

融合文本与分类信息的重复缺陷报告检测方法

范道远¹ 孙吉红² 王 炜^{1,3} 涂吉屏¹ 何 欣¹

(云南大学软件学院 昆明 650500)¹ (云南省科学技术院 昆明 650091)²

(云南省软件工程重点实验室 昆明 650500)³

摘 要 软件缺陷是软件出现错误、故障的根源。软件缺陷是需求分析不合理、编程语言不严谨、开发人员缺少经验等因素导致的。软件缺陷不可避免,提交缺陷报告是发现缺陷并改进缺陷的重要途径。缺陷报告是描述缺陷的载体,对缺陷报告的修复是完善软件的必要手段。维护人员和用户因同一缺陷重复提交报告,导致缺陷报告库中存在大量冗余的报告,手动分诊已无法适应越来越复杂的软件系统。重复缺陷报告检测能过滤缺陷报告库中冗余的重复报告,并将人力与时间投入到新的缺陷报告上。当前研究方法的预测准确率始终不高,其难点在于寻找一个合适且全面的方法来衡量缺陷报告之间的相似性。借鉴集成方法的思想,提出了一种基于文本信息、分类信息相融合的重复缺陷报告检测方法——BSO(combination of BM25F、LSI and One-Hot)。在数据预处理的基础上,文中将重复缺陷报告分割为文本信息域与分类信息域。在文本信息域上使用 BM25F 与 LSI 算法,得到两个方法的相似性打分,运用相似性融合方法将两个方法的相似性打分进行整合;在分类信息域上使用 One-Hot 算法得到相似性打分。运用相似性融合方法,融合文本信息域与分类信息域的相似性打分,为每个缺陷报告对应一个重复缺陷报告推荐列表,并计算重复缺陷报告检测的准确率。利用 Python 语言,在公开的数据集 OpenOffice 上与基线方法以及较新水平方法 REP、DBTM 进行对比。实验结果表明,与 DBTM 相比,本文方法的准确率平均提高了 4.7%;与 REP 方法相比,本文方法的准确率平均提高了 6.3%;与基线方法相比,本文方法的准确率提升较高。实验结果充分证明了 BSO 方法的有效性。

关键词 重复缺陷报告,信息检索方法,主题模型,One-Hot,相似性融合

中图法分类号 TP311.5 文献标识码 A DOI 10.11896/jsjx.181102232

Detection Method of Duplicate Defect Reports Fusing Text and Categorization Information

FAN Dao-yuan¹ SUN Ji-hong² WANG Wei^{1,3} TU Ji-ping¹ HE Xin¹

(College of Software,Yunnan University,Kunming 650500,China)¹ (Academy of Sciences in Yunnan Province,Kunming 650091,China)²

(Key Laboratory for Software Engineering of Yunnan Province,Kunming 650500,China)³

Abstract Software defect is the root of software errors and failures. Software defect is caused by unreasonable requirement analysis,imprecise programming language and lack of experience of developers. Software defects are inevitable,and submitting defect reports is an important way to find and improve defects. Defect report is the carrier of describing defects,and the repair of defect report is the necessary means to improve software. Maintenance personnel and users submit reports for the same defect repeatedly,resulting in a large number of redundant reports in the defect report library. Manual triage is unable to adapt to more and more complex software systems. The detection of duplicate defect reports can filter redundant duplicate reports from defect report libraries and invests human and time in new defect reports. The prediction accuracy rate of current research methods is not high,and the difficulty is to find a suitable and comprehensive method to measure the similarity between defect reports. Based on the idea of the integration method and the python language,a new method named BSO (combination of BM25F,LSI and One-Hot) for detecting duplicate defect report was proposed by using text information and categorization information. On the basis of data preprocessing,duplicate defect report is divided into text information domain and categorization information domain. BM25F and LSI algorithms are used to get similarity scores in text information domain,and One-Hot algorithm is used to get similarity scores in categorization information domain. The similarity fusion method is used to synthesize the similarity score be-

到稿日期:2018-11-30 返修日期:2019-05-29
范道远(1993—),女,硕士生,CCF 学生会员,主要研究领域为软件工程、软件演化、数据挖掘,E-mail:1367178901@qq.com;孙吉红(1983—),男,硕士,助理研究员,CCF 学生会员,主要研究领域为计算机应用,E-mail:3217357113@qq.com(通信作者);王 炜(1979—),男,博士,副教授,CCF 会员,主要研究领域为软件工程、软件演化、数据挖掘;涂吉屏(1994—),女,硕士,CCF 学生会员,主要研究领域为软件工程、软件演化、数据挖掘;何 欣(1994—),女,硕士生,CCF 学生会员,主要研究领域为软件工程、软件演化、数据挖掘。

tween text information domain and categorization information domain,and a recommendation list for each defect report corresponds to a duplicate defect report. The accuracy of the duplicate defect report detection is calculated. Compared with the baseline method and the state-of the art methods including REP and DBTM on OpenOffice. The experimental results show that the accuracy of the proposed method is 4.7% higher than that of DBTM,6.3% higher than that of REP,and higher than that of baseline method. Experiment results fully prove the effectiveness of BSO method.

Keywords Duplicate defect report,Information retrieval method,Topic model,One-Hot,Similarity fusion

1 引言

缺陷报告作为跟踪软件缺陷的载体,是描述软件缺陷、故障或者错误的一种结构化文档^[1-2]。维护人员与用户会因同一缺陷重复提交报告,导致缺陷报告库中存在大量冗余的重复报告^[3]。2005 年,Mozilla 重复缺陷报告数量占有所有报告数量的 23%;Eclipse 和 Firefox 中有 20%~40% 的缺陷报告被标记为重复报告。当一个报告提交后,相关人员将判断该报告是新的缺陷报告还是重复的缺陷报告^[4]。随着缺陷报告库规模越来越大,人工处理越来越困难,耗费的时间、人力、费用大大增加,降低了软件的维护效率。相关人员将重复的缺陷报告独立地解决会导致资源的浪费。因此,一种合适与全面的重复缺陷报告检测方法是具有重要意义的。

重复缺陷报告检测方法是指选择缺陷报告的部分信息或是执行信息作为输入,运用一系列的相关性算法衡量缺陷报告之间的相似性,为某一个缺陷报告推荐与它最相似的一个或多个缺陷报告或是标记出与某一报告重复的缺陷报告^[5]。根据研究对象的不同,当前的重复缺陷报告检测方法可分为 3 类,分别是基于自然语言的重复缺陷报告检测方法^[5-6]、基于执行信息^[7]的重复缺陷报告检测方法和基于实证研究的重复缺陷报告检测方法^[8]。由于包含执行信息的缺陷报告数量极少且执行信息很难获取,实证研究需要缺陷报告库的相关实践知识,本文针对基于自然语言的重复缺陷报告进行研究。针对此类研究,国内外学者自 2007 年起已提出多种重复缺陷报告检测方法,目前比较主流的方法有 BM25F^[9-11],以 TF-IDF 为基础构建向量空间模型^[1]、REP^[12]、二元分类^[13-15]、神经网络^[16-17]、主题模型^[18]、DBTM^[19]等。当前方法主要存在以下不足:1)同一缺陷导致不同的错误现象,以及对同一错误不同的人会用不同的词语表述,这使得重复缺陷报告的文本并不相似。经典的相关性信息检索方法只有在文本相似的情况下表现得,因此在这种情况下,信息检索方法的准确率^[9,12]。虽然主题模型方法能挖掘潜在语义信息,但基于词频的主题模型方法无法消除同义词对相似性刻画的影响,预测准确率^[20-21]。2)没有区别对待分类信息相似性与文本信息相似性^[19]对重复检测的作用大小。3)预测准确率不够高。本文借鉴了集成学习的思想,提出了一种基于文本信息和分类信息的重复缺陷报告检测方法——BSO。在文本信息上将流行文本相关性算法 BM25F 与潜语义索引算法 LSI^[22-23]相结合,并线性融合文本信息相似性与分类信息相似性,得到缺陷报告之间的相似性。

本文的贡献如下:1)弥补了用不同术语描述同一缺陷的情况下,经典的相关性信息检索方法准确率降低以及主题模

型方法预测准确率低的缺点;2)考虑了分类信息相似性与文本信息相似性对重复检测的贡献大小不一致的问题;3)通过实验验证了本文方法的有效性。

本文第 2 节阐述了研究背景及相关工作;第 3 节对本文方法的具体内容进行了阐述;第 4 节对本文方法进行了实验验证,介绍数据集及评价指标,并对实验结果进行了分析;第 5 节介绍了本文方法的局限和不足;最后总结全文。

2 研究背景及相关工作

2.1 研究背景

缺陷报告组成部分一般如表 1 所列^[24],其中,Product,Component,Version,Priority,Type,Status,Resolution 为分类信息域,Summary 和 Description 为文本信息域。表 2 列出了缺陷报告示例。

表 1 缺陷报告组成部分

Table 1 Compositions of defect report		
组成部分	类型	含义
ID	整型	缺陷报告编号
Summary	文本	问题的简要概述
Description	文本	对问题的详细描述
Type	枚举	缺陷报告的类型
Product	枚举	缺陷报告涉及的产品
Component	枚举	缺陷报告涉及的产品部分
Version	枚举	缺陷报告涉及的产品版本
Priority	枚举	缺陷报告被解决的优先权
Severity	枚举	所描述缺陷的严重程度
Status	枚举	缺陷报告的状态
Resolution	枚举	缺陷报告处理方式

表 2 缺陷报告示例

Table 2 Examples of defect report	
报告示例	
ID	850
Summary	Users can not find the reader's files correctly Click "search",do not enter the query or do not fill in the query conditions,there will be corresponding warning information. If the query condition does not exist,there is a corresponding prompt statement,"the content of your query is not correct. Please check it again."
Description	
Type	Nonfunctional
Product	word
Component	UI
Version	unspe
Priority	P3
Severity	defect
Status	new
Resolution	fixe

缺陷报告重复的情况为^[1]:缺陷报告 BG₁ 和 BG₂ 用不同术语或相同术语描述相同的问题,该问题来源于同一个缺陷;

缺陷报告 BG_1 和 BG_2 描述不同的问题,该问题来源于同一个缺陷。具体如图 1 所示。

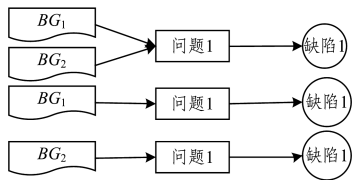


图 1 重复缺陷报告
Fig. 1 Duplicate defect reports

2.2 相关工作

最早的重复缺陷报告检测方法是以词频为依据建立的向量空间模型法^[24],用向量之间的余弦值衡量缺陷报告之间的相似度。该方法噪音数据多、维度高。Runeson 等^[1]提出了 NLP 方法,其与向量空间模型法的不同之处在于采用了词干提取、去停用词等自然语言处理进程来对数据进行预处理,去除噪音数据。Sun 等^[12]提出以 BM25F 算法为核心的 REP 方法。REP 方法考虑了不同文档域、文档长度的单词对缺陷报告相关性的贡献。BM25F 算法的核心是对文档进行语素分解,将每个语素与另一个文档的相关性得分加权求和,得到两文档之间的相似性。REP 表现较优的前提是重复缺陷报告之间的术语表达是非常相似的,然而缺陷报告来自于有着不同书写习惯的提交者,导致相同缺陷有着不同的言语表达。Somasundaram 等^[18]首次将 LDA 模型运用到重复缺陷报告检测上,LDA 主题模型算法能在一定程度上解决相同缺陷用不同术语表达的问题,但仅是 LDA 主题模型算法的预测准确率远远低于 REP 的预测准确率^[25]。大量实践表明,LSI^[26]在文本相关性打分上优于 LDA。Nguyen 等^[19]提出了 DBTM 方法,该方法将主题模型方法与传统相关性检索方法结合。DBTM 方法将文本信息与分类信息^[27]作为重复检测的数据对象,但是其没有区别对待分类信息相似性与文本信息相似性对重复检测的作用大小。Alipour 等^[8]提出基于上下文信息的重复缺陷报告检测方法。它将组成缺陷报告的单词或词语分成功能性的、非功能性的、描述性的等多类^[28],运用 LDA 分别计算各类主题概率分布。该方法在实际运用中操作复杂,难以实现。

借鉴以上研究方法以及集成方法的思想,本文提出了一种基于文本信息与分类信息的重复缺陷报告检测方法。在文本信息上将流行文本相关性算法 BM25F 与潜语义索引 LSI 算法相结合,弥补了用不同术语描述同一缺陷的情况下,经典的相关性信息检索方法准确率降低以及主题模型方法预测准确率低的缺点。MAP^[29]线性融合文本信息相似性与分类信息相似性,解决了分类信息相似性与文本信息相似性对重复检测的作用大小不一致的问题。由于分类信息数据的特殊性,首次使用了 One-Hot^[30]方法进行编码。它保证每一个取值使一种状态处于“激活态”,各个取值地位对等,公平表现差异。BM25F 是典型 BM25^[31]的改进算法。BM25 在计算相关性时把文档当作整体来考虑,而结构化文档由多个独立的域组成。BM25F 算法对不同的区域赋予不同的权值,以此反映不同区域的不同重要性。LSI 利用 SVD 降维的方法将词

项和文本映射到一个新的二维空间,从而得到文档的主题分布。

3 本文 BSO 方法

本文提出的 BSO 方法的框架如图 2 所示。为了减小噪音的影响,本文采用统一小写形式、去除符号、词根还原、停词处理等方法,对数据进行预处理。

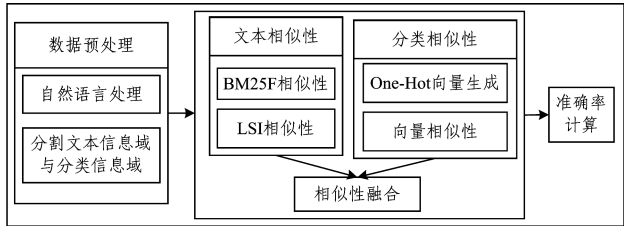


图 2 研究方法框
Fig. 2 Research method framework

本文方法将重复报告分割为文本信息域和分类信息域。针对文本信息域,使用 BM25F 算法以及 LSI 算法,得到相似性打分矩阵 Sim_1 与 Sim_2 。运用 MAP 方法确定线性结合的系数,将两个相似性打分矩阵融合,得到文本信息相似性。针对分类信息域,使用 One-Hot 编码方法,将每一个缺陷报告转换成向量形式,计算向量余弦值,输出相似性打分矩阵 Sim_3 ,该矩阵表示分类信息相似性。最后使用 MAP 融合算法融合文本信息相似性和分类信息相似性,得到最终相似性打分矩阵 Sim 。根据最终相似性打分矩阵 Sim ,为每一缺陷报告返回与之相似度较高的前 L 个缺陷报告,前 L 个缺陷报告的 ID 组成该报告的推荐列表。最终计算本文方法的检测准确率。

3.1 数据预处理

一个缺陷报告可分为两部分:Summary 为文本信息域数据;Product, Component, Version, Priority, Type, Status, Resolution 为分类信息域数据。

为了减小噪音的影响,提高计算准确率,需要对原始语料库进行预处理,主要包括以下 5 个方面:1)同一个单词的大小写形式语义相同,但是在进行相似性计算时,不同的书写形式会导致不同的结果。因此,本文将大写转换成小写,统一采用小写形式。2)去除句号、逗号、括号、连字符等一切符号。符号对相似性计算没有任何意义,甚至会产生负影响。3)词根还原。词语的形态不同,表达的语义却是一致的,因此将一个词的不同形态进行还原,如将 worked 和 working 都写成 work。4)停词处理。本文的主要工作为计算缺陷报告之间的语义相似性,冠词、谓词、程度副词、人称代词、介词、连接词等类型的词携带语义信息很少,对相似性计算的意义不大,因此进行去停用词处理。5)由于实验的需要,所有单词之间以一个空格分隔。

3.2 相似性计算

3.2.1 文本相似性计算

(1)BM25F 相似性计算

将 BM25F 算法运用于缺陷报告相关性计算的原理为:

对缺陷报告 q 进行语素解析,生成 m 个语素;对于缺陷报告 d ,计算语素 q_i 与 d 的相关性得分;将 q_i 相对于 d 的相关性得分加权求和,得到缺陷报告 d 与缺陷报告 q 的相关性得分,如式(1)所示:

$$BM25F\text{-}score(d,q)=\sum_1^mIDF(q_i)\cdot R(q_i,d)$$

(1)

$$IDF(q_i)=\log\frac{N-n(q_i)+0.5}{n(q_i)+0.5}$$

(2)

式(1)中, $IDF(q_i)$ 是语素 q_i 的普遍重要性度量,为逆文本频率指数, IDF 值越大说明该语素区分能力越好, $R(q_i,d)$ 为加权前语素 q_i 与缺陷报告 d 的相关性得分, m 为缺陷报告 q 的语素数量。式(2)中, N 为全部缺陷报告文档数, $n(q_i)$ 为包含了 q_i 的缺陷报告数量。 N 为 10,假设包含了语素 files 的缺陷报告数量为 4,由式(2)可得 $IDF(files)=0.15$ 。对于给定的缺陷报告集合,包含 q_i 的报告数量越多, q_i 的区分度不高,根据 q_i 判断相关性的重要度较低,故使用 $IDF(q_i)$ 调节权重。

$$R(q_i,d)=\frac{f(q_i,d)(k_1+1)}{f(q_i,d)+k_1(1-b+b\cdot\frac{dl}{avgdl})}$$

(3)

$$f(q_i,d)=\sum_1^sv_s\cdot f'(q_i,d,S)$$

(4)

$$f'(q_i,d,S)=\frac{f(q_i,d,S)}{1-b+b\cdot(l(d,S)/l_s)}$$

(5)

式(3)一式(5)中, k_1 和 b 为调节因子,通常 $k_1=2,b=0.75$; $f(q_i,d)$ 为 q_i 在缺陷报告 d 中的出现频率,等同于 q_i 在报告 d 所有区域的频率加权之和; dl 为缺陷报告 d 的长度; $avgdl$ 为所有报告的平均长度; v_s 为每个区域的权重; $f(q_i,d,s)$ 为语素 q_i 在文档 d 的 s 域中出现的次数; $l(d,s)$ 为报告 d 的 s 区域的长度; l_s 为所有区域的平均长度。

给定 10 个缺陷报告集 D ,表 3 列出了数据预处理后两个缺陷报告 d 与 q 的文本信息,文本信息仅包含 Summary 一个域。报告 q 生成 5 个语素:users,find,readers,files,correct。以语素 files 为例,计算与报告 d 的相关性得分。语素 files 在报告 d 中出现一次,故 $f(files,d)=1$ 。已知 $k_1=2,b=0.75,dl=7$,假设 10 个报告的平均长度为 8,即 $avgdl=8$,可得 $R(files,d)=1.176$ 。语素 files 与报告 d 的相关性得分为:

$$BM25F\text{-}score(files,d)=IDF(files)\cdot R(files,d)$$

$$IDF(files)\cdot R(files,d)=0.15\cdot 1.176=0.176$$

表 3 缺陷报告文本示例

Table 3 Text sample of defect report

报告	文本内容
d	open remote revision files default text edit
q	users find readers files correct

同理,计算余下 4 个语素 users,find,readers,correct 与报告 d 的相关性得分,分别为 0,0,0,0,因此报告 d 与报告 q 的相关性得分为:

$$BM25F\text{-}score(d,q)=\sum_1^5IDF(q_i)\cdot R(q_i,d)$$

$$\sum_1^5IDF(q_i)\cdot R(q_i,d)=0+0+0+0.176+0=0.176$$

缺陷报告集上的 BM25F 相关性计算的伪代码如算法 1

所示。相似性打分矩阵如表 4 所列。

算法 1 BM25F 相似性计算

```
输入:缺陷报告数据集 B,数量 N
输出:相似性打分矩阵 Sim1
For i=1 to N
    For j=1 to N
        Sim1=BM25F (b1,bj)
    End For
```

表 4 相似性打分矩阵

Table 4 Similarity scoring matrix

ID	Bug ₁	Bug ₂	Bug ₃	Bug ₄	...	Bug _n
Bug ₁	1	0.246	0.009	0.12	...	0.005
Bug ₂	0.246	1	0.0	0.02	...	0.003
Bug ₃	0.009	0.0	1	0.211	...	0.18
Bug ₄	0.12	0.02	0.211	1	...	0.2
...	...	...	...	...	...	...
Bug _n	0.005	0.03	0.18	0.2	...	1

(2)LSI 相似性计算

LSI 方法的思想是假设所有的缺陷报告包含 K 个技术缺陷,每个技术缺陷为一个 Topic,计算每个缺陷报告的 Topic 概率分布,通过计算概率分布的余弦相似性得到缺陷报告之间的相似性。统计语料库中的文本信息,将其转换为 TF-IDF 矩阵,再使用奇异值分解,得到每个缺陷报告的 Topic 概率分布。缺陷报告数据集可以表示为一个 $m\times n$ 的矩阵 $\mathbf{A}$, m 为缺陷报告文档数量, n 为每个缺陷报告文档的单词数, A_{ij} 为第 i 个缺陷报告的第 j 个单词的值。在不降维的情况下进行奇异值分解,可以得到如式(6)所示的 3 个矩阵:矩阵 $\mathbf{U}$ 表示 m 个文档与 m 个主题之间的相关度,矩阵 $\mathbf{\Sigma}$ 表示 m 个词与 m 个主题之间的相关度,矩阵 $\mathbf{V}$ 表示 n 个词之间的相关度。

$$\mathbf{A}_{m\times n}=\mathbf{U}_{m\times m}\mathbf{\Sigma}_{m\times n}\mathbf{V}_{n\times n}^T$$

(6)

TF-IDF 和奇异值分解的定义与基本理论请参考文献 [32-33]。

为了将维度降低到 $K(K$ 为设定的 Topic 数),奇异值降维分解后如式(7)所示,矩阵表示与式(6)一样,其中 $\mathbf{U}$ 表示第 m 个文档与第 k 个主题的相关度,根据矩阵 $\mathbf{U}$,就可以进行余弦相似性计算。

$$\mathbf{A}_{m\times n}=\mathbf{U}_{m\times k}\mathbf{\Sigma}_{k\times k}\mathbf{V}_{k\times n}^T$$

(7)

表 5 是对应于表 3 的 TF-IDF 矩阵,列向量对应于缺陷报告,行向量对应于语料库中出现过的词汇。将表 5 中的矩阵进行奇异值分解,假设主题数 Topic 为 2(即降到 2 维),得到矩阵 $\mathbf{U}_{2\times 2}$,即每个缺陷报告的 Topic 概率分布如表 6 所列。通过计算 Topic 概率分布的余弦相似性,即把 d 与 q 看成两个向量,计算向量的余弦值,得到 d 与 q 的相似性为 0。

表 5 TF-IDF 矩阵示例

Table 5 Matrix sample of TF-IDF

	d	q		d	q
open	0.14	0	text	0.14	0
remote	0.14	0	edit	0.14	0
revision	0.14	0	users	0	0.25
files	0.098	0.14	find	0	0.25
default	0.14	0	readers	0	0.25
			correct	0	0.25

表 6 Topic 概率分布示例

Table 6 Sample of Topic probability distribution

	<i>d</i>	<i>q</i>
Topic1	0.5205	0
Topic2	0	0.3548

$$sim(d,q)=\frac{0.5205*0+0*0.3548}{\sqrt{0.5205^2+0^2}*\sqrt{0^2+0.3548^2}}=0$$

LSI 相似性计算的伪代码如算法 2 所示。第 1 行由 LSI 生成得到大小为 $N \times K$ 的 Topic 概率分布矩阵。第 2—5 行计算 N 个向量(N 个缺陷报告)之间的余弦相似性。相似性打分矩阵 $\mathbf{Sim}_2$ 如表 4 所列。

算法 2 LSI 相似性计算

输入:缺陷报告数据集 B,数量 N,Topic 数为 k

输出:相似性打分矩阵 $\mathbf{Sim}_2$

1. $P_{N \times k} = \text{LSI}(B)$

2. For i=1 to N

3. For j=1 to N

4.
$$\mathbf{Sim}_2 = \frac{\sum_{z=1}^k x_{iz} y_{jz}}{\sqrt{\sum_{z=1}^k x_{iz}^2} \sqrt{\sum_{z=1}^k y_{jz}^2}}$$

5. End for

3.2.2 分类相似性计算

由于缺陷报告分类信息域的属性取值由单一的单词组成,在进行相似性计算之前,本文采用 One-Hot 方法进行向量生成。缺陷报告分类信息域中的每一种属性对应一个状态寄存器,每一种属性的取值对应一种寄存器状态。将多个寄存器状态相连,形成相似性计算的向量,以余弦值衡量报告之间的相似性。假设每个缺陷报告的分类信息域包括 3 个属性,可能的取值情况如下:

1)Priority:low,medium,lowest,high

2)Resolution:fixe,dupl

3)Version:inher,unspe,maj

存在 3 个缺陷报告,取值情况如表 7 所列。One-Hot 编码如表 8 所列。

表 7 缺陷报告取值

Table 7 Defect report value

	报告 1	报告 2	报告 3
Priority	low	medium	high
Resolution	fixe	dupl	fixe
Version	maj	inher	unspe

表 8 One-Hot 编码

Table 8 One-Hot code

	Priority(4)	Resolution(2)	Version(3)
报告 1	1000	10	001
报告 2	0100	01	100
报告 3	0001	10	010

因此,生成向量如下:

报告 1—>[1,0,0,0,1,0,0,0,1]

报告 2—>[0,1,0,0,0,1,1,0,0]

报告 3—>[0,0,0,1,1,0,0,1,0]

以 One-Hot 方法为基础的相似性计算的伪代码如算法 3 所示。第 1—5 行收集 M 个属性的取值情况。第 6—7 行根据属性情况,为各个缺陷报告进行 One-Hot 编码,生成 N 个向量。第 8—11 行计算 N 个缺陷报告间的相似性。

算法 3 One-Hot 相似性计算

输入:缺陷报告集 B,数量 N,属性数 M 输出:相似性打分矩阵 $\mathbf{Sim}_3$

1. For i=1 to M

2. For j=1 to N

3. Collect state_{ij}

4. Return state₁, ..., state_M

5. End for

6. For i=1 to N

7. Vec_{N×k}=One-Hot (bi,state₁, ..., state_M)

8. For i=1 to N

9. For j=1 to N

10.
$$\mathbf{Sim}_3 = \frac{\sum_{z=1}^k x_{iz} y_{jz}}{\sqrt{\sum_{z=1}^k x_{iz}^2} \sqrt{\sum_{z=1}^k y_{jz}^2}}$$

11. End for

3.3 相似性融合

本文采用线性方法将两个缺陷报告相似性矩阵相结合,如式(8)所示:

$$\mathbf{y} = \alpha_1 \times \mathbf{y}_1 + \alpha_2 \times \mathbf{y}_2 \tag{8}$$

其中, $\alpha_1 + \alpha_2 = 1$, $\mathbf{y}_1$ 和 $\mathbf{y}_2$ 为文档相似性矩阵。单一相似性打分矩阵内容不再改变,因此线性结合的重点在于确定 α_1 的取值,使得结合后的相似性打分矩阵性能最优,依托相似性打分矩阵预测的准确率最优。本文将数据集随机分为 10 份,其中 9 份作为训练集,1 份作为测试集。通过训练集确定最优的参数 α_1 ,将该 α_1 作为测试集的确定参数输入。在训练过程中,返回 MAP 值最大时的 α_1 的取值,该值为最优取值。MAP 方法的思想如式(9)所示:

$$MAP(L_{\text{real}}, L_{\text{pred}}) = \frac{1}{|L_{\text{pred}}|} \sum_{i=1}^{|L_{\text{pred}}|} \frac{1}{index_i} \tag{9}$$

其中, L_{real} 表示缺陷报告之间实际的重复关系列表, L_{pred} 表示本文方法预测得到的重复关系列表, $index_i$ 表示在缺陷报告 i 的推荐列表中与它实际重复的缺陷报告索引位置。缺陷报告的索引位置越靠前,索引效果越好,MAP 的值越大,因此返回 MAP 值最大时的 α_1 。

MAP 结合相似性打分矩阵 $\mathbf{Sim}_1$ 与 $\mathbf{Sim}_2$ 的伪代码如算法 4 所示。令 α_1 从 0 开始,每次增加 0.01,每次生成一个新的 $\mathbf{Sim}$ 相似性打分矩阵。根据 $\mathbf{Sim}$ 矩阵,按相似性打分由高到低,为每一个缺陷报告生成一个推荐列表 L_{pred} ,计算 MAP 值,直至 α_1 为 1。返回 MAP 值最大时的 α_1 。

算法 4 MAP 结合

输入:缺陷报告集 B,重复关系列表 L_{real} ,相似性打分矩阵 $\mathbf{Sim}_1$ 和

$\mathbf{Sim}_2$

输出:相似性打分矩阵 $\mathbf{Sim}$

1. MAP=0

2. For $\alpha_1=0$ to 1

3. Increase α_1 by 0.01

```
4.    Sim= $\alpha_1$  * Sim1 +  $\alpha_2$  * Sim2
5.    Lpred = index_list(Sim)
6.    MAP(Lreal * Lpred)
7. End for
8. Return  $\alpha_1$  when maximum MAP
9. Renew Sim
```

4 实证研究

本节将利用公开数据集 OpenOffice 对本文方法的有效性进行实证研究,主要包括以下几个方面:1)数据集描述及评价指标选择;2)根据本文贡献以及需要回答的问题进行实验设计,描述实现工具;3)依据评价指标对实验结果进行分析,并回答 3 个研究问题。

针对本文的实证研究设计的 3 个问题如下:

RQ1:与单一方法相比,集成方法即流行文本相关性算法与潜语义索引模型算法集成是否能表现出较优的检测性能?

RQ2:相似性融合即区别对待分类信息相似性与文本信息相似性后,检测方法性能是否优于将分类信息与文本信息作为整体考虑的检测性能?

RQ3:与基线方法以及与较新方法 DBTM 和 REP 相比,BSO 方法是否能表现出较优的检测性能?

4.1 数据集及评价指标

4.1.1 数据集

本文缺陷报告数据集来源于 OpenOffice¹⁾。该类缺陷报告描述了在使用 OpenOffice 的文字处理、电子表格、演示文稿、公式、绘图、资料库六大功能时遇到的问题。缺陷报告数量为 12912,重复缺陷报告数量为 12912。对于每一个缺陷,至少存在一个及以上的重复关系。12912 个报告中有 4204 个含有属性缺失,经过数据预处理后,可用缺陷报告数量为 8708 个。原始数据集为 2 个,数据集 1 为 8708 个缺陷报告,数据集 2 记录了 8708 个报告之间的重复关系。数据集 1 被分割成两部分,一部分是文本信息,另一部分是分类信息。数据集 2 进行简单的重复关系处理,每一个报告的重复报告会在其之后列出。

分类信息共有 Product,Component,Status,Version,Priority,Severity,Resolution 7 个特征,由于 Product 特征的取值都是 OpenOffice,没有区分性能,因此分类特征为 6 个。每个特征下不同情况的取值(由于特征取值属性较多,只列出了 4 个取值)如表 9 所列。

表 9 特征取值

Table 9 Feature values

特征	取值 1	取值 2	取值 3	取值 <i>n</i>	状态数
Component	calc	libreoff	basic	...	28
Status	reso	clos	new	...	9
Resolution	fixe	dupl	work	...	8
Version	inher	unspe	maj	...	81
Priority	low	medium	lowest	...	5
Severity	enh	tri	min	...	7

4.1.2 评价指标

为检测本文方法实验结果的有效性,依据实验结果属性,用该领域常用的准确率为评价指标,准确率越高,检测性能越好。根据相似性打分矩阵,为每一个缺陷报告返回长度为 *L* (取值为 1 到 20)的重复缺陷报告推荐列表,按相似性打分由高到低排列。若推荐列表中存在 1 个及以上的缺陷报告与该报告重复,则对于该报告的重复报告的检测为成功。

准确率=查找成功的缺陷报告数量/重复缺陷报告总数量

4.2 实验设计

本文实验采用 Python3.5 实现,使用 gensim 库创建单一及基线方法,利用 OpenOffice 数据集进行了实验验证。

本文设计了 3 组对照实验:1)针对提出的 RQ1,设计了第一组对照实验。将分类信息与文本信息作为一个整体信息,在整体信息上分别使用 LSI 方法以及 BM25F 方法作为单一方法。将用 MAP 方法线性集成的 LSI&BM25F 方法作为集成方法,并验证集成方法是否能表现出较优的检测性能。2)针对提出的 RQ2,设计了第二组对照实验。首先在整体信息上分别使用 LSI,BM25F,LSI&BM25F 方法;其次在文本信息上分别使用 LSI,BM25F,LSI&BM25F 方法,在分类信息上使用 One-Hot 方法,并用相似性融合方法线性集成 LSI&One-Hot,BM25F&One-Hot,BSO。验证区别对待分类信息相似性与文本信息相似性后,检测方法性能是否优于将分类信息与文本信息作为整体考虑的检测性能。3)针对提出的 RQ3,设计了第三组对照实验。在文本信息上,分别使用 LSI 与 BM25F 方法,并用相似性融合方法线性集成为 LSI&BM25F;在分类信息上,使用 One-Hot 方法。再次运用相似性融合方法集成 LSI&BM25F 与 One-Hot,即本文提出的 BSO 方法。将 DBTM,REP,BM25F,LDA,LSI 作为基线方法,其中 DBTM 与 REP 方法是较新较优的相似性检测方法。验证 BSO 方法与这些方法相比,检测性能是否更优。

表 10 对照实验

Table 10 Controlled experiments

RQ1	LSI&BM25F	VS	LSI
	LSI&BM25F	VS	BM25F
RQ2	LSI&One-Hot	VS	LSI
	BM25F&One-Hot	VS	LSI
	BSO	VS	LSI&BM25F
	BSO	VS	DBTM
RQ3	BSO	VS	BM25F
	BSO	VS	LDA
	BSO	VS	LSI
	BSO	VS	REP

4.3 实验结果分析

推荐列表数过高,检测准确率高,检测性能优;推荐列表数过低,导致检测准确率过低,检测性能不优。为了让检测准确率与检测性能平衡,且与该领域的相关研究保持一致,本文的推荐列表个数的范围为 1~20。

在上述 3 组对照实验之外,本文就参数 *K* (Topic 主题数)取值的不同,对 BSO 方法实验结果的准确率的影响进行

¹⁾ https://zenodo.org/record/579547#.Wd93YPmCyM_

了分析。实验结果如图 3 所示。

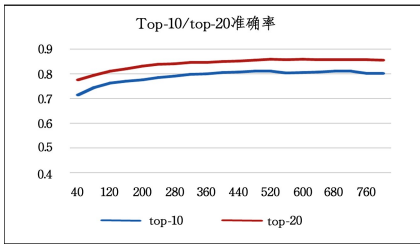


图 3 实验展示结果(1)(电子版为彩色)

Fig. 3 Display of experimental results(1)

此次实验由两个小实验组成，一个是在推荐列表数为 10 的情况下,BSO 方法准确率随参数 K 的变化而变化的情况，另一个是在推荐列表数为 20 的情况下,BSO 方法随参数 K 的变化而变化的情况。由于数据集较大,根据经验,本文从 K 为 40 开始实验,每次 K 增加 40,直到 760 左右。从图 3 可以看出,当 K 较小时,BSO 方法的准确率较低。在 K 小于 520 时,BSO 方法的准确率随 K 的增加而增加,特别是在 K 小于 280 时,增长较快。这是因为当数据集较大而主题数较少时,较少的主题特征不能很好地区分缺陷报告之间的缺陷问题。在 K 大于 520 时,准确率增长缓慢,甚至在个别点上出现下降。这是因为主题数高,特征过细,缺陷报告之间在很多缺陷方面出现相同的问题,导致缺陷报告之间的区分不明显,从而准确率下降。因此,在下文的对照实验中,我们选取实验结果最好的 K 值进行实验。

(1)RQ1 分析

这里主要验证与单一方法相比,集成方法即流行文本相关性算法与潜语义索引算法集成是否能表现出较优的检测性能。图 4 给出了针对 RQ1 的实验结果,表 11 列出了针对 RQ1 的指标提升。

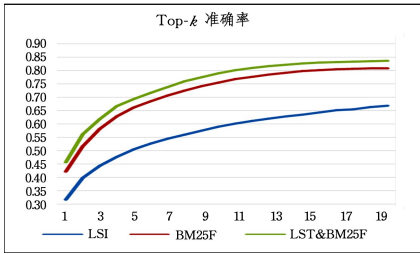


图 4 实验展示结果(2)(电子版为彩色)

Fig. 4 Display of experimental results(2)

表 11 针对 RQ1 指标的提升情况

Table 11 Index promotion for RQ1

(单位: %)

方法	LSI 为基础的 指标提升	BM25F 为基础的 指标提升
Top1—5	40	6.8
Top6—10	35	4.6
Top11—15	32	4.0
Top16—20	27	3.4
平均提高	33.5	4.7

从图 4 以及表 11 中可以看出:1)3 种方法的准确率都随着推荐数量的增多而持续增长。Top 数为 12 之后,增长比较

缓慢,其原因可能是在 Top 数为 12 之后,相似度排名趋于平稳,大部分的重复缺陷报告被检测到,因此之后准确率提升不明显。2)3 种方法中准确率最高的是 LSI&BM25F 方法,在单一方法中,BM25F 检测性能较 LSI 主题模型方法更高。但 LSI&BM25F 方法在 LSI 和 BM25F 的基础上分别有 33.5% 和 4.7% 的提高,且在推荐列表数较少的情况下,指标提高更加明显。其原因可能是 Top 数较多后,大部分重复报告已被检测到,各方法之间的准确率的差距趋于变小。3)主题模型方法 LSI 的准确率虽然较低,但是经过 MAP 方法集成之后,能够弥补在用不同术语描述同一缺陷的情况下,经典的相关性信息检索方法 BM25F 不太适用的缺点。

综上可得:与单一方法相比,集成方法即流行文本相关性算法 BM25F 与潜语义索引 LSI 算法集成能表现出较优的检测性能。

(2)RQ2 分析

这里主要验证相似性融合方法线性结合即区别对待分类信息相似性与文本信息相似性后,检测方法性能是否优于将分类信息与文本信息作为整体考虑的检测性能。为了方便,图中 One-Hot 简称为 HOT。图 5 给出了针对 RQ2 的实验结果,表 12 列出了针对 RQ2 的指标提升情况。

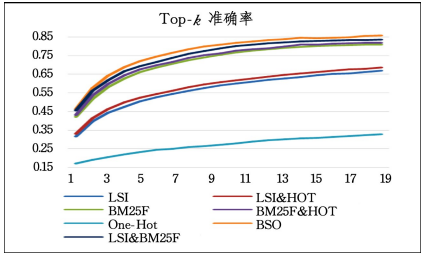


图 5 实验结果展示(3)(电子版为彩色)

Fig. 5 Display of experimental results(3)

表 12 针对 RQ2 指标的提升情况

Table 12 Index promotion for RQ2

(单位: %)

方法	LSI 为基础的 指标提升	BM25F 为基础的 指标提升	LSI&BM25F 为 基础指标提升
Top1—5	4.3	2.8	3.2
Top6—10	3.4	1.6	3.3
Top11—15	2.9	1.1	2.1
Top16—20	2.8	1.2	2.0
平均提高	3.3	1.7	2.6

从图 5 以及表 12 中可以看出:1)One-Hot 方法检测准确率非常低,这是因为该方法只运用了分类信息,而分类信息携带的信息量远远少于文本信息。2)相似性融合线性集成的方法与 LSI,BM25F,LSI&BM25F 方法相比准确率有了一定提高,以 LSI 为基础的指标提升较其他更明显。其原因可能是信息检索方法 BM25F 的相似性打分排名比 LSI 方法相似性打分稳定,因此分类信息的影响较 LSI 影响小。3)LSI&BM25F 方法已取得了较高的准确率,但是在集成 One-Hot 方法后,依然有 2.6% 的平均指标提高。即使 One-Hot 方法的检测率极低,但是 One-Hot 方法的参与对 LSI&BM25F 方法的相似性打分进行一定程度的重新排名。因此,该方法的平均提高介于其他两者之间。

综上可得:相似性融合方法线性结合即区别对待分类信息相似性与文本信息相似性后,检测性能优于将分类信息与文本信息作为整体考虑的检测性能。

(3)RQ3 分析

这里主要验证与基线方法以及较新较优方法相比,BSO 方法是否能表现出较优的检测性能。其中基线方法分别为 DBTM,REP,BM25F,LDA 和 LSI,其中 DBTM,REP 是当前用于重复缺陷报告检测性能较好的方法。图 6 给出了针对 RQ3 的实验结果,表 13 列出了针对 RQ3 的指标提升情况。

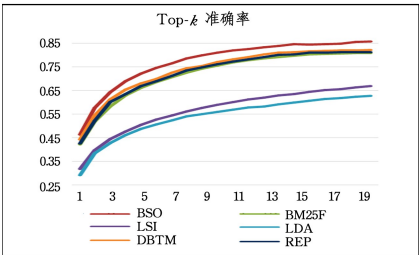


图 6 实验结果展示(4)(电子版为彩色)

Fig. 6 Display of experimental results(4)

表 13 针对 RQ3 指标的提升情况
Table 13 Index promotion for RQ3

方法	(单位:%)				
	LSI 为基础的 指标提升	LDA 为基础的 指标提升	BM25F 为基础的 指标提升	DBTM 为基础的 指标提升	REP 为基础的 指标提升
Top1-5	45	51.5	10	5.4	8.5
Top6-10	40	45.8	8	5.7	7.1
Top11-15	34	42.6	6	4	5.4
Top16-20	30	37.6	5.5	3.8	4.8
平均提高	37	44	7.3	4.7	6.3

从图 6 和表 13 可以看出:1)6 种方法的准确率都随 Top 数的增长而增长,Top 数为 12 之后,增长缓慢。2)DBTM 和 REP 的准确率略高于 BM25F,但差距较小。其原因是 DBTM 和 REP 方法是在 BM25F 方法上的改进,但是 DBTM 性能较 REP 更好。LSI 与 LDA 准确率的差距较小,两种方法都能够挖掘潜在语义信息。LSI 准确率高于 LDA,验证了本文相关部分指出的 LSI 在文本相关性打分上优于 LDA 的结论。3)信息检索方法的检测性能优于 LDA、LSI 方法的检测性能。4)在 Top1 到 Top20 的范围内,本文提出的 BSO 方法都有着最高的检测准确率,在主题模型方法上的提升最为明显,相比当前的主流方法 DBTM,平均指标提高了 4.7%;相比 REP 方法,平均指标提高了 6.3%。

综上可以得出结论:与基线方法以及较新较优方法相比,BSO 方法能表现出较优的检测性能。

5 存在的局限和不足

本文提出的 BSO 方法还存在一些局限性和不足,具体包括以下方面:1)本文提出的 BSO 重复缺陷报告的检测方法只用于用英语书写的缺陷报告的检测,对于其他语言不适用。2)参数的设置问题,参数设置的好坏与模型性能优劣有很大关系。本文算法的参数都为默认值,通过调参可能会进一步提高重复缺陷报告检测的性能。但是默认的参数能够提供相同的实验环境,更利于对对照实验的分析。3)本文用了 One-Hot 编码方法进行向量生成,虽然能够比较公平地比较向量之间的差异,但是属性取值过多且实验数据集较大时,将占用较多的空间资源以及耗费较长的时间。4)本文方法只用了一个数据集,这对验证方法的一般性有一定的局限性,未来可以尝试在更多数据集上进行实验。

结束语 当一个错误报告被修复时,将重复的对象独立地解决会耗费大量时间。当考虑到流行软件的日常报告缺陷的数量时,手动分诊需要大量时间,实际中无法完成。重复缺陷报告检测方法能够有效地提高工作者的工作效率。目前,

对重复缺陷报告的检测方法,已经从最初的向量空间模型方法发展到了 REP 和 DBTM 方法,检测效果得到了巨大的提高。本文提出的 BSO 方法在数据集上进行了 OpenOffice 实验,回答了 3 个问题。其在 DBTM 方法的基础上有着平均 4.7%的提高,在 REP 方法的基础上有着平均 6.3%的提高,从而证明了该方法的有效性。

在未来的研究中,我们将进一步从 3 个方面进行探究:1)探究参数设置问题;2)探究分类信息的向量生成问题,尝试用其他构建空间向量的方法;3)本文只在一个数据集上进行了验证,未来将尝试在更多数据集上进行实验。

参 考 文 献

[1] RUNESON P,ALEXANDERSSON M,NYHOLM O. Detection of Duplicate Defect Reports Using Natural Language Processing [C]// International Conference on Software Engineering. IEEE Computer Society,2007:499-510.

[2] BETTENBURG N,PREMJAR R,ZIMMERMANN T,et al. Extracting structural information from bug reports[C]// International Working Conference on Mining Software Repositories. ACM,2008:27-30.

[3] XIA X,LO D,SHIHAB E,et al. Automated Bug Report Field Reassignment and Refinement Prediction[J]. IEEE Transactions on Reliability,2016,65(3):1094-1113.

[4] HUANG X L. Research on Automatic Distribution of Software Defects[D]. Shanghai:Fudan University,2011.

[5] JALBERT N,WEIMER W. Automated duplicate detection for bug tracking systems[C]// IEEE International Conference on Dependable Systems and Networks with Ftics and DCC. IEEE, 2008:52-61.

[6] CUBRANIC D. Automatic bug triage using text categorization [C]// International Conference on Software Engineering & Knowledge Engineering. USA:KSI Press,2004:92-97.

[7] WANG X,ZHANG L,XIE T,et al. An approach to detecting

- duplicate bug reports using natural language and execution information[C]//ACM/IEEE International Conference on Software Engineering. IEEE,2008;461-470.
- [8] ALIPOUR A,HINDLE A,STROULIA E. A contextual approach towards more accurate duplicate bug report detection [C]//Mining Software Repositories. IEEE,2013;183-192.
- [9] ROBERTSON S,ZARAGOZA H,TAYLOR M. Simple BM25 extension to multiple weighted fields[C]//Thirteenth ACM International Conference on Information and Knowledge Management,2004;42-49.
- [10] CHENG Z Y,HUNG H D,W S S,et al. Duplication Detection for Software Bug Reports Based on BM25 Term Weighting[C]//2012 Conference on Technologies and Applications of Artificial Intelligence (TAAD). 2012.
- [11] PÉREZ-AGÜERA J R,GREENBERG A J,et al. Using BM25F for semantic search[C]//Proceeding of the 3rd International Semantic Search Workshop. 2010;1-8.
- [12] SUN C,LO D,KHOO S C,et al. Towards more accurate retrieval of duplicate bug reports[C]//IEEE/ACM International Conference on Automated Software Engineering. IEEE,2011;253-262.
- [13] LAZAR A,RITCHEY S,SHARIF B. Improving the accuracy of duplicate bug report detection using textual similarity measures [C]//Working Conference on Mining Software Repositories. ACM,2014;308-311.
- [14] PODGURSKI A,LEON D,FRANCIS P,et al. Automated Support for Classifying Software Failure Reports[C]//International Conference on Software Engineering. IEEE,2003;465-475.
- [15] SUN C N,LO D,WANG X Y,et al. A discriminative model approach for accurate duplicate bug report retrieval[C]//2010 ACM/IEEE 32nd International Conference on Software Engineering. 2010.
- [16] SANGUANSAT P. Paragraph2Vec-based sentiment analysis on social media for business in Thailand[C]//International Conference on Knowledge and Smart Technology. IEEE,2016;175-178.
- [17] WANG B. Research on detection method of duplicate defect report[D]. Shanghai:East China Normal University,2016.
- [18] SOMASUNDARAM K,MURPHY G C. Automatic categorization of bug reports using latent Dirichlet allocation[C]//India Software Engineering Conference. ACM,2012;125-130.
- [19] NGUYEN A T. Duplicate bug report detection with a combination of information retrieval and topic modeling[C]//Proceedings of International Conference on Automated Software Engineering. IEEE,2012;70-79.
- [20] REN Y G,YANG R J,YIN M F. Text feature selection algorithm based on the correlation between feature weights and words[J]. Computer Application and Software,2012,29(9):33-36.
- [21] BLEI D M,NG A Y,JORDAN M I. Latent dirichlet allocation [J]. JMLR,2003,3(1):993-1022.
- [22] BELLEGARDA J R. Exploiting latent semantic information in statistical language modeling[J]. Proceedings of the IEEE,2000,88(8):1279-1296.
- [23] HE L,WU J,FANG D T,et al. Speaker Adaptation Method Based on Maximum Posterior Estimation and Nearest Neighbor Linear Regression[J]. Acta Electronic Sinica,2000,28(11):55-58.
- [24] BETTENBURG N,PREMRAJ R,ZIMMERMANN T. Duplicate bug reports considered harmful. Really? [C]//Proc. International Conference on Software Maintenance 2008. 2008;337-345.
- [25] HUANG X L,YU Z S,GUAN J H. Software Defect Assignment Method Based on LDA Topic Model [J]. Computer Engineering,2011,37(21):46-48.
- [26] TIAN Y,LO D,XIA X,et al. Automated prediction of bug report priority using multi-factor analysis[J]. Empirical Software Engineering,2015,20(5):1354-1383.
- [27] ALENEZI M,BANITAN S,ZAROOUR M. Using Categorical Features in Mining Bug Tracking Systems to Assign Bug Reports[J]. International Journal of Software Engineering & Applications,2018,9(2):29-39.
- [28] KARIM M R,IHARA A,XIN Y,et al. Understanding Key Features of High-Impact Bug Reports[C]//2017 8th International Workshop on Empirical Software Engineering in Practice (IWESEP). IEEE,2017;53-58.
- [29] RAKHA M S,BEZEMER C P,HASSAN A E. Revisiting the Performance Evaluation of Automated Approaches for the Retrieval of Duplicate Issue Reports[J]. IEEE Transactions on Software Engineering,2017,PP(99):1-1.
- [30] CHREN W A. One-hot residue coding for low delay-power product CMOS design[J]. IEEE Transactions on Circuits & Systems II Analog & Digital Signal Processing,1998,45(3):303-313.
- [31] SHI Z,KEUNG J,SONG Q. An empirical study of BM25 and BM25F based feature location techniques[C]//International Workshop on Innovative Software Development Methodologies and Practices. 2014:106-114.
- [32] DEERWESTER S,DUMAIS S T,FURNAS G W,et al. Indexing by latent semantic analysis[J]. Journal of the Association for Information Science & Technology,2010,41(6):391-407.
- [33] BELLEGARDA J R. Exploiting latent semantic information in statistical language modeling[J]. Proceedings of the IEEE,2000,88(8):1279-1296.