为输入,输出的是抽取规则,SRV 的规则有较强的表达能力,且不需进行语法分析。RAPIER 将半结构化文本作为输入,然后学习出抽取规则。WIEN 是一个辅助包装器生成的系统,它主要抽取页面中表格里的结构化数据,样例文档通过启发式规则进行自动标注,标注后的文档通过归纳算法生成包装器,该方法学习和抽取数据的速度快,但不能处理排列顺序的改变及缺失内容,并需要对整个页面进行标注。LIXTO 系统所需的训练数据需要用户进行手工标注,然后根据训练数据学习出抽取规则,最后根据获得的抽取规则进行数据抽取,LIXTO 对抽取规则的定义是使用 Datalog[13]的 Elog 语言,该语言的实现以及优化都很困难,而且也不容易理解,使用起来较为困难。

总体来说,半自动数据抽取方法一般都具有较高的抽取准确度,但是半自动的 Web 数据抽取方法需要用户的手工参与,主要是要进行手工标注训练集。

1.3 全自动的抽取方法

全自动的抽取方法不需要用户的参与,不需要进行手工标注就可以生成抽取规则。典型的方法有文献[14 - 21]提出的方法。

RoadRunner 通过归纳总结多个页面结构的特征来生成一个能用正则表达式表示的页面的模板,然后利用这个模板抽取其他的同类型的页面。该方法实现全自动的数据抽取,但是由于抽取的数据有可能包含噪音数据,准确率相对较低。ViNTs 通过利用搜索引擎返回的结果页面的视觉特征来总结内容的规律,然后结合 HTML 的标记结构进行抽取。Omini 及 Embley 方法首先将页面解析为 DOM 树,然后通过运用启发式规则来识别记录边界。

胡东东等提出了一种基于语义块结构模型的数据抽取方法,Web 数据主要存在于页面的语义块中,对于同一数据源的页面,页面间的区别主要在于语义块间的区别。杨少华等提出了模板检测问题,基于模板产生的网页的结构特征,提出了一种模板检测方法,对于由模板生成的 Web 页面,抽象出其模板从而全自动的抽取页面中的数据。对于从 Deep Web 页面中抽取结构化数据记录的问题,刘伟等提出了一种完全基于页面的视觉信息的方法来实现自动的数据抽取,首先提出 Deep Web 页面中的若干有效视觉特征,然后基于这些特征,把 Deep Web 页面转换为一棵视觉树,最后在这棵树上实现了对数据记录和数据项的自动抽取。丁艳辉等提出基于集成学习和二维关联边条件随机场的 Web 数据语义标注方法,首先利用已抽取的信息和目标网站训练页面中呈现的特征构造多个分类器,使用 Dempster 合成法则合并分类器结果,区分训练页面中的属性标签和数据元素,然后利用二维关联边条件随机场模型对 Web 数据元素间的长距离依赖联系和短距离依赖联系进行建模,实现数据元素的自动语义标注。

全自动的 Web 实体信息抽取方法自动化程度最高,抽取过程不需要用户的手工参与,但是对于一些结构复杂的页面,其抽取准确度不高。针对这种情况,本文提出一种基于 SVM 和 AdaBoost 的 Web 实体信息抽取方法,与相关研究工作的思想类似,本文方法也是利用 Web 实体信息在页面的特征来构造抽取规则,但是同时提出利用已有的 Web 实体全局模式来指导 Web 实体信息的抽取,实验表明本文方法能从较大程度上提高查准率、查全率。

2 基于 SVM 的 Web 页面主数据区域识别

根据网页中的数据与主题的相关性,可以将网页分为数据区域及非数据区域。数据区域包含与用户指定的主题相关的信息,非数据区域包含导航信息等与用户指定的主题无关的信息。在本文中,主数据区域是一组与用户指定的 Web 实体类型相关的数据区域。图 1 所示部分即为网页的主数据区域。

图 1 Web 页面主数据区域示例

在抽取 Web 实体信息前,准确地识别出待抽取页面的主数据区域,有助于提高 Web 实体抽取的准确度与效率。

2.1 方法框架

本节提出一种基于 SVM 的 Web 页面主数据区域识别方法,首先将 Web 页面利用分割规则划分为若干数据区域,然后通过对主数据区域特征的学习获得 SVM 分类器,最后使用 SVM 分类器对 Web 页面的数据区域进行识别以判定出页面的主数据区域。

方法的算法描述如下:

Algorithm 1: Web 页面主数据区域识别算法
Input: 已爬取的 Web 页面集合{Page},全局模式 GS
Output: 主数据区域集合{$Region_M$}
1: 将 Web 页面{Page}解析为 DOM 树:
{DOM} = HTMLParse({Page});
2: 根据规则将 DOM 树分割为若干数据区域:
{Region} = GetRegions({DOM});
3: 生成 SVM 训练集 td:

```
    td = GenerateTrainingDataSet({Region}, GS);
4:  由训练集构造 SVM 分类器 CM:
    CM = GenerateCM(td);
5:  for each Page in {Page} do
6:    DOM = HTMLParse(Page);
7:    {Region} = GetRegions(DOM);
8:    Region_M = GenerateMainRegion({Region}, CM);
9:    put Region_M into {Region_M};
10:  end for
11:  output {Region_M};
```

2.2 Web 页面解析

将 Web 页面解析为一棵 HTML 标记树 DOM,相关定义如下:

定义 1　同一层的一个或相邻的若干个标记节点所构成个节点集合为一个数据块。形式化表示为数据块 $Block = \{n_1, n_2, \cdots, n_i\}, i>0$,其中 n_i 为一个标记节点,并且 $Parent(n_1) = Parent(n_2) = \cdots = Parent(n_i)$。

定义 2　同一层的一个或相邻的若干个数据块所构成的数据块集合为一个数据区域,即 $Region = \{Block_1, Block_2, \cdots, Block_i\}, i>0$。

定义 3　主数据区域是指与目标 Web 实体相关的若干个数据区域所构成的集合,即 $Region_M = \{Region_1, Region_2, \cdots, Region_i\}, i>0$。

基于上述的定义,Web 页面主数据区域的识别可描述为在一棵 HTML 标记树 DOM 中找出与目标 Web 实体相关的数据区域所构成的主数据区域 $Region_M = \{Region\}$。

2.3 Web 页面数据区域分割

Web 页面数据区域分割是指根据一定的规则对将 Web 页面的 HTML 解析树分割成数据区域集合 $\{Region\}$。

当前 Web 页面主要使用 DIV, TABLE 等标记来展现相关的信息,因此为了获得更好的数据区域(这里指包含更多的相关的数据并且包含的数据噪声更小),在数据区域分割过程中定义以下几种分割规则:

规则 1　数据区域的分割从 DOM 的根节点开始进行;

规则 2　每个划分的数据区域必须包含一定的文本信息;

规则 3　对于 DIV 标记节点 $node_{div}$,如果其子节点集合不包含 DIV 节点,则节点 $node_{div}$ 分割为一个数据区域;否则,$node_{div}$ 子节点集合中连续的相同或相似的节点分割为一个数据区域,DIV 节点继续分割;

规则 4　将每个 TABLE 标记节点 $node_t$ 分割为一个数据区域;

规则 5　对于其他标记节点 $node_{others}$,如果该节点的子节点集合包含叶子节点,则将该节点分割为一个数据区域,否则继续分割其子节点。

分割过程可由如下算法描述:

```
Algorithm 2: Web 页面数据区域分割算法
Input: Web Page
Output: 数据区域集合 R = {Region}
1:   使用 HTML 解析器将 Web 页面解析为 DOM 树,并去除无用节点:
     DOM = HTMLParse(Web Page);
2:   初始化待分割节点队列 Node 及数据区域集合 R:
3:   Node = {root},root 为 DOM 根节点;
4:   R = {};
5:   While(Node 非空) do
6:     node = FirstNode(Node);
7:     If(node 不符合规则 2) then
8:       从 Node 中移除 node;
9:       Continue;
10:    If(node 为 node_t 节点) then
11:      把 node 放入 R;
12:      从 Node 中移除 node;
13:    Else if(node 为 node_div 节点) then
14:      If(node 不包含 node_div 节点) then
15:        把 node 放入 R;
16:        从 Node 中移除 node;
17:      Else then
18:        把 Block = {n1,n2,n3,…,ni} 放入 R, ni is not node_div;
19:        从 Node 中移除 node;
20:        把 node_div 放入 Node;
21:      End if
22:    Else if(node 为 node_others 节点) then
23:      If(node_others contains leaf node)
24:        把 node 放入 Node;
25:        从 Node 中移除 node;
26:      Else
27:        把 node 子节点放入 Node;
28:        从 Node 中移除 node;
29:      End if
30:    End if
31:  End while
32:  Return R;
```

其中 Block 为连续的相同或相似的节点集合所构成的数据块。

2.4 SVM 分类器构建

根据对 Web 页面分割获得的数据区域 $\{Region\}$ 及已有的 Web 实体全局模式信息生成训练集 *td*,并以此训练集来构建 SVM 分类器 CM。

根据 Web 页面及目标数据区域的特点,本文选取若干个重要的特征,以 5 元组的形式表示为 $F = \{text, link, attr, len, nodes\}$,特征的详细描述如表 1 所示。

表 1　Web 页面特征描述

特征	含义	描述
text	文本大小	数据区域包含的文本数
link	超级链接数	数据区域包含的超级链接数
attr	Web 实体属性标签数	数据区域包含的 GS 中的属性标签数
len	路径长度	数据区域的路径长度
nodes	子孙节点数	数据区域包含的所有子孙节点数

基于特征 F 的特征向量 $V = (v_{text}, v_{link}, v_{attr}, v_{len}, v_{nodes})$ 的计算方法如下:

(1) $v_{text} = Region_{text}/DOM_{text}$,即数据区域的文本数与 Web 页面文本数的比值;

(2) $v_{link} = Region_{link}/DOM_{link}$,即数据区域的链接数与 Web 页面链接数的比值;

(3) $v_{attr} = Region_{attr}$,即数据区域包含的属性标签数;

(4) $v_{len} = Region_{len}/AVG_{len}$,即数据区域的路径长度与 DOM 节点的平均路径长度的比值;

(5) $v_{nodes} = Region_{nodes}/DOM_{nodes}$,即数据区域包含的节点数

与 DOM 包含的节点数的比值。

基于机器学习方法来学习分类器,首先需要构造合适的训练集,然后将训练集输入到 SVM 学习出分类器。分类器构造方法如下所示:

(1) 输入:页面 DOM 树结构,已有的全局模式 GS,页面的数据区域{Region};

(2) 根据特征向量计算方法计算 DOM_{text}, DOM_{link}, GS_{attr}, AVG_{len}, DOM_{nodes};

(3) 对于数据区域集合{*Region*}计算其中的每个数据区域的特征向量 V;

(4) 将所有的特征向量写入训练集文件,并进行手工标注;

(5) 将标注过的训练集输入到 SVM,学习出分类器模型 CM。

2.5 主数据区域识别

由 SVM 学习出的分类模型 CM 后,将 Web 页面的数据区域集合{*Region*}输入到模型中进行分类,根据分类结果值与阈值 θ 进行比较,如果大于阈值则判定该数据区域 $Region \in Region_M$。算法描述如下:

```
Algorithm 3: 主数据区域识别算法
Input: {Region}, td.arff;
Output: Region_M;
1:  初始化主数据区域集合 Region_M = {};
2:  Load td.arff into SVM
3:  For each Region in {Region} do
4:    res = CM(Region);
5:    If(res > θ) then
6:      把 Region 放入 Region_M;
7:    End if
8:  Return Region_M;
```

其中 θ 为设定的阈值,由实验获得。

3 基于 AdaBoost 的 Web 实体信息抽取

在确定 Web 实体实例信息所在的主数据区域后,需要对数据区域中的文本节点进行判断,以区分属性标签、属性标签值及其他信息。

3.1 方法框架

本节提出一种基于 AdaBoost 的机器学习方法,充分利用数据区域自身的特征、数据区域在一组页面中的特征及 Web 实体全局模式的特征来提高从主数据区域抽取 Web 实体属性信息的准确率。

本方法将 Web 实体属性信息的抽取问题转化为文本分类问题,即对于主数据区域 *RegionM* 中所有的待识别文本 $\forall t_i \in T$,计算相似度 $P(t_i, c_j), cj \in C, C = \{c_1, c_2\}$,其中 c_1 代表属性标签,c_2 代表属性值。如果相似度 P 小于阈值 θ,则认为待识别文本 t_i 既不是属性标签也不是属性值,作为其他信息处理。

本方法首先利用选取的特征及特征向量计算方法来生成训练集;然后利用 AdaBoost 算法构造一个识别待识别文本的强分类器;最后根据分类器结果,将相关数据作为 Web 实体实例。

3.2 特征选取与训练集生成

3.2.1 特征选取

本方法主要使用三种类型的特征用于构造训练集。第一种类型的特征是待识别文本所在标记节点自身的特征,如使用的标记、路径深度、是否为链接等;第二种类型的特征是待识别文本与已有的 Web 实体实例信息及其全局模式信息的关系,如是否包含全局模式中属性标签、是否与已有的信息相似等;第三种类型的特征是基于统计的特征,如待识别文本是否在同类型的 Web 页面(来自相同的数据源的页面)中多次出现。本方法所选取的特征如表 2 所示。

表 2 待识别文本的特征描述

特征名称	特征含义	特征描述
tag	HTML 标记	待识别文本使用的 HTML 标记
link	是否为链接	待识别文本是否为链接
length	长度	待识别文本长度
deep	路径深度	文本节点在 HTML 标记树的路径深度
pathA	节点路径是否与已知标签的路径相同	是/否
pathB	节点路径是否与已知标签值的路径相同	是/否
attr	与全局模式中每个属性的相似度	取最大值
attrValue	与已有的属性值的相似度	值相似度;类型相似度;长度相似度等,取最大值
count	在相同类型的页面中出现的次数	相同类型的页面指来自相同数据源的 Web 页面

综合使用以上三类特征来为待识别文本构建特征向量,为构造分类器奠定基础。特征向量 $V = (v_{tag}, v_{link}, v_{length}, v_{deep}, v_{pathA}, v_{pathB}, v_{attr}, v_{attrValue}, v_{count})$ 的计算方法如下:

(1) $v_{tag} = tag_i \in \{\text{HTML 标记}\}$;

(2) $v_{link} = bool \in \{0,1\}$;

(3) $v_{length} = len, 0 < len < \infty$;

(4) $v_{deep} = deep, 0 < deep \leqslant$ Max deep of DOM;

(5) $v_{pathA} = bool \in \{0,1\}$;

(6) $v_{pathB} = bool \in \{0,1\}$;

(7) $v_{attr} = MaxSim(v_{attr}, attr_i \in GlobalSchema)$;

(8) $v_{attrValue} = MaxSim(v_{attrValue}, attrValue_i \in$ Web 实体实例);

(9) $v_{count} = count, 0 < count < \infty$。

3.2.2 特征选取

基于以上的特征及特征向量的计算方法,训练集的生成方法如下所示:

(1) 输入:Web 页面{*Page*};主数据区域{$Region_M$};已有的全局模式 *GS*;已有的 Web 实体实例信息{*inst*};

(2) 对于主数据区域中的节点文本 $node_{text}$,根据特征向量计算方法计算 $v_{tag}, v_{link}, v_{length}, v_{deep}, v_{pathA}, v_{pathB}, v_{count}$;

(3) 计算 $node_{text}$ 与全局模式 *GS* 中的所有属性的最大相似度值作为 v_{attr};

(4) 计算 $node_{text}$ 与 Web 实体实例 *inst* 中属性值的最大相似度值作为 $v_{attrValue}$;

(5) 将特征向量 $V = (v_{tag}, v_{link}, v_{length}, v_{deep}, v_{pathA}, v_{pathB}, v_{attr}, v_{attrValue}, v_{count})$ 写入训练集文件;

(6) 对训练集进行手工标注。

3.3 AdaBoost 分类器构造

Adaboost 是一种迭代算法,其核心思想是针对同一个训练

集训练不同的分类器(弱分类器),然后把这些弱分类器集合起来,构成一个更强的最终分类器(强分类器)。

AdaBoost 强分类器构造方法如下所述:

(1) 从 N 个训练样本学习出第一个弱分类器;

(2) 从分错的样本与新样本组成的 N 个样本中学习出第二个弱分类器;

(3) 从第一步分错的与第二步分错的样本以及新的样本组成的 N 个样本中学习出第三个弱分类器;

(4) 最后获得一个强分类器。

算法描述如下所示:

Algorithm 4: AdaBoost 分类器构造算法

Input: 训练集 td. arff, TD = $\{(x_1, y_1), \cdots, (x_N, y_N)\}$;迭代次数 T

Output: 分类器 f(x)

1: 初始化样本权重 $w_i = 1/N$; N 为样本总数,$0 < i < N$;

2: For t = 1, …, T do

3: (1) 调用弱学习算法来训练一个弱学习分类器 h_t;哦

4: (2) 计算分类器 h_t 的错误率:

5: $$\varepsilon_i = \sum_{i=1}^{N} W_i^t, y_i \quad y_i \neq h_t(x_i);$$

6: (3) 计算分类器 h_t 权重:

7: $$\alpha_t = \frac{1}{2} \ln \frac{1 - 2_t}{\varepsilon_t}$$

8: (4) 更新训练样本权重:

9: $$w_i^{t+1} = \frac{w_i^t \exp\{-\alpha_t y_i h_t(x_i)\}}{c_t} \quad i = 1, \cdots, N;$$

10: 是归一化常数,并且;

11: Output: f(x) = ;

3.4 Web 实体信息抽取算法

首先使用训练好的 AdaBoost 分类器对主数据区域中的待识别文本进行分类,然后根据分类结果判断待识别文本是属性标签还是标签值或是其他信息,最后将识别出的信息抽取出。

算法描述如下所示:

Algorithm 5: Web 实体信息抽取算法

Input: 主数据区域集合$\{Region_M\}$, 分类器 f(x)

Output: Web 实体集合 WebEntity

1: 初始化 WebEntity = {};

2: For each $Region_M$ in $\{Region_M\}$ do

3: For each $node_{text}$ in $Region_M$ do

4: 计算 $V = (v_{tag}, v_{link}, v_{length}, v_{deep}, v_{pathA}, v_{pathB}, v_{attr}, v_{attrValue}, v_{count})$;

5: 计算 $P(node_{text}, c_1)$, $P(node_{text}, c_2)$ based on f(V);

6: If($P(node_{text}, c_1) > P(node_{text}, c_2) > \theta$) then

7: $node_{text}$为标签;

8: Else if($P(node_{text}, c_2) > P(node_{text}, c_1) > \theta$) then

9: $node_{text}$为标签值;

10: Else if($P(node_{text}, c_2) < \theta$and $P(node_{text}, c_1) < \theta$) then

11: $node_{text}$为其它信息;

12: End if

13: End for;

14: 生成 Web 实体基于所有的 $node_{text}$, 然后放入 WebEntity;

15: End for;

16: Output WebEntity;

4 实　验

本文的实验分别在“图书”和“工作”两个领域的数据集上进行,对于每个领域,我们选取 21 个中文网站作为数据源,从每个网站随机抽取若干 Web 页面作为实验数据集。

对于算法性能的评估,使用查全率(precision),查准率(recall),F-测度(F-measure)作为评估标准。

基于 SVM 的主数据区域识别方法,在两个领域数据集的实验结果表 3 所示。从表 3 中可以看出,基于 SVM 的主数据区域识别方法在“图书”及“工作”领域都可以达到较高的查准率、查全率及 F-测度。

表 3　基于 SVM 的主数据区域识别方法效果

数据集	Precision (%)	Recall (%)	F-measure (%)
图书	94.3	95.1	94.7
工作	95.0	95.6	95.3

我们将基于 SVM 的主数据区域识别方法与基于位置信息的 Web 页面分割方法 PO SPS 进行比较,其 F-测度比较结果如图 2 所示。

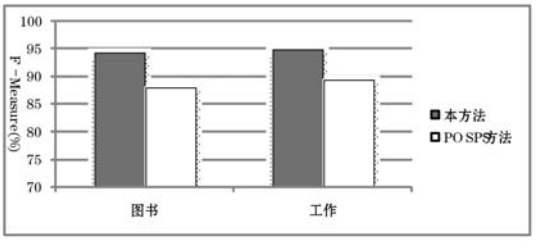

图 2　本方法与 PO SPS 方法的主数据区域识别结果比较

从图 2 可以看出,本文的基于 SVM 的主数据区域识别方法比基于位置信息的 Web 页面分割方法达到了更好的实验结果。因为 PO SPS 方法只利用了 Web 页面本身的特征信息,而本文的方法不仅利用了 Web 页面的自身特征还借助了已有的全局模式的信息。

为了验证 Web 实体全局模式对基于 SVM 的 Web 页面主数据区域识别方法的重要作用,我们做了一个 F-测度对比实验,图 3 的实验 1 在训练集的生成及分类器的学习方面不使用 Web 实体全局模式信息,实验 2 则将 Web 实体全局模式作为重要的特征用于训练集的生成及分类器的学习。

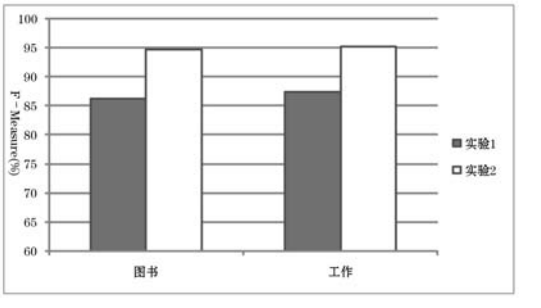

图 3　全局模式对主数据区域识别效果的影响对比

从图 3 中可以看出,Web 实体全局模式对于 Web 页面主数据区域的识别起着重要的作用,借助于全局模式的指导,本方法达到了更好的查准率及查全率。因为在 Web 页面中的 Web 实体实例主要使用属性标签来表达其自身信息,所以数据区域是否是主数据区域关键在于判断其是否包含相关 Web 实体的属性信息,因而通过判断数据区域与全局模式之间的关系,可以为判断数据区域是否为主数据区域提供重要的依据。

基于主数据区域的识别结果,Web 实体信息抽取在两个领域数据集的实验结果如表 4 所示。从表 4 可以看出,基于 Ada-

Boost 的 Web 实体属性信息抽取方法在查全率、查准率上都达到了较高的指标。

表 4 基于 AdaBoost 的 Web 实体信息抽取效果

数据集	Precision (%)	Recall (%)	F-measure (%)
图书	93.3	91.1	92.2
工作	94.0	92.6	93.3

我们选取经典的全自动 Web 数据抽取方法 RoadRunner 与本文提出的方法进行比较,RoadRunner 是一个全自动的 Web 数据抽取工具,它通过总结两个或多个样本页面结构所呈现出来的规律,从中抽取出能够用一个正则表达式表示的该类页面的模板,再使用该模板进行同类页面的数据抽取,该方法根据页面结构中 HTML 标记间的关系,以嵌套的方式表示抽取出的数据。实验结果如表 5 所示。

表 5 本文方法与 RoadRunner 的比较结果

数据集	本方法			RoadRunner		
	P (%)	*R* (%)	*F* (%)	*P* (%)	*R* (%)	*F* (%)
图书	93.3	91.1	92.2	66.5	61.2	63.9
工作	94.0	92.6	93.3	67.1	62.5	64.8

从表 5 可以看出,本文提出的基于 AdaBoost 的 Web 实体属性信息抽取方法在查全率、查准率上都优于 RoadRunner 方法,因为 RoadRunner 从 Web 页面中抽取 Web 实体实例信息时,只依据网页的代码来学习抽取规则,而没有考虑网页的其它特征,例如 Web 实体的模式、Web 实体实例的展示特征等。本文提出的方法充分利用了文中提到的三种特征,并且借助已有的 Web 实体全局模式的指导,从而有效地提高了抽取的准确率。

本方法是基于集成学习的方法来构造分类器,本文将其与单分类器 SVM 方法进行比较,比较结果如表 6 所示。由表 6 看出,本方法在查全、查准率上都优于基于 SVM 的 Web 实体信息抽取方法。

表 6 本文方法与 SVM 方法的比较结果

数据集	本方法			SVM		
	P (%)	*R* (%)	*F* (%)	*P* (%)	*R* (%)	*F* (%)
图书	93.3	91.1	92.2	87.5	85.2	86.3
工作	94.0	92.6	93.3	89.1	86.5	87.8

本文将 Web 实体全局模式信息最为重要的特征用于分类器的构造,为了检验 Web 实体全局模式在属性抽取过程中的作用,我们做了 F-测度对比实验,图 4 的实验 1 是本文提出的 Web 实体信息抽取方法,实验 2 使用的抽取方法没有使用 Web 实体全局模式信息。从图 4 可以看出,Web 实体全局模式在 Web 实体属性信息抽取过程中起着重要的作用。

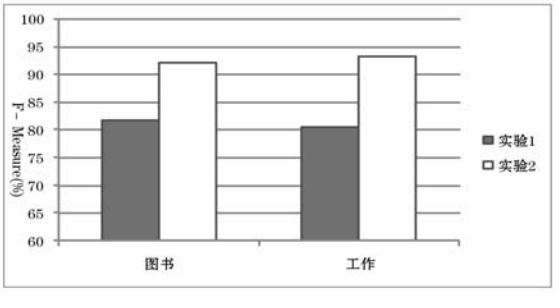

图 4 全局模式对 Web 实体抽取效果的影响对比

5 结 语

本文提出一种基于 SVM 和 AdaBoost 的 Web 实体抽取方法。为了提高 Web 实体信息抽取的准确性,首先提出一种基于 SVM 的 Web 页面主数据区域识别方法,基于 Web 实体实例在页面中的展示特征,有效地将 Web 页面进行数据区域分割,识别出 Web 实体实例所在的主数据区域,然后基于 Web 实体属性标签的特征,提出一种基于 AdaBoost 的集成学习方法,从页面的主数据区域自动地抽取 Web 实体信息。实验结果说明,本文方法能够取得良好的抽取效果。

参 考 文 献

[1] Wong T L, Lam W. Learning to adapt web information extraction knowledge and discovering new attributes via a Bayesian approach[J]. IEEE Transactions on Knowledge and Data Engineering, 2010, 22(4): 523-536.

[2] Chang C H, Kayed M, Girgis M R, et al. A survey of Web information extraction systems[J]. IEEE Transactions on Knowledge and Data Engineering, 2006, 18(10): 1411-1428.

[3] Laender A H F, Neto B A R, Silva A S, et al. A brief survey of Web data extraction tools[J]. SIGMOD Record, 2002, 31(2): 84-93.

[4] Eikvil L. Information extraction from World Wide Web-a survey[R]. Report #945. ISBN82-539-0429-0.

[5] Hammer J, Hugh J M, Molina H G. Semistructured data: the TSMMIS experience [C]//Proceedings of the First East-European Workshop on Advances in Databases and Information Systems, 1997: 1-8.

[6] Arocena G O, Mendelzon A O. WebOQL: Restructuring documents, databases, and Webs [C]//Proceedings of the 14th IEEE International Conference Data Engineering, 1998: 24-33.

[7] Saiiuguet A, Azavant F. Building intelligent Web applications using lightweight wrappers[J]. Data and Konwledge Engineering, 2001, 36(3): 283-316.

[8] Freitag D. Information extraction from HTML: application of a general machine learning approach[C]//Proceedings the 15th International Conference on Artificial Intelligence, 1998: 517-523.

[9] Califf M, Mooney R. Relational learning of pattern-match rules for information extraction [C]//Proceedings of the Sixteenth National Conference on Artificial intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference, 1999: 328-334.

[10] Kushmerick N, Weld D, Doorenbos R. Wrapper induction for information extraction [C]//Proceedings of the 15th International Conference on Artificial Intelligence, 1997: 729-735.

[11] Robert B. Supervised wrapper generation with lixto [C]//Proceedings of 27th International Conference on Very Large Database, 2001: 715-716.

[12] Rpbert B, Gartmer. Visual Web information extraction with lixto [C]//Proceedings of 27th international conference on Very Large Database, 2001: 119-128.

[13] Levy A Y, Srivastava D, Kirk T. Data model and query evaluation in global information systems[J]. Journal of Intelligent Information Systems, 1995, 5(2): 121-143.

[14] Crescenzi V, Mecca G, Merialdo P. RoadRunner: Towards automatic data extraction from large Web sites [C]//Proceedings of the 26th International Conference on Very Large Database Systems, 2001: 109-118.

(下转第 152 页)

2009 的数据),可见网络模型的评估值与期望值一致,评估准确率达到了 98.7%,表明建立的中药现代化产业发展能力综合绩效评估模型是有效的。为了验证通过基于 Levenberg-Marquardt 优化算法的 BP 神经网络建立评估模型的高效性,使用了基于其他算法的 BP 神经网络来构建同样的实例评估模型进行比较,对比结果如表 3 所示,效率最高的是 Levenberg-Marquardt 算法,训练次数仅为 7 次,效率最低的是附加动量法,需要进行 16 510 次的训练才能建立出达到同样标准精度的模型,时间相当于采用 Levenberg-Marquardt 算法所需时间的 2 358 倍,且效率相对较高的算法所需时间也将近为 Levenberg-Marquardt 算法的 6 倍,从而表明本文采用的 Levenberg-Marquardt 算法是优越的。

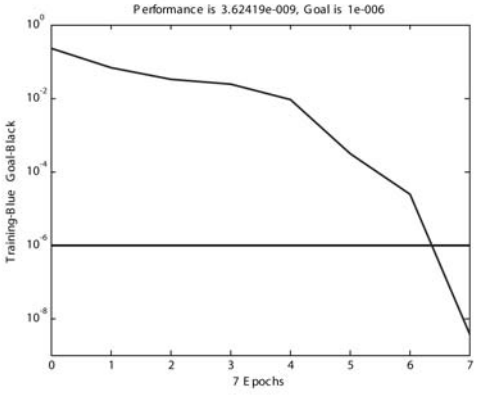

图 1 网络训练过程

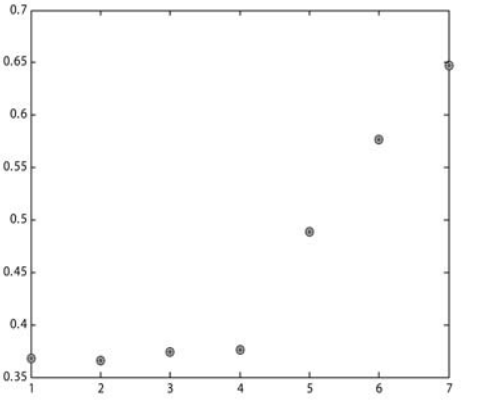

图 2 仿真结果与期望值比较

表 2 2001 – 2009 年中药产业现代化发展数据仿真结果

年份	2001	2002	2003	2004	2005	2006	2007	2008	2009
原产业发展指数	0.3675	0.3658	0.3742	0.3763	0.4887	0.5762	0.6474	0.4343	0.5638
仿真评价结果	0.3675	0.3658	0.3742	0.3763	0.4886	0.5761	0.6473	0.4415	0.5827

表 3 建立网络模型的各算法训练次数对比(epochs)

附加动量法	自适应调整参数法	bfgs 拟牛顿法	一步正切拟牛顿法	共轭梯度法	Levenberg-Marquardt
16510	2976	41	153	44	7

4 结 语

本文通过采用基于 Levenberg-Marquardt 优化算法的 BP 神经网络构建中药现代化产业发展能力综合绩效评估模型,经过实验验证,模型训练时间短,评估结果准确。构建的评估模型实例具有代表性,能够将该构建方法应用到中药现代化产业其他方面的综合绩效评估,构建出相应的综合绩效评估模型,为我国中药现代化产业相关评估工作和政策的制定提供帮助和指导。由于我国中药现代化产业涉及面广,具有一定复杂性,相关数据的采集和统计也存在一定困难,本文用于训练网络模型的历史数据可能不够全面,待采集到更多的数据时,可以进一步训练和完善该实例模型。

参 考 文 献

[1] 陕西中药现代化信息网. 中药现代化科技产业基地发展规划(2010 – 2020 年)[EB/OL]. 2010 – 04 – 15[2012 – 03 – 10]. http://www.snzy.gov.cn/e/DoPrint/? classid = 73&id = 5198.

[2] 张良均. 神经网络实用教程[M]. 北京:机械工业出版社,2008.

[3] 蔡秋茹,柳益君,蒋红芬,等. 基于 Levenberg-Marquardt 神经网络的企业资信评估方法研究[J]. 计算机与数字工程,2009(7):154 – 156.

[4] 王赟松,许洪国. 快速收敛的 BP 神经网络算法[J]. 吉林大学学报:工学版, 2003,33(4):79 – 84.

[5] 马爱霞,李勇,余伯阳. 我国中药产业现代化发展的实证研究——对我国中药产业现代化发展的评价[J]. 产业经济研究,2011(3):88 – 94.

(上接第 106 页)

[15] Zhao P, Meng W, Wu Z, et al. Fully automatic wrapper generation for search engines [C]//Proceedings of Fourth International Conference on World Wide Web, 2005:66 – 75.

[16] Buttler D, Liu L, Pu C. Omini: A fully automated object extraction system for the World Wide Web[R]. Technical Report, Georgia Institute of Technology, GIT – CC – 00 – 22, 2000.

[17] Embley D W, Jiang Y S, Ng Y. Record-boundary discovery in Web documents[C]//Proceedings of the 18th ACM SIGMOD International Conference on Management of Data, 1999:467 – 478.

[18] 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.

[19] 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 – 223.

[20] Liu W, Meng X F, Meng W Y. ViDE: A Vision-based approach for deep Web data extraction[J]. IEEE Transactions on Knowledge and Data Engineering, 2010,22(3):447 – 460.

[21] 丁艳辉, 李庆忠, 董永权, 等. 基于集成学习和二维关联边条件随机场的 Web 数据语义标注方法[J], 计算机学报, 2010,33(2):267 – 278.