

中文非功能需求描述的识别与分类方法研究^{*}

贾一获, 刘 璘

(清华大学 软件学院, 北京 100084)

通讯作者: 刘璘, E-mail: linliu@tsinghua.edu.cn



摘 要: 非功能需求描述系统质量相关的属性,是软件设计决策的重要依据和评估标准.与功能需求的描述相比,非功能需求描述通常比较分散,且往往是隐含的.当采集到的需求原始描述内容较多时,逐一进行人工判别和整理需要耗费大量的时间和精力.针对中文自然语言描述的需求文本,提出了自动化的非功能需求识别和分类方法.其中,识别过程旨在从文本中提取出可能包含非功能需求的语句,尽可能地缩小人工检查的范围.分类过程则进一步将包含非功能需求的语句划分到性能、可靠性、可用性、安全性、可维护性这五大类非功能需求类型之中.分类训练和实验测试基于课题组工作项目整理的实验数据集进行,最终,基于一个工业界的实际应用案例数据,验证了方法的有效性.

关键词: 需求工程;非功能需求;中文文本分类;需求识别;特征词提取

中图法分类号: TP311

中文引用格式: 贾一获,刘璘.中文非功能需求描述的识别与分类方法研究.软件学报,2019,30(10):3115-3126. <http://www.jos.org.cn/1000-9825/5791.htm>

英文引用格式: Jia YD, Liu L. Recognition and classification of non-functional requirements in Chinese. Ruan Jian Xue Bao/ Journal of Software, 2019, 30(10): 3115-3126 (in Chinese). <http://www.jos.org.cn/1000-9825/5791.htm>

Recognition and Classification of Non-functional Requirements in Chinese

JIA Yi-Di, LIU Lin

(School of Software, Tsinghua University, Beijing 100084, China)

Abstract: Non-functional requirements state quality related expectations to a system, is used as important criteria for software design decision making and design evaluation. Comparing to functional requirements descriptions, non-functional requirements statements are often scattered and implicit, which may require much effort to group and interpret manually. This study proposes an automated non-functional requirements recognition and classification approach, which identifies the sentences that are likely to include information about non-functional requirements, to save manual efforts. Then, given non-functional requirements statement is classified into five possible non-functional requirements types, including: performance, reliability, usability, security, and maintainability. The training of classification model and experiment are based on a requirements dataset from the past projects, and then the proposed approach is validated with an industrial project requirements data.

Key words: requirement engineering; non-functional requirement; Chinese natural language processing; requirement recognition; feature words extraction

在软件开发过程中,软件需求分析工作占有重要地位.软件需求工程的一个主要任务就是将系统的功能需

^{*} 基金项目: 国家自然科学基金(61432020)

Foundation item: National Natural Science Foundation of China (61432020)

本文由“面向 DevOps 的软件工程新技术”专题特约编辑荣国平、白晓颖、岳涛推荐.

收稿时间: 2018-08-31; 修改时间: 2018-10-31; 采用时间: 2018-12-14; jos 在线出版时间: 2019-05-22

CNKI 网络优先出版: 2019-05-22 15:25:49, <http://kns.cnki.net/kcms/detail/11.2560.TP.20190522.1525.001.html>

求和非功能需求尽可能完整、准确地表示出来.与描述系统能力的功能需求相比,用于约束系统质量属性的非功能需求更容易被忽视,却也造成了许多项目的最终失败^[1].Bajpai 等人^[2]更是在实际调研后,得出了如下结论:相较于具备明确非功能需求描述的项目,缺少非功能需求分析的项目会在失败率上高出 60%.李凤麟等人^[3]提出了一种从质量属性的角度分析非功能需求的方法.Boehm 等人^[4]的研究则指出,质量需求之间同样存在冲突,需要进行权衡和优化取舍.由此可见,在软件开发的各个阶段关注非功能需求,尤其是在软件开发的前期明确相应的质量属性,会对项目的最终结果产生积极的影响.

相较于被集中撰写的功能需求,软件的非功能需求描述通常分散甚至隐含在不同语句、段落、文档之中,因此极容易被忽视.而面对复杂的系统和巨大的文本量,人工分析、提取非功能需求意味着大量的时间成本,且难以避免疏漏之处.所以,将上述过程在一定程度上自动化,即通过软件辅助完成非功能需求的识别、分类工作,对保障需求分析过程的效率和质量具有重要意义.

非功能需求相关的概念本体和分类模型研究在国内外有一定的基础,但在软件工程实践中,用户在表达非功能需求时没有明确的规范^[5].本文重点围绕性能、可靠性、可用性、安全性、可维护性这 5 类典型、常见的非功能需求展开研究.在识别非功能需求描述的基础上,将其划分到最为近似的分类.具体来说,本文的主要贡献包括:

(1) 针对性能、可靠性、可用性、安全性、可维护性这 5 类非功能需求,整理了其通用描述词汇,并针对中文需求文本,整理了功能需求和非功能需求数据集.

(2) 针对中文需求文本,提出了一种自动化的非功能需求识别方法,并对其效果进行了实际测试.

(3) 针对中文非功能需求描述,提出了一种自动化的分类方法,并对其效果进行了实际测试.

值得一提的是,上述方法主要适用于功能需求和非功能需求界限不清晰的中文需求描述,如非结构化的需求文档、用户评论、专家意见等.而结构化很强的需求文本,通常已经区分了功能需求和非功能需求,甚至区分了非功能需求的种类.在这样的情况下,一般可以直接依据需求模板的格式提取相应内容,无需采用本文的方法.

此外,本文提出的识别方法主要针对非结构化的中文需求文本.例如,可以将需求文档作为输入,然后采用本文的识别方法标识出其中可能包含非功能需求的语句,从而缩小人工检查的范围.而本文提出的分类方法则用于进一步确定非功能需求的分类.例如,针对人工确认包含非功能需求的语句,或是已区分功能需求和非功能需求的结构化文档,可以使用此方法完成非功能需求语句的类别划分.本文将识别和分类的过程分开完成,一方面,可以使上述方法在不同的应用场景下更为灵活;另一方面,如果在识别和分类过程之间加入人工确认的步骤,则能够以较低的人力成本有效地提升分类效果,具体的讨论和相关实验结果将在后文中进行阐述.

本文第 1 节概述国内外现有的研究成果.第 2 节对实验数据集进行描述.第 3 节介绍中文文本中的非功能需求识别方法.第 4 节阐释非功能需求分类的具体过程.第 5 节总结全文,并对后续可进行的研究工作进行初步探讨.

1 研究背景

为了高质量、高效地完成需求分析阶段的各项工作,自动化的需求文本分析工具曾一度是需求工程领域的研究热点.而随着软件系统的不断复杂化,非功能需求的重要性也逐渐显现出来.

针对英文需求文本,已有不少研究围绕非功能需求的提取和分类展开.Cleland-Huang 等人^[6]提出了一种基于特征词的非功能需求识别方法,从人工标注的训练数据中提取各类非功能需求的特征词,并根据特征词的出现频率对文本进行分类.在面对新的领域文本时,上述方法可以达到 79.9%的召回率,能够有效缩小人工识别的范围.但是,其精确率只有 20.7%,并且需要人工标注部分数据才能完成迭代训练,因此仍然需要较多的人工参与.为了降低人力成本,Casamayor 等人^[7]提出了一种半监督学习方法,并且达到了 70%的准确率.Hindle 等人^[8]采用了不同的思路,直接基于话题模型获得词汇列表,进而将其对应到各类非功能需求.近年来,Sumner 等人^[9]使用基于遗传算法的神经网络,实现了非功能需求和功能需求的分类.Ramnani 等人^[10]基于文本模式,实现了非功能需

求识别.基于模糊规则的问题本方法也在非功能需求分类中得到了有效应用^[11].

整体来看,针对英文需求文本自动检测的研究已颇有进展.但是,由于语言之间的巨大差异性,这些研究成果无法直接对中文需求文本提供支持.中文词语间无空格分隔、歧义现象普遍、实词运用灵活等特点,更是形成了其特有的复杂性和难度.刘婷婷等人^[12]以情感分析和主题分类两种任务为背景,基于 IMDB、RCV1 这两个英文数据集和豆瓣影评、搜狐新闻这两个中文数据集,分别应用传统机器学习方法和深度学习方法进行实验.使用传统的机器学习方法进行情感分类,英文的分类准确率达到 90.70%,而中文的分类准确率仅达 84.00%.引入深度学习方法后,虽然部分实验结果表明中文可能获得更高的准确率,但是英文分类的整体效果依旧好于中文.考虑到本文所使用的训练数据量难以满足深度学习模型所需,我们将重点关注传统的机器学习方法.虽然对软件非功能需求代价权衡方法^[13]和对非功能性元需求集^[14]的研究已经体现出国内对非功能需求的逐渐重视,但是与国外相比,针对中文非功能需求的自动化分析方法,研究程度尚显不足.倪瑜泽等人^[15]提出了一种基于用户评论的演化需求发现方法,达到了 93.32% 的平均精确率和 78.42% 的平均召回率.由于我们基本上没有找到对中文非功能需求进行自动分类的相关工作,因此我们的实验结果将主要对比上述国外相关工作和国内的中文文本分类工作.考虑到我们的训练数据量以及前文提及的中文文本处理上的难度,我们认为,近似或略低于上述实验效果的结果均是可接受的.

本文的研究聚焦于两个方面.第一,识别中文需求文本中的非功能需求.前文已经提到,针对非功能需求的描述往往较为分散,且很有可能隐含于系统的功能描述之中.例如,“系统在用户登录过程中,验证其身份和权限”这句话,就隐含了安全性这一非功能需求.诸如此类识别工作如果均由人工完成,则需要大量的时间成本.因此,本文将首先探讨非功能需求的自动化识别方法,并在这一阶段重点考虑非功能需求的召回率,以期缩小人工检查的范围.第二,针对包含非功能需求的语句,确定相应非功能需求的具体分类,即判定语句是否包含性能、可靠性、可用性、安全性、可维护性这 5 类非功能需求中的某一类或某几类,并通过准确率对分类结果进行评价.

2 实验数据集

本文所使用的实验数据可大致分为 3 个部分.

第 1 部分是需求文本数据,主要来源于清华大学软件需求工程课程的学生作业,也有部分来源于互联网上的博客和文档.我们对上述数据进行了筛选和整理,数据所属的系统包括信息管理系统、网络通信软件、车载导航软件、在线教育软件、游戏软件、微信小程序、手机应用软件等多种类型,尽可能地保证文本覆盖较多的项目领域.上述系统的规模相对较小,复杂度相对较低,但其需求描述较为全面.我们将包含非功能需求或纯粹描述某一功能模块的几句或几段话作为用于实验的 1 条数据.最终,我们获得训练数据 400 条,其中,功能需求描述和非功能需求各占 200 条,不同类别非功能需求的数据量见表 1.考虑到识别非功能需求可以将各类别数据整合使用,本质上属于二分类问题;而非功能需求分类则基于初始特征词和语句中的关键词.因此,这一数据量基本上能够满足实验和测试要求.

Table 1 Amount of data for various NFRs

表 1 各类非功能需求的数据量

非功能需求 数据量(条)	性能	可靠性	可用性	安全性	可维护性
	65	30	51	31	23

第 2 部分是用于非功能需求分类的初始特征词.本文以需求工程领域对非功能需求的常用描述词汇为基础,实现基于文本相似度的分类方法.上述描述词汇主要整理、提取自相关研究文献^[16-18].同时,考虑到同一英文词汇可能对应多种中文表达,因此,我们适当进行了同义词扩充,其结果见表 2.

第 3 部分是在实际测试中使用的数据.这一部分的测试数据及结果来自工业界实际项目,项目规模大,复杂度较高,需求描述也比较规范和准确.

Table 2 NFRs and descriptive words

表 2 非功能需求及其描述词汇

非功能需求	描述词汇
性能	性能,高性能,时间,响应,反应,响应时间,反应时间,延迟时间,带宽,空间,容量,等待,等待时间,吞吐,吞吐量,吞吐率,计算,计算能力,执行,速度,执行速度,快速,延迟,延时,传输,传输延迟,通过,通过延迟,负载,工作负载,工作量,工作负荷,负荷,资源,利用,占用,运用,使用率,利用率,功耗,加速,内存,存储,准确度,准确,精度,精准,准确性,准确率,效率,高效,推迟,秒,s,小时,h,分钟,min,次,并发,同时,同步,及时,准时
可靠性	可靠性,可靠,稳定,稳定性,完整,一致,稠度,相容,相合,有效,整体,健全,正确性,正确,检查,成熟,完备,容错性,容错,误差,恢复,复原,合规,故障,故障率,失效,失误,失效率,误失,误失率,致命,崩溃,关键,关键失效,关键错误,错误,严重程度,预测,缺失,缺失率,丢失,损失,遗失,丢失率,损失率,遗失率,异常,报错
可用性	可用性,可用,使用,容易,简单,学习,易学,易用,可理解,理解,易理解,易懂,操作,可操作,可操作性,吸引力,吸引,好用,产出,生产,产出率,生产率,生产力,有用,有效,有益,实用,体验,界面,视觉,整洁,混乱,感受,美观,方便,简便,简易,友好,人性化,偏好,喜好,喜欢,喜爱,记忆,可记忆性,记忆性
安全性	安全性,安全,机密,保密,密码,加密,解密,密钥,访问,控制,访问控制,存取,存取控制,权限,身份,验证,身份验证,认证,身份认证,鉴别,认定,防护,入侵,识别,病毒,恶意,后门,防火墙,口令,锁定,攻击,抵抗,抵御,泄漏,非法,合法,隔离,可信,信任,保护,保障,指纹,窃取,限制
可维护性	可维护性,维护,测试,易测,检测,可测,修改,变更,分析,易变,可变,变化,内聚,耦合,内聚性,耦合性,模块,移植,可移植,可移植性,迁移,复用,重用,质量,代码,文档,可读性,设计模式,重构,解藕,扩展,可扩展性,软件缺陷,架构

3 基于中文文本的非功能需求识别方法

中文需求文本中,非功能需求描述可能分散、隐含在各个语句中,但也有部分语句与非功能需求无关.例如,“用户可以对文本进行查看,增加,删除,编辑等操作.”就是纯粹的功能描述.因此,需要首先识别出与非功能需求相关的语句,即对语句是否包含非功能需求进行判断,本质上是一个二分类问题.本节将重点介绍中文非功能需求识别的具体流程,并对实现效果和测试结果进行总结和说明.为了尽可能地缩小人工检查的范围,这一部分工作将重点关注非功能需求的召回率.

3.1 非功能需求识别的流程

中文非功能需求识别包含分词和提取关键词、选择文本特征、训练分类模型、测试文本分类这 4 个主要步骤,其流程如图 1 所示.接下来,我们将对每一步骤的实现方法进行具体介绍.

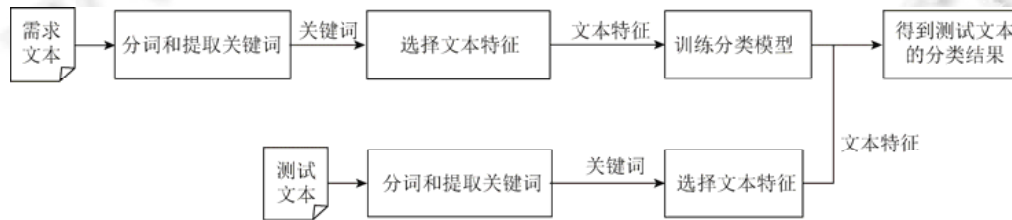


Fig.1 Chinese non-functional requirements identification process

图 1 中文非功能需求识别流程

(1) 分词和提取关键词

本文采用的非功能需求识别方法基于文本关键词,因此,首先需要对需求文本进行分词和词性标注处理,并提取其中的关键词. ICTCLAS^[19]、HanLP^[20]等近年来广泛应用的中文文本分析工具均提供了上述功能,因此,本文将直接使用 HanLP 工具完成上述工作.

(2) 选择文本特征

在获得需求文本中的关键词后,需要确定文本特征的表达方式.常用的文本特征模型包括 BoW(bag-of-words)和 TF-IDF 模型.BoW 模型统计文档中各词汇的出现次数,并使用无序词集合表示整个文档.TF-IDF 模型的主要思路是:如果一个词语在某一文档中出现次数较多,且出现该词语的文档数量较少,则该词语对文档的区分能力较强.我们分别尝试使用上述两种特征进行后续实验,并对其结果进行了比较,具体实验结果将在第 3.2

节给出.

(3) 训练分类模型

将文本以特征形式表示后,即可基于此训练分类模型.我们分别尝试了逻辑斯蒂回归(LR)、朴素贝叶斯(NB)、NBSVM(Naïve Bayes-support vector machine)^[21]等分类模型,并在参数调整后选取了最优模型用于后续的测试文本分类,具体的实验结果同样在第 3.2 节给出.

(4) 测试文本分类

使用步骤(1)中的分词和关键词提取方法及步骤(2)中的文本特征选择方法,将测试文本以特征的形式进行表述.之后,将特征作为分类模型的输入,模型会给出“包含非功能需求”和“不包含非功能需求”两类标签.

3.2 实验结果及总结

我们组合使用不同的文本特征和分类模型,实现非功能需求识别,并采用五折交叉验证的方法进行测试,即将全部数据平均划分为 5 份,每次取其中的 4 份用于训练,1 份用于测试.实验和测试均采用第 2 节中所描述的数据集.5 次实验的精确率、召回率及其均值见表 3.

Table 3 Results of non-functional requirements identification
表 3 非功能需求识别的实验结果

特征选择方法及分类模型	实验结果						平均值
	实验 1	实验 2	实验 3	实验 4	实验 5		
BoW+LR	Precision (%)	81.40	83.33	66.10	73.47	71.15	75.09
	Recall (%)	87.50	87.50	97.50	90.00	92.50	91.00
BoW+NB	Precision (%)	84.21	80.56	81.82	74.42	85.00	81.20
	Recall (%)	80.00	72.50	90.00	80.00	85.00	81.50
BoW+NBSVM	Precision (%)	91.43	80.00	80.00	80.49	78.05	81.99
	Recall (%)	80.00	80.00	90.00	82.50	80.00	82.50
TF-IDF+LR	Precision (%)	76.09	82.93	77.55	76.60	70.91	76.82
	Recall (%)	87.50	85.00	95.00	90.00	97.50	91.00
TF-IDF+NB	Precision (%)	68.09	82.86	86.84	71.74	72.34	76.37
	Recall (%)	80.00	72.50	82.50	82.50	85.00	80.50
TF-IDF+NBSVM	Precision (%)	68.89	85.00	89.74	77.27	70.21	78.22
	Recall (%)	77.50	85.00	87.50	85.00	82.50	83.50

为了对实验结果形成直观认识,将不同模型的平均精确率和平均召回率以簇状柱形图的方式加以呈现,其结果如图 2 所示.可以看到,结合使用 BoW 和 NBSVM 模型取得了最高的精确率,使用 BoW 或 TF-IDF 表示文本特征,使用 LR 进行模型训练均得到了 91.00%的召回率.

由于我们的实验文本已经覆盖了较多的项目领域,并且划分数据的过程中采用了随机抽取的方法,因此,实验结果具有一定的普适性.而表 3 中列出的单次实验结果,也能够一定程度上帮助我们判断模型在不同数据集上的稳定性.

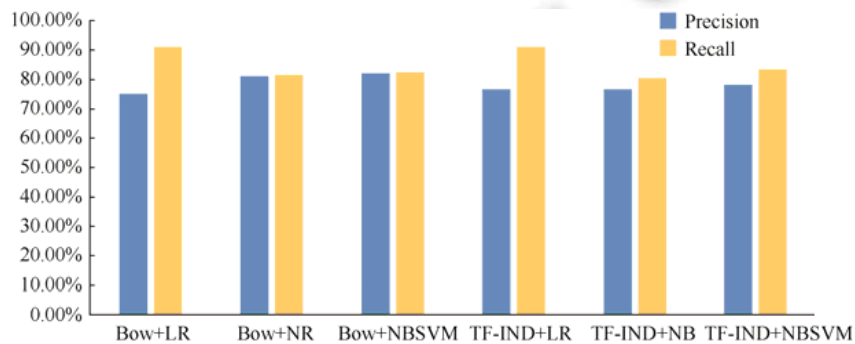


Fig.2 Average precision and average recall rate for different models

图 2 不同模型的平均精确率和平均召回率

这一部分工作的主要目的是提取可能包含非功能需求的语句,从而缩小人工检查的范围,所以我们将重点关注非功能需求的召回率,并在此基础上尽可能地保证精确率以及模型效果的稳定性.因此,最终选用 TF-IDF 表示文本特征,并使用逻辑斯蒂回归实现分类.我们使用上述模型,在普通配置(Intel(R) Core(TM) i5-4200U CPU @ 1.60GHz,8.00GB RAM)的笔记本电脑上进行实验,针对包含 5 万字左右的需求文本,可以在 20s 内完成非功能需求的识别和导出,其时间成本远低于人工检查所需时间,可以满足实用要求.

最后,由领域专家对上述分类方法进行生产环境中的测试.测试数据来源于实际项目,项目规模大,复杂度较高,需求描述也比较规范和准确.使用的训练数据共计 200 条,其中非功能需求 100 条,功能需求 100 条;测试数据共计 200 条.非功能需求识别的召回率达到 88%,精确率约为 73%.因此,上述方法的召回率基本上满足实用要求,可以有效缩小人工检查的范围.

整体来看,针对不同系统的需求描述,上述模型均取得了较好的测试结果,比较适用于对召回率要求较高的应用场景.例如,可将需求文档作为输入,然后采用本文的识别方法标识或导出可能包含非功能需求的语句,从而缩小人工检查的范围.在其他应用场景下,若需要重点关注精确率,则应选取其他特征表示方法或分类模型,本文的实验结果也可以提供一定的参考.

4 基于中文文本的非功能需求分类方法

识别出包含非功能需求的文本后,可将其进一步对应用到性能、可靠性、可用性、安全性、可维护性这 5 类非功能需求中的某一类或某几类.值得一提的是,目前非功能需求的分类并不存在统一的标准^[22],因此,在和领域专家交流后,我们最终选取了上述典型、常见的非功能需求.另外,safety 与 security 虽然均会被翻译为安全性,但是两者的实际侧重有所差别.前者主要关注人身安全,而后者则强调信息安全.由于软件涉及的安全性需求主要围绕后者,而对前者的涉及较少,因此,我们参考 ISO/IEC 25010^[23]中的软件质量属性分类,不对 safety 和 security 进行区分.本节将重点介绍实现中文非功能需求分类的具体流程,并对实现效果和测试结果进行总结和说明.

4.1 非功能需求分类的流程

中文非功能需求分类包含分词和提取关键词、划分关键词类别、扩充特征词、计算特征词权重、文本分类这 5 个主要步骤,其流程如图 3 所示.接下来,我们将对每一步骤的实现方法进行具体介绍.

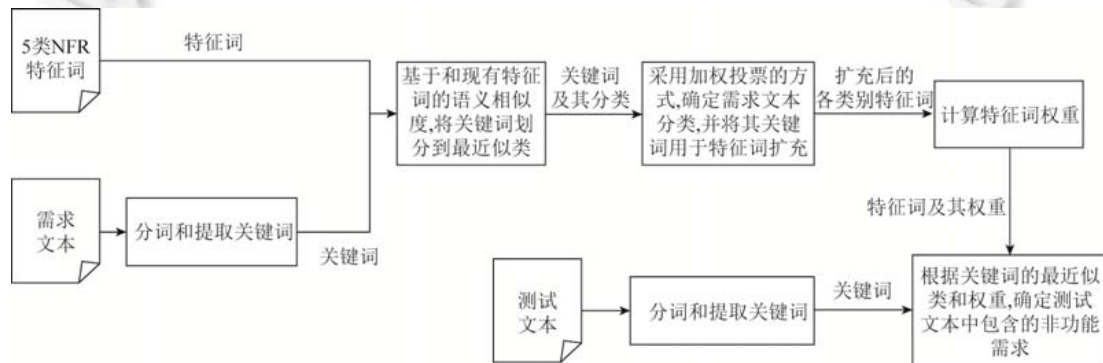


Fig.3 Chinese non-functional requirements classification process

图3 中文非功能需求分类流程

(1) 分词和提取关键词

本文采用的非功能需求分类方法基于特征词和文本关键词,因此,首先需要对需求文本进行分词和词性标注处理,并提取其中的关键词.这一部分工作同样使用 HanLP 工具来完成.

(2) 划分关键词类别

在获得需求文本中的关键词后,可以计算出每一个关键词与现有特征词的语义相似度.具体来讲,词语间语

义相似度的计算需要从《哈工大信息检索研究室同义词词林扩展版》中获得词语编码,之后即可采用田久乐等人^[24]提出的计算方法,依照公式求得两个词语间的语义相似度.最后,各个关键词分别被划分到与其语义相似度最高的特征词所属的非功能需求类别.

(3) 扩充特征词

由于词语相似度的计算没有考虑到上下文的影响,因此,同一语句中的关键词很可能属于不同的非功能需求类别.例如,从“高峰时段快速响应用户请求.”这一语句中,可以提取出“响应”“快速”“高峰”“用户”等多个关键词.接下来,使用步骤(2)中的方法计算语义相似度,并将关键词划分到其最近似特征词所在的分类.前两个关键词会被划分为性能;后两个词语的最相似特征词分别为“指纹”和“身份”,因此均被划分至安全性.

显然,上述语句描述的是性能这一非功能需求.“用户”这一名词本质上可以在各类非功能需求描述中使用,因此不应该作为特征词出现在任何类别中;而根据经验,“高峰”倾向于出现在描述性能的语句中,因此应该作为性能的扩充特征词.这样一来,对上述两个词语的期望处理方式产生了分歧,且其中应用了大量的人工经验,所以计算机很难对此进行分辨.但是,考虑到非特征关键词可能出现在任何类别的非功能需求中,因此在不断增加训练数据的情况下,该关键词会被加入到各个分类,所以可以考虑在后续步骤中降低其权重,而在本步骤中将所有关键词作为扩充特征词.

接下来,需要确定语句中关键词的最近似分类.为了尽可能地在各分类特征词中引入上下文信息,将一个语句中的全部关键词作为一个整体,确定其最近似分类.“响应”和“快速”均包含在初始特征词中.而根据语义相似度的计算结果来看,“指纹”和“身份”与其最近似词语的相似度并不是很高.所以,相较于直接采用多数投票的方式决定最终分类,对不同的关键词赋予不同的权重更为合理,其计算公式如下:

$$y = \arg \max_{v \in V} \sum_{i=1}^k w_i \delta(v, f(x_i)) \quad (1)$$

其中, y 表示分类结果, v 表示分类, V 表示全部非功能需求类别, k 表示从文本中提取的关键词数量, $\delta(v, f(x_i))$ 判定关键词 x_i 是否属于类别 v , w_i 则表示关键词权重.考虑到尚未通过大量训练完成特征词的扩充,因此,在现阶段的实验中,初始特征词的相同关键词被分配较高的权重,相似关键词的权重则随着其相似程度的下降而递减.

(4) 计算特征词权重

步骤(3)中已经提到,“用户”等非特征关键词有可能会作为扩充特征词,被加入到各个非功能需求类别中.为了降低这些词语在实际分类过程中造成的干扰,同时提高重要特征词的影响力,需要对不同特征词赋予不同的权重,其计算公式如下:

$$w_i = \log \frac{|V|}{|\{j: t_i \in v_j\}|} \quad (2)$$

其中, w_i 表示特征词权重, $|V|$ 表示非功能需求类别总数, v 表示具体的非功能需求分类, t 表示特征词.这样一来,当特征词出现在越多的分类中时,其权重就会越小,对最终分类结果的影响也就越小.

(5) 文本分类

针对待分类文本,同样先进行分词和关键词的提取.接下来,计算关键词的最近似类.如果该关键词已存在于特征词中,则使用步骤(4)中计算所得的权重;否则,使用统一的基础权重.这一步骤的计算依旧使用公式(1),只是权重的计算方式与步骤(3)有所差异.

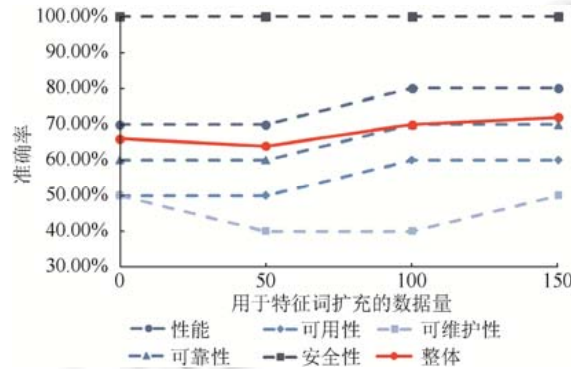
4.2 实验结果及总结

本节的实验和测试均采用第2节中所描述的数据集.首先,从每类非功能需求中随机抽取10条数据用于测试,共计50条,其余150条数据用于特征词扩充.我们分别使用不同数据量进行特征词扩充,值得一提的是,用于扩充的数据将从各类非功能需求中分别随机抽取,尽量保证各类别数据分布均匀.用于特征词扩充的数据量分别为0、50、100、150,因此,理想情况下,每次总计需要抽取数据50条,即从各类别未使用数据中随机抽取10条.但是,考虑到数据本身分布不均匀,因此,当某一类非功能需求的数据不足时,平均使用其他类的数据进行补充即可.上述实验的结果见表4.

Table 4 Results of non-functional requirements classification**表 4** 非功能需求分类的实验结果

用于特征词扩充的数据量	准确率(%)					
	性能	可靠性	可用性	安全性	可维护性	整体
0	70.00	60.00	50.00	100.00	50.00	66.00
50	70.00	60.00	50.00	100.00	40.00	64.00
100	80.00	70.00	60.00	100.00	40.00	70.00
150	80.00	70.00	60.00	100.00	50.00	72.00

为了对实验结果形成直观认识,将使用不同数据量实现特征词扩充后的准确率以折线图的方式进行呈现,其结果如图 4 所示.

**Fig.4** Accuracy for non-functional requirements classification**图 4** 非功能需求分类准确率折线图

通过上述实验可以发现,随着用于特征词扩充的数据量的增加,整体的准确率呈现出先降后升的趋势.分析造成这一现象的原因,特征词扩充的初期之所以会引入一些错误,很大程度上是由于扩充了一些非特征词汇,并且这些词汇初始可能只出现在少量类别的文本中,因此未被调整到适宜权重.例如,“用户”这一名词本质上可以出现在任何类别的非功能需求之中,因此不应作为任何一类的特征词.但是,由于其语义与“身份”最为近似,因此会被划分至安全性.在完成第 1 次扩充时,其权重为 0.510 825 6,而在完成最后一次扩充后,其权重则下降到 0.223 143 55.可见,这类非特征词的加入可能会对整体分类效果产生一定的影响,但是,随着数据量的增加,影响将会逐渐降低.

接下来,使用全量数据进行测试,并分别使用其中的 0、50、100、150、200 条数据进行特征词扩充.这一实验的目的主要分为两方面:首先,增加测试数据量,使测试结果更为准确.其次,对于测试数据,判断在逐一对文本进行分类前,先将其用于特征词扩充的可行性.因为这一做法可能会降低非特征词汇对测试结果的影响.上述实验的结果见表 5.

Table 5 Results of non-functional requirements classification (test with full dataset)**表 5** 非功能需求分类的实验结果(全量数据测试)

用于特征词扩充的数据量	准确率(%)					
	性能	可靠性	可用性	安全性	可维护性	整体
0	80.00	60.00	66.67	96.77	60.87	74.00
50	83.08	60.00	66.67	90.32	60.87	74.00
100	83.08	63.33	68.63	93.55	60.87	75.50
150	84.62	60.00	68.63	93.55	60.87	75.50
200	84.62	60.00	68.63	96.77	60.87	76.00

对于上述实验结果,使用如图 5 所示的折线图加以直观呈现.

整体来看,在文本分类前进行特征词扩充会对实验结果产生一定的积极影响.但是,具体来看,随着用于特征词扩充的数据量的增加,可靠性的准确率呈现出先升后降的趋势,可维护性的准确率则维持稳定.经过分析,

我们认为,这在一定程度上是由于数据分布不均造成的.由于性能这一分类的非功能需求数据最多,且语句中的特征较为明显,因而会造成其他类别词汇或非特征词汇被加入其中.相比之下,可靠性和可维护性的数据量较少,所以,最后两次数据扩充中这两个类别的数据占比较低,因而会影响到扩充的词汇量.

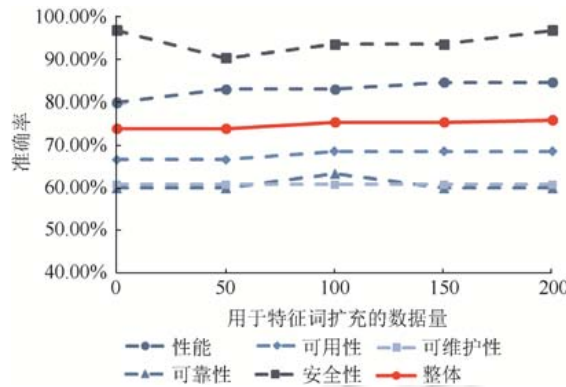


Fig.5 Accuracy for non-functional requirements classification (test with full dataset)

图5 非功能需求分类准确率折线图(全量数据测试)

上述方法主要针对包含非功能需求的语句.本文之所以将识别和分类的过程分开完成,一方面,可以使上述方法在不同的应用场景下更为灵活;另一方面,如果在识别和分类过程之间加入人工确认的步骤,则能够以较低的人力成本有效地提升分类效果.第3节的实验结果已经表明,选用 TF-IDF 表示文本特征,并使用逻辑斯蒂回归实现分类,基本能够达到接近 90%的召回率和 70%以上的精确率.而逻辑斯蒂回归本身,能够表示出分类结果的置信度.因此,在人工确认的过程中,只需重点关注置信度相对较低的语句,所以不会过多地增加人力成本.

作为对比实验,我们将上述两个过程合并进行,即省略识别过程,添加“功能需求”这一类别.考虑到功能需求本身的特征词汇难以定义,我们采用了设置阈值的方法.具体来讲,特征词扩充的过程与上文描述的方法相同,使用全量数据进行特征词扩充.但是,在对测试文本进行分类时,针对加权投票的得分设定阈值.如果投票所得分类的得分超过该阈值,则将文本分类为相应的非功能需求.反之,如果所有分类的得分均低于该阈值,则将文本划分至功能需求.使用不同阈值的实验结果见表 6.

Table 6 Results of requirements classification (contains functional requirements)

表 6 需求分类的实验结果(包含功能需求)

阈值	准确率(%)					
	性能	可靠性	可用性	安全性	可维护性	功能需求
2.00	83.08	50.00	62.75	90.32	47.83	21.00
4.00	80.00	50.00	60.78	90.32	47.83	37.50
6.00	80.00	50.00	60.78	90.32	47.83	42.50
8.00	80.00	50.00	60.78	90.32	47.83	43.00
9.00	56.92	30.00	35.29	67.74	13.04	67.50
10.00	47.69	30.00	37.25	58.06	13.04	81.00

可以看到,随着阈值的增加,文本更倾向于被分类至功能需求,因此,功能需求的识别准确率不断上升,非功能需求的识别准确率则有所下降.与之前的实验相比,当阈值为 8.00 时,非功能需求的整体准确率略有下降,功能需求的准确率则与之前的实验结果存在明显的差距.而当阈值为 9.00 和 10.00 时,功能需求的整体准确率接近于之前的实验结果,非功能需求的整体准确率却明显不及.所以,综合来看,拆分非功能需求的提取和分类过程会获得更好的效果.

最后,由领域专家对上述分类方法进行生产环境下的实际测试.测试数据共计 300 条.在不进行特征词扩充的情况下,准确率约为 70%.在进行特征词扩充的情况下,准确率约为 74%,因此,本方法针对非功能需求分类具备一定的有效性,且扩充特征词能够在一定程度上提升分类效果.

整体来看,本节所述的分类方法,比较适用于无法提供额外标注数据时,非功能需求文本的类别划分.例如,针对已区分功能需求和非功能需求的结构化文档,可以直接依照需求模板抽取其中的非功能需求,并将其作为输入,使用本节所述的方法进行需求描述词汇扩充,进而实现输入文本的分类.此外,由于本文方法的实现基于特征词权重,所以可以对需要人工自定义核心词汇的场景提供良好的支持.

相较于已有的国内外工作,本文将非功能需求的提取和分类过程分开完成.提取过程旨在定位非功能需求,从而缩小人工检查的范围.针对这个二分类问题,仅需要较少地标注数据训练模型,因而更符合工业界的实际情况.接下来,对确认包含非功能需求的语句进行分类,上述过程基于我们整理的非功能需求描述词汇来进行,并且特征词会在系统的使用过程中不断地自动扩充,不再需要额外地标注数据.

5 总结

随着软件系统的日益复杂化,在软件开发过程中尽早明确非功能需求,对于项目具有积极意义.针对英文需求描述,已有不少研究围绕其提取和分类展开.但是,由于不同语言需求表述之间的巨大差异性,这些研究成果无法直接对中文需求文本提供支持.

本文重点研究中文非功能需求描述的识别和分类方法.识别过程采用机器学习方法,从文本中提取出可能包含非功能需求的语句,尽可能地缩小人工检查的范围.分类过程则基于语义距离和相似度计算,将包含非功能需求的语句划分为性能、可靠性、可用性、安全性、可维护性这5类.本文的分类训练和实验测试基于课题组工作项目整理的实验数据集来进行,并通过工业界的实际领域应用案例,验证了方法的有效性.

我们也注意到,在实验过程中,对于同一语句,不同的中文文本分析工具可能给出不同的分词和词性标注结果.而对于部分领域术语,上述工具也无法很好地进行识别.例如,HanLP工具会将“用户体验”这一术语拆分为“用户”和“体验”两个词语,其中,前者作为名词,后者作为动词.这样无疑会对特征词的扩充和识别产生一定的影响.但是,现有的工具基本上支持用户自定义词典功能.因此,构建一个非功能需求的领域术语库,并用其辅助分词和关键词提取工作,有望优化特征词的质量,从而进一步提升非功能需求识别和分类的效果,相关工作和实验将在未来的研究中继续进行.

References:

- [1] Yin B, Jin Z. Extending the problem frames approach for capturing non-functional requirements. In: Proc. of the 11th IEEE/ACIS Int'l Conf. on Computer and Information Science. IEEE, 2012. 432–437. [doi: 10.1109/ICIS.2012.47]
- [2] Bajpai V, Gorthi RP. On non-functional requirements: A survey. In: Proc. of the 2012 IEEE Students' Conf. on Electrical, Electronics and Computer Science. IEEE, 2012. 1–4. [doi: 10.1109/SCECS.2012.6184810]
- [3] Li FL, Horkoff J, Mylopoulos J, *et al.* Non-functional requirements as qualities, with a spice of ontology. In: Proc. of the 22nd IEEE Int'l Requirements Engineering Conf. (RE). IEEE, 2014. 293–302. [doi: 10.1109/RE.2014.6912271]
- [4] Boehm B, Franch X. Conflicts and synergies among quality requirements. In: Proc. of the 2017 IEEE Int'l Conf. on Software Quality, Reliability and Security Companion (QRS-C). IEEE, 2017. 507–508. [doi: 10.1109/QRS-C.2017.87]
- [5] Mairiza D, Zowghi D, Nurmaliani N. An investigation into the notion of non-functional requirements. In: Proc. of the 2010 ACM Symp. on Applied Computing. ACM, 2010. 311–317. [doi: 10.1145/1774088.1774153]
- [6] Cleland-Huang J, Settini R, Zou X, *et al.* Automated classification of non-functional requirements. Requirements Engineering, 2007,12(2):103–120. [doi: 10.1007/s00766-007-0045-1]
- [7] Casamayor A, Godoy D, Campo M. Identification of non-functional requirements in textual specifications: A semi-supervised learning approach. Information and Software Technology, 2010,52(4):436–445. [doi: 10.1016/j.infsof.2009.10.010]
- [8] Hindle A, Ernst NA, Godfrey MW, *et al.* Automated topic naming to support cross-project analysis of software maintenance activities. In: Proc. of the 8th Working Conf. on Mining Software Repositories. ACM, 2011. 163–172. [doi: 10.1145/1985441.1985466]
- [9] Sunner D, Bajaj H. Classification of functional and non-functional requirements in agile by cluster neuro-genetic approach. Int'l Journal of Software Engineering and its Applications, 2016,10(10):129–138. [doi: 10.14257/ijseia.2016.10.10.13]

- [10] Ramnani RR, Sharma VS, Sengupta S, *et al.* Identifying and classifying non-functional requirements in text. Int Cl: G06F17/30 U.S. Patent 9, 886, 499, 2018-2-6.
- [11] Shi L, Chen C, Wang Q, *et al.* Understanding feature requests by leveraging fuzzy method and linguistic analysis. In: Proc. of the 32nd IEEE/ACM Int'l Conf. on Automated Software Engineering (ASE). IEEE, 2017. 440–450. [doi: 10.1109/ASE.2017.8115656]
- [12] Tingting L, Wendong Z, Guangyi L. Advances in deep learning based text classification. Electric Power Information and Communication Technology, 2018,16(3):1–7 (in Chinese with English abstract). [doi: 10.16543/j.2095-641x.electric.power.ict.2018.03.000]
- [13] Zhang X, Wang X, Li T, Bai C, Kang YN. Trade-off costs of software non-functional requirements. Ruan Jian Xue Bao/Journal of Software, 2017,28(5):1247–1270 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/5106.htm> [doi: 10.13328/j.cnki.jos.005106]
- [14] Liu SQ. Acquisition and analysis of non-functional meta-requirement sets based on meta-requirement model [MS. Thesis]. Changsha: Central South University, 2010 (in Chinese with English abstract). [doi: 10.7666/d.y1914241]
- [15] Yuze NI, Rong P, Dong S, *et al.* Potential evolution requirements detect method based on user comments. Journal of Wuhan University (Natural Science Edition), 2015,61(4):347–355 (in Chinese with English abstract). [doi: 10.14188/j.1671-8836.2015.04.009]
- [16] Mairiza D, Zowghi D. Constructing a catalogue of conflicts among non-functional requirements. In: Proc. of the Int'l Conf. on Evaluation of Novel Approaches to Software Engineering. Berlin, Heidelberg: Springer-Verlag, 2010. 31–44. [doi: 10.1007/978-3-642-23391-3_3]
- [17] Rao AA, Gopichand M. Four layered approach to non-functional requirements analysis. Int'l Journal of Computer Science Issues, 2011,8(6):371–379. [doi: 10.13140/RG.2.1.2991.3841]
- [18] Cleland-Huang J, Settini R, Zou X, *et al.* The detection and classification of non-functional requirements with application to early aspects. In: Proc. of the 14th IEEE Int'l Requirements Engineering Conf. IEEE, 2006. 39–48. [doi: 10.1109/RE.2006.65]
- [19] Zhang HP, Yu HK, Xiong DY, *et al.* HHMM-based Chinese lexical analyzer ICTCLAS. In: Proc. of the 2nd SIGHAN Workshop on Chinese Language Processing, Vol. 17. Association for Computational Linguistics, 2003. 184–187. [doi: 10.3115/1119250.1119280]
- [20] Shanghai Linyuan Information Technology Co., Ltd. HanLP: HanLP v1.2.8. 2017 (in Chinese). <http://hanlp.linrunsoft.com/>
- [21] Wang S, Manning CD. Baselines and bigrams: Simple, good sentiment and topic classification. In: Proc. of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers, Vol. 2. Association for Computational Linguistics, 2012. 90–94.
- [22] Luo XX, Li ZH, Zhao YJ. Overviews on software non-functional requirements at home and abroad. Application Research of Computers, 2015,32(4):972–977 (in Chinese with English abstract). [doi: 10.3969/j.issn.1001-3695.2015.04.003]
- [23] ISO/IEC. (2011) ISO/IEC 25010: Systems and Software Engineering—Systems and Software Product Quality Requirements and Evaluation (SQuaRE)—System and Software Quality Models. International Organization for Standardization, Geneva, p. 43.
- [24] Tian J, Zhao W. Words similarity algorithm based on tongyici cilin in semantic Web adaptive learning system. Journal of Jilin University, 2010,28(6):602–608 (in Chinese with English abstract). [doi: 10.3969/j.issn.1671-5896.2010.06.011]

附中文参考文献:

- [12] 刘婷婷,朱文东,刘广一.基于深度学习的文本分类研究进展.电力信息与通信技术,2018,16(3):1–7. [doi: 10.16543/j.2095-641x.electric.power.ict.2018.03.000]
- [13] 张璇,王旭,李彤,白川,康燕妮.软件非功能需求权衡代价.软件学报,2017,28(5):1247–1270. <http://www.jos.org.cn/1000-9825/5106.htm> [doi: 10.13328/j.cnki.jos.005106]
- [14] 刘书琴.基于元需求模型的非功能性元需求集的获取与分析研究[硕士学位论文].长沙:中南大学,2010. [doi: 10.7666/d.y1914241]
- [15] 倪瑜泽,彭蓉,孙栋,等.基于用户评论的潜在演化需求发现方法.武汉大学学报(理学版),2015,61(4):347–355. [doi: 10.14188/j.1671-8836.2015.04.009]
- [20] 上海林原信息科技有限公司.HanLP 汉语处理包:HanLP v1.2.8.2017. <http://hanlp.linrunsoft.com/>

- [22] 罗新星,李柱辉,赵玉洁.软件非功能需求国内外研究综述.计算机应用研究,2015,32(4):972-977. [doi: 10.3969/j.issn.1001-3695.2015.04.003]
- [24] 田久乐,赵蔚.基于同义词词林的词语相似度计算方法.吉林大学学报(信息科学版),2010,28(6):602-608. [doi: 10.3969/j.issn.1671-5896.2010.06.011]



贾一荻(1994—),女,北京人,硕士,主要研究领域为需求工程,中文文本分析.



刘璘(1973—),女,博士,副研究员,主要研究领域为需求工程,信息系统工程,医疗数据分析.

www.jos.org.cn

www.jos.org.cn