# 基于正则表达式构建学习的网页信息抽取方法

朱文琰 郑肖雄
(复旦大学计算机科学技术学院智能信息处理重点实验室 上海 200433)

**摘 要** 正则表达式作为信息抽取领域中的一种常用方法已经被广泛应用多年。然而构建高质量并且复杂度较高的正则表达式通常需要耗费大量人工成本,为此,提出一种基于正则表达式状态转换的算法来学习复杂正则表达式的构建过程。该算法需要给定输入初始正则以及正反例样本,初始正则表达式在经过析取分离与合并交叉两大类正则表达式状态转换之后,得到候选正则表达式集合,利用 F 值评估候选项的信息抽取效果,通过贪心的启发式策略选择一个最优正则表达式作为输出。在多种数据集上对算法进行测评。实验表明,该算法性能与准确度均优于常规的机器学习方法。尤其在较小规模训练集和跨数据集上依然有较好的效果。

**关键词** 正则表达式构建 状态转换 Web 信息抽取

**中图分类号** TP3 **文献标识码** A DOI:10.3969/j.issn.1000-386x.2017.02.003

## A WEBPAGE INFORMATION EXTRACTION METHOD BASED ON REGEX CONSTRUCTION LEARNING

Zhu Wenyan Zheng Xiaoxiong
(*Shanghai Key Lab of Intelligent Information Processing, School of Computer Science, Fudan University, Shanghai 200433, China*)

**Abstract** As one of the main methods in the field of information extraction, the method based on regular expression has been widely used for many years. However, the construction of regular expressions is with high quality and high complexity, it is usually required to spend a lot of manual efforts. Therefore, a method based on regular expression state transition is proposed to learn the construction of complex regular expressions. The method takes in a given initial input RegEx and both positive and negative labeled samples, a collection of candidate RegEx is got after applying two main kind of regular expressions transformation on the input RegEx, based on F value assessment of the candidate RegEx on the information extraction task, the algorithm selects an optimal regular expressions as output by greedy heuristic strategy. The performance of this algorithm is evaluated on multiple datasets. Experiments show that the performance and accuracy of the proposed method outperforms those of the standard machine learning methods. And it still has a good effect on condition of small scale training set and cross domain data set.

**Keywords** RegEx construction State transition Web information extraction

## 0 引 言

大部分的信息抽取任务可以通过精心构建的正则表达式来抽取相应的实体。例如邮箱地址、电话号码、信用卡号码和身份证号码、基因和蛋白质名称等。在标准的正则表达式构建过程中,以上提到的这些实体所代表的特征模式均能够被很好地表现出来。构建这样的正则表达式原型是非常简单直接的,但事实上,考虑到方法的健壮性要求,通常就需要更加复杂的规则。尽管信息抽取领域已有许多机器学习的方法,包括最大熵马尔科夫模型[1]、监督学习[2]、字符类模型[3]、半监督学习[4]、无监督学习[5]等。但是在实际的信息抽取任务中,人工构建的正则表达式依然被广泛采用[6]。事实上,如何通过自动学习的方法来减少在构建正则表达式过程中的人工消耗是信息抽取领域的一个难题。本文提出了一个基于正则表达式学习的信息抽取方法,将标注过的正反例样本以及一个初始的正则表

收稿日期:2016-01-22。朱文琰,硕士,主研领域:金融信息处理。郑肖雄,硕士。

达式作为输入,通过一系列正则表达式的状态变换得到一组备选正则集合,在多组备选集合上根据其在标注样本上的信息抽取效果,选出最优正则表达式作为输出。通过实验可以证明,该方法能够显著减少在信息抽取初期构造复杂正则表达式的人工成本,在一些特定类型的信息抽取任务中具有较好的效果。尤其是在较小规模的训练集以及跨数据集上有不错的表现,能够构造出高质量的正则表达式。

## 1　相关工作

信息抽取是自然语言处理研究中的重要部分,网页信息抽取是指在半结构化的网页内容中抽取结构化数据的过程。目前,信息抽取系统在实现上主要分为两大类,分别是基于统计学习的方法和基于规则模式的匹配方法。两者互有优劣,统计学习的方法具有可移植性和鲁棒性,但是都需要大量训练数据来完成对参数的设置以及对系统的优化。基于规则的匹配方法具有较高的效率和准确率。但是规则的生成需要较高的代价。所以规则的自动生成是信息抽取领域的重要问题。

第一个实现规则的机器学习方法的是 Cristal 信息抽取系统[7]。这个系统先从训练样本中生成规则集合,抽取方法是每一个实例提取出一个原始规则。然后循环从规则集合中选择两个相似度最高的规则进行合并,最后得到最小规则集。Crystal 系统目前只能够支持单槽的信息抽取,它的缺陷是无法确定目标字段的界限。WHISK[8] 抽取系统通过将规则的约束条件不断增加来得到最终的结果。此系统首先确定能够能覆盖所有样例的规则最普通范式,然后通过训练样本对规则增加特征和限制进行拓展,满足一定的错误率要求后停止训练,得到最终的集合。AutoSlog 是基于模板词典的规则构造器,它能够自动地构造指定领域的词典,这样的模板也叫做概念节点。一个概念节点包含如下部分:概念位元、语言规则以及触发条件[9]。其中位元包含了一系列用于触发的词组,触发条件对生成的语言规则在语法上进行了一些约束。RAPIER[10] 是基于逻辑的一种信息抽取系统,从训练语料上归纳出所需要的抽取规则。RAPIER 采用的是自底向上的学习算法,从具体某一个样本的规则归纳为 覆盖全集的范式。RAPIER 系统在执行规则生成的过程中运用了语义和句法的信息。SRV[11] 是一种基于关联的信息抽取系统,采用了自顶向下的归纳式算法进行信息抽取。该系统应用分类算法来完成抽取任务,具有相同大小的文本数据被选取为候选项,这些候选项将在后续作为分类器的输入。候选短语在文本训练数据上的覆盖率是其在规则中出现的概率估计。

近年来,关于通过正负样本来学习正则语言的方法有很多[12-15]。其中大部分的工作通常假设所学习的目标表达式比较简单。例如在 DNA 定序应用的模式学习中[16],输入序列被看作是由序列间隙隔开的多个原子事件。每一个原子事件都可以通过一个简短的正则表达式来表示,于是问题就简化为简单正则表达式的学习问题。在 XML DTD 推断中[17],通常 XML 文件会使用简单的 DTD 来描述文件主题内容,利用 DTD 就可以对 XML 文件信息进行提取。然而,考虑到方法的鲁棒性和结果的准确性,通常需要构建更加复杂的正则表达式来获得更有效的信息抽取。

在信息抽取领域,传统正则学习的方法大多着眼于在相对小的字符表上进行正则表达式的学习[18]。常见的情况是在词性标注[19]、形态分析[20]、词典匹配[21]等文本处理过程之后产生的标注词上进行正则表达式的学习,字符表的大小就由以上分析步骤产生的标注结果所决定。另外,之前的几乎所有工作都将问题限制在一个特定的正则类型中[22],禁用或限制了某些正则符号和操作的使用。

## 2　问题描述

对于一个信息抽取任务,需要获取实体 $\varepsilon$,令 $R_0$ 表示初始输入的正则表达式,$M(R_0,D)$ 表示将 $R_0$ 作用于文本 $D$ 上所得到的结果。令:

$$M_p(R_0,D)=\{x\in M_p(R_0,D):x\text{ 是实体 }\varepsilon\text{ 的实例}\} \quad (1)$$

$$M_n(R_0,D)=\{x\in M_n(R_0,D):x\text{ 不是实体 }\varepsilon\text{ 的实例}\} \quad (2)$$

分别表示 $R_0$ 的正例和反例的匹配结果。任务目标就是输出一个比初始输入的 $R_0$ 效果更好的正则表达式。

对于一个候选正则 $R$,及其所应用的目标文档 $D$,如果想要判断 $R$ 是否比 $R_0$ 有更好的效果,那么就必须对 $R$ 所匹配的结果进行正负分类。如果 $R$ 所匹配的结果是所有已有结果的子集,那么就能够进行判断,因此有如下假定:

**假定 1**　对于字符集 $\Sigma$ 上的正则表达式 $R_0$,任何其他的正则表达式 $R$ 是一个更优候选者当且仅当 $M(R,D)\subseteq M(R_0,D)$。

在此假定的基础上,仍然会得到近乎无限多的候选正则表达式,为了进一步通过学习得到一个最优正则式,需要一个在不同正则表达式之间进行转换的方法,以及评价一个正则表达式抽取信息效果的目标函数。下面给出两者的定义:

**定义 1** 令 $R$ 表示字符集 $\Sigma$ 上所有的正则表达式的集合,正则表达式的转换就是一个函数 $F: R \rightarrow 2^R$, $\forall R_0 \in F(R)$, $L(R_0) \subseteq L(R)$。$L(R)$ 表示 $R$ 所匹配的结果。

举例来说,对于正则表达式 (\d + \ -) + \d + 来说,可以将第二个数量符号 + 替换为一个特定的范围,例如{1,2} 或{3,4},于是就得到了新的表达式(\d + \ -){1,2}\d + ,(\d + \ -){3,4}\d + 。将数量符号 + 替换为一个具体的范围就是正则表达式转换的一个特例,后面会进一步说明,现阶段可以将正则的转换看作是能够产生一系列候选正则集合的函数。下面给出本文学习问题的搜索空间的定义:

**定义 2** 给定一个输入正则表达式 $R_0$ 和一系列的转换 $T$,学习问题的搜索空间为 $T(R_0)$,即将转换 $T$ 重复作用在 $R_0$ 上所得到的所有正则表达式的集合。

选择使用信息检索领域的常用指标 F1-Measure 来比较检索空间中不同正则表达式抽取信息的效果,使用之前提到的 $M_p(R,D)$ 和 $M_n(R,D)$ 分别定义一下指标:

$$p(R,D) = \frac{M_p(R,D)}{M_p(R,D) + M_n(R,D)} \tag{3}$$

$$r(R,D) = \frac{M_p(R,D)}{M_p(R,D) + M_n(R,D)} \tag{4}$$

$$F(R,D) = \frac{2 \times p(R,D) \times r(R,D)}{p(R,D) + r(R,D)} \tag{5}$$

最终的正则表达式学习任务就可以由如下最优化问题来定义:

**定义 3** 对于给定的输入正则表达式 $R_0$,文档集合 $D$,正反例样本集合 $M_p(R,D)$ 和 $M_n(R,D)$ 以及一系列正则转换 $T$, 输出一个最优正则表达式 :

$$R_f = \text{argmax}_{R \in T(R_0)} F(R,D) \tag{6}$$

# 3 正则表达式转换

## 3.1 现代正则引擎

现代正则引擎主要可以分为基本不同的两大类[18]:一种是 DFA(确定型有穷自动机),另一种是 NFA(不确定型有穷自动机)。使用 DFA 的工具主要有 egrep、awk、lex 和 flex。而使用 NFA 的工具包括. NET、PHP、Ruby、Perl、Python、GNU Emacs、ed、sec、vi、grep 的多数版本,甚至还有某些版本的 egrep 和 awk。也有一些系统采用了混合引擎,它们会根据任务的不同选择合适的引擎,甚至对同一表达式中的不同部分采用不同的引擎,以求得功能与速度之间的平衡。NFA 和 DFA 都发展了很多年了,产生了许多不必要的变体,以致现在的情况比较复杂。POSIX 标准的出台,就是为了规范这种现象,POSIX 标准清楚地规定了引擎中应该支持的元字符和特性。

## 3.2 正则表达式转换定义

下面介绍如何利用现代正则引擎的句法规则实现正则表达式的转换,首先考虑这样一个信息抽取任务:从一段文本中找出所有的软件名称,一个简单的模式是 $R$ = ([A - Z]\w * \s * ) + [Vv]? (\d + \.?) +,即为以一个或多个大些字母开头的单词,后面跟一个数字版本号。将 $R$ 应用于实际数据中会发现,一方面 $R$ 确实能够提取出正确的结果,如 Office 2010, Python 2.7, Adobe GoLive 5, Deelex Installer v1.0,另一方面,一些非软件名称的结果也会被提取到,例如 Building 17, Chapter 3.2, ENGLISH 202。针对以上出现的非软件名称结果,大体可以从两个角度来改进:

1) 对于像 ENGLISH 202 这样的全大写单词的课程代码,可以将 R 的前半部分单词的表达式改为 $R_1$ = ([A - Z][a - z] * \s * ) +,即变为以一个大写字母开头的单词,这样就能够过滤掉之前匹配到的错误结果。

2) 另一个改进的方法是利用现代正则引擎中的前向否定符"?!"来显式地去除掉 Building, Chapter, ENGLISH 之类的单词,前向否定就是除了正常的搜索匹配之外继续查找,检查是否出现某些特定内容以达到在正常匹配到内容的基础上,过滤掉某种不想要的内容。举例来说:(?! $R_a$)$R_b$ 返回的是匹配 $R_b$ 并且不匹配 $R_a$ 的内容。回到上面的例子一个改进的方法是将前半部分表达式改为 (?! Building | Chapter | ENGLISH)[A - Z]\w * \s * 。

以上两个角度给出了正则表达式转换的基本原则,通过抽取出一个正则表达式的一部分并对其进行一定的修改以获得原正则匹配结果的真子集。这样的修改分为析取分离与合并交叉两类。析取分离主要作用在由一系列析取项组成的子表达式上,通过分离掉一个或多个析取项从而进行正则的转换。合并交叉则是将某一限定的子表达式与其他表达式进行合并,从而得到转换后的表达式。具体定义如下:

**定义 4** 令 $R$ 是属于 $R_\Sigma$ 中的一个正则表达式,且有 $R = R_a\lambda(X)R_b$,$\lambda(X)$ 表示正则集合 $X = \{R_1, R_2, \cdots, R_n\}$ 上的析取 $R_1 | R_2 | \cdots | R_n$,对于某一 $Y \subset X$,$Y$ 作用于 $R$ 的析取分离转换后的正则为 $DD(R,X,Y)$ =

$R_a\lambda(Y)R_b$。

**定义 5**　令 $R$ 是属于 $R_\Sigma$ 中的一个正则表达式,且有 $R = R_aXR_b$,对于某一 $R_\Sigma$ 中的 $Y$,合并交叉转换后的正则为 $II(R,X,Y) = R_a(X \cap Y)R_b$。

### 3.3　正则表达式转换实现

下面介绍具体如何运用不同的句法规则来实现析取分离和合并交叉的正则表达式转换,包含以下部分:

1) 字符类限制约束

字符类通常表示一系列字符析取结果的缩写,例如元字符\d 表示(0|1|…|9),\w 表示(a|…|z|A|…|Z|0|1…|9|_)。字符类可以用层级结构来表示,其中每一层的节点所包含的字符范围都比他的父亲节点更少。在正则中对某一字符类用其后代节点对其代替就是析取分离转换的一种实现形式。

2) 数量符约束

数量符用来定义一段重复序列出现的次数范围,例如 $x\{a,b\}$ 表示的是序列 $x$ 出现至少 $a$ 次,至多 $b$ 次。由于数量符也可以被看成是其范围中每一项的析取,如 $a\{1,3\}$ 相当于 $a|aa|aaa$,所以如定义4所描述,将某一正则表达式 $R\{m,n\}$ 替换为 $R\{m',n'\}$ $(m \leqslant m' \leqslant n' \leqslant n)$ 也是一种析取分离转换的形式。需要注意的是,在进行这样的析取分离转换之前,需要对 $a+$ 和 $a*$ 这样的通配符限制在一个实现设置好的范围 max 中,转变为 $a\{1,\max\}$ 和 $a\{0,\max\}$。

3) 否定词典

对于一个给定的正则表达式,交叉合并转换可以作用在其每一个有效子表达式上,对于一个正则表达式 $R = R_aXR_b$ 和它的子表达式 $X$,交叉合并转换需要另一个表达式 $Y$ 来得到转换后的表达式 $R_a(X \cap Y)R_b$。由于在转换后所得到的表达式会比原来的表达式有更好的效果,那么可以通过构建所有 $R$ 的反例匹配,找出其中与 $X$ 所对应的匹配字串,然后通过启发式规则找出其中的一部分构成否定词典 $Y'$。于是利用前向否定符构造的正则表达式 $R_a(?!\ Y')XR_b$ 就是利用交叉合并 $R_a(X \cap \neg\ Y')R_b$ 后的结果。

4) 否定词典的启发式构造策略

交叉合并转换过程需要建立在否定词典合理构造的基础上,$S(X)$ 表示正则表达式 $R$ 的反例匹配中对应子表达式 $X$ 的所有字符串集合,理论上说,所有 $S(X)$ 的子集都有可能构成一个否定词典,这样就会有指数级数量的转换可能。为了减少转换的数量,使用一种贪心的启发式策略,对于每一个元素,如果能够单独提高 $F$ 值,那么就将其加入否定词典。

## 4　基于正则表达式学习的信息抽取算法

### 4.1　算法流程

算法 1 描述了对于定义 3 的正则表达式学习问题的算法流程。总体来说,该算法是一个不断迭代的过程,通过对初始输入的正则表达式进行不同形式的转换,在每一轮迭代过程中,得到一组候选正则表达式的集合,并从中贪心地选择在训练集上具有最高 F 值的正则表达式 $R$。为了避免过拟合的情况,当出现以下两种情况中的任一种时,算法就会终止。(1) 当 $R$ 在训练集上的效果没有提升;(2) 当 $R$ 在测试集上的效果下降。

**算法 1**　正则表达式构建学习算法

Input:

$M_{train}$: set of labeled matches used as training data

//训练集数据

$M_{test}$: set of labeled matches used as test data

//测试集数据

$R_0$: user-provided regular expression

//用户输入初始正则表达式

T : set of regular expression transformation

//正则转换集合

Output : one regular expression with highest F-measure

//输出结果

Procedure RegexLearning($M_{train}$, $M_{test}$, $R_0$, T)

0. $R_{new} = R_0$
1. while(true)
2. for each transformation $t_i \in T$　　//遍历正则转换集合
3. $Candidate_i$ = Transformation($R_{new}$, $t_i$)
4. Candidates = $\cup_i$ $Candidate_i$
5. $R' = \text{argmax}_{R \in Candidates} F(R, M_{train})$　　//选择最优结果
6. if ($F(R', M_{train}) <= F(R_{new}, M_{train})$) return $R_{new}$
7. if ($F(R', M_{test}) < F(R_{new}, M_{test})$) return $R_{new}$
8. $R_{new} = R'$
9. end while

End procedure

### 4.2　复杂度分析

下面讨论算法 1 的时间复杂度,根据之前提到的两个算法终止条件,在算法的每一轮迭代中选出的具有最高 F 值得正则表达式 $R'$ 都是严格优于 $R_{new}$ 的,由假定 1 可知,$R'$ 所得到的匹配结果是 $R_{new}$ 匹配结果的子集,所以 $R'$ 所匹配的反例结果严格小于 $R_{new}$ 所匹配的反例结果,因此,总的迭代次数至多为 $|M_n(R_0, M_{train})|$。

对于一个正则表达式 $R$,令 $n_{cc}(R)$ 和 $n_q(R)$ 分别表示 $R$ 中字符类和数量符的个数,$R$ 中子表达式的个数至多为 $|R|^2$,令 $MaxQ(R)$ 表示自定义的 $R$ 中单个数量符所限定的数量范围,$F_{cc}$ 表示字符类层级树中的最大分支数,令 $R_i$ 表示在 $i$ 轮迭代开始时的输入正则,经过正则表达式转换后的候选正则数量为:

$$NumREG(R_i, M_{\text{train}}) \leqslant n_{cc}(R_i) \times F_{cc} + n_q(R_i) \times MaxQ(R_i) + |R_i|^2$$

令 $T_{RE}(D)$ 表示将正则表达式作用于文档 $D$ 上的平均时间。对于字符类替换和数量符限制的析取分离转换时间是与候选正则数量成正比的,而对于通过构造否定词典的交叉合并转换,由于否定词典的贪心式启发式规则,所需要的时间也与子表达式以及反例匹配数量相关。总的构造候选正则表达式集合的时间为:

$$T_{\text{Candidate}}(R_i, M_{\text{train}}) \leqslant c \times (n_{cc}(R_i) \times F_{cc} + n_q(R_i) \times MaxQ(R_i) + |R_i|^2 \times M_n(R_i, M_{\text{train}}) \times T_{RE}(M_{\text{train}}))$$

最后,从候选正则表达式集合中选出最优正则表达式,包括将候选集中的每一个正则作用于训练集以及测试集上的时间:

$$T_{\text{Pick}}(R_i, M_{\text{train}}, M_{\text{test}}) = NumREG \times (T_{RE}(M_{\text{train}}) + T_{RE}(M_{\text{test}}))$$

$$T_i(R_i, M_{\text{train}}, M_{\text{test}}) = T_{\text{Candidate}}(R_i, M_{\text{train}}) + T_{\text{Pick}}(R_i, M_{\text{train}}, M_{\text{test}})$$

$$T_{\text{total}}(R_i, M_{\text{train}}, M_{\text{test}}) = \sum T_i(R_i, M_{\text{train}}, M_{\text{test}}) \leqslant |M_n(R_0, M_{\text{train}})| \times t_0$$

所以每一轮迭代的总时间为:

$$T_i(R_i, M_{\text{train}}, M_{\text{test}}) = T_{\text{Candidate}}(R_i, M_{\text{train}}) + T_{\text{Pick}}(R_i, M_{\text{train}}, M_{\text{test}})$$

由于每一轮迭代的总时间是单调递减的,所以最终算法的时间复杂度为( $t_0$ 是第一轮迭代所需要的时间):

$$T_{\text{total}}(R_i, M_{\text{train}}, M_{\text{test}}) = \sum T_i(R_i, M_{\text{train}}, M_{\text{test}}) \leqslant |M_n(R_0, M_{\text{train}})| \times t_0$$

## 5 实验分析

在本节中,将通过四种不同的信息抽取任务来评估本文中的算法在复杂正则表达式学习问题上的效果,并与常规的机器学习方法进行了比较。实验中用到了数据集有三个,分别为从 yahoo finance 选取的 2000 多个上市公司网页进行爬取所得内容,以及从密歇根大学网站[23]上爬取的 5000 个页面数据和 3000 个的电子邮件内容信息。后两者均为互联网上的公开数据。实验的四个信息抽取任务分别为电话号码、课程代码、超链接、公司名称的信息抽取。实验结果通过 F 值来评估,每个数据集被分为 10 个大小相同的子数据集,分别使用 10%、30%、70% 的数据作为训练集,剩下的作为测试集。实验结果如图 1 – 图 4 所示。

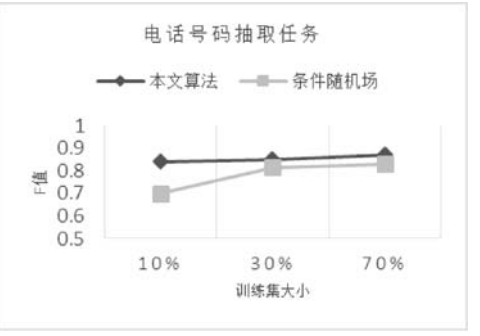


图 1 电话号码任务抽取效果

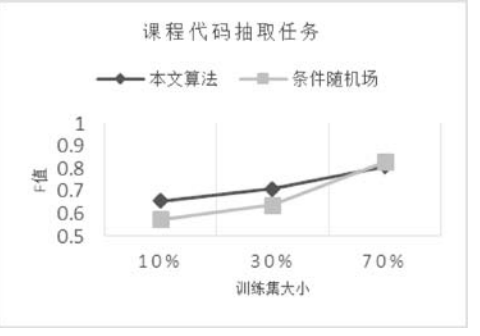


图 2 课程代码任务抽取效果

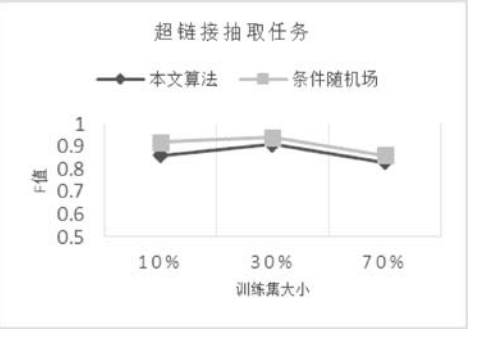


图 3 超链接任务抽取效果

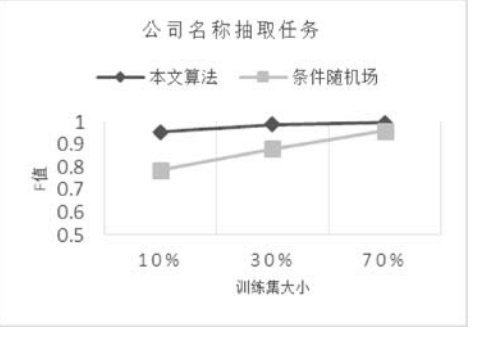


图 4 公司名称任务抽取效果

分析实验结果可见:

如果使用 10% 的数据作为训练数据,对于公司名称、课程代码以及电话号码三个信息抽取任务,本文的算法比传统的条件随机场方法 F 值会有显著提高。

随着训练数据的增加,两种方法的抽取效果均有

提升,两者之间的差别也随之减小。在使用较大的训练集时,本文的方法在公司名称以及电话号码两个信息抽取任务上效果较好,而传统机器学习方法则在另外两个任务上有更优的表现。

以上实验结果说明在有限的训练数据上,本文的算法比传统的机器学习算法有一定的提升。通常情况下,由于获取标注数据通常需要花费大量的时间,所以如何能够在有限的训练集上获得高质量的信息抽取效果是非常重要的。也正因为如此,理想中的情况是在一个训练集上训练出的抽取器能够在其他的数据集上也有较好的效果,表 1 展示了两种方法在跨数据集上的信息抽取效果。可以看出,在不同规模的数据集上,本文的算法均有较好的效果,相比于同数据集上的实验结果,传统的条件随机场方法在跨数据集上的表现有显著下降。

**表 1　跨数据集测试(F 值)**

| 训练集 / 任务 | 10% | | 30% | | 70% | |
|---|---|---|---|---|---|---|
| | RL | CRF | RL | CRF | RL | CRF |
| 公司名称 | 0.865 | 0.356 | 0.927 | 0.427 | 0.841 | 0.354 |
| 电话号码 | 0.368 | 0.159 | 0.664 | 0.213 | 0.489 | 0.167 |
| 超链接 | 0.745 | 0.213 | 0.782 | 0.283 | 0.692 | 0.512 |

## 6　结　语

在信息抽取领域,基于正则表达式的实体抽取一直是一种实际应用中广泛使用的有效方法。本文提出了一种基于正则表达式学习的信息抽取方法,在给定输入初始正则的前提下,通过机器学习的方法找到最优结果。介绍了正则表达式状态转换的概念及其实现方法,并且给出了选取最优正则表达式的算法,通过实验验证了该方法在一些特定类型的信息抽取任务中具有较好的效果。尤其是在较小规模的训练集以及跨数据集上有不错的表现。本文的方法仍有许多不足之处,例如在某些信息抽取任务中无法保持较高的准确率与召回率,可能的原因是在选择最优正则表达式使用贪心的启发式策略,但这也是与时间复杂度的一个权衡。

## 参　考　文　献


[1] McCallum A, Freitag D, Pereira F C N. Maximum entropy Markov models for information extraction and segmentation [C]//Proceedings of the Seventeenth International Conference on Machine Learning, 2000: 591 - 598.

[2] Soderland S. Learning information extraction rules for semi - structured and free text [J]. Machine Learning, 1999, 34 (1): 233 - 272.

[3] Klein D, Smarr J, Nguyen H, et al. Named entity recognition with character-level models [C]//Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003. Association for Computational Linguistics, 2003: 180 - 183.

[4] Carlson A, Betteridge J, Wang R C, et al. Coupled semi-supervised learning for information extraction [C]//Proceedings of the Third ACM International Conference on Web Search and Data Mining. ACM, 2010: 101 - 110.

[5] Wang J, Lochovsky F H. Wrapper induction based on nested pattern discovery: HKUST-CS-27-02 [R]. Technical Report of Hong Kong University of Science and Technology, 2002.

[6] Chiticariu L, Li Y, Reiss F R. Rule-based information extraction is dead! Long live rule-based information extraction systems! [C]//Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013: 827 - 832.

[7] Sekine S. On-demand information extraction [C]//Proceedings of the COLING/ACL on Main Conference Poster Sessions. Association for Computational Linguistics, 2006: 731 - 738.

[8] Kluegl P, Toepfer M, Beck P D, et al. UIMA Ruta: rapid development of rule-based information extraction applications [J]. Natural Language Engineering, 2016, 22(1): 1 - 40.

[9] Fagin R, Kimelfeld B, Reiss F, et al. Spanners: a formal framework for information extraction [C]//Proceedings of the 32nd Symposium on Principles of Database Systems. ACM, 2013: 37 - 48.

[10] 郑家恒,王兴义,李飞. 信息抽取模式自动生成方法的研究[J]. 中文信息学报,2004,18(1):48 - 54.

[11] Fagin R, Kimelfeld B, Reiss F, et al. Document spanners: a formal approach to information extraction [J]. Journal of the ACM (JACM), 2015, 62(2): 1 - 51.

[12] 杨博,蔡东风,杨华. 开放式信息抽取研究进展[J]. 中文信息学报,2014,28(4):1 - 11,36.

[13] Denis F. Learning regular languages from simple positive examples [J]. Machine Learning, 2001, 44(1): 37 - 66.

[14] Denis F, Lemay A, Terlutte A. Learning regular languages using RFSAs [J]. Theoretical Computer Science, 2004, 313 (2): 267 - 294.

[15] Fernau H. Algorithms for learning regular expressions [C]// 16th International Conference on Algorithmic Learning Theory. Springer, 2005: 297 - 311.

[16] Garofalakis M, Gionis A, Rastogi R, et al. XTRACT: a system for extracting document type descriptors from XML documents [N]. ACM SIGMOD Record, 2000, 29(2): 165 - 176.


(下转第 79 页)

分析 ECA 规则能够在系统正式实施前有效地发现集合中规则交互问题并且具有较高的效率。

## 4 结　语

基于 ECA 规则建模的系统以其灵活、使用方便等特点已经应用于相关领域,但是由于规则间的交互问题导致还没有在工业领域得到广泛的应用。本文从实际需求出发,提出将 ECA 规则转换为相应的时间自动机模型,利用形式化方法成熟的验证工具对规则系统进行分析、验证,充分发挥 ECA 规则建模方面的优势,同时利用时间自动机验证方面的优势。

本文提出的方法还有不足之处,在今后的工作中,还需要进一步对方法进行优化和改进,提高方法的有效性和实用性。

## 参　考　文　献


[1] Dittrich K R, Gatziu S, Geppert A. The active database management system manifesto: a rulebase of ADBMS features [C]//Second Workshop on Rules in Database Systems. Springer, 1995: 1 - 17.

[2] 卢涛,刘晓伶. 普适服务冲突检测方法研究[J]. 哈尔滨工程大学学报,2013,34(11):1402 - 1408.

[3] 孙政. 一种基于 ECA 规则的审批工作流模型的浅析[J]. 电子技术与软件工程,2015(15):77 - 79.

[4] 耿盼盼,丁香乾,陶冶,等. 一种通用物联网数据分析平台的设计与实现[J]. 计算机应用与软件,2013,30(11):115 - 118.

[5] Zhang J, Moyne J, Tilbury D. Verification of ECA rule based management and control systems [C]//2008 IEEE International Conference on Automation Science and Engineering, 2008: 1 - 7.

[6] 张立臣,王小明,窦文阳. 基于扩展 Petri 网的 ECA 规则集表示及终止性分析[J]. 通信学报,2013,34(3):157 - 164.

[7] Jin X, Lembachar Y, Ciardo G. Symbolic termination and confluence checking for ECA rules [M]//Transactions on Petri Nets and Other Models of Concurrency IX. Springer, 2014: 99 - 123.

[8] Cano J, Delaval G, Rutten E. Coordination of ECA rules by verification and control [C]//Proceedings of the 16th IFIP WG 6. 1 International Conference on Coordination Models and Languages. Springer, 2014: 33 - 48.

[9] Schordan M, Prantl A. Combining static analysis and state transition graphs for verification of event-condition-action systems in the RERS 2012 and 2013 challenges[J]. International Journal on Software Tools for Technology Transfer, 2014, 16(5): 493 - 505.

[10] Larsen K G, Pettersson P, Yi W. UPPAAL in a nutshell[J]. International Journal on Software Tools for Technology Transfer (STTT), 1997, 1(1): 134 - 152.

[11] Alur R, Dill D L. A theory of timed automata[J]. Theoretical Computer Science, 1994, 126(2): 183 - 235.

[12] Lotfi A, Langensiepen C, Mahmoud S M, et al. Smart homes for the elderly dementia sufferers: identification and prediction of abnormal behaviour[J]. Journal of Ambient Intelligence and Humanized Computing, 2012, 3(3): 205 - 218.

[13] Moshnyaga V, Osamu T, Ryu T, et al. An intelligent system for assisting family caregivers of dementia people[C]//Computational Intelligence in Healthcare and e-health (CICARE), 2014 IEEE Symposium on. IEEE, 2014: 85 - 89.

[14] 侯刚,周宽久,勇嘉伟,等. 模型检测中状态爆炸问题研究综述[J]. 计算机科学,2013,40(6A):77 - 86,111.


(上接第 19 页)


[17] Galassi U, Giordana A. Learning regular expressions from noisy sequences[C]//6th International Symposium on Abstraction, Reformulation and Approximation. Springer, 2005: 92 - 106.

[18] Bex G J, Neven F, Schwentick T, et al. Inference of concise DTDs from XML data[C]//Proceedings of the 32nd International Conference on Very Large Data Bases. VLDB Endowment, 2006: 115 - 126.

[19] Wu T, Pottenger W M. A semi-supervised active learning algorithm for information extraction from textual data: research articles[J]. Journal of the American Society for Information Science and Technology, 2005, 56(3): 258 - 271.

[20] Minkov E, Wang R C, Cohen W W. Extracting personal names from email: applying named entity recognition to informal text[C]//Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2005: 443 - 450.

[21] Chen H, Chiang R H L, Storey V C. Business intelligence and analytics: from big data to big impact[J]. MIS Quarterly, 2012, 36(4): 1165 - 1188.

[22] Cohen W, McCallum A. Information extraction from the world wide web[C]//Tutorial Note at the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2003), 2003.

[23] Intranet Transactional Search[OL]. http://www.eecs.umich.edu/db/transactionalSearch/.