一种构件标签自动提取方法及其实现

刘 飞,王立杰,李 戈,赵俊峰,谢 冰
(北京大学信息科学技术学院软件研究所,北京 100871)

摘 要: 基于构件的软件复用的前提之一是存在并能够找到大量可复用的构件,软件构件库是对软件构件进行管理的基础设施,其作用是对大量构件进行管理,并辅助软件开发者找到合适的构件.在构件库中,基于构件标签的分类管理是一种新型的构件信息分类管理方法,该方法使用构件标签(Tag)对构件进行管理,并支持用户通过选择标签进行构件检索,该方法能够更直接的反应构件的特性,并能够有效提高检索效率.然而,由于构件库中许多构件没有构件标签,而通过人工的方法为构件库中存在的构件资源添加标签需要耗费大量的时间和人力资源,特别是当构件数量较大时,通过人工方式为构件添加标签是难以实现的.因此本文提出了一种基于分类的构件标签自动提取方法,该方法能够根据构件描述信息自动提取构件标签.本文对基于该方法的构件标签自动提取工具的实现进行了论述,并通过实验验证了该工具的有效性.

关键词: 软件复用; 构件库; 构件标签
中图分类号: TP311 **文献标识码:** A **文章编号:** 0372-2112 (2010) 2A-045-05

An Automatic Component Tag Extracting Method and Implementation

LIU Fei, WANG Li-jie, LI Ge, ZHAO Jun-feng, XIE Bing
(*Institute of Software, School of Electronic Engineering and Computer Science, Peking University, Beijing* 100871, *China*)

Abstract: The existence and being able to find a large number of reusable components is one of the prerequisites of component-based software reuse. Software component library is the infrastructure of component management. It can help developers find appropriate components. Component tag based taxonomy is a new method to manage components in component library. It uses component tag to manage component and supports users using tags to find components. This method is efficient in component management because component tag represents the characteristics of components. However, many components don't have component tags in component library. Adding tags for these components manually needs a lot of time and workload. It's almost impossible when there are a lot of such components. So, this paper proposed an automatic component tag extracting method based on classification. This method extracts component tags from component descriptions automatically. This paper described a tool based on this method and verified the effectiveness of the tool.

Key words: software reuse; component library; component tag

1 引言

软件复用是公认的并被实践证明了的能够切实有效地提高软件开发效率和软件质量的有效途径.软件复用中产品复用的基本单元是构件(Component),构件是指软件系统中具有相对独立功能,可以明确辨识的构成成分[1].随着软件产业的发展和人们对构件技术研究的进展,出现了大量可复用构件.对构件进行有效地管理成为人们有效利用这些构件资源的关键.构件管理是对构件进行描述、分类、存储和检索的过程[2].构件库能够为构件管理提供全面的支持.

使用构件标签对构件进行分类管理是一种新型的构件分类管理方法,该方法支持用户为构件添加标签并通过标签对构件进行检索.构件标签(以下简称标签)是描述构件特点和构件功能的关键词.当用户向构件库中发布一个构件时,用户可以为该构件添加相应的标签,一个用户也可以向其他用户发布的构件添加标签.当用户使用标签方式查找构件时,用户可以通过查看其添加过的标签以浏览他发布的构件.同时,用户也可以通过查看其他人添加的标签以查找其他人发布的构件.由于标签反映了构件本身的特点和功能,并且标签本身具有表达上高效的特点,使用标签方式管理构件能够使用户

收稿日期:2009-09-07;修回日期:2010-01-21
基金项目:国家高技术研究发展计划(863 计划)(No.2007AA010301,2009AA010307);国家创新研究群体(No.60821003);国家自然科学基金(No.60803010,60803011)

更方便的找到他想要的构件.

然而,构件库中已有的很多构件,用户没有为该构件添加相应的标签.这使得用户在使用标签方式查找构件时,很多构件因标签的缺失而不能被查找到.为了使这些构件能够更好的被用户查找并复用,需要对构件库中没有标签的构件添加标签.由于构件库中不含有标签的构件数量很多,人工为这些构件添加标签需要很大的工作量,并且用户需要理解构件的特点及用途后才能给构件添加标签,这进一步增加了添加标签的难度.因此,本文提出了一种基于分类的标签自动提取方法,并实现了基于该方法的标签自动提取工具,自动为构件库中已有的不存在标签的构件自动添加标签.最后,本文通过实验验证了该工具的有效性.

本文接下来的结构如下:第二部分介绍本文的研究背景.第三部分介绍标签自动提取方法.第四部分对工具进行介绍并通过实验验证了该工具的有效性.最后,本文给出总结和未来的工作.

2 研究背景

软件复用和软件构件库在软件开发中的巨大作用已经得到了广泛的重视.随着互联网技术的发展,出现了很多基于互联网的构件库管理系统,例如 ComponentSource[3]、Sourceforge[4]等,在国内比较有代表性的有北京大学研制的青鸟软件构件库管理系统(JBCLMS)[5].

构件库为支持用户有效地利用构件资源进行复用,对构件进行分类和提供检索.对构件进行分类有便于组织管理,方便检索和辅助理解的好处.构件库中常用的分类方法有刻面分类法、枚举分类法、属性-值分类法和关键词分类法.刻面分类法从不同的角度对构件进行精确的分类,刻面分类法存在的问题是刻面分类结构很难做出修改;枚举分类法是将一个领域划分为不相交的子领域,子领域再进行进一步划分而构成的层次结构分类,这种方法存在的问题是对构件的查找定位不准确;属性-值分类法根据构件属性和对应的取值对构件进行分类;关键词分类法是对构件的赋予一组关键词而进行检索的方法.

近些年来,基于标签的方法在构件管理中被广泛的使用,标签方法是一种特殊的关键词分类方法,它与关键词分类方法不同之处在于所有的用户都可以为构件添加标签.标签的使用十分灵活,用户通过标签方式能够有效对构件库中的资源进行管理,增加了构件被复用的可能.

3 标签自动提取方法

对构件库中不存在标签的构件提取标签的方法的整体框架如图 1 所示.其基本思想是:将构件库中的构件分成两部分,一部分是含有标签的构件,另一部分是不含有标签的构件.通过预处理把含有标签的构件转化成构件向量,将构件库中存在的每个标签作为一个标签类别并添加其他一些可能的标签类别.对于一个待提取标签的构件,将其转化为构件向量,利用判断这个构件是否在某些标签类别中,以确定该构件是否应该包含相应的标签.

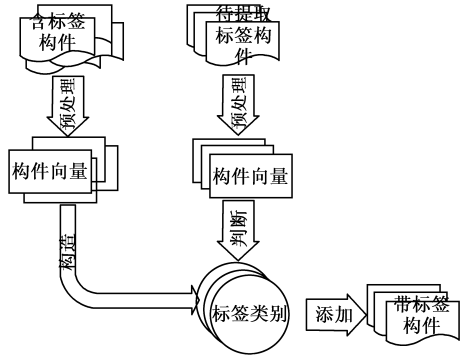


图1 标签自动提取算法框架

3.1 标签自动提取方法概述

为便于方法描述,此处给出相应方法描述部分各个术语的定义:

构件:由一个三元组表示,$comp$ = 〈构件名称,构件描述信息,{构件标签}〉,构件描述信息是文本;

构件向量:是构件 $comp$ 的向量表示,$compv$ = 〈w_1, w_2,…, w_m〉,其中每一维表示一个词,w_i 表示第 i 维上的权重.

标签类别训练数据:是标签类别的集合,所有的标签类别存在于这个集合中,$TagTrainSet$ = {T_1, T_2,…, T_n},n 表示标签类别集合中的包含的标签类别的个数.

标签类别:是一个集合,该集合 T_i 对应标签 Tag_i,集合中包含的元素是构件向量,构件向量满足其对应的构件包含标签 Tag_i 或构件向量在 Tag_i 维度上的权值超过阈值.

标签的提取步骤如下:

(1)从构件库中获取构件及构件的描述信息,将含有标签的构件作为训练数据.

(2)通过构件的预处理将构件转化为构件向量.

(3)根据构件中包含的标签及其他候选的标签生成标签类别训练数据.

(4)对于每一个待提取标签的构件,通过预处理生成构件向量.

(5)判断待提取标签的构件是否属于某些标签类别中,如果属于某标签类别,则将该标签加入到该构件的标签集合中.

下面,对各个步骤做详细的说明.

3.2 构件描述信息的预处理

3.2.1 构件描述词的提取

从构件库中获取构件名字、描述信息以及该构件所包含的标签.其中含有标签的构件作为训练数据.首先提取构件的描述信息:一些构件的描述信息是纯文本形式,不用做特殊处理;一些构件描述信息保存在PDF文档中,使用工具 PDFBox[6]对 PDF 中的文本内容进行提取;一些构件描述信息是保存在 WORD 文档中,使用工具 POI[7]对 WORD 文档中文字内容进行提取.将提取出的构件描述信息全部以纯文本形式保存,以便后续处理.

将构件描述信息文本进行分词,去除标点,停用词等.对于英文文本,分词只需按照空格进行分割,同时还需要提取英语单词的词根,采用基于规则的 Porter Stemmer[8]算法来进行词根的提取.对中文的处理比较困难,中文分词基本方法有基于词表的方法和统计的方法.本文采用基于词表的方法中的最大双向匹配的方法.分词之后需要去除没有意义的词以及标点符号等.另外,由于有一些词经常出现且对提取标签没有帮助,如"的"、"是"等,将其去除.

3.2.2 构件描述词的筛选

通过对构件的描述词进行提取之后,得到的构件向量空间维数非常高,需要对构件向量空间进行构件描述词的筛选以降低构件向量空间中的维数.对构件描述词筛选采用特征提取的方法,特征提取不仅可以降低构件特征向量空间的维数,还能提高标签提取的速度和准确度.特征提取常用的方法有:互信息法,信息增益,CHI 等方法[9].本文中选用 CHI 的特征提取算法.CHI 使用如下公式计算词 w 和标签 t 的相关性:

$$\chi^2(w,t)=\frac{N\times(AD-CB)^2}{(A+C)\times(B+D)\times(A+B)\times(C+D)} \quad (1)$$

其中 A 为词 w 和标签 t 同时出现的次数;B 为 w 出现而 t 没有出现的次数;C 为 w 没有出现而 t 出现的次数;D 为 w 和 t 都没有出现的次数.取词 w 在多个标签类别中的 χ 的最大值作为词 w 的特征值,将特征词按照特征值由大到小排序后,取前 20% 的特征词作为特征项.

3.2.3 构件描述词词权重的计算

经过构件特征词筛选之后,计算构件特征词在构件向量中的权重.如何计算词的权重会直接影响构件向量间距离的结果,对词的权重的计算是十分重要的[10].对每个词的权重计算方法采用 TF * IDF 计算方法,公式如下所述,其中 $df(w)$为描述信息中包含词 w 的构件的个数,$num(c,w)$表示词 w 在构件 c 的描述信息中出现的次数,$maxNum(c)$表示在构件 c 的描述信息中出现次数最多的词的出现次数.在计算 $tf(c,w)$时除以 $maxNum(c)$是为了防止某些构件的描述信息很长而做的归一化.

$$weight(c,w)=tf(c,w)*idf(w) \quad (2)$$

$$tf(c,w)=num(c,w)/maxNum(c) \quad (3)$$

$$idf(w)=\log(N/df(w)) \quad (4)$$

3.3 标签自动提取

3.3.1 标签类别训练数据的构造

通过构件描述信息预处理步骤后,每一个构件形成相应的构件向量,接下来构造标签类别训练数据.将每一个标签作为一个标签类别,如果一个构件包含该标签,则将这个构件的构件向量加入到这个标签类别之中.对每一个经过筛选过的构件描述词,如果该描述词不是标签,且存在某构件向量在该描述词的维度上的归一化后的权值超过一定阈值,那么也将其作为构件标签类别,这个类别中包含的元素为所有在该描述词的维度上归一化后权重超过该阈值的构件向量.

这样,就形成了标签类别训练数据 *TagTrainSet*,其中标签类别 T_i 对应标签名称 Tag_i.

3.3.2 构件标签的提取

对每个待提取标签的构件,利用构件描述词的抽取、筛选和权值计算后得到构件向量.判断构件是否应该添加某标签可以通过判断该构件是否属于该标签类别来决定.如果该构件属于该标签类别,则将这个标签类别对应的标签加入到该构件的标签集合之中;否则,这个标签不应被加入到该构件的标签集合之中.

对于某一构件 *comp*,计算其构件向量 *compv* 与某一标签类别 T_i 的距离,该构件向量 *compv* 与该标签类别 T_i 间的距离为 *compv* 到该标签类别 T_i 中包含的所有构件向量的距离的平均值,即

$$avg\_dis(compv,t_i)=\frac{\sum_{comp_t\in t_i} dis(compv,comp_t)}{|t_i|} \quad (5)$$

其中 $|t_i|$为 t_i 中包含的构件向量的个数,$dis(\vec{x},\vec{y})$表示两个构件向量间的距离,计算公式如下:

$$dis(\vec{x},\vec{y})=\frac{\sum_k x_k\cdot y_k}{\sqrt{\sum_i x_i^2}\cdot\sqrt{\sum_j y_j^2}} \quad (6)$$

```
输入:某构件向量表示 compv,
      构件标签类别训练集合 tagTrainSet,
      构件向量与标签类别阈值 threshold,
      标签个数 K
输出:该构件的构件标签列表
1.  extractComponentTag(compv, tagTrainSet, threshold, K)
2.    result←{}
3.    for each ti in tagTrainSet, then
4.      dis = calAvageDistance(compv, ti);
5.      if (dis > threshold) then
6.        result←result∪{<ti, dis>}
7.      endif
8.    endfor
9.    sort result according to distance
10. return max K result;
```

图 2　标签自动提取算法

当计算了 $compv$ 与 T_i 之间的距离之后,如果该距离值大于一定的阈值,那么将 T_i 所对应的标签 Tag_i 加入到构件 $comp$ 的候选标签列表中.当计算完该构件向量与所有标签类别距离后,将候选标签列表中的标签按照距离值由大到小的顺序排序,选取距离值最大的 K 个标签作为该构件的标签.该过程的伪代码如图 2 所示.

4 工具实现及实验验证

4.1 工具实现

标签自动提取工具使用 Java 开发,图 3 是本工具的实现,该工具有三个面板,分别对应三个视图.第一个视图为构件训练数据视图,在构件训练数据视图中,可以查看作为训练数据的构件所包含的标签,也可以通过该视图的添加或修改标签的功能来修改这些构件的标签.在标签类别视图中,可以查看所有标签类别以及在每个标签类别中所包含的构件.第三个视图为提取标签结果视图,可以在这个视图中查看对没有标签的构件自动提取的标签的结果.

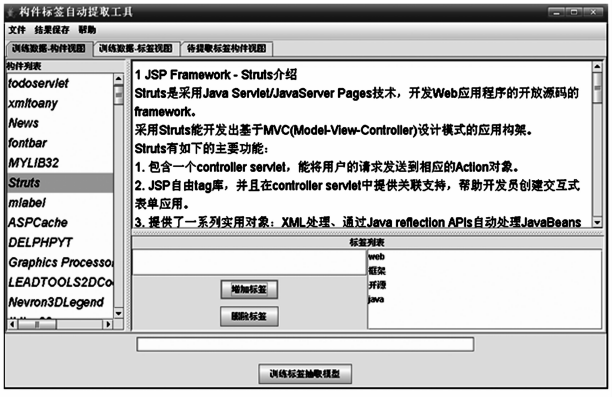


图3 构件标签自动提取工具

4.2 实验验证

首先本文对标签提取工具提取标签的准确性进行了实验验证.然后本文挑选了一些构件库中不包含标签的构件作为例子,使用标签自动提取工具为这些构件提取标签,以此来进一步说明本文工作的有效性.

本文使用查准率、查全率和 $F1$ 值来评价该工具提取标签的准确性.查准率反映该工具提取的标签的准确程度.查全率反映该该工具是否能够把应该有的标签全部提取出来. $F1$ 是查准率和查全率的综合考量.查准率,查全率和 $F1$ 值计算如下:

$$查准率 = \frac{提取的正确构件标签的个数}{提取出的总构件标签的个数} \quad (7)$$

$$查全率 = \frac{提取的正确构件标签的个数}{构件应有的构件标签的个数} \quad (8)$$

$$F1 = \frac{2 \times 查准率 \times 查全率}{查准率 + 查全率} \quad (9)$$

从北京大学软件构件库中选取出带有标签的构件共 105 个.将这些构件随机分成 A, B, C 三组,每组包含 35 个构件,分别选取两组作为训练数据,剩下的一组作为测试数据,使用标签自动提取工具分别对测试数据进行标签的提取.方法中训练数据构造时的阈值为 0.41,标签提取算法中 K 取 4, $threshold$ 取 0.17,得到的结果见表 1.

表 1 交叉验证试验结果

训练/测试	查准率	查全率	$F1$
$A, B/C$	0.725	0.846	0.781
$A, C/B$	0.756	0.883	0.814
$B, C/A$	0.744	0.881	0.807

从实验结果中可以看出,该标签自动提取工具可以得到比较好的查准率和查全率.其中,有一些标签提取的结果存在错误,原因是含有这个标签的构件较少,导致当判断一个构件是否应该包含这个标签时容易出现偏差.当某个标签类别所含有的构件的数量比较多时,判断该构件是否应该包含该标签就比较准确了.

为了进一步说明本文工作的有效性,下面举几个为不含标签的构件提取标签的例子.将从构件库中取出的含有标签的 105 个构件作为训练数据,从北京大学软件构件库中取出 4 个不包含标签的构件,使用标签自动提取工具为这些构件自动提取标签.其中训练数据构造时的阈值为 0.41,标签提取算法中 K 取 4, $threshold$ 取 0.17.例子如下表 2 所示:

表 2 提取标签结果实例

构件名称	标签个数	标签列表
JUnit	4	测试,框架,工具,开源
Hibernate	3	数据库,开源,框架
Spring	4	web,框架,开源,IoC
Resin	4	Java,xml,ejb,服务器

从表 2 中可以看出,该使用该工具为构件提取的标签是比较很好的.

5 结束语

本文提出了一种构件标签自动提取方法,实现了基于该方法的构件标签自动提取工具,并通过实验验证了该工具的有效性.

构件标签自动提取方法需要构件库中存在一定数量含有构件标签的构件作为训练数据.当训练数据较少时,可以通过人工对一些不含标签的构件加入标签,并将这些加入构件标签的构件作为训练数据.训练数据数量的多少对自动提取构件标签的准确性的影响,是未来研究的问题之一.

资源库中不同类型的构件各自存在着一些特点,可以利用这些特点来进行标签的抽取,如 Web Service 的描述文件 WSDL 中含有所提供功能方法和服务的名

字,这里面存在一些词可以作为该 Web Service 的标签.下一步可以在该方法的框架下,具体考虑不同形态构件之间的标签的抽取.

参考文献:

[1] 杨芙清,梅宏,李克勤.软件复用与软件构件技术[J].电子学报,1999,27(2):68 – 75.
Yang Fuqing, Mei Hong, Li Keqin. Software reuse and software component technology[J]. Acta Electronica Sinica, 1999, 27(2):68 – 75. (in Chinese)

[2] 赵俊峰.构件库反馈管理及运行时应用支持技术的研究[D].北京:北京大学,2005.

[3] Component Source. The Definitive Source of Software Components[DB/OL]. http://www.componentsource.com/, 2009.

[4] Geeknet Inc. Sourceforge[DB/OL]. http://sourceforge.net/, 2009.

[5] 杨芙清,梅宏,李克勤.支持构件复用的青鸟 III 型系统概述[J].计算机科学,1999,26(5):50 – 55.
Yang Fuqing, Mei Hong, Li Keqin. An introduction to JB3 system supporting component reuse[J]. Computer Science, 1999, 26(5):50 – 55. (in Chinese)

[6] The Apache Software Foundation. Apache PDFBox[DB/OL]. http://pdfbox.apache.org/, 2009.

[7] The Apache Software Foundation. The Apache POI Project [DB/OL]. http://poi.apache.org/, 2009.

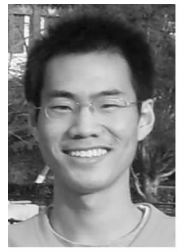
[8] Martin Porter. The Porter Stemming Algorithm [DB/OL]. http://tartarus.org/~martin/PorterStemmer/, 2009.

[9] Yang Y, Pedersen J O. A comparative study on feature selection in text categorization[A]. Proceedings of the Fourteenth International Conference on Machine Learning[C]. Nashville, Tennessee: Morgan Kaufmann Publishers, 1997. 412 – 420.

[10] C Buckley. The importance of proper weighting methods[A]. Proceedings of the workshop on Human Language Technology [C]. Princeton, New Jersey: Association for Computational Linguistics, 1993. 349 – 352.

作者简介:

刘　飞　男,1986 年生,辽宁人.2008 年毕业于北京大学计算机科学系,其后在北京大学软件研究所攻读硕士学位,主要研究领域为软件复用与软件构件技术.
E-mail: liufeipekingu@gmail.com



王立杰　男,1986 年生,江苏人.北京大学信息科学技术学院软件工程研究所博士研究生,主要研究方向为基于复用的软件开发.
E-mail: wanglj07@sei.pku.edu.cn