一种软/硬模板相结合的定义抽取算法

钱 菲,袁春风
(南京大学 计算机科学与技术系,江苏 南京 210046)

摘 要:术语定义抽取是信息抽取研究领域的重要内容之一。文中提出了一种结合硬模板匹配和软模板匹配技术的综合术语定义自动抽取方法。文中首先使用硬模板库对待抽取文本进行了初步的定义句匹配抽取。接着,通过使用基于 N 元语言模型的软模板匹配模型来计算待匹配文本中每个句子与软模板之间的匹配度,并通过设定匹配得分阈值来抽取定义句或过滤掉错误召回的非定义句。实验结果表明文中的术语定义抽取方法远远优于单纯的硬模板匹配或软模板匹配方法。

关键词:定义抽取;硬模板匹配;软模板匹配;N 元语言模型;词类格
中图分类号: TP391 **文献标识码**:A **文章编号**:1673-629X(2012)09-0032-05

A Definition Extraction Algorithm Combining Hard Pattern Matching and Soft Pattern Matching

QIAN Fei, YUAN Chun-feng
(Department of Computer Science and Technology, Nanjing University, Nanjing 210046, China)

Abstract: Definition extraction is an important topic in the field of information extraction. It proposes a definition extraction method based on both hard pattern matching and soft pattern matching. Firstly, conduct hard matching on candidate sentences and hard patterns. Secondly, n-gram based soft pattern matching model is used to get a matching score between the candidate sentence and the soft pattern. In the second step, an upper threshold is set to recall candidate sentences with a high matching score; A lower threshold is used to rule out some wrongly-recalled sentences by hard matching. The experimental results show that the proposed definition extraction method is far superior to both pure hard pattern matching and soft pattern matching method.

Key words: definition extraction; hard pattern matching; soft pattern matching; N-gram language model; word class lattice

0 引 言

术语定义的自动抽取是信息抽取研究领域的重要内容之一,它在很多信息检索和信息抽取问题中都有重要的应用,例如术语抽取[1,2]、领域本体构建[1,3]、语义关系获取[4]以及自动问答系统[5,6]等。

术语定义的自动抽取是一个相对较新的领域,目前很多相关研究依赖于基于规则的定义抽取方法。基于规则的方法一般是通过手工建立或者机器学习出术语定义的模板,通过模板的硬匹配,抽取出文本中的术语定义[7~9]。然而,手工编写定义模板的工作量大,并且较为主观,难以穷尽语言现象。为此,文献[10]提出了一种基于词类格(Word Class Lattice, WCL)的规则泛化模型。研究结果表明词类格获取的模板信息具有较高的精确率[11]。但是,该方法对训练语料有较强的依赖性,泛化能力不如手工模板,因而在召回率方面的表现差强人意。

由于在自然语言文本中作者可能会使用不同的表述方式去表达同一个意思,手工模板往往由于一个字(词)的增加或减少而造成匹配失败。针对上述问题,文献[11]提出了一种基于 *N* 元语言模型(n-gram)的软模板匹配方式来进行术语定义的抽取,该方法运用概率模型来反映语言变化,而不要求与模板完全匹配,避免了硬模板匹配时非 0 即 1 的机械性。但是,由于该模型通过设置阈值来界定定义句和非定义句,阈值设定的过高或者过低都会带来问题。若阈值过高,语料数据的稀疏问题会使得非典型定义句匹配得分很低,从而无法召回;若阈值过低,又会使其识别精度难以得到保证。

针对上述不同方法各自存在的问题,文中提出了一种结合硬模板匹配和软模板匹配技术的综合术语定义自动抽取方法。一方面,通过将手工模板和基于词

收稿日期:2012-02-16;**修回日期**:2012-05-25
基金项目:国家自然科学基金资助项目(61072152,61021062)
作者简介:钱 菲(1989-),女,泰州人,硕士,主要研究方向为自然语言处理;袁春风,教授,CCF 高级会员,主要研究方向为 Web 信息检索与文本挖掘技术、多媒体文档处理等。

类格的泛化模板(下文简称“词类格模板”)结合构成一个硬模板库,采用硬模板匹配的方式进行术语定义自动抽取;另一方面,通过采用基于 N 元语言模型的概率计算来得到匹配度的方式进行术语定义句的自动识别。

1　术语定义的自动抽取系统框架

文中综合考虑了手工模板、词类格模板以及基于 N 元语言模型的软模板的优缺点,提出了一种将硬模板匹配和软模板匹配技术相结合的术语定义抽取方式。

图 1 给出了文中的术语定义抽取总体框架。

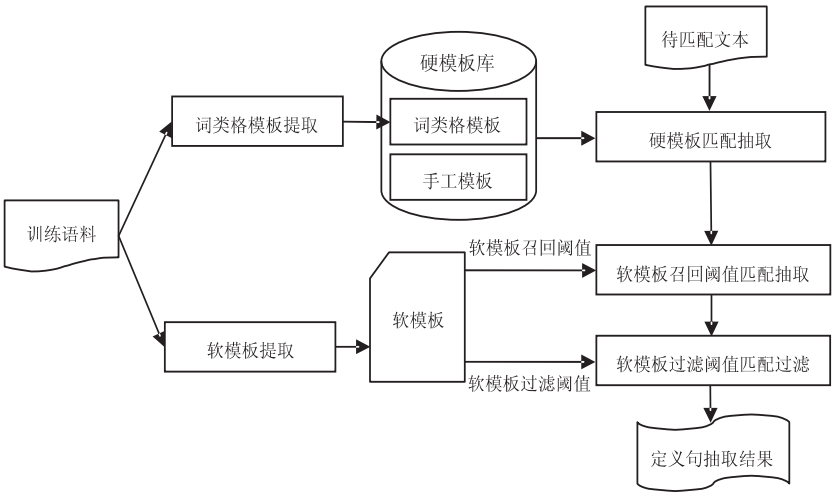

图 1　术语定义抽取总体框架

如图 1 所示,基于硬模板匹配部分所用模板包括手工模板和词类格模板两种。文中参考文献[1,12],对中文文本中术语的定义句结构进行了分析和总结,给出了一组具有代表性的手工模板,并通过定义句的主体部分提取了词类格模板,将这两者共同构成硬模板库。硬模板库中的规则使用正则表达式描述。首先使用硬模板库对待抽取文本进行了初步的定义句匹配抽取。对于软模板匹配部分,采用基于 N 元语言模型的概率模型来表示模板,通过计算待匹配文本中每个句子与模板之间的匹配度来抽取定义句或过滤掉错误召回的非定义句。在进行模板提取处理前,需要对训练语料进行手工标注以及预处理,同样,在进行模板匹配操作前,也需要对待匹配文本进行预处理。

首先为训练语料中的每个定义句进行手工标注,将每个定义句细分为“被定义词”、“连接词”、“定义体”、“补充定义”四个部分,并去除句子中属于补充定义的部分,剩下的用于获取词类格模板。接着,对训练语料和待抽取文本中的所有句子都进行了泛化预处理,后续的模板提取和模板自动匹配工作都是基于泛化后的句子进行的。在对训练语料进行分词和词性标注后,首先将被定义词用<目标>代替,并对每个词进行词频统计,设定阈值以获取频繁词集。将句子中不属于频繁词集的词汇用其相应的词性替代,这样就把一个词泛化为它的词性,将其称为词类(Word Class),并将泛化后的句子的每个词或其词性、标点符号等称为一个标识符(token)。最后删除形容词以及助词标识符,并合并连续相同的标识符。

2　硬模板库的获取与匹配

2.1　手工模板的获取

文中参考文献[1,12]对中文文本中术语的定义句结构进行了总结和分析,得到几种有代表性的定义句模板,使用正则表达式描述如下:

1)(.*)(称为|称|称之为|称作|称做|叫|叫做|即|就是)(“)?<目标>(”)?

2)(“)?<目标>(”)?(主要)?(包括|包含)(.*)

3)(“)?<目标>(”)?,(当且仅当|若|如果)(.*)

4)(如果|若)?(.*),(则)?称(.*)为(“)?<目标>(”)?

5)(所谓)?(“)?<目标>(”)?,(是指|是|指|特指|指的是|就是|即)(.*)

6)(所谓)?(“)?<目标>(”)?(.*)(是|即)

7)(也|又|简)?(称为|叫|称)(“)?<目标>(”)?,它(.*)(是|即)

其中“?”表示可出现一次或者零次的部分,“.*”用于通配所有非换行符号。

2.2　词类格模板的获取

词类格模板是从手工标注以及预处理过的训练语料的定义句中提取得到的。首先对训练语料中的每个定义句按照一定的规则提取出类模板,然后将定义句按照提取出的类模板进行聚类,进而得到每个模板类的一个词类格,该词类格保留了对应模板类中所有定义句的句式信息。一个词类格就是一个硬匹配模板,所有模板类的词类格构成词类格模板库。

假设训练语料中所有定义句的集合为 T,对于每个定义句 $s\in T$,通过下面的方式得到类模板 $\sigma(s)$。假设 $s=w_1,w_2,\cdots,w_{|s|}$,其中 w_i 表示句子中的第 i 个词。F 表示在预处理中得到的 T 的频繁词集。对于 $w_i\notin F$,用 * 代替 w_i,并将相邻的 * 合并。得到训练语料中所有的类模板后,将定义句 $s\in T$ 按照各自对应的类模板 $\sigma(s)$ 进行聚类。假设得到的聚类集合为 $C=(C_1,C_2,\cdots,C_m)$,为每个类 $C_i\in C$ 构建其对应的词类格。

假设 $C_i = \{s_1, s_2, \cdots, s_{|C_i|}\}$，其中，第 k 个句子为 $s_k = \{(w_1^k, w_2^k, \cdots, w_{|s_k|}^k)\}$，$w_j^k$ 表示第 k 个句子的第 j 个标识符。

为了得到 C_i 类的词类格，首先根据 C_i 中的第一个句子 s_1 构建关于 C_i 的一个初始的有向图 $G=(V, E)$，其中，$V = \{w_1^1, w_2^1, \cdots, w_{|s_1|}^1\}$，$E = \{(w_1^1, w_2^1), (w_2^1, w_3^1), \cdots, (w_{|s_1|-1}^1, w_{|s_1|}^1)\}$。然后，对 C_i 其它句子 $s_j (j=2, \cdots, |C_i|)$，通过动态规划的方法计算句子 s_j 与 $s_k (s_k \in C_i, k<j)$ 之间的对齐度(alignment)。

$$A_{m,n} = \begin{cases} \max\{A_{m-1,n-1} + S_{m,n}, A_{m-1,n}, A_{m,n-1}\}, \\ \quad 0 < m < |s_k|, 0 < n < |s_j| \\ 0, \ m = 0 \ || \ n = 0 \end{cases} \quad (1)$$

式(1)中的 $S_{m,n}$ 通过句子 s_k 的第 m 个标识符 w_m^k 以及 s_j 的第 n 个标识符 w_n^j 计算得到：

$$S_{m,n} = \begin{cases} 1, w_m^k = w_n^j \\ 0, o.w. \end{cases} \quad (2)$$

最终的对齐度(alignment)由 $A_{|s_k|,|s_j|}$ 给出，它代表了两个标识符序列的未对齐位置的最小数目。对每个 $s_k (s_k \in C_i, k=1, \cdots, j-1)$ 重复上述计算，并最终选择与 s_j 的对齐度得分最高的那个句子。然后根据动态规划计算的矩阵由 $A_{|s_k|,|s_j|}$ 回溯到 $A_{0,0}$，根据回溯路径将 s_j 加入到图 G 中。将 s_j 中未与 s_k 对齐的标识符加入到顶点集合 V 中，并在 E 中加入边 $(w_1^j, w_2^j), (w_2^j, w_3^j), \cdots, (w_{|s_j|-1}^j, w_{|s_j|}^j)$。

为了提高匹配效率，将得到的词类格模板转化成正则表达式。这样，就获得了手工模板和词类格模板共同构成的硬匹配模板库。

3 基于 N 元语言模型的软模板匹配模型

文中采用基于 N 元语言模型的概率模型来表示软模板，由于语料集大小有限，文中使用了简单的一元(unigram)和二元(bigram)语言模型，将它们有机结合以得到最终的匹配分数。

3.1 软模板的获取

文中的软模板是通过对泛化预处理后的训练语料进行学习而得到的。为了获取有关“被定义词”的上下文信息，选择一个以<目标>为中心的窗口，左、右长度各为 w，这样，就获得了一个包括<目标>在内的大小为 $2w+1$ 的片段。这些定义句的片段被用来生成软模板 SP(Soft Pattern)。

为了表示学习得到的软模板框架，将<目标>两侧的位置称为槽(Slot)。假定选择的左、右窗口都为 w，则得到一个长度为 $2w+1$ 的软模板框架 $<\text{Slot}_{-w}, \cdots, \text{Slot}_{-1}, <\text{目标}>, \text{Slot}_1, \cdots, \text{Slot}_w>$。

对于训练语料中的所有句子，按照上述软模板框架，将其相应片段中的标识符按相对于<目标>的不同位置排列，就得到每个槽 $\text{Slot}_i (i=-w, \cdots, -1, 1, \cdots, w)$ 对应的标识符集合 $\{\text{token}_{i1}, \text{token}_{i2}, \cdots, \text{token}_{im}\}$。根据每个标识符 $\text{token}_{ik} (k=1, 2, \cdots, m)$ 在 Slot_i 处的出现次数可以得到相应的出现概率 p_{ik}。将所有可能在 Slot_i 处出现的标识符及其相应的出现概率用一个集合 $\text{Token\_in\_Slot}_i (i=-w, \cdots, -1, 1, \cdots, w)$ 表示，所有 Token_in_Slot_i 构成的 $2w$ 个集合看成是定义句的软模板 SP，即 $\text{SP} = \{\text{Token\_in\_Slot}_{-W}, \cdots, \text{Token\_in\_Slot}_{-1}, \text{Token\_in\_Slot}_1, \cdots, \text{Token\_in\_Slot}_W\}$。在得到软模板 SP 以后，就可以用它来计算一个待匹配句子与软模板之间的匹配程度。

3.2 软匹配得分的计算

将待匹配的候选句进行同样的泛化以及窗口截取处理后，可以得到候选句的待匹配片段 $C<\text{token}_{-w}, \cdots, \text{token}_{-1}, <\text{目标}>, \text{token}_1, \cdots, \text{token}_w>$。待匹配片段 C 与软模板 SP 之间的匹配分数由两部分组成。

第一部分基于一元语言模型来计算 C 和 SP 之间的相似度，即按照单独的槽来计算，使用以下朴素贝叶斯公式将各槽 Slot_i 的匹配概率综合起来得到一个匹配得分，并在针对各槽的概率计算中加入了拉普拉斯平滑因子：

$$\text{Score}_{\text{slots}} = Pr(C \mid \text{SP}) = \prod_{i=-w}^{w} Pr(\text{token}_i \mid \text{Slot}_i) \quad (3)$$

由于用这个公式计算匹配度时只考虑了单独的槽，因而是十分灵活的，即便有些槽匹配得不好，它仍然可以根据其他槽来给出相应的匹配程度，而不会像硬模板匹配中那样直接认定匹配失败。

第二部分使用二元语言模型来计算待匹配片段 C 对应的标识符序列的出现概率。给定一个 token 序列 T，条件概率 $Pr(T|\text{SP})$ 表示通过 SP 计算得到的 T 的出现概率。对于待匹配片段 C，分别计算从<目标>开始左、右两个标识符序列的概率值，即 $Pr(\text{left}_{\text{bigram}}|\text{SP})$ 和 $Pr(\text{right}_{\text{bigram}}|\text{SP})$。整个片段 C 的第二部分的得分由<目标>左右两侧的标识符序列的出现概率加权相加得到。

$$\text{Score}_{\text{bigram}} = (1-\alpha).Pr(\text{left}_{\text{bigram}}|\text{SP}) + \alpha.Pr(\text{right}_{\text{bigram}}|\text{SP}) \quad (4)$$

根据对术语定义句的观察分析，发现<目标>右侧蕴含了更多有助于定义句判定的上下文信息，因而将 α 设定为 0.7。由于二元语言模型会有数据稀疏的问题，加入了拉普拉斯平滑。

最后，将两部分的得分综合加权相加得到最终的匹配分数：

$$\text{Score}_{\text{match}} = \frac{\gamma.\text{Score}_{\text{slots}} + (1-\gamma).\text{Score}_{\text{bigram}}}{\text{fragment\_length}} \quad (5)$$

实验中,将式(5)中的 γ 值设置为 0.9。

4　实验及分析

4.1　数据集

在实验中,使用了两个语料资源。

第一个是从中文维基百科中抽取出的 2500 个句子。其中包括 969 个定义句,以及 1531 个非定义句。定义句来自随机抽取的维基页面的第一个句子。为了能够得到具有代表性的、领域无关的定义句模式,抽取的维基页面包含了计算机、地形、气象、医药、社会学、流体力学 6 个领域。非定义句则是这些维基页面中包含了页面标题(即该页面所对应词条)的其它句子。记为 $\text{Corpus}_{\text{wiki}}$。

第二个是从 Web 网页中随机抽取出的 9436 个句子。其中包括了 1208 个定义句,以及 8228 个非定义句。从与 $\text{Corpus}_{\text{wiki}}$ 中相同的 6 个领域选取出 250 个术语,并将它们分别作为关键词使用 Google 搜索引擎进行检索。对每个关键词抓取返回的前 20 个网页,并抽取出包含关键词的句子。记为 $\text{Corpus}_{\text{web}}$。

对语料集进行了手工标注以及预处理。具体步骤已在第 1 节中说明,在此不再赘述。

4.2　实验设计

为了证明文中的术语定义抽取方法的有效性,对表 1 所示的几组定义抽取方案进行了实验。其中,定义抽取方案 1、2、3 是指单独使用软模板、手工模板或者词类格模板进行定义句的匹配抽取,用作文中方法实验结果的参照。

表 1　定义抽取方案

编号	定义抽取方案
1	软模板
2	手工模板
3	词类格模板
4	硬模板库(手工模板 + 词类格模板)
5	硬模板库 + 软匹配召回阈值 θ_1
6	硬模板库 + 软匹配召回阈值 θ_1 + 软匹配过滤阈值 θ_2

在实验中,首先使用硬模板匹配的方式进行初步的定义句抽取,抽取结果分为两部分:

1、手工模板召回的句子集合,记为 $R_{\text{Handcraft}}$;

2、词类格模板召回的句子集合,记为 R_{WCL}。

在对语料中大量句子的软模板匹配得分进行统计分析后,发现定义句和非定义句的软模板匹配得分在统计上有较显著的差别,定义句的平均得分要远高于非定义句的得分。因此,分别设置软匹配召回阈值 θ_1 以及过滤阈值 $\theta_2(\theta_1 > \theta_2)$ 进行定义句的补充召回和过滤。阈值 θ_1 召回的定义句集合记为 R_{θ_1}。由于词类格模板的精确率较高,我们对其结果不作进一步的过滤。阈值 θ_2 用于对 $R_{\text{Handcraft}} - R_{\text{WCL}}$ 中的句子进行过滤,过滤后的集合记为 $R_{\text{Handcraft}+\theta_2}$。最终,$R_{\text{WCL}} \cup R_{\theta_1} \cup R_{\text{Handcraft}+\theta_2}$ 为系统抽取出的定义集合。

为了更真实地反映文中的定义抽取方法的效果,实验采用了 10 重交叉验证法。

4.3　实验结果与分析

表 2 和表 3 分别给出了表 1 中的术语定义抽取方案在中文语料 $\text{Corpus}_{\text{wiki}}$ 和 $\text{Corpus}_{\text{web}}$ 上的实验结果,有关各指标的最高取值用粗体标出。除了传统的 P、R 以及 F1 指标,定义 A = 标记正确的句子/测试集句子总数 * 100%。

表 2　在 $\text{Corpus}_{\text{wiki}}$ 上的实验结果

编号	P	R	F1	A
1	71.64	82.77	76.57	80.98
2	68.40	90.71	77.99	80.15
3	**94.39**	35.32	51.40	71.12
4	68.59	91.95	78.57	80.56
5	67.66	**96.1**	79.41	80.68
6	78.87	88.95	**83.61**	**86.48**

表 3　在 $\text{Corpus}_{\text{web}}$ 上的实验结果

编号	P	R	F1	A
1	55.04	69.22	61.32	88.81
2	47.41	82.90	60.31	86.04
3	**83.72**	40.15	54.27	91.33
4	48.46	86.92	62.23	86.49
5	48.68	**91.63**	63.58	86.56
6	75.21	74.84	**75.02**	**93.62**

从实验结果可以得出以下几个结论:

1)实验结果证实了词类格模板的高精确性以及在提升定义抽取效果上的有效性。在获取词类格模板的过程中,保证了某种定义模式只要出现过,其相应的句式信息就会被保留在词类格模板里面,因而能够精确地将某些非典型定义句补充召回。

2)实验表明在大部分情况下,软模板和手工模板是等效的,能成功匹配手工模板的句子,往往软模板匹配得分也高。这一方面表明了软模板的合理性,另一方面,这一等效性刚好可以用来对基于硬模板的定义句抽取结果进行过滤以及补充召回。

3)对比方案 1、2、6 的实验结果,发现在精确率这个指标上,将手工模板与软模板匹配过滤阈值相结合远优于单独的基于手工模板或者软模板的定义句抽取方法,尤其是在 $\text{Corpus}_{\text{web}}$ 上。这主要是由于这种结合方式融合了两种定义句抽取方法的优点,相当于对候选句进行了“双重认证”。

4)对比方案 2、4、5 的实验结果,可以看出,将基于硬模板库的定义句匹配抽取与软模板匹配的召回阈值

相结合后,在两个语料集上都使得召回率达到了实验中的最大值。然而,精确率指标在 $Corpus_{wiki}$ 上在此过程中不升反降,这说明基于阈值 θ_1 的补充召回带入了较大的噪声,降低了精确度。

5)由方案 5 和 6 的结果可以看出,通过设定软匹配阈值来进行候选定义句的召回和过滤,存在着对精确率和召回率方面的权衡,一个指标的上升总是伴随着另一个指标的下降。这也从一个侧面说明,试图在单特征上使用线性的分类指标去判别定义句和非定义句是不合理的,要达到更好的定义句抽取效果,势必需要加入更多的特征。这也是未来研究工作的一个方向。

实验结果表明,文中提出的术语定义句抽取方法在两个语料集上都达到了较好的效果,是明显优于单独的硬模板匹配以及软模板匹配方式的。

5 结束语

文中在分析现有的基于规则的术语定义句抽取方法不足的基础上,提出了一种将硬模板匹配与软模板匹配技术相结合的综合术语定义句抽取方法。在两个语料集上的实验结果验证了文中方法的有效性,以及相对于单独的硬模板匹配以及软模板匹配方式的优越性。

在未来的研究工作中,会尝试加入更多的语料和其他新的特征,来综合进行定义句和非定义句的分类,以期达到更好的效果。

参考文献:

[1] 荀恩东,李 晟. 采用术语定义模式和多特征的新术语及定义识别方法[J]. 计算机研究与发展,2009,46(1):62-69.

[2] 张 榕,宋 柔. 一种被定义项的识别策略[J]. 当代语言,2007,9(1):33-38.

[3] Gangemi A, Navigli R, Velardi P. The OntoWordNet project: extension and axiomatization of conceptual relations in WordNet[C]//Proceedings of the International Conference on Ontologies, Databases and Applications of Semantics(ODBASE 2003). Catania, Italy:[s. n.],2003:820-838.

[4] Snow R, Dan Jurafsky D, Ng A Y. Learning syntactic patterns for automatic hypernym discovery[C]//Proceedings of Advances in Neural Information Processing Systems. [s. l.]:MIT Press,2005:1297-1304.

[5] Cui Hang, Kan Min-Yen, Chua Tat-Seng. Soft pattern matching models for definitional question answering[J]. ACM Transactions on Information Systems(TOIS),2007,25(2):1-30.

[6] 陈 议. 开放域的自动问答系统的研究[D]. 重庆:重庆大学,2006.

[7] Liu Bing, Chin C W, Ng H T. Mining topic-specific concepts and definitions on the web[C]//Proceedings of the 12th international conference on world wide web. Budapest, Hungary:[s. n.],2003.

[8] 贾爱平. 科技文献中术语定义语言模式研究[D]. 北京:北京语言文化大学,2002.

[9] 张 艳,宗成庆,徐 波. 汉语术语定义的结构分析和提取[J]. 中文信息学报,2003,17(6):9-16.

[10] Navigli R, Velardi P. Learning word-class lattices for definition and hypernym extraction[C]//Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Uppsala, Sweden:[s. n.],2010:1318-1327.

[11] Cui Hang, Kan Min-Yen, Chua Tat-Seng. Unsupervised learning of soft patterns for generating definitions from online news[C]//Proceedings of the 13th international conference on world wide web. New York, NY, USA:[s. n.],2004.

[12] 张 榕,宋 柔. 基于互联网的汉语术语定义提取研究[C]//全国第八届计算语言学联合学术会议. 北京:清华大学出版社,2005:428-434.

(上接第 31 页)

McCulloch-Pitts Neural Model and Its Applications[J]. IEEE Trans. on Neural Networks,1999,10(4):925-929.

[7] Wu Tao, Mao Junjun, Gao Liang, et al. Covering Algorithm Based on Neighborhood Search and Its Applications[C]// Third International Conference on Natural Computation (ICNC 2007). [s. l.]:[s. n.], 2007:115-119.

[8] 李丽芳,周鸣争. 一种基于构造性核覆盖的聚类算法[J]. 计算机技术与发展,2009,19(1):88-91.

[9] Wang Di. Fast constructive-covering algorithm for neural networks and its implement in classification[J]. Applied Soft Computing,2008(8):166-173.

[10] 贾瑞玉,冯伦阔. 基于集成学习的覆盖算法[J]. 计算机技术与发展,2009,19(7):76-79.

[11] 李文娟,胡春生. 基于聚类优化覆盖的集成学习方法[J]. 计算机技术与发展,2010,20(11):51-54.

[12] 赵 姝,张燕平. 覆盖聚类算法[J]. 安徽大学学报(自然科学版),2005,29(2):28-32.

一种软/硬模板相结合的定义抽取算法

作者：钱菲， 袁春风
作者单位：南京大学 计算机科学与技术系,江苏 南京 210046
刊名：计算机技术与发展
英文刊名：Computer Technology and Development
年，卷(期)：2012(9)