

doi: 10.12194/j.ntu.20201013002

引文格式: 贾焱鑫, 林浩, 陈翔, 等. 一种变粒度缺陷报告严重程度预测方法[J]. 南通大学学报(自然科学版), 2022, 21(1): 44-51.

一种变粒度缺陷报告严重程度预测方法

贾焱鑫¹, 林浩¹, 陈翔^{1*}, 濮雪莲², 葛骅¹

(1. 南通大学 信息科学技术学院, 江苏 南通 226019; 2. 南通大学 经济与管理学院, 江苏 南通 226019)

摘要:已有工作一般将缺陷报告严重程度预测问题建模为粗粒度分类问题(即二分类问题)或细粒度分类问题(即多分类问题),更细的预测粒度意味着对分类准确性提出更高的要求,因而预测性能通常会下降。通过有效利用粗粒度分类的高预测性能与细粒度分类更为精细的预测粒度,提出一种变粒度缺陷报告严重程度预测(variable granularity bug report severity prediction, VG-BSP)方法, VG-BSP 方法首先对缺陷报告进行基于二分类的粗粒度划分,然后根据二分类的结果,选择对应的预测模型以预测出具体的严重程度(即基于多分类的细粒度划分)。为了验证 VG-BSP 方法的有效性,搜集了来自 Eclipse 的两个子项目 JDT 和 CDT 的缺陷报告和来自 Mozilla 项目的缺陷报告;并将 VG-BSP 方法与逻辑回归(logistic regression, LR)方法、K 近邻(K-nearest neighbors, KNN)方法、决策树(decision tree, DT)方法、随机森林(random forest, RF)方法、支持向量机(support vector machine, SVM)方法等经典细粒度分类方法进行了对比。实验结果表明:基于 macro-F1 指标, VG-BSP 方法的性能与已有的细粒度方法相比平均可提升 19.6%。除此以外,使用 LR 分类算法和 Spacy 词向量可以使得 VG-BSP 方法取得最好的性能。

关键词:软件仓库挖掘;缺陷报告;严重程度预测;机器学习

中图分类号: TP311.5

文献标志码: A

文章编号: 1673-2340(2022)01-0044-08

A severity predication method for variable granularity bug report

JIA Yanxin¹, LIN Hao¹, CHEN Xiang^{1*}, PU Xuelian², GE Hua¹

(1. School of Information Science and Technology, Nantong University, Nantong 226019, China;

2. School of Economics and Management, Nantong University, Nantong 226019, China)

Abstract: The previous studies generally model the bug report severity prediction problem as a coarse-grained classification problem (i.e., binary classification problem) or a fine-grained classification problem (i.e., multi-class classification problem). Finer prediction granularity means higher requirements for classification accuracy, so prediction performance tends to decline. Since using the coarse granularity classification can achieve high prediction performance while using the fine granularity can obtain finer prediction granularity, a variable granularity bug report severity prediction method (VG-BSP) is proposed. Firstly, the bug report is classified into two categories via the coarse-grained classification. Then according to the previous classification results, the corresponding prediction model is utilized to predict the specific severity of the bug report via fine-grained multi-class classification. To verify the effectiveness of VG-BSP, bug reports from two Eclipse subprojects (i.e., JDT and CDT), and bug reports from the Mozilla project are collected. The proposed VG-BSP method is compared with the previous classical fine-grained classification methods, such as logistic regression (LR), K-nearest neighbor (KNN), decision tree (DT), random forest (RF), support vector

收稿日期: 2020-10-13

基金项目: 国家自然科学基金青年科学基金项目(61202006)

第一作者简介: 贾焱鑫(1996—), 女, 硕士研究生。

* 通信联系人: 陈翔(1980—), 男, 副教授, 博士, 主要研究方向为智能软件工程、经验软件工程、软件仓库挖掘。E-mail: xchencs@ntu.edu.cn

machine (SVM). The results show that compared with the previous methods, the performance of the proposed method can be improved by 19.6% on average in terms of the macro-F1 measure. Moreover, the LR classifier and the Spacy word embedding can enable VG-BSP to achieve the best performance.

Key words: software repository mining; bug report; severity prediction; machine learning

软件内存在的缺陷会给项目带来不可预知的风险。当前,项目通常会利用缺陷跟踪系统(bug tracking system)来搜集和维护缺陷报告。其中,按照缺陷报告的严重程度进行优先级排序,并尽早修复高风险缺陷是降低项目风险的一种可行手段。然而,精准设置缺陷报告所对应的严重程度并不容易。由于受到缺陷报告提交者的能力和经验等多方面因素的影响,当提交者对严重程度的设置不重视,或者没有足够的信心时,在提交缺陷报告时经常会选择默认取值(即 normal),因此,normal 取值并不能真实反映缺陷报告所对应的实际严重程度。除此之外,由于不同提交者对缺陷报告严重程度的理解并不完全相同^[1],因此即使是遇到了同一缺陷,不同提交者在提交缺陷报告时也可能设置不同的严重程度。上述问题的存在,使得提交者不得不人工审查项目缺陷报告严重程度的准确性。而缺陷报告严重程度的评估工作专业性较强,需要有经验的提交者投入大量的时间和精力,因此,构建高质量的缺陷报告严重程度预测模型^[2],对节约人力和提高软件项目质量具有重要的研究意义。

目前,研究人员针对缺陷报告严重程度预测问题,一般将其建模为粗粒度分类问题(即二分类问题)或细粒度分类问题(即多分类问题)。具体来说,二分类问题是以严重程度 normal 为界限,将超过 normal 程度的类型归为一类;低于 normal 程度的类型归为另一类。由于二分类问题的粒度较粗,因此构建出的分类模型通常可以取得不错的分类性能。而多分类问题则可以直接预测出缺陷报告所对应的具体严重程度类型,因此可以根据多分类的预测结果,更准确地确定缺陷修复的优先级^[3]。但多分类问题一般受到训练数据集规模的不足、数据集内存在的类不平衡问题等因素影响,导致构建出的分类模型的性能不理想。基于上述分析,不难看出粗粒度分类与细粒度分类各有优势和不足,并互为补充,因此本文提出变粒度预测(variable granular-

ity bug report severity prediction, VG-BSP)方法。变粒度指随着预测阶段的改变,对模型预测粒度和预测性能的需要产生的变化,对在不同阶段中所采用的预测粒度进行的改变。具体来说:首先对新缺陷报告进行基于二分类的粗粒度划分,然后根据二分类的结果进一步细分(即基于多分类的细粒度划分)。即先使用粗粒度二分类预测模型将缺陷报告所对应的严重程度分为严重或不严重两大类,然后使用细粒度多分类预测模型在已预测出的两大类严重程度结果基础上,继续细分严重程度的类别。与已有的多分类方法相比,VG-BSP 方法的优势在于可以利用粗粒度二分类的高准确性,并在一定程度上缓解直接使用细粒度多分类所带来的低准确性问题。

本文选择来自实际开源项目的缺陷报告(来自 Eclipse 的两个子项目 JDT 与 CDT 的缺陷报告和来自 Mozilla 项目的缺陷报告)^[2,4],并分别设置了粗粒度类别和细粒度类别,与逻辑回归(logistic regression, LR)方法、K 近邻(K-nearest neighbors, KNN)方法、决策树(decision tree, DT)方法、随机森林(random forest, RF)方法、支持向量机(support vector machine, SVM)方法等经典细粒度分类方法进行比较。基于 macro-F1 指标, VG-BSP 方法的性能平均可提升 19.6%。除此之外,研究还发现,采用 LR 方法作为 VG-BSP 方法的分类方法、采用 Spacy 词向量作为文本建模方法,可以使得 VG-BSP 方法取得最好的预测性能。

1 相关工作和研究背景

1.1 相关工作

Tan 等^[5]通过从 Stack Overflow 网站中提取缺陷相关问题贴和回复贴,来对缺陷报告的内容进行扩充,在此基础上使用逻辑回归算法预测缺陷报告所对应的严重程度,最终在预测性能上超过了基准方法中效果最好的朴素贝叶斯方法。Sharma 等^[6]认为模型性能可能会因为缺陷报告的严重程度在修复

过程中产生的变化而受到影响,同时认为缺陷报告的优先级、评论数量等属性也应当被纳入考虑范围,最终基于上述两点提出了基于支持向量机、朴素贝叶斯和k近邻等算法的分类模型;Chaturvedi等^[6]尝试分析了朴素贝叶斯、K近邻、支持向量机等算法在NASA项目中的缺陷报告严重程度预测的适用性,并使用各种性能评测指标进行了验证。Lamkanfi等^[7]比较了4种著名的文本挖掘算法(朴素贝叶斯、朴素贝叶斯多项式、K近邻和支持向量机)之间的准确性及所需训练集的规模。Menzies等^[8]则基于标准的文本挖掘方法和机器学习方法提出了一种高效的缺陷报告严重程度预测方法SEVERIS。Lamkanfi等^[9]则利用来自Mozilla、Eclipse和GNOME 3个开源项目的缺陷报告,验证了在规模足够大的训练集上,使用文本挖掘方法是行之有效的。Tian等^[10]则转换了研究思路,尝试预测缺陷报告所对应的具体严重程度,提出了一种基于BM-25文本相似度计算公式的自动化预测方法,并在细粒度严重性预测方面超越了基准方法。刘文杰^[11]针对严重性实证分析问题和严重性预测问题,分别提出了缺陷报告严重性影响分析框架、面向领域特征的粗粒度严重性预测方法和基于特征序列重构的细粒度严重性预测方法,提升了细粒度严重性定量预测方法的性能。王婧宇等^[12]则在相关工作的基础上对基于分类的严重程度预测方法进行了总结,提炼了技术框架,并对框架中的主要步骤进行了介绍。

1.2 对文本内容的建模

缺陷报告的摘要是一种非结构化文本,目前将非结构化数据转为机器学习方法可处理的数据格式,需要对文本内容进行建模,常用的文本建模方法有以下4种:1)Spacy词向量,由Python第三方工业级自然语言处理库Spacy支持,基于OntoNotes 5和GloVe Common Crawl数据集训练生成的685 000个300维词向量;2)Word2Vec词向量,是使用Word2Vec模型中的Skip-gram模型训练生成的,可用来表示单词对之间的关系的词向量;3)BOW词向量,是使用词袋模型表示法,通过构建基于数据集的不重复单词集合,从而将每句话表示成单词集合中各单词在该句子内的出现次数所组成的向量;

4)TF-IDF词向量,是根据一个词在单份语料中出现的次数与在语料库中出现的频率来判断一个词语的重要性,并在此基础上生成的词向量。

1.3 缺陷报告的粒度设置

本文分析的开源项目都使用Bugzilla这一缺陷跟踪系统来维护缺陷报告。与已有研究工作^[9]保持一致,在粗粒度二分类问题中,我们将blocker、critical、major这3种缺陷报告严重程度归为严重类别;将minor、trivial、enhancement这3种缺陷报告严重程度归为不严重类别。

在细粒度多分类问题中,在粗粒度二分类得到的预测结果为严重程度时,则将借助对应细粒度多分类模型,进一步细分为blocker、critical或major类型。在粗粒度二分类得到的预测结果为不严重程度时,则将借助对应的细粒度多分类模型,进一步细分为minor、trivial或enhancement类型。

2 变粒度缺陷报告严重程度预测方法

本节首先给出了文本预处理方法,随后给出本文所提出的变粒度缺陷报告严重程度预测(variable granularity bug report severity prediction, VG-BSP)方法的实现细节,包含了变粒度操作的实施流程与具体操作规则。

2.1 文本预处理

缺陷报告的摘要是文本,因此需要对其进行文本预处理。首先将文本进行分词;其次借助停用词列表过滤掉停用词,停用词是指在文本中经常出现,但含有极少文本信息的高频词(例如英语中的the, at, which, on等);然后对分词进行规范化,即去除词缀得到单词对应的词根,例如,将effective规范为effect;最后利用Spacy词向量文本建模方法针对该摘要信息进行文本建模。

2.2 方法过程细节分析

2.2.1 训练粗粒度二分类预测模型

根据文本预处理后生成的词向量和缺陷报告中的严重程度粗粒度类别,采用二分类算法,构建粗粒度预测模型MC,预测缺陷报告对应的严重程度是否为严重。

2.2.2 训练细粒度多分类预测模型

通过训练粗粒度二分类预测模型,预测并得到

准确度较高的粗粒度类型结果,以此为基础进一步训练细粒度多分类预测模型,即本文所提方法中的变粒度操作,从而保证在预测准确度较高的前提下进一步细化预测粒度。首先根据粗粒度类型将训练数据集细分为两个数据集,其中第一个数据集包含了 blocker、critical、major 等 3 种类型的缺陷报告,并基于多分类方法构建出细粒度分类模型 MF1;第二个数据集包含了 minor、trivial、enhancement 等 3 种类型的缺陷报告,并基于多分类方法构建出细粒度分类模型 MF2。

2.2.3 模型组合

将粗粒度预测模型 MC 与两个细粒度预测模型 MF1、MF2 进行组合,构成最终的缺陷报告严重程度预测模型。VG-BSP 方法的整体框架如图 1 所示。当面对新的缺陷报告时,先使用粗粒度预测模型 MC 预测其所对应的粗粒度类别,若粗粒度类别为不严重,则进入粗粒度为不严重所对应的细粒度预测模型 MF1 进行细粒度的分类;若粗粒度类别为严重,则进入粗粒度为严重所对应的细粒度预测模型 MF2 进行细粒度的分类。

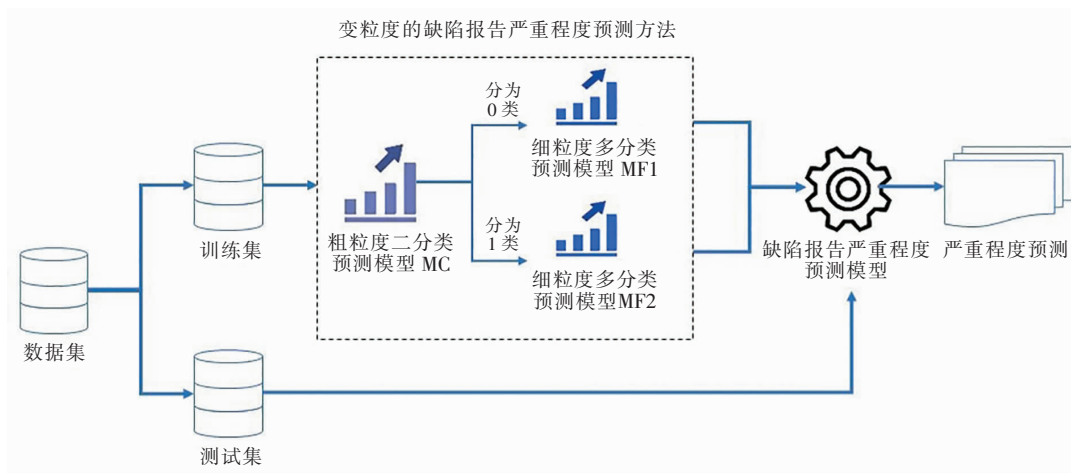


图 1 VG-BSP 方法的整体框架图

Fig. 1 Overall framework of the VG-BSP method

3 实验设计

3.1 研究目的

为了验证本文所提 VG-BSP 方法的有效性和方法内部设置的合理性,提出以下 3 个研究问题:

RQ1 变粒度缺陷报告严重程度预测方法的性能是否优于现有的基准方法?

RQ2 不同分类算法对变粒度缺陷报告严重程度预测方法的性能影响如何?

RQ3 与其他文本建模方法相比,使用 Spacy 词向量是否能提高变粒度缺陷报告严重程度预测方法的性能?

3.2 评测对象

为了验证 VG-BSP 方法的有效性,选择来自实际开源项目的缺陷报告数据集,包括来自 Eclipse 的两个子项目 JDT 和 CDT 的缺陷报告及来自 Mozilla

项目的缺陷报告。其中 JDT 子项目是 Java 开发工具的简称,其作用是为高级用户提供工具插件;CDT 子项目是 C/C++ 开发工具的简称,其提供了一个基于 Eclipse 平台的功能完整的 C 和 C++ 集成开发环境。Mozilla 项目包含 Core、Firefox、Thunderbird 等。

以上 3 个项目采用 Bugzilla 进行缺陷报告的管理,我们首先从 Bugzilla 中搜集这些项目的历史缺陷报告,针对每一个缺陷报告,抽取该缺陷报告的摘要信息(summary)和严重程度(severity)这两个属性的信息,并构成数据集。数据集的统计特征如表 1 所示,包括项目的名称、总的缺陷报告数,以及不同缺陷严重程度对应的缺陷报告数。在以前的研究过程中,研究人员发现由于 normal 被设置为缺陷报告严重程度标签的默认取值,因此一些对自己的经验和能力缺乏信心的提交者会将不符合 normal 取值

标准的缺陷报告默认标注为 normal 类型^[1]。在本研究工作中,我们认为引入 normal 取值的数据很容易

为数据集引入噪音,所以在数据集中去除了所有严重程度标签为 normal 的缺陷报告。

表 1 研究工作中使用的缺陷报告数据集的统计特征

Tab. 1 Statistical characteristics of the bug report datasets used in the study

项目名称	#缺陷报告数	#blocker	#major	#critical	#minor	#trivial	#Enhancement
Eclipse JDT	2 567	12	144	27	159	2 214	11
Eclipse CDT	1 690	35	329	65	281	943	37
Mozilla	1 181	8	241	322	523	87	0

3.3 评测指标

由于本文关注的问题是多元分类问题,因此采用 macro-F1 作为模型性能的评测指标。

macro-F1 指分别计算每个类别的 F1 值,然后求均值。假设计算类别 A 的 F1 值,在属于类别 A 的实例为正例,不属于类别 A 的实例为反例的情况下,F1 值的计算公式为

$$F1 = 2 \times \frac{P \times R}{P + R}, \tag{1}$$

其中:P 为查准率(Precision);R 为查全率(Recall)。计算公式为

$$P = \frac{TP}{TP + FN}, \tag{2}$$

$$R = \frac{TP}{TP + FP}. \tag{3}$$

该情况下对应的混淆矩阵,如表 2 所示。

表 2 分类结果混淆矩阵

Tab. 2 Confusion matrix of classification results

真实情况	预测结果	
	正例	反例
正例	TP(真正例)	FN(假反例)
反例	FP(假正例)	TN(真反例)

3.4 基准方法

本文所提方法通过粗粒度预测阶段得到二元分类结果,随后细化预测粒度,最终得到细粒度严重程度预测结果。因此本文主要选择已有的细粒度预测方法作为基准方法。在软件缺陷预测严重程度预测问题中,若建模为多元分类问题时,逻辑回归^[13]、K 近邻^[14]、决策树^[15]等机器学习方法是目前主流的建模方法^[16]。除此之外,额外考虑了随机森林^[17]、支持向量机^[18]等其他经典的多分类方法。

多元分类任务的拆分策略有如下 3 种:一对一、一对其余和多对多。在本文中,我们采用基于一对

其余方式的多分类方法^[19]。

除此之外,论文使用了第三方提供的成熟框架^[16],主要基于 Python 机器学习包 scikit-learn 对如下基准方法进行了实现:

1)逻辑回归(logistic regression,LR)^[13] 一种对数几率模型,是离散选择法模型之一,属于多元变量分析范畴,是社会学、生物统计学、临床、数量心理学、计量经济学、市场营销等统计实证分析的常用方法。

2)K 近邻(K-nearest neighbors,KNN)^[14] 一种用于分类和回归的非参数统计方法,在这两种情况下,输入包含特征空间中的 k 个最接近的训练样本。基于相同类别案例彼此相似度高这一原理并采用向量空间模型来实现分类的效果,借由计算与已知类别案例之间的相似度来评估未知类别案例的分类。

3)决策树(decision tree,DT)^[15] 在决策论中,决策树由一个决策图和可能的结果(包括资源成本和风险)组成,用来规划到达目标的路径。决策树是一种特殊的树结构,是一个包括随机事件结果、资源代价和实用性的利用了像树一样的图形或决策模型的决策支持工具,主要被用来辅助决策。决策树经常在运筹学中使用,特别是在决策分析中,它能帮助确定一个最可能达到目标的策略。

4)随机森林(random forest,RF)^[17] 在机器学习中,随机森林是一个包含多个决策树的分类器,并且其输出的类别由个别树输出的类别的众数而定。随机森林天然可用来对回归或分类问题中变量的重要性进行排序。。

5)支持向量机(support vector machine,SVM)^[18] 在机器学习中,支持向量机是在分类与回归分析中分析数据的监督式学习模型与相关的学习算法。给

定一组训练实例,每个训练实例被标记为属于两个类别中的一个或另一个,SVM 训练算法将创建一个将新的实例分配给两个类别之一的模型,使其成为非概率二元线性分类器。除了进行线性分类之外,SVM 还可以使用所谓的核技巧有效地进行非线性分类,将其输入隐式映射到高维特征空间中。

3.5 实验设置

为了合理评估本文所提 VG-BSP 方法的预测性能,我们将数据集进行随机分层抽样,即针对事先已经标注好相应缺陷报告严重程度标签的数据集,依据各类数据在整体中所占的比例,确定从各类中抽取的数据数量,并进一步从中随机抽取样本数据,最终划分出整个数据集中 70% 的数据作为训练数据,并将剩余 30% 数据作为测试数据,以防止训练数据和测试数据产生数据重叠问题。为避免数据集划分过程的随机性,重复上述数据集划分方法 10 次。为了确保实证研究的可重现,使用相同的随机种子,来完成数据集的 10 次划分。

4 实验结果与分析

4.1 针对 RQ1 的分析

在 3 个数据集上分别使用基准方法进行预测,计算得到 macro-F1 值,并与本文提出的方法结果进行比较,统计结果如表 3 所示。与 LR、KNN、DT、RF 和 SVM 方法相比,在数据集 JDT 上,本文提出的 VG-BSP 方法在 macro-F1 指标上最高可提升 37.8%,最低可提升 12.5%,平均可提升 26.6%;在数据集 CDT 上,VG-BSP 方法最高可提升 33.5%,最低可提升 4.6%,平均可提升 18.7%;在数据集 Mozilla 上,VG-BSP 方法最高可提升 19.9%,最低可提升 7.4%,平均可提升 13.3%。基于上述分析,可以看出,VG-BSP 方法的性能明显优于 LR、KNN、DT、RF、SVM 方法,平均可提升 19.6%。从原理上来解释,本文所提的 VG-BSP 方法,首先应用预测性能较好而预测粒度较粗的粗粒度预测方法对数据集进行分类,在很大程度上确保了预测性能后又应用细粒度预测方法进一步地细化了预测粒度,最终得到了细粒度严重程度预测结果。本文所提 VG-BSP 方法结合了粗粒度方法与细粒度方法的优点,弥补了各自的不足,因而整体表现优于直接采用细粒度预测策

略的基准方法。

表 3 VG-BSP 方法与基准方法的性能比较

Tab. 3 Performance comparison between the method VG-BSP and baselines

方法	Eclipse JDT	Eclipse CDT	Mozilla	均值	标准 方差
LR	0.217	0.247	0.363	0.276	0.063
KNN	0.192	0.263	0.350	0.268	0.065
DT	0.185	0.183	0.326	0.231	0.067
RF	0.162	0.198	0.347	0.236	0.080
SVM	0.155	0.229	0.377	0.254	0.092
VG-BSP	0.248	0.276	0.407	0.310	0.069

注:基于 macro-F1 评测指标,并对最好的结果进行加粗。

4.2 针对 RQ2 的分析

在 3 个数据集上分别使用 6 种常用机器学习方法作为粗细粒度预测模型的分类型算法,目的是验证并探寻是否存在一个分类方法,可以在本文提出的 VG-BSP 方法中取得最好的预测性能,最终结果如表 4 所示。总体来说 LR 分类算法在 VG-BSP 方法上可以取得最好的预测性能,尽管在 Mozilla 数据集上,其性能稍弱于 SVM 方法,但差距几乎可以忽略不计。因此,当 VG-BSP 方法选择 LR 分类算法时,模型的预测性能最好。

表 4 不同分类算法对变粒度缺陷报告严重程度预测方法的性能影响

Tab. 4 Performance influence of different classifiers on variable granularity bug report severity prediction method

分类算法	Eclipse JDT	Eclipse CDT	Mozilla	均值	标准 方差
LR	0.248	0.276	0.407	0.310	0.069
KNN	0.196	0.238	0.362	0.265	0.070
DT	0.199	0.208	0.325	0.244	0.057
RF	0.187	0.196	0.396	0.260	0.096
SVM	0.158	0.221	0.409	0.263	0.107

注:最好的结果加粗表示。

4.3 针对 RQ3 的分析

由针对 RQ2 的分析结果可知,基于 LR 分类算法可使变粒度缺陷报告严重程度预测方法取得最好的性能。针对 RQ3,我们尝试分析文本建模方法对 VG-BSP 方法性能的影响。在 3 个数据集上分别使用 4 种常用文本建模方法对数据集的文本进行

建模,并随后基于LR 分类算法来构建缺陷报告严重程度预测模型并计算得到 macro-F1 值。最终结果如表 5 所示,不难看出,Spacy 词向量在所有数据集上都可以取得最好的性能,因此我们选取 Spacy 词向量作为 VG-BSP 方法的文本建模方法。

表 5 不同文本建模方法对变粒度缺陷报告严重程度预测方法的性能影响

Tab. 5 Performance influence of different text modeling methods on variable granularity bug report severity prediction method

文本建模方法	Eclipse JDT	Eclipse CDT	Mozilla	均值	标准方差
Spacy	0.248	0.276	0.407	0.310	0.069
Word2Vec	0.157	0.119	0.124	0.133	0.017
BOW	0.215	0.253	0.399	0.289	0.079
TF-IDF	0.241	0.265	0.405	0.304	0.072

注:最好的结果加粗表示。

5 总结与展望

通过构建缺陷报告严重程度预测模型,可以有效地降低项目开发人员和用户面临的风险,节省提交者在缺陷报告严重程度设置上所花费的时间和精力。本文提出一种变粒度缺陷报告严重程度预测方法 VG-BSP,尝试通过融合细粒度分类与粗粒度分类来兼顾预测的准确性与预测粒度,并基于实际大规模开源项目的缺陷报告,在 macro-F1 指标上对 VG-BSP 方法的有效性进行了验证。除此之外,还探讨了 VG-BSP 方法内最优分类方法和文本建模方法的设置。

该方法仍然存在以下工作有待于今后继续研究:

1) 尝试搜集来自其他开源项目的缺陷报告数据集,对论文结论的一般性进行进一步的探索和验证;

2) 当前数据集内,不同严重程度类型的缺陷报告分布不均衡,即一些严重程度类型的缺陷报告数量很少。因此,如何引入类不平衡学习方法^[20]来进一步提升模型的预测性能值得关注。

参考文献:

[1] SAHA R K, LAWALL J, KHURSHID S, et al. Are these bugs really "normal"?[C]//Proceedings of the 2015 IEEE/

ACM 12th Working Conference on Mining Software Repositories, May 16–17, 2015, Florence, Italy. New York:IEEE Xplore, 2015:258–268.

[2] ZHANG T, YANG G, LEE B, et al. Predicting severity of bug report by mining bug repository with concept profile[C]//Proceedings of the 30th Annual ACM Symposium on Applied Computing, Salamanca, Spain. New York:ACM, 2015:1553–1558.

[3] NGUYEN T T, AN T Q, HAI V T, et al. Similarity-based and rank-based defect prediction[C]//Proceedings of the 2014 International Conference on Advanced Technologies for Communications (ATC 2014), October 15–17, 2014, Hanoi, Vietnam. New York:IEEE Xplore, 2014:321–325.

[4] SHARMA M, KUMARI M, SINGH R K, et al. Multiattribute based machine learning models for severity prediction in cross project context[C]//Proceedings of the International Conference on Computational Science and Its Applications (ICCSA 2014), June 30–July 3, 2014, Guimarães, Portugal. Cham, Switzerland:Springer, 2014:227–241.

[5] TAN Y S, XU S J, WANG Z W, et al. Bug severity prediction using question-and-answer pairs from Stack Overflow[J]. Journal of Systems and Software, 2020, 165:110567.

[6] CHATURVEDI K K, SINGH V B. Determining Bug severity using machine learning techniques[C]//Proceedings of the 2012 CSI Sixth International Conference on Software Engineering (CONSEG), September 5–7, 2012, Indore, India. New York:IEEE Xplore, 2012:1–6.

[7] LAMKANFI A, DEMEYER S, SOETENS Q D, et al. Comparing mining algorithms for predicting the severity of a reported bug[C]//Proceedings of the 2011 15th European Conference on Software Maintenance and Reengineering, March 1–4, 2011, Oldenburg, Germany. New York:IEEE Xplore, 2011:249–258.

[8] MENZIES T, MARCUS A. Automated severity assessment of software defect reports[C]//Proceedings of the 2008 IEEE International Conference on Software Maintenance, September 28–October 4, 2008, Beijing, China. New York:IEEE Xplore, 2008:346–355.

[9] LAMKANFI A, DEMEYER S, GIGER E, et al. Predicting the severity of a reported bug[C]//Proceedings of the 2010 7th IEEE Working Conference on Mining Software Repositories (MSR 2010), May 2–3, 2010, Cape Town, South Africa. New York:IEEE Xplore, 2010:1–10.

- [10] TIAN Y, LO D, SUN C N. Information retrieval based nearest neighbor classification for fine-grained bug severity prediction[C]//Proceedings of the 2012 19th Working Conference on Reverse Engineering, October 15–18, 2012, Kingston, ON, Canada. New York:IEEE Xplore, 2012: 215–224.
- [11] 刘文杰. 软件缺陷报告严重性预测研究[D]. 大连:大连理工大学, 2020.
LIU W J. Software defect reports severity prediction[D]. Dalian:Dalian University of Technology, 2020. (in Chinese)
- [12] 王婧宇, 张欣, 邹卫琴. 基于分类的软件缺陷严重性预测[J]. 计算机与数字工程, 2016, 44(8):1532–1534.
WANG J Y, ZHANG X, ZOU W Q. Severity Prediction of software defects based on classification[J]. Computer & Digital Engineering, 2016, 44(8):1532–1534. (in Chinese)
- [13] GRIMM L G, YARNOLD P R. Reading and understanding multivariate statistics[M]. Washington, DC:American Psychological Association, 1995:217–244.
- [14] FUKUNAGA K, NARENDRA P M. A branch and bound algorithm for computing k-nearest neighbors[J]. IEEE Transactions on Computers, 1975, 24(7):750–753.
- [15] FRIEDL M A, BRODLEY C E. Decision tree classification of land cover from remotely sensed data[J]. Remote Sensing of Environment, 1997, 61(3):399–409.
- [16] 陈翔, 赵英全, 顾庆, 等. 基于文件粒度的多目标软件缺陷预测方法实证研究 [J]. 软件学报, 2019, 30(12): 3694–3713.
CHEN X, ZHAO Y Q, GU Q, et al. Empirical studies on multi-objective file-level software defect prediction method [J]. Journal of Software, 2019, 30(12):3694–3713. (in Chinese)
- [17] LIAW A, WIENER M. Classification and regression by RandomForest[J]. R News, 2002, 2/3:18–22.
- [18] SUYKENS J A K, VANDEWALLE J. Least squares support vector machine classifiers[J]. Neural Processing Letters, 1999, 9(3):293–300.
- [19] 周志华. 机器学习[M]. 北京:清华大学出版社, 2016:63.
- [20] 曲豫宾, 陈翔. 基于代价敏感主动学习的软件缺陷预测方法[J]. 南通大学学报(自然科学版), 2019, 18(1):9–15.
QU Y B, CHEN X. Software defect prediction method based on cost-sensitive active learning[J]. Journal of Nantong University (Natural Science Edition), 2019, 18(1): 9–15. (in Chinese)

(责任编辑:仇慧)