面向软件仓库挖掘的数据驱动特征提取方法

李晓晨 江 贺 任志磊
(大连理工大学软件学院 大连 116621)

摘 要 在软件仓库挖掘领域,通常将软件工程任务转换成数据挖掘问题进行解决。领域特征的使用严重影响了软件任务的解决效果。然而,如何根据特定任务从软件仓库数据中提取有价值的特征,在软件仓库挖掘领域尚缺乏系统的研究。数据驱动特征提取方法是一种新的特征提取方法。对于给定的软件工程任务,该方法从任务的数据集中选取部分数据(如源代码、缺陷报告等),招募若干志愿者人工完成该任务,并要求志愿者说明在人工完成特定软件工程任务时所考虑的因素。通过分析这些因素,可以提取所需的领域特征。以缺陷报告摘要任务为例进行实验,结果表明新方法能够发现高效的领域特征,并取得比现有方法更好的预测效果。

关键词 软件仓库挖掘,数据驱动方法,特征提取,缺陷报告摘要
中图法分类号 TP311.5 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.8.031

Data Driven Feature Extraction for Mining Software Repositories

LI Xiao-chen JIANG He REN Zhi-lei
(School of Software, Dalian University of Technology, Dalian 116621, China)

Abstract In mining software repositories(MSR), software tasks are usually transformed into data mining problems for solving. Domain-specific features heavily impact the solving of software tasks. However, no systematic investigation has been conducted on the issue of extracting features for specific software tasks. In this study, data driven feature extraction(DDFE) is a new feature extraction approach. For a software task, DDFE extracts a set of software data(e.g., source code, bug reports) and employs some volunteers to manually accomplish this software task. During the process, these volunteers are requested to submit their reasons under consideration. From these submitted reasons, DDFE can extract domain-specific features for software tasks. The experimental results on the task of bug report summarization demonstrate that DDFE may find effective features and achieve better predictive results against the state-of-the-art algorithm in the literatures.

Keywords Mining software repositories, Data driven approach, Feature extraction, Bug report summarization

1 引言

软件仓库挖掘(Mining Software Repositories)是近年来软件工程研究中的前沿领域,其通过挖掘软件仓库内形式多样的数据中蕴含的知识,来提高软件工程任务的完成效率。软件仓库挖掘的流程一般包括:收集数据、预处理数据(提取数据中的特征)、寻找/改进/设计合适的数据挖掘算法、运用挖掘结果解决软件工程任务[1-3]。在上述流程中,提取符合当前任务特性的领域特征是成功运用数据挖掘算法的前提条件,对于高效完成软件任务具有重要意义[4]。特征提取(Feature Extraction)指针对特定的软件仓库挖掘任务,寻找满足领域特点的维度刻画软件数据,以构建有辨别性的特征向量供数据挖掘算法使用。以软件仓库中的缺陷报告(Bug Report)为例,用于刻画缺陷报告的维度(特征)包括报告长度、报告提交者、报告所属的组件等[5]。针对不同的任务,需要提取不同的维度形成特征向量。提取特征的质量显著影响数据挖掘算法在软件仓库挖掘任务上的性能。比如,在软件仓库挖掘的典型任务——缺陷报告分派(Bug Report Triage)中,Arivik等人把缺陷报告看作普通文本,以词项为特征训练多种数据挖掘模型预测缺陷报告的修复者[6]。但缺陷报告与普通文本之间存在一定区别,近年来各国学者专门分析缺陷报告分派任务的特点,提出Tossing图、开发者优先级等领域特征,使该任务的准确率有显著提升[7,8]。

尽管在软件仓库挖掘研究中很多学者使用了大量高效的领域特征[5-8],但据我们所知,该领域学者尚未明确指出他们是如何确定并提取领域特征的。这种现象在一定程度上限制了软件仓库挖掘研究的发展。通过分析现有的研究成果,发现目前软件仓库挖掘中的(潜在但未指明)特征提取方法大致包括两类:基于少数研究者对特定任务的理解,提出面向软件工程任务的特征[7-10];直接迁移其它领域相似任务的经典特

到稿日期:2014-09-16 返修日期:2014-12-21 本文受教育部新世纪优秀人才支持计划(NCET-13-0073),国家自然科学基金(61175062,61370144)资助。

李晓晨(1989-),男,博士生,主要研究方向为软件仓库挖掘,E-mail:li1989@mail.dlut.edu.cn;江 贺(1980-),男,教授,主要研究方向为软件仓库挖掘、基于搜索的软件工程,E-mail:jianghe@dlut.edu.cn(通信作者);任志磊(1984-),男,博士后,主要研究方向为基于搜索的软件工程、演化计算。

征[6,11,12]。然而,前者要求研究者有深厚的领域背景,从而限制了该类方法的使用范围;后者难以发现满足领域数据特性的特征。因此,有必要研究一种新的有效的特征提取方法,该方法需从多个角度提取出有价值且面向特定任务的领域特征,进而提高软件仓库挖掘任务的完成效率。

针对上述挑战,本文提出一种面向软件仓库挖掘的数据驱动特征提取方法。数据驱动方法是一种利用特定领域数据的常见方法[13,14]。对于给定的软件工程任务,本文从该任务的数据集中选取一定量的数据,如源代码、缺陷报告等。然后寻找多名志愿者人工完成该任务,并要求志愿者说明在人工完成特定软件工程任务时所考虑的因素。最后研究者通过这些因素确定领域特征,构建特征向量。本文以缺陷报告摘要(Bug Report Summarization)任务为案例,实际验证该方法的效果。实验结果表明,利用数据驱动方法能够发现高效的领域特征,多名志愿者的辅助能从更多角度认识数据并提取有效的领域特征。

本文的贡献包括以下几个方面:

(1)首次系统研究影响软件仓库挖掘任务性能的特征提取问题,并提出一种数据驱动特征提取方法。该方法通过多名志愿者完成软件工程任务,从具体的软件数据中提取有价值的领域特征。

(2)本文以典型的软件仓库挖掘任务——缺陷报告摘要为例,验证了所提出的方法在具体任务上的效果。实验结果表明,新方法能够发现富有价值的领域特征,同时多名志愿者的辅助能够从多角度认识软件数据。

(3)基于本文方法得到了领域特征,从而获得了一种更加有效的基于监督学习的缺陷报告摘要方法。该方法比当前最好的监督学习方法在准确率、召回率等 4 个指标上均有改进。

本文第 2 节阐述数据驱动特征提取方法的研究动机;第 3 节介绍该方法的一般流程;第 4 节以缺陷报告摘要任务为案例,具体介绍该方法的应用方式;第 5 节对实验结果进行分析;第 6 节介绍本文的相关工作。

2 研究动机

本节从 3 个方面阐述面向软件仓库挖掘的数据驱动特征提取方法的研究动机。该方法以特定的软件仓库挖掘任务为基础,通过分析志愿者完成软件工程具体任务时的判断标准,从多个角度提取领域特征。

(1)软件仓库中数据类型丰富,对于不同的数据类型所能提取的特征往往不同,有必要结合特定的软件工程任务,以数据为驱动进行特征提取。Xie 等人总结了软件仓库中的数据类型[3],包括序列数据、关系图数据与文本数据等。不同类型的数据往往拥有特定的特征提取方式,以文本数据为例,其中源代码数据通常需要提取代码度量元作为特征,而缺陷报告数据通常需要提取词项、报告属性等作为特征[5,16]。另外,即使是相同类型的数据,由于软件工程任务不同,提取的特征也不相同[5,12]。因此需要结合特定的软件仓库挖掘任务,以数据为中心提取有辨别性的特征构造特征向量,并以此为基础训练机器学习模型完成软件工程任务。

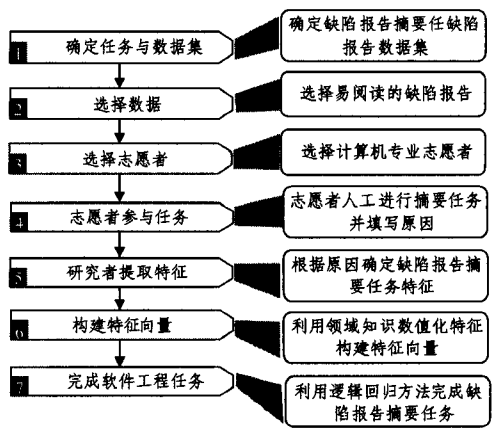
(2)少数研究者往往对数据的分析具有一定片面性,因此需要多名志愿者辅助,从更多的维度获得特征。研究者通常对特定任务的数据有深刻理解,对于给定的特征能够选择高效的算法进行计算,但在发现特征方面,单个研究者对数据的分析具有一定片面性,难以全面地提取领域特征。而不同的人可以相互弥补,从多个角度分析数据,最终能够提取出更全面的特征(第 5.2 节详细分析此现象)。因此本文利用多名志愿者辅助研究者的数据分析工作,从不同的角度分析数据,最终提取丰富的领域特征。

(3)志愿者往往对软件工程领域了解较少,因此需要简化志愿者的辅助工作。对于特定的软件工程任务,志愿者通常很难像研究者一样直接提出便于计算的领域特征,但每名志愿者在人工完成软件工程任务时通常具有一定的判断准则。因此我们让志愿者说明这些判断准则,研究者通过这些准则理解志愿者在完成某个任务时分析数据的角度,最终从每个角度中提取不同的特征。

综合上述 3 点,本文提出一种多名志愿者辅助的面向软件仓库挖掘的数据驱动特征提取方法,并在缺陷报告摘要任务上加以验证。

3 数据驱动特征提取方法的流程

本节着重介绍数据驱动特征提取方法的一般流程,具体过程如图 1(a)所示,可以分为 7 个步骤。



(a) 数据驱动特征提取方法流程　(b) 缺陷报告摘要任务的数据驱动特征提取方法流程

图 1　特征抽取方法的流程

(1)确定任务与数据集:首先确定所需解决的软件工程任务,并获取该任务的数据集。

(2)选择数据:从数据集中选择一个子集供志愿者分析。受志愿者的领域知识所限,在选择数据时,既要确保保留数据全貌,又要尽可能降低数据的使用难度。以软件仓库中缺陷报告数据为例,大部分软件仓库挖掘过程均需要处理其标题与描述信息,因此在选择缺陷报告时,需要向志愿者提供这两部分内容的全貌。但软件仓库中缺陷报告的阅读难度存在一定的差异,需要研究者控制所选报告难度。

(3)选择志愿者:本步骤需要选择志愿者参与任务。针对特定任务,需确立志愿者的选择标准。考虑到需要志愿者人工完成某个软件工程任务,在面向软件仓库挖掘的特征提取时,要求志愿者有一定的程序设计基础。

(4)志愿者参与任务:研究者向志愿者提供数据集,解释待完成的软件工程任务,要求志愿者人工完成特定的软件工程任务,并要求志愿者填写完成此任务时的判断标准或对任务进行某种标注的原因。此过程需要志愿者尽量填写有意义且易于理解的内容,为后续分析工作提供支持。

(5)研究者提取特征:特征提取工作分3步进行。(a)研究者收集志愿者填写的原因,并删除其中无意义的内容。对于每条原因,研究者需初步确定其可能反映的特征。(b)研究者向志愿者解释特征含义,通过与志愿者交流及修改特征使得特征能够反映志愿者的判断标准。该过程需尽量确保还原志愿者的真实想法,以减少特征提取的主观性问题。(c)研究者汇总并合并特征,最终提取出人工完成某软件工程任务时所考虑的特征。

(6)构建特征向量:对于每个特征,研究者需要选择方法度量特征以构建特征向量。此过程可以使用两种方法:(a)根据其它任务中相似特征的计算方式进行计算;(b)根据研究者对特定任务的理解确定特征的计算方式。

(7)完成软件工程任务:研究者选择合适的数据挖掘模型,利用步骤(6)的特征向量进行训练、预测,完成软件工程任务。

4 案例分析

本节以软件仓库挖掘中的典型任务——缺陷报告摘要为例,研究上述特征提取方法的具体应用过程。

4.1 缺陷报告摘要任务

Rastkar等人首先提出缺陷报告摘要任务,该任务是指从软件缺陷报告的描述与评论中选取部分句子(摘要句),使得软件开发者仅需阅读少量句子便可了解整个缺陷报告的内容,以减少软件开发者阅读缺陷报告的时间[12]。目前针对缺陷报告摘要的研究分为监督学习方法[12,15]和无监督学习方法两类[9,10]。本文研究主要针对基于监督学习的方法。

在文献[12]中,Rastkar等人手工从Eclipse、Gnome、Mozilla和KDE 4个开源项目中选择36个缺陷报告,为了便于阅读,报告中不包含大段栈信息和源代码。作者选择一定数量的标注者标注缺陷报告中的摘要句。由于标注工作的主观性,作者设定每个缺陷报告由3名标注者标注。对于缺陷报告中的每个句子,只有至少两名标注者认为是/不是摘要句时才确定结果。通过此方法,Rastkar等人得到36个缺陷报告的标准集。在此基础上,Rastkar等人迁移会议/邮件文本摘要领域经典的24个特征[17],使用逻辑回归(Logistical Regression)模型,采用留一交叉验证法(Leaving One Out Cross-Validation),即每次选择35个报告作为训练集,预测1个报告的摘要句,从准确率、召回率、F值、Pyramid精度4个指标上验证利用上述24个特征进行缺陷报告摘要的效果。Rastkar等人简称此算法为BRC算法。后文将以BRC算法为对比算法。

4.2 缺陷报告摘要的数据驱动特征提取方法

结合图1(a)的流程,本文分为7个步骤进行缺陷报告摘要任务的特征提取工作(见图1(b))。

(1)确定任务与数据集:确定软件工程任务为缺陷报告摘要任务,数据集为软件仓库中的缺陷报告。该任务需要处理缺陷报告的描述与评论文本内容。

(2)选择数据:从Eclipse等4个开源项目中选择15个缺陷报告供志愿者分析,选择报告的数量由参与者数目决定。所选的缺陷报告包括标题、描述和评论等信息,并且不包含大段的栈信息和源代码。

(3)选择志愿者:根据第3节的标准招募9名志愿者,其中3名曾经从事软件工程的研究工作(在本文中称为有领域知识志愿者,记为V_1、V_2、V_3),6名为未从事软件工程的研究工作的学生(在本文中称为无领域知识志愿者,记为V_4,…,V_9)。

(4)志愿者参与任务:把15个缺陷报告随机分配给9名志愿者,规定每名志愿者标注5个缺陷报告,同时确保每个缺陷报告由3名志愿者标注。把缺陷报告以纯文本形式通过邮件发送给志愿者,并附以简要的任务说明。采用文献[12]方式让志愿者进行标注。要求志愿者对每个标注结果写明原因,即志愿者为什么判断该语句是/不是摘要句。要求志愿者在一周内完成所有标注并提交结果。

(5)研究者提取特征:研究者收集所有结果,进行特征提取工作。该过程分为3步进行。

首先初步确立特征。研究者逐条阅读标注者填写的原因并初步判断其可能代表的特征,填写的原因可分为3类(见表1)。无意义原因指标注者没有对某句填写原话的标注结果因,或者填写的原因难以理解,忽略该原因。冲突原因指对于某句话一名标注者的标注与另外两名标注者的结果不一致,认为这名标注者(如表1(b)V_3)对该语句的判断标准不够准确,因此忽略其填写的内容。其它情况为满足要求原因。

表1 志愿者填写原因与提取特征举例

原因类型	例句	志愿者填写内容	提取特征	计算方法
(a)无意义原因	The news on the front page will have a date next to each item。	V_6:不是摘要句,不知所云。	忽略该句	
(b)冲突原因	Fixed in 2004. 03. 08 commit on 1. 8-branch and HEAD。	V_2:是,说明了bug修复的版本。 V_3:不是,不太确定,这是说主bug的解决情况。 V_8:是,说明了这个bug已经在哪个版本修复。	忽略V_3内容。根据V_2和V_8内容,确定关于缺陷报告状态更改语句可能是摘要句,如缺陷被修复。	判断语句是否包含"fixed/fix"等词,如果包含,特征值为1,否则为0。
(c)满足要求原因	When scheduling a transaction without entering an amount, GNUcash will hang when it try's to schedule this transaction。	V_3:是,和标题内容一样,解释bug。	句子与缺陷报告标题的相似度可能作为判断摘要句的特征。	利用向量空间模型把当前语句与标题的余弦相似度值作为特征,取值范围为0~1。
	OK。	V_9:不是,OK而已。	初步确定句子是否含语气词作为特征。经交流知志愿者意图是句子过于简短。修改后把句子长度作为特征。	句子字符数除以当前报告最长语句字符数,结果值作为特征,范围为0~1。

此后研究者与志愿者进行面对面交流,询问志愿者所提取的特征是否能够反映其填写的原因内容,如果不能则对特征进行修改,如表1(c)V_9所示。

最终,志愿者在缺陷报告上标注的所有满足要求的原因

均存在一个特征与其对应。志愿者填写的原因中,有些反映的是同一个特征(见表 1(b)V_2 和 V_8)。将所有的特征进行合并,最终得到 11 个缺陷报告摘要特征(见表 2)。这些特征是志愿者从真实数据集中反馈的,因此在一定程度上确保了所获得的特征具有价值。

表 2 缺陷报告摘要特征

	特征	含义	计算方法
1	SLEN	句子长度	统计句子字符数并归一化,范围 0~1。
2	CCW	是否包含特定词语	判断句子是否包含 http、build id 等词语,范围 0~1。
3	SI	句子重要性	计算句子词项的文档频率乘逆文档频率值并求和与归一化,范围 0~1。
4	TONE	句子语气	判断句子是否含有 what,how,guess 等疑问语气词,范围 0~1。
5	SWS	与标题相似度	计算句子与标题的 VSM 相似度,范围 0~1。
6	SWO	与训练集语句相似度	判断当前语句与训练集中最相似的句子集合是否为摘要句,范围 0~1。
7	CLOC	句子位置	计算句子在缺陷报告的位置并归一化,范围 0~1。
8	CLEN	会话长度	计算当前所在的会话的句子数量并归一化,范围 0~1。
9	INS	句子影响力	把与当前语句相邻语句的长度信息作为当前语句被影响的程度特征,范围 0~1。
10	CHD	修改历史	判断句子是否含有提示修改报告状态的词,如 fix/fixed 等,范围 0~1。
11	STR	重现步骤	计算当前会话中描述缺陷重现步骤的句子比例,范围 0~1。

(6)构建特征向量:研究者需要对每个特征数值化以构建特征向量。同一特征通常存在多种度量方法,特征计算方式不同往往会影响数据挖掘模型的效果。本文并不深入研究某个特征的计算方法,因此对于每个特征,均按照第 3 节(6)中的方法,使用较为直接的方式计算(见表 2),但这样可能会降低算法性能。在未来的研究工作中,将专门研究计算方式的影响。

(7)完成软件工程任务:本文把提取的特征应用到 4.1 节介绍的缺陷报告摘要流程中。具体做法是,替换文献[12]中使用的 24 个特征为数据驱动方法提取的 11 个特征,采用逻辑回归方法在 Rastkar 等人提供的 36 个已标注的缺陷报告上运用留一交叉验证方式对每个缺陷报告预测摘要句。所有的参数与文献[12]的对比算法 BRC 相同。

作为对比,本文重现 BRC 算法,并从准确率、召回率、F 值、Pyramid 精度 4 个指标上对比特征效果。如表 3 所列,在 4 个指标上,数据驱动方法得到的特征均优于 BRC 算法。可见,数据驱动方法能够得到有效的特征,并优于文献[12]采用的领域迁移方法。

表 3 BRC 算法与数据驱动方法的比较

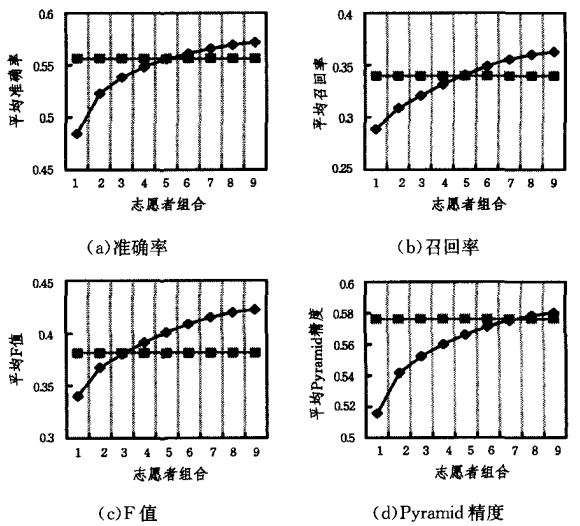
算法	准确率(%)	召回率(%)	F 值(%)	Pyramid 精度(%)
BRC 算法	55.66	34.00	38.15	57.61
数据驱动方法	57.53	36.23	42.32	58.12

5 实验结果分析

5.1 参与者的数量对任务的影响

本节研究参与者数量与软件工程任务完成情况的关系,分析是否少数志愿者就足以得到相似的结果。

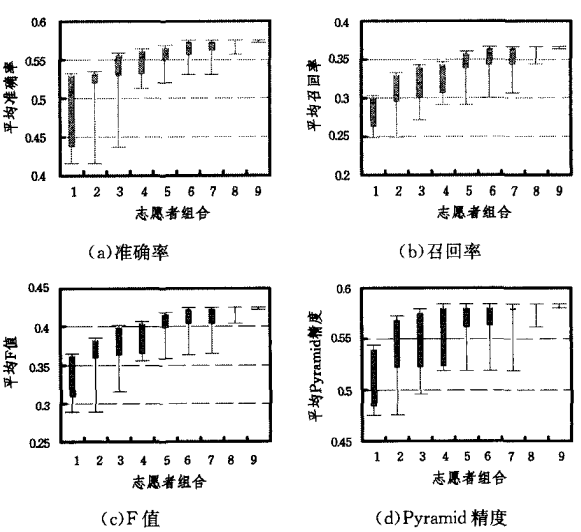
本文以组合的方式遍历 9 人中任意 N 人的组合。以 $N=3$ 为例,组合数目为 $(\frac{9}{3})=9*8*7/3!=84$。对于所有 84 种组合,统计利用每种组合中 3 个人所提取的特征产生的摘要结果并求平均值,它代表仅使用 3 个人挖掘特征时能够发现特征的平均效果。如图 2 所示,黑色曲线为人数与各指标的关系,为了便于对比,以灰色直线标记 BRC 算法结果。图中随着人数增长各指标逐渐提高,但增长趋势变缓。从图中可以看出,当志愿者超过 6 人时,准确率、召回率和 F 值均超过对比算法。当志愿者人数超过 8 时,Pyramid 精度超过对比算法。对于当前实验,当组合的人数增加到 8 人时,特征质量改善程度变缓。因此有如下结论:(1)增加志愿者能够发现有价值的特征,进而提高摘要效果,(2)少数人并不能确保得到有价值的特征,随着人数的增加,挖掘出的领域特征趋于稳定。



(a)准确率 (b)召回率

(c)F 值 (d)Pyramid 精度

图 2 志愿者组合的特征效果



(a)准确率 (b)召回率

(c)F 值 (d)Pyramid 精度

图 3 志愿者组合的特征效果盒图

用盒图(见图 3)表示 N 个人每种组合各指标值的分布情况。可以看出,当人数较少时不同的组合方式得到的特征效果差异较大,随着人数增加,志愿者彼此提取的特征相互弥补,得到的结果更加稳定。在只有少数志愿者(人数少于 3)时,盒图的最大值与最小值相差很大,表明所得到的特征质量

不够稳定；当人数超过 3 时，随着人数增多，特征质量逐渐稳定；达到 8 人时，盒图的长度明显变短，表明各指标明显趋于稳定。这与图 2 观察到的结果一致。

根据以上现象，应用数据驱动特征提取方法时需要招募较多的志愿者，以确保得到较好的特征。

5.2 领域知识对志愿者的影响

本节分析志愿者领域知识对特征质量的影响，分析志愿者是否需要足够的领域知识以完成任务。

以组合的方式计算 N 名有/无领域知识志愿者得到特征的平均指标。有领域知识志愿者用黑色柱形表示，无领域知识志愿者用灰色柱形表示。由于有领域知识志愿者只有 3 名，因此不能计算 3 人以上的组合结果。如图 4 所示，在同等人数下，有领域知识志愿者标注特征质量较高，但随着无领域知识志愿者的增加，当无领域知识志愿者人数达到 6 时，得到的特征在准确率、召回率、F 值 3 个指标上均超过有领域知识的 3 名志愿者所构建的特征。即对于本实验而言，当无领域知识志愿者是有领域知识志愿者的 2 倍时，无领域知识志愿者能够抽取更高质量的特征。因此可以发现：(1)有领域知识志愿者并不能保证得到全部特征；(2)随着无领域知识志愿者人数的增加，发现的特征的质量甚至超过有领域知识志愿者。所以志愿者并不需要过多的领域知识。

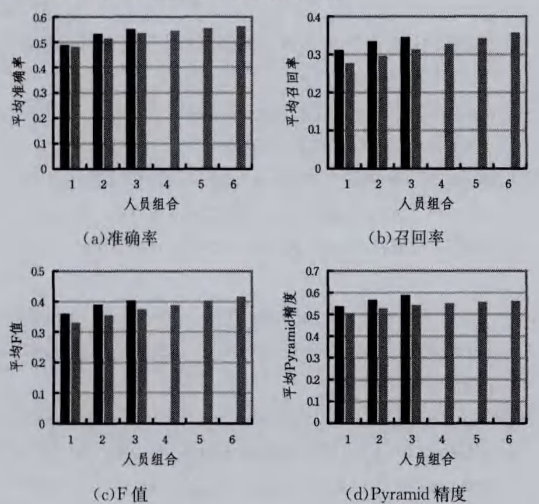


图 4 有领域知识志愿者与无领域知识志愿者标注特征提取的比较

为解释上述结论，进一步应用特征排序理论对所有 11 个特征进行排序。具体做法是计算每个特征的 Fscore[18]，需要指出的是本段中 Fscore 与文中提到的 F 值指标不同，特征排序的 Fscore 是一种计算特征重要性的方式，它通过计算每个特征在训练集中区分正例/负例的能力来评价特征的重要性。计算式如式(1)所示。

$$F(i)=\frac{(\bar{x}_i^{(+)}-\bar{x}_i)^2+(\bar{x}_i^{(-)}-\bar{x}_i)^2}{\frac{1}{n_+-1}\sum_{k=1}^{n_+}(x_{k,i}^{(+)}-\bar{x}_i^{(+)})^2+\frac{1}{n_--1}\sum_{k=1}^{n_-}(x_{k,i}^{(-)}-\bar{x}_i^{(-)})^2} \tag{1}$$

其中，$\bar{x}_i$，$\bar{x}_i^{(+)}$，$\bar{x}_i^{(-)}$ 表示第 i 个特征在训练集中所有特征值平均值、正例特征值平均值、负例特征值平均值；$x_{k,i}^{(+)}$，$x_{k,i}^{(-)}$ 表示第 i 个特征在第 k 个正例或负例的值。

图 5 中横坐标是根据 Fscore 值以升序方式对特征进行排序，最右侧特征 Fscore 值最大。纵坐标代表每名志愿者，志愿者如果发现某一特征，则在图中加以标记。有领域知识志愿者发现的特征分布更趋向于右侧即更有价值，因此在人数相同时，有领域知识的志愿者所抽取的特征质量会明显高于无领域知识志愿者。但由于思维局限，每名志愿者只能发现少数几个特征，随着人数的增多，志愿者发现的特征相互补充，最终能抽取出更丰富的特征，从图中可以看出，当把 6 名无领域知识志愿者发现的特征汇总时，他们发现了绝大多数重要的特征，只漏掉 CLEN 和 STR。而 3 名有领域知识志愿者却漏掉相对重要的特征 SI 和 DNS，因此此时无领域知识志愿者能够构建更高质量的特征，这也体现了多名志愿者的优势。在实验中每个缺陷报告要求 3 名志愿者阅读，从图 5 可以看出，有些特征只有一名有/无领域知识的志愿者发现(如 STR，SI)，这说明即使阅读同样的报告，不同志愿者也会发现不同的特征。领域知识并不是招募志愿者的硬性要求。

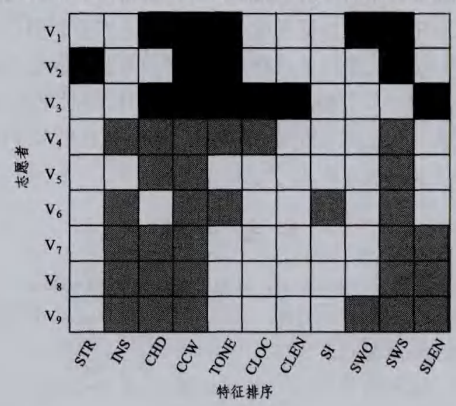


图 5 特征分布

6 相关工作

6.1 缺陷报告摘要

软件仓库挖掘任务引起了越来越多研究者的重视，通过深入分析软件仓库数据，研究者进行重复缺陷报告检测、缺陷报告摘要、软件下一版本功能预测等多种任务[11,12,19]。本文分析的缺陷报告摘要是典型的软件仓库挖掘任务。已有的研究工作可以分为监督学习和非监督学习两类。在监督学习方面，Rastkar 等人把会议/邮件文本摘要的 24 个特征迁移到缺陷报告摘要上，并使用逻辑回归模型完成该任务[12,15]。在非监督学习方面，Mani 等人发现缺陷报告中包含大量代码段、提问等内容，它们通常不是摘要句，利用这一领域特征对缺陷报告的语句进行过滤[9]。Lotufo 等人通过评估开发者对报告中每个语句的关注程度进行缺陷报告摘要[10]，然而并没有研究者对不同特征的获取方式进行专门深入分析，因此本文提出数据驱动的方法用于提取领域特征。

6.2 特征提取

特征提取是数据挖掘领域的关键问题之一，其目的在于提取有辨别性的特征训练数据挖掘模型[20]。在图像识别和文本分类领域有大量的特征提取研究工作。由于软件缺陷仓库中大部分数据以文本形式存储，因此着重介绍文本方面的特征提取工作。

通用文本中最常见的特征模型是词袋模型(Bag of Words)，即把文本中词项作为特征，通过计算词项间的 tf-idf

值等信息得到特征向量[21]。除此之外，如词项重要性、词项熵和词项贡献度等内容均被提出作为通用文本特征[22]。除了依赖统计方法得到文本特征，LSI 和单词聚类等方法也被使用进而得到高层的文本特征。Tasci 等人比较了常见的文本特征，提出了一个自适应模型对不同的应用选择合适的特征[23]。Sourcy 等人提出词项置信度概念，以词项归属于某一文本类的置信程度为重要的文本特征[24]。然而特征的提取与领域有关，通用的文本特征往往难以在特定领域发挥作用，因此本文研究如何提取符合特定的软件工程任务的领域特征。

结束语 本文以软件仓库为研究对象，分析如何提取软件工程任务特征，提出了一种数据驱动特征提取方法，通过志愿者对特定问题的标注与原因解释，辅助研究者从多角度提取领域特征。本文以缺陷报告摘要任务为案例，在具体软件工程任务中分析该特征提取方法的效果。实验表明，该方法提取出的特征优于特征迁移方式。进一步分析表明，增加志愿者人数可以显著提高特征提取的稳定性，同时志愿者能够辅助研究者从更多的角度发现特征。未来将在更多的软件工程任务上实践该方法，以验证该方法的效果。

参 考 文 献

[1] Xie T, Pei J, Hassan A E. Mining software engineering data[C]// Proceedings of the 29th International Conference on Software Engineering(ICSE'2007). 2007:172-173

[2] Hassan A E, Xie T. Software intelligence: the future of mining software engineering data[C]// Proceedings of the FSE/SDP workshop on Future of Software Engineering Research(FoSER' 2010). 2010:161-166

[3] Xie T, Thummalapenta S, Lo D, et al. Data mining for software engineering [J]. Computer, 2009, 42(8):55-62

[4] Srinivasa K G, Venugopal K R, Patnaik L M. Feature extraction using fuzzy c-means clustering for data mining systems[J]. International Journal of Computer Science and Network Security, 2006, 6(3A):230-236

[5] Sun C, Lo D, Khoo S C, et al. Towards more accurate retrieval of duplicate bug reports[C]// Proceedings of 2011 26th IEEE/ACM International Conference on Automated Software Engineering(ASE'11). 2011:253-262

[6] Anvik J, Hiew L, Murphy G C. Who should fix this bug? [C]// Proceedings of the 28th International Conference on Software Engineering(ICSE'06). 2006:361-370

[7] Jeong G, Kim S, Zimmermann T. Improving bug triage with bug tossing graphs[C]// Proceedings of the 7th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on The Foundations of Software Engineering (FSE'09). 2009:111-120

[8] Xuan J, Jiang H, Ren Z, et al. Developer prioritization in bug repositories[C]// Proceedings of the 34th International Conference on Software Engineering(ICSE'12). 2012:25-35

[9] Mani S, Catherine R, Sinha V S, et al. Ausum: approach for unsupervised bug report summarization[C]// Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering(FSE'12). 2012:11-21

[10] Lotufo R, Malik Z, Czarnecki K. Modelling the 'hurried' bug report reading process to summarize bug reports[C]// Proceedings of the 28th IEEE International Conference on Software Maintenance(ICSM'12). 2012:430-439

[11] Runeson P, Alexandersson M, Nyholm O. Detection of duplicate defect reports using natural language processing[C]// Proceedings of the 29th International Conference on Software Engineering(ICSE'07). 2007:499-510

[12] Rastkar S, Murphy G C, Murray G. Summarizing software artifacts: a case study of bug reports[C]// Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering (ICSE'10). 2010, 1:505-514

[13] Yin S, Ding S, Xie X, et al. A review on basic data-driven approaches for industrial process monitoring [J]. IEEE Transactions on Industrial Electronics, 2014, 61(11):6418-6428

[14] Yin S, Wang G, Karimi H R. Data-driven design of robust fault detection system for wind turbines [J]. Mechatronics, 2014, 24 (4):298-306

[15] Rastkar S, Murphy G, Murray G. Automatic Summarization of Bug Reports[J]. IEEE Transactions on Software Engineering, 2014, 40(4):366-380

[16] 王青，伍书剑，李明树. 软件缺陷预测技术[J]. 软件学报，2008, 19(7):1565-1580
Wang Q, Wu S J, Li M S. Software defect prediction [J]. Journal of Software, 2008, 19(7):1565-1580

[17] Murray G, Carenini G. Summarizing spoken and written conversations[C]// Proceedings of the Conference on Empirical Methods in Natural Language Processing(EMNLP'08). 2008:773-782

[18] Chen Y W, Lin C J. Combining SVMs with various feature selection strategies[M]// Feature Extraction. Springer Berlin Heidelberg, 2006:315-324

[19] Xuan J, Jiang H, Ren Z, et al. Solving the large scale next release problem with a backbone-based multilevel algorithm[J]. IEEE Transactions on Software Engineering, 2012, 38(5):1195-1212

[20] Srinivasa K G, Venugopal K R, Patnaik L M. Feature extraction using fuzzy c-means clustering for data mining systems[J]. International Journal of Computer Science and Network Security, 2006, 6(3A):230-236

[21] Salton G, Wong A, Yang C S. A vector space model for automatic indexing[J]. Communications of the ACM, 1975, 18(11): 613-620

[22] Aggarwal C C, Zhai C. A survey of text clustering algorithms [M]// Mining Text Data. Springer US, 2012:77-128

[23] Taşcl Ş, Güngör T. Comparison of text feature selection policies and using an adaptive framework [J]. Expert Systems with Applications, 2013, 40(12):4871-4886

[24] Sourcy P, Mineau G W. Beyond TFIDF weighting for text categorization in the vector space model[C]// Proceedings of the 19th international joint conference on Artificial intelligence(IJCAI'05). 2005:1130-1135