一种有效的标签抽取和匹配方法

邹显春[1] 吴春明[1] 李盛瑜[2]
(西南大学计算机与信息科学学院 重庆 400715)[1] (重庆工商大学计算机与信息工程学院 重庆 400067)[2]

摘 要 标签抽取和匹配是查询接口理解的重要组成部分。提出了一种基于视觉的标签抽取和匹配方法,深入分析了相关匹配因子,给出了一种对查询接口表单进行重构的方法,它能依据接口 HTML 源代码自动还原出该表单的视觉布局特征。在最终的匹配算法中,综合考虑了基于 label 标记的匹配、基于文本语义的匹配以及基于位置特征的匹配。在 8 个领域共计 277 个查询接口上的实验证明了所提方法能取得较高的匹配精度。

关键词 标签抽取,位置特征,表单布局,元素-标签匹配
中图法分类号 TP391 **文献标识码** A

Effective Approach to Label Extraction and Matching

ZOU Xian-chun[1] WU Chun-ming[1] LI Sheng-yu[2]
(College of Computer and Information Science, Southwest University, Chongqing 400715, China)[1]
(School of Computer Science and Information Engineering, Chongqing Technology and Business University, Chongqing 400067, China)[2]

Abstract Label extraction and matching are an important part of the query interface understanding. A vision-based label extraction and matching approach was proposed in this paper. First, the factors which affect label matching were deeply analyzed, and then, a method of reconstructing query interface by analyzing its html code was given correspondingly which can restore the visual layout of form effectively. Finally, the element-label matching was realized which comprehensively considers label tag, text semanteme and position feature. Experiments on 277 query interfaces in 8 domains demonstrate the feasibility of our proposed approach.

Keywords Label extraction, Position feature, Form layout, Element-label matching

查询接口理解是许多基于 Web 应用的基础和前提,如表单语义模式抽取和匹配、Deep Web 索引、Web 数据库分类/聚类以及查询结果注释等[1,2]。标签抽取和匹配是查询接口理解的重要组成部分,即为每一个表单元素寻找一个最适合描述其语义的文本标签[7]。查询接口的设计是面向用户的,因此人们可以基于视觉习惯来方便地进行元素-标签匹配,但这一判断过程对于机器处理来说,面临着许多困难:首先,计算机面对的是查询接口的 HTML 代码,从视觉上相近的表单组件在 HTML 代码中则可能处于较远的位置;其次,表单在设计上缺乏统一的布局模式,往往带有较大的随意性;第三,表单元素与文本标签并不是一一匹配,可能有多个表单元素共用一个标签,个别元素没有独立的标签,也可能存在不与任何元素相匹配的标签,这都为标签的自动抽取和匹配带来了极大的挑战。

近年来,有关查询接口标签抽取和匹配问题引起了广泛关注,学者们从不同角度对该问题进行了研究,并提出了许多解决方案[3-11],但目前仍然缺乏统一高效的实用技术。本文对该问题展开研究,提出了一种基于视觉的标签抽取和匹配方法,通过 8 个领域共计 277 个表单数据的实验,证明了该方法的有效性。

1 相关研究

按照所采取技术的不同,目前已有工作可以分为两大类:基于规则/启发式的方法和基于模型的方法[8]。

基于规则/启发式的方法基于这样的假设:表单元素与其语义标签在布局上存在着某种通用且有限的模式,而查询接口的构建正是基于这些模式,如文献[3]指出表单元素与其语义标签间的条件模式共有 25 种,且仅有少数会被频繁使用。基于这一假设,目前大多工作都采用这一方法,如文献[12]设计了一个用于 Hidden-Web 搜索的爬虫——HiWE,在其表单分析模块中使用了布局引擎来计算表单元素与候选标签间的像素距离,并据此产生一系列候选匹配集,再综合考虑文本内容、文本样式以及表单布局等属性来生成启发式规则,最后利用这些规则来完成匹配;文献[3]将标签匹配看成是一种对表单句法进行解析的过程,他们开发了称为 2P-grammar 和 Best-effort 的软件解析器,其允许用户自己来指定抽取规则;文献[7]则首先将查询接口表示成一个字符串,称之为 IEXP (Interface Expression),用"t"代表任意文本标签,"e"代表任意表单元素,"|"代表 HTML 换行标记,如 P、BR、TR 等,然后,利用定义的 5 条启发式规则,基于文本标签和表单元素在

到稿日期:2011-06-23 返修日期:2011-09-20 本文受重庆市自然科学基金(CTS2009817),重庆市教委教育教学改革项目(303155)资助。
邹显春(1965—),男,硕士,副教授,主要研究方向为 Web 技术应用,E-mail:zouxc@swu.edu.cn;**吴春明**(1972—),男,博士,副教授,主要研究方向为计算机网络、Web 信息获取;**李盛瑜**(1972—),女,硕士,讲师,主要研究方向为计算机网络应用、Web 技术应用。

IEXP 中的位置进行标签匹配。

基于规则的方法缺乏动态适应性，而且这些规则或启发式通常需要手工创建[9]。因此，部分学者提出了基于模型的方法，其采用机器学习来进行标签抽取和匹配。如文献[6]提出一种有指导的机器学习方法——LabelEx，即首先基于元素与标签的位置关系产生一系列候选匹配集，然后基于布局特征，利用朴素贝叶斯算法和决策树算法分别完成剪除错误匹配和选择正确匹配的工作；文献[10]描述了一种基于隐马尔可夫模型(HMM)的接口分片方法，其首先将查询接口解析成一棵 DOM 树，然后将该树转换为相应的观察序列，最后利用样本数据对 HMM 进行训练和测试，将隐含的接口分片知识编码到基于 HMM 的人工设计器中，完成训练和测试的设计器即可以自动完成标签匹配与表单分片。

综观已有文献，标签抽取和匹配过程大体可以分为两个阶段(见图 1)：

1)建模：给定一个查询接口的 HTML 源代码，首先对其进行建模和重构，将其处理成适合机器处理的数据结构，如树状结构[9,11]、字符串表达式[7,10]或者一系列候选匹配集[3,6]等；

2)解析：这一阶段是对重构的查询接口进行解析，运用某种算法进行标签抽取和匹配，如基于规则[3,7,12]或基于机器学习[6,10]等方法，最后输出匹配结果。

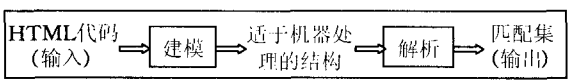

图 1　标签抽取过程

本文在这两个方面进行了深入研究，在建模阶段，提出了一种 TBIExp 表单重构方法；在解析阶段，提出了一种 R3LEX 算法，实验表明其能取得较好的匹配效果。

2　标签抽取和匹配因子

如前所述，人们依靠视觉和布局习惯来完成元素-标签匹配，而在 HTML 代码中，元素与其语义标签间的这种匹配关系却并没有清晰地定义。基于对大量查询接口的分析，本文提出以下 3 类用于标签抽取和匹配的影响因子。

(1)基于 label 标记的匹配

HTML 标准提供了 label 标记来修饰各表单元素，其 FOR 属性唯一地对应于该标签所修饰元素的 ID 属性值。但实际使用该标记的情况并不多见，本文以 UIUC 的 TEL-8 表单数据集为统计对象[13]，在 277 个查询接口中，共有标签 2584 个，其中使用了 label 标记的有 57 个，所占比例仅为 2.21%。虽然 label 标记使用得较少，但由于它是 HTML 语言提供的唯一一种确定性的匹配标识，因此，本文将首先考虑这种基于 label 标记的匹配，并将其记为 LabelMap(t;e)。

(2)基于语义的匹配

表单元素通常具有 ID、NAME 和 VALUE 3 个属性(可以部分为空)。这些属性值虽然对用户不可见，但往往与其标签间存在着一定的语义关系，这为元素-标签匹配提供了很好的基于语义的证据。本文将这种基于语义的匹配记为 SemanticMap(t;e)。为了对其进行量化，本文采用了归一化的“最长公共子序列”(Longest Common Subsequence，LCS)作为衡量元素-标签间的语义相似度，其定义如下[6]：

$$\text{SemanticMap}(t;e)=\frac{\text{LCS}(t;e)}{\text{Min}(\text{Length}(t);\text{Length}(e))}$$

式中，t 代表标签文本，e 代表元素的 ID、NAME 或 VALUE 3 个属性文本，若 3 者均存在，则取最大值；并且规定，如果 t 为空或者 e 为空，则 SemanticMap 值为 0。

基于该量化指标，如果计算出的语义相似度大于设定的阈值，则认为该文本是该表单元素的语义标签。

(3)基于视觉的标签匹配

从视觉上，虽然表单布局并没有统一的标准，但有一个基本规律：标签总是位于表单元素的附近，毫不相关的控件和标签不可能也不允许出现在一起；依据人们的视觉习惯，表单布局往往存在一定的规律性。基于对大量查询接口的统计和观察，本文对元素-标签间的这种视觉特征进行了总结，并给出如下 7 条相应的启发式规则。

- 如果元素与标签出现在同一表格单元格内，则它们匹配，且标签通常位于 textbox 与 select list 的前面、radiobutton 和 checkbox 的后面；
- 如果表单元素与其语义标签出现在一行的不同列中，则它们位于相邻的两列；
- 如果一行中有多个表单元素和文本标签，则它们通常交替出现；
- 如果表单元素与其语义标签没有被布局在同一行，则两者通常被布局在垂直相邻的列上；
- 如果在一行中有多个相邻的元素，则它们通常共享同一语义标签，且该标签或者位于该行最左侧，或者位于最左侧表单元素的上方；
- 在表单或表格首部出现的独立的长文本标签，通常用于标识该标记范围内的所有内容，而不与任何表单元素匹配；
- 在表单或表格尾部出现的独立的长文本标签，通常为说明文字，不与任何表单元素匹配。

如果能通过对表单的 HTML 源代码进行建模和重构还原出这种视觉布局特征，则可方便地利用这些规则进行标签匹配。本文将这种基于视觉的匹配记为 VisionMap(t;e)。

基于以上讨论，元素-标签匹配可以通过如下模型表示：

$m(t,e)$=LabelMap∪SemanticMap∪VisionMap

3 个匹配因子的重要性排序如下：LabelMap＞SemanticMap＞VisionMap，这是本文提出的 R3LEX 匹配算法的基础。

3　特征抽取与接口表单重构

为了利用以上 3 类因子进行标签匹配，应首先对查询接口的 HTML 代码进行处理，从中提取出相应的特征项，然后对接口进行重构，还原出元素-标签间的视觉特征。

3.1　表单预处理

本文关注的是为表单元素匹配适合的语义标签，因此首先去除了与元素和标签无关的内容，然后对文本进行了基于语义的变换。具体包括：

- 去除 HTML 源文件中的图像、超链接、CSS 样式表、JavaScript 脚本以及 Hidden 控件等干扰内容；
- 去掉括号以及其中的文本，因为括号内的文本往往用于其他对象的解释或说明，通常并无实际的意义；
- 去除停用词；
- 对文本进行词根还原；

• 去除所有非字母字符，并用空格替代；

• 将所有字符转为小写。

接下来，对预处理后的 HTML 代码进行特征提取，并将表单处理为表格化的数据结构。

3.2 特征抽取

文本特征抽取　在 HTML 中，所有文本标签均位于“>”和“<”之间。为此，本文首先对 HTML 代码中所有位于“>”和“<”之间的文本进行提取，将其作为候选文本标签，同时考查了两类特征：1）若文本有 label 标记作为修饰，则提取其 FOR 属性值；2）考查该标签是否有样式标记作为修饰，如 B、I、FONT、BIG、EM、STRONG、U 等，这些样式标记往往标识了候选标签的重要性。基于以上考虑，本文将候选文本标签表示为一个三元组：$t=$(text, forID, StyleTag)，其中，text 代表文本内容，forID 代表 label 标记的 FOR 属性值，StyleTag 代表了该文本的样式标记。

表单元素抽取　对于表单元素，为了同时进行以上 3 类匹配，将其表示为一个四元组：$e=$〈type, id, name, value〉，其中，type 表示元素的类型，可以为 textbox, select list, radio button, checkbox 之一，ID、VALUE 和 NAME 则分别代表该元素对应的属性，若无则为空。

3.3 接口表单重构

为了从 HTML 文档中还原出表单的原始布局特征，文献[7]提出用“t”代表表单中的任意文本标签，“e”代表表单中的任意表单元素，而将 BR、P、TR 等标记转换为符号“|”，用以标识换行，得到形如“$t|tt|ee|t$”的接口表达式（IEXP）。该方法虽然提供了对表单的高度抽象表示，但也存在以下几点不足：1）IEXP 忽略了元素与标签的细节特征，不能进行基于语义的匹配；2）IEXP 不能表示元素与标签间原有的上下邻近关系，不能进行基于视觉特征的匹配；3）IEXP 没有对复杂布局情况进行处理，往往会导致对表单的错误解析。本文对 IEXP 方法进行了完善，在提取了元素和标签的相应特征后，对 HTML 中的定界符进行解析，提出了一种基于表格的接口表达方法（Table-Based Interface Expression, TBIExp），较好地保留了查询表单的原始布局信息。

下面结合图 2 所示的查询接口片断示例，介绍 TBIExp 的具体生成方法及主要思想。

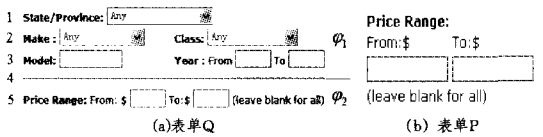

图 2　查询接口片断

查询接口通常由表单元素、文本标签及一系列定界符构成。其中，典型的定界符包括 BR、P 和 TR 等；此外，HTML 还提供了如 HR、DIV、UL、H1-H6 等几个带有自动换行功能的标记，这些标记的使用为表单布局带来了灵活性，但也为分析 HTML 文档以自动获得表单布局信息增加了复杂度。本文首先将这些具有自动换行功能的标记视为换行符；其次，对下面一种特殊的布局方式进行了特别处理：少数表单会在表格单元格内采用换行标记来进行布局，例如在表单 P 中，标签“From: $”和“To: $”分别与下方的 textbox 位于表格的同一单元格内，并使用了 BR 标记用于换行，如果按文献[7]所提方法，则 P 的 IEXP 表示为“$t|t|et|e|t$”，这显然是错误的。为此，本文采用了忽略所有位于单元格内的换行标记的处理方法。这样，P 就可以表示为“$t|te|te|t$”，虽然处理后的接口表达式对表单的原始布局进行了改变，但却准确地保留了标签与元素间的位置关系，保证了匹配的准确性。现在，顺序扫描查询接口的 HTML 源代码：

• 当遇到文本标签时，用三元组 $t=$(text, forID, StyleTag)进行表示；

• 当遇到表单元素时，用四元组 $e=$〈type, id, name, value〉进行表示；

• 当遇到 BR、TR、DIV、UL 标记时，用符号“|”进行表示；

• 当遇到 HR、P、H1-H6 标记时，用“‖”（两个“|”）进行表示。

对于最后一种情况，主要考虑了这几个标记的显示特点以及人们的布局习惯。HR（横线）、P（段落）和 H1-H6（标题）都可以从视觉上将表单划分为若干逻辑单元，如 Q 中的第 4 行将表单划分为 $\varphi1$ 和 $\varphi2$ 两个区域，而基于人们的常识，不同逻辑单元间的标签和元素间不可能存在匹配关系。为此，本文用“‖”来表示对逻辑单元的划分。

在得到的接口表达式中，有可能出现多于两个连续“|”的情况，此时应将多余的“|”删除，如对于接口 Q，最终的接口表达式 Q=“$te|tete|tettete\|\|ttete$”。观察 Q 的接口表达式，可以发现 Q 被“|”划分为 $n(n=5)$ 个片断$\{f_1|f_2|\cdots|f_n\}$，其中 $f_1=te$，$f_2=tete$，$f_3=tettete$，$f_4=$null，$f_5=ttete$，假设所有片断中的最大长度 $m=\sum_1^n \text{length}(f_i)$，则 Q 可以被方便地表示成为一个 n 行 m 列的表格，由此得到该接口的 TBIExp。如对于 Q，其 TBIExp 表示如表 1 所列。其中，第 1 行中的 $t=$(State Province, , B)，$e=$〈select, , state id, 〉，其他的 t 和 e 依次表示后继的文本和表单元素。第 4 行的空行是因为此处采用了 HR 标记，因而被转换成为“‖”；在第 5 行中，原来的文本“(leave blank for all)”由于处于括号内，现已被移除了。

表 1　接口 Q 的 TBIExp 表示

行号	TBIExp							
1	t	e						
2	t	e	t	e				
3	t	e	t	t	e	t	e	
4								
5	t	t	e	t	e			

由表 1 可以看出，TBIExp 不仅保留了文本和表单元素的所有特征，更为重要的是较好地保留了原始表单的视觉布局信息，这为后继的匹配算法提供了依据。

4 匹配算法

基于查询接口的 TBIExp，本文提出 R3LEX（Round-3 Label Extraction）匹配算法。算法的基本思想是：给定一个查询接口的 TBIExp，首先扫描 t 中是否存在 forID 值，若存在，则进行基于 LabelMap(t;e)的匹配，输出匹配结果，将匹配项置空；其次，扫描 e 中是否含有 ID、NAME 或 VALUE 3 个属性，若存在，则进行基于 SemanticMap(t;e)的匹配，输出匹配结果，将匹配项置空；第三，若 TBIExp 中尚存在 e，则根据启发式规则，进行基于 VisionMap(t;e)的匹配，输出匹配结果，

若无匹配项，则认为 e 没有语义标签。

具体的算法描述如图 3 所示。其中，$a,b\in[-1,+1]$，这是因为元素标签一定与元素相邻，且通常位于元素的上、下、左、右 4 个位置上；第二轮中的“θ”代表阈值，为了减少错误匹配的机率，本文选择 $\theta=0.7$。

```
R3LEX(T)→H:
Input: T, the TBIExp of a query interface
Output: H: element-label matching set
1.For each t(i,j) in T do   /*Round 1: LabelMap-based matching*/
   if t(i,j).forID != Φ and LabelMap(t(i,j).forID;c(i+a,i+b).id):
      /*Output matching and set matching items as null*/
      H← <c(i+a,i+b) ; t(i,j)>; t(i,j)= Φ; c(i+a,j+b)= Φ;
2.For each e(i,j) in T do   /*Round 2: SemanticMap-based matching*/
   if Max(LCS(e(i,j), t(i+a,j+b).text))> θ
      H← <e(i,j), t(i+a,j+b)>; t(i+a,i+b)= Φ; e(i,j)= Φ;
3.For each e(i,j) in T do   /*Round 3: VisionMap-based matching*/
   if satisfy the heuristic rules in Table 1:
      H← <e(i,j), t(i+a,j+b)>; t(i+a,i+b)= Φ; e(i,j)= Φ;
   else:
      H← <e(i,j), Φ>
4.Return H
```

图 3　R3LEX 算法描述

在算法的评价标准上，本文采用了信息检索中常用的 3 个指标：准确率（P）、召回率（R）以及综合 F 测度（*F-measure*）。假设 En 表示由本文所提算法识别出的匹配数量，Rn 表示实际的元素-标签匹配数量，则：

$$P=En\cap Rn/En$$

$$R=En\cap Rn/Rn$$

$$F\text{-}measure=2\times P\times R/(P+R)$$

本文中，由于是基于表单元素来进行标签匹配，因此无论匹配结果正确还是错误，总能为每个表单元素找到一个匹配项，也就是说，En 的值与 Rn 的值实际上是一致的。本文中的 P、R 以及 *F-measure* 均相同，这里仅选择 *F-measure* 作为衡量指标。

5　实验与结果分析

为了验证以上方法的可行性和有效性，本文利用 UIUC 大学的 TEL-8 查询接口数据集进行了实验[13]，从 8 个领域中共提取了 277 个接口表单。数据集的领域分布情况及相关统计特征如表 2 所列。

表 2　实验所用数据集的统计特征

领域	表单	元素数	标签数	平均控件数	平均标签数
airfares	46	528	529	11.5	11.5
automobiles	28	248	256	8.9	9.14
books	44	301	365	6.8	8.30
carrentals	16	223	269	13.9	16.8
hotels	35	370	345	10.6	9.86
jobs	29	253	355	8.7	12.2
movies	39	280	307	7.2	7.87
music	40	168	158	4.2	3.95
Totel	277	2371	2584	8.97	9.95

5.1　实验结果

首先对每个表单的 HTML 代码进行了预处理，提取了文本标签与表单元素的相关特征属性，并利用本文提出的 TBIExp 方法对每一个查询接口都进行了重新表示，最后利用 R3LEX 匹配算法对 TBIExp 进行了标签抽取和匹配，最终结果如图 4 所示。

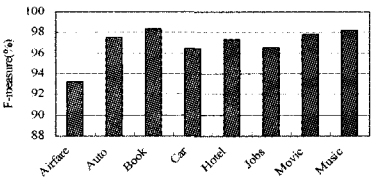

图 4　标签匹配结果

由实验结果可以看出，利用本文所提方法在 8 个领域均获得了较高的匹配精度，其中，Book 和 Music 领域的 *F-measure* 值均超过了 98%，8 个领域的平均 *F-measure* 值为 96.93%，这证明了本文所提方法的有效性。

5.2　实验分析

从图 4 可以看出，不同领域间的匹配结果存在着一定的差异，这可以通过分析表 2 中的数据来发现其中的原因：首先，不同领域查询接口中元素的平均个数有较大不同，如 Music 领域的元素平均个数仅为 4.2，而 Carrentals 领域的元素平均个数则达到了 13.9。显然，随着元素数量的增多，查询接口的布局变得相对复杂，出现不规则布局的机率就越大；其次，不同领域间元素与文本标签的数量存在不匹配现象。例如，Hotel 与 Music 领域的元素个数均大于候选文本标签的数量，即无匹配标签的情况较多，而 Jobs 领域的标签数量则远远大于元素个数（比例约为 1.4:1），这进一步加大了表单布局的复杂度，使得在对表单进行基于 TBIExp 的重构时出错机率变大，最终致使匹配精度下降。

结束语　标签抽取和匹配是查询接口理解的前提和基础。本文对此问题展开了研究，提出了一种基于视觉的标签抽取和匹配方法。主要工作如下：

1）提出了一种基于表格的接口表达式方法——TBIExp，它能基于表单的 HTML 文档对表单进行表格化的重新表示，较好地保留了原始表单的视觉布局信息；

2）对元素与其标签的视觉特征进行了全面总结，并给出了相应的启发式规则；

3）提出了一种三轮匹配算法——R3LEX，其综合考虑了 label 标记的匹配以及表单元素与其文本标签间的语义关系和位置关系。

实验证明了本文所提方法的可行性。

参 考 文 献

[1] He B, Patel M, Zhang Z, et al. Accessing the deep web: A survey [J]. Communications of the ACM, 2007, 50(5): 95-101

[2] Lawrence S, Giles C L. Searching the World Wide Web [J]. Science, 1998, 5360(280): 98-100

[3] Zhang Z, He B, Chang K. Understanding web query interfaces: best-effort parsing with hidden syntax [C] // Proceedings of ACM SIGMOD. 2004: 107-118

[4] Chang K, He B, Zhang Z. Metaquerier over the deep web: Shallow integration across holistic sources [C] // Proceedings of the VLDB Workshop on Information Integration on the Web. 2004

[5] Wu W, Doan A, Yu C. WebIQ: Learning from the web to match deep-web query interfaces [C] // Proceedings of ICDE. 2006: 44

[6] Nguyen H, Nguyen T, Freire J. Learning to extract form labels [C] // Proceedings of PVLDB. 2008: 684-694

[7] He H, Meng W, Lu Y, et al. Towards deeper understanding of the search interfaces of the deep Web [J]. World Wide Web,

2007,10(2):133-155

[8] Khare R, An Yuan, Song I-Y. Understanding Deep Web Search Interfaces: A Survey[J]. ACM SIGMOD Record, 2010, 39(1): 33-40

[9] Wu W, Doan A, Yu C, et al. Modeling and Extracting Deep-Web Query Interfaces[C]// Advances in Information and Intelligent Systems. Springer Berlin, Heidelberg, 2009: 65-90

[10] Khare R, An Y. An Empirical Study On Using Hidden Markov Model for Search Interface Segmentation[C]//Proc. of the 18th Int'l Conf. on Information and Knowledge Management. Hongkong, China, ACM Press, New York, NY, 2009: 17-26

[11] Dragut E C, Kabisch T, Yu C, et al. Hierarchical Approach to Model Web Query Interfaces for Web Source Integration[C]// Proc. of the 35th Int' Conf. on VLDB (Lyon, France). IEEE Computing Society, Washington DC, 2009: 325-335

[12] Raghavan S, Garcia-Molina H. Crawling the hidden Web[C]// Proc. of the 27th International Conference on very Large Data Bases. 2001: 129-138

[13] UIUC Web integration repository[EB/OL]. http://metaquerier.cs.uiuc.edu/repository/, 2010-10

(上接第 179 页)

Blog 数据集。该数据集采集于 2005 年 8 月 29 日，一共包含 3000 个垃圾博客网页，其中有 700 个 splog 网页、700 个 blog 网页，剩余 1600 个网页作为测试集。博客内容涵盖科学研究、个人情感、公共事业、新闻、图片、国际政治以及博客链接等，其完全可以模拟真实的博客世界。

SVM 分类器已经用于垃圾博客过滤任务中[3]，本文基于这个方法进行对比实验。所有的实验在相同环境下进行。实验环境为：Windows XP 系统，2.66GHz 奔腾四处理器，512MB 内存，VC 环境。采用的评价指标是准确率(Precision)、召回率(Recall)和 F 值，以下所指的准确率都综合考虑了准确率、召回率及 F 值。

所设计的实验方法如下：设定特征值数量从 100 到 400。分别比较组合特征算法和传统分类算法的准确率。实验结果如图 2 所示。

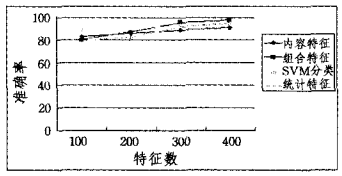

图 2 准确率结果对比

从图 2 来看，当特征数量比较少时，SVM 的准确率比较好；随着特征数量的增加，组合特征的准确率缓慢提高并逐渐显现出组合特征的优势。这是因为 SVM 需要大量人工标注的训练语料，由于训练集有限，而特征规模不断增大，使得 SVM 的分类面会出现较大的误差。特征数量增加，会完善 CFDSD 算法，使之具有高达 97.7%的准确率。

选定 400 个特征，比较分类算法的准确率(Precision)、召回率(Recall)和 F 值，如图 3 所示。

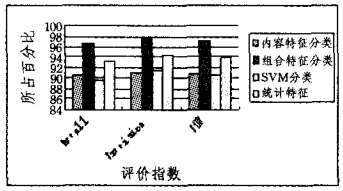

图 3 评价指数对比

当选定特征数量时，组合特征分类的准确率和召回率都高于传统特征分类、统计特征和 SVM。因组合特征是提取垃圾博客的关联特征，且特征之间存在互补性，所以其提高了算法对垃圾博客分类的能力；内容特征分类算法提取独立词频特征而忽略了特征的关联性；统计特征缺乏动态性而 SVM 不容易找到一个最佳的分类面。

结束语 本文分析了博客的关联特征和词频特征，比较了特征融合后分类算法与现有算法的分类效果。结果表明，基于组合特征的动态垃圾博客过滤算法，在同等时间复杂度下大大提高了过滤的准确率，具有更好的泛化能力，适应了博客的内容随着时间的不同而不断更新的特点。下一步工作是改进 CFDSD 算法阈值的提取，使阈值是由算法根据不同训练集而自适应地提取的。同时将 CFDSD 算法扩展到其他语言，如中文，以及将博客的 ping 时间系列特征加入到 CFDSD 算法中，以进一步提高 CFDSD 算法的泛化能力。

参 考 文 献

[1] Nanno T, Fujiki T, Suzuki Y. Automatically collecting, monitoring, and mining Japanese weblogs[C]//Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters. ACM Press(WWW Alt. '04), 2004: 320-321

[2] Sato Y, Utsuro T, Fukuhara T. Analysing features of Japanese splogs and characteristics of keywords[C]//Proc. 4th AIRWeb. 2008

[3] Kolari P, Finin T, Joshi A. SVMs for the blogosphere: Blog identification and splog detection [C]// Proc. of the AAAI Spring Symp. on Computational Approaches to Analyzing Weblogs. California: AAAI Press, 2006: 92-99

[4] Melville P, Gryc W, Lawrence R D. Sentiment Analysis of Blog by Combining Lexical Knowledge with Text Classification[C]// Proc KDD 09. June 2009

[5] Ru Yu, Sundaram L H, Chi Yun. Splog Detection Using Self-similarity Analysis on Blog Temporal Dynamics[C]// Proc 5th AIRWeb Press. 2007

[6] Katayama T, Utsuro T, Sato Y. An Empirical Study on Selective Sampling in Active Learning for Splog Detection[C]// Proc 4th AIRWeb Press. 2009

[7] Kolari P, Finin T, Joshi A. Svms for the blogosphere: Blog identification and splog detection[C]// AAAI Spring Symposium on Computational Approaches to Analysing Weblogs. Baltimore County: Computer Science and Electrical Engineering. University of Maryland, March 2006

[8] Cormack G V, Smucker M D, Clarke C L A. Efficient and effective spam filtering and re-ranking for large Web datasets[J]. Computing Research Repository, 2010, 14(5): 441-465

[9] 魏红宁. 决策树剪枝方法的比较[J]. 西南交通大学学报, 2005, 40(1): 44-48

[10] 刘玮, 廖祥文, 许洪波. 基于统计特征的垃圾博客过滤[J]. 中文信息学报, 2008, 22(6): 86-91

[11] http://ebiquity.umbc.edu/resource/

[12] http://technorati.com