

智能需求获取与建模研究综述

汪 烨¹ 陈骏武¹ 夏 鑫² 姜 波¹

¹(浙江工商大学计算机与信息工程学院 杭州 310018)

²(澳大利亚蒙纳士大学信息技术学院 澳大利亚墨尔本 3800)

(yewang@zjgsu.edu.cn)

Intelligent Requirements Elicitation and Modeling: A Literature Review

Wang Ye¹, Chen Junwu¹, Xia Xin², and Jiang Bo¹

¹(School of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou 310018)

²(Faculty of Information Technology, Monash University, Melbourne, Australia 3800)

Abstract Requirements elicitation and modeling refer to the process of obtaining explicit or implicit requirements from the requirements text described in natural language, and constructing the corresponding models through tabular, graphical, and formulaic methods. Requirements elicitation and modeling is an extremely critical step in software development process, which paves the way for subsequent system design and implementation, improves the efficiency and quality of software development, and improves the stability and feasibility of software systems. Researchers have obtained a series of research achievements in requirements elicitation and modeling. Requirements elicitation and modeling can be generally divided into three steps: requirements knowledge extraction, requirements knowledge classification and requirements model construction. Due to the fact that traditional requirements elicitation and modeling approaches have problems in terms of accuracy and efficiency of model construction, in recent years, more and more researchers have integrated widely applicable artificial intelligence techniques with these approaches, and put forward a series of intelligent requirements elicitation and modeling approaches, so as to make up for the deficiencies of the traditional methods. This paper focuses on the perspective of intelligent requirements elicitation and modeling, and sorts out and summarizes the research progress of requirements elicitation and modeling in recent years. The main contents include: 1) statistics and analysis of the artificial intelligence techniques applied in requirements knowledge extraction, requirements knowledge classification and requirements model construction; 2) summarizing the verification and evaluation methods used in the process of intelligent requirements elicitation and modeling; 3) summarizing the key issues of intelligent requirements elicitation and modeling from two aspects of scientific problems and technical difficulties, and elaborating on the six research trends including integrated and dynamic model construction, mining the relationships among intelligent requirements elicitation and modeling and other software engineering activities, refining the granularity of intelligent requirements modeling, data sets construction, evaluation metrics construction and industrial practice as the possible solutions to the above problems.

收稿日期:2020-09-16;修回日期:2020-12-21

基金项目:浙江省自然科学基金项目(LY21F020011, LY20F020027, LY19F020003);国家自然科学基金项目(61672459)

This work was supported by the Natural Science Foundation of Zhejiang Province (LY21F020011, LY20F020027, LY19F020003) and the National Natural Science Foundation of China (61672459).

通信作者:夏鑫(Xin.Xia@monash.edu)

The future development trend of intelligent requirements elicitation and modeling research is also discussed.

Key words requirements modeling; artificial intelligence; requirements elicitation; natural language processing; machine learning; deep learning

摘要 需求获取和建模是指从需求文本或记录中获取显式和隐式的需求,并通过表格化、图形化、形式化等方法构建相应模型的过程,是软件开发过程中极为关键的一步,为后续系统设计与实现铺平道路,提高软件开发效率和质量,提升软件系统稳定性和可行性.研究者在需求获取与建模方面获得了一系列研究成果,根据其关注阶段不同,可以将它们分为需求知识提取、需求知识分类和需求模型构建3个方面.鉴于传统方法在知识获取、模型构建的准确性和效率方面一直存在弊端,近年来,越来越多的研究者将具有广泛应用性的人工智能技术与需求获取、需求分类、需求建模方法相结合,提出了一系列智能需求获取与建模的方法和技术,从而弥补了传统方法的不足.着重从智能需求获取与建模角度着手,对近年来的研究进展进行梳理和总结.主要内容包括:1)统计并分析人工智能技术在需求知识提取、需求知识分类和需求模型构建中使用的方法和技术;2)总结了智能需求获取与建模过程中采用的验证方法和评估方法;3)从科学问题和技术难点2个方面归纳得出目前智能需求获取与建模的关键问题,围绕集成化和动态化模型构建、与其他软件工程活动关联、智能需求知识分类的粒度、数据集构建、评估指标构建和工具支持6部分,阐述了上述问题的可能解决思路和未来发展趋势.

关键词 需求建模;人工智能;需求获取;自然语言处理;机器学习;深度学习

中图法分类号 TP311

智能软件工程(intelligent software engineering, ISE)是近年来软件工程科学研究中出现的新兴领域^[1-3].谢涛^[4-5]从2个角度对ISE进行了定义:1)人工智能(artificial intelligence, AI)技术在软件工程学科中的应用;2)为智能软件开发提供软件工程的解决方案.越来越多的软件工程顶级会议中单独设立了与ISE相关的研讨会.诸如IEEE/ACM International Conference on Automated Software Engineering (ASE)的the International Workshop on Intelligent Software Engineering(WISE), International Conference on Software Engineering(ICSE)的International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering(RAISE), IEEE International Requirements Engineering Conference(RE)的International Workshop on Artificial Intelligence for Requirements Engineering(AIRE), International Conference on Predictive Models in Software Engineering(PROMISE),都显示出软件工程领域对ISE方向研究的重视.

需求工程(requirements engineering, RE)是软件工程中的重要研究领域,其目标是指通过现有的技术、方法进行用户需求分析,帮助技术人员理解问题并明确系统目标^[6].需求获取与建模是需求工程

中的重要步骤,旨在通过表格化、图形化、形式化等方法,对获取的结构化或非结构化的需求知识进行模型构建.通过模型构建,软件开发人员可以:1)根据用户需求对软件系统进行合理的可视化;2)更详细地说明软件系统的结构和行为;3)为软件系统的设计提供清晰化指导;4)有利于涉众之间的相互交流;5)为后期系统维护和升级提供了便利.

自20世纪90年代,研究者们已提出很多传统需求获取和建模方法,包括问卷、访谈、文档分析^[7]、团队启发^[8]、原型设计^[9]、模型驱动^[10]等获取方法,以及数据建模^[7]、企业建模^[11]、行为建模^[12]、非功能需求建模^[13]、领域建模^[14]、用例建模^[15]、特征建模^[16]、面向目标的建模^[17]、问题框架^[18]、面向方面的建模^[19]等需求建模方法,分别对用户需求或系统需求的不同侧面、不同层次进行提取、抽象和刻画,但仍然存在一些问题与挑战.首先,由于用户通常使用自然语言描述软件系统的需求^[20],而自然语言描述中存在着模糊、冗余等特性,传统方法所提取的知识或构建的模型中往往会出现重点不明确等问题.其次,传统的需求获取与建模过程需要大量依靠人工实施,这种方式在面对大规模的、复杂的需求文档和需求之间的关系时,将耗费巨大的时间成本,需求的准确性、一致性和完整性会大幅降低^[21].

鉴于此,研究者们致力于研究并解决需求在获取与建模过程中的不完整^[22]、不准确^[23]和效率低下^[24]问题,并提出了智能需求获取与建模方法,将人工智能技术如机器学习^[25]、深度学习^[26]、自然语言处理^[27]等与传统的需求获取和建模方法相融合,模拟人工处理的方式,能极大地在需求知识提取^[28]、需求知识分类^[29]、需求模型构建^[30]方面提出新思路,解决人工方式下模型生成效率低、准确率差等弊端^[21],在需求管理、需求追踪、需求变更、需求预测等方面发挥了重要作用^[31-32]。

目前,国内外对于智能需求工程的研究正在日渐成熟,杨卓群等人^[33]针对自适应系统中的需求工程,对其建模方法、需求质量属性和应用领域等方面进行了充分的调查研究,总结了其最新发展趋势;栗元邦等人^[34]对不同的情景维度进行分析,总结出其对需求获取与建模的支持作用,统计并深入分析了情景感知下的需求获取与建模中常用方法,Perkusich等人^[35]综合分析了不确定下的推理、基于搜索的解决方案和机器学习用于敏捷软件开发领域的技术现状,包括工作评估、需求优先级划分、资源分配、需求选择和需求管理,Gea等人^[36]调查了当前RE工具的支持功能和支持程度,提出这些工具在需求建模、开放数据模型和数据集成性上仍有改进的空间,然而,目前对于系统性归纳人工智能背景下需求获取与建模的工作鲜有涉及,为了使研究者对智能需求获取与建模研究现状有更清晰的把握,了解其研究

进展,本文采用文献综述的方法,对近15年来的研究进行了系统的分析整理,对这些新方法进行了统一的调研和分析。

在文献选取方式方面,我们选取文献方式遵循的标准有:1)该文献针对智能需求工程中的需求知识提取、需求知识分类和需求模型构建方面提出新的自动化技术和方法,或该文献对智能需求获取与建模的新技术和新方法进行了实际的案例研究和应用研究;2)该文献已公开发表在国内外的会议、期刊或者专著中。

根据这2个标准,本文通过4步索引和筛选文献:

1) 根据研究方向确定检索的关键字,确定的英文关键字为: requirements engineering 或 requirements survey 或 requirements investigation 或 requirements challenge 或 requirements mining 或 requirements modeling 或 requirements extraction 或 requirements classification 等,确定的中文关键字包括:需求建模、需求获取、需求分类、需求挑战、需求提取、需求工程等。主要用到的数据库包括:EI Database(简称EI)、Web of Science Database(简称Web of Science)、IEEE-CS Database(简称IEEE)、ACM Database(简称ACM)、ScienceDirect (SD) Database(简称ScienceDirect)、Springerlink Database(简称Springer)、中国知网(CNKI)。检索时间区间定义在2005年1月~2020年6月,初步检索出18137篇文献,具体分布如图1所示:

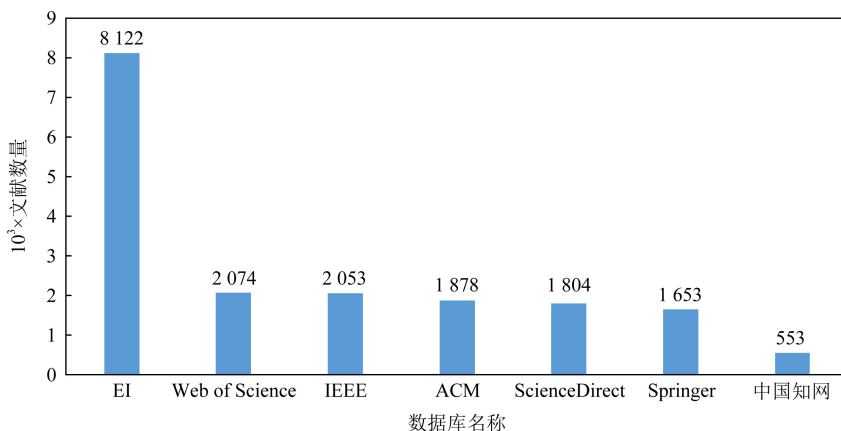


Fig. 1 Database search

图1 数据库检索

2) 由第1作者和第2作者对上述文献按照标题、关键词、摘要、结论和来源进行筛选,我们筛选文献的标准为:对于英文文献,根据《中国计算机学会推荐国际学术会议和期刊目录》,选择B类(含)以

上的权威软件工程期刊和会议;对于中文文献,根据《CCF推荐中文科技期刊目录》选择A类的科技期刊,由此得到95篇文献。

3) 由上述2个作者对95篇文献进行全文查看,

若根据纳入标准 2 名研究人员无法达成一致意见时,向课题组另外 2 名研究人员陈述纳入和排除理由,并由另外 2 名研究人员做出最终决定.经过此步骤,保留相关文献 45 篇.

4) 由第 1 作者和第 2 作者从保留文献的参考文献中通过滚雪球的方式^[37]进一步筛选出符合要求的相关文献,滚雪球共找出 23 篇可能文献,再经过 2 轮筛选,得到 12 篇相关文献.

通过上述 4 步文献的检索与筛选后,目前共有 57 篇相关文献纳入本文的调研中,这些文献与智能需求获取与建模技术直接相关,为需求知识提取、需求知识分类、需求模型构建、方法评估和应用提供了新思路 and 新技术.图 2 展示了本文所总结的文献分布情况,其中包含 JSS(Journal of Systems and Software),ASE 各 9 篇,RE 论文 7 篇,REJ(Requirements Engineering)论文 6 篇,《软件学报》5 篇,CAiSE(International Conference on Advanced Information Systems Engineering)4 篇,ICSE, ASEJ(Automated

Software Engineering),EMSE(Empirical Software Engineering),IST(Information and Software Technology),IETS(IET Software),ESEC/FSE(European Software Engineering Conference/ACM SIGSOFT Conference on the Foundations of Software Engineering)各 2 篇,SoSyM(Software System Modeling),TOSEM(ACM Transactions on Software Engineering Methodology),MoDELS(Conference on Model Driven Engineering Languages and Systems)、《电子学报》和《计算机学报》各 1 篇.图 3 展示了本文所总结文献发表年份的分布情况.

本文的综述过程如图 4 所示,根据上面 4 步获得 57 篇论文集后,对选取的文献进行分类和归纳,分为智能需求获取与建模的新技术新理论、实证研究 2 个方面.对于提出新技术新理论的文献,从 3 个步骤进行分类:需求知识提取、需求知识分类和需求模型构建.对于实证研究方面,对各类文献进行系统分析、比较,总结了此类文献用来衡量其方法的评估

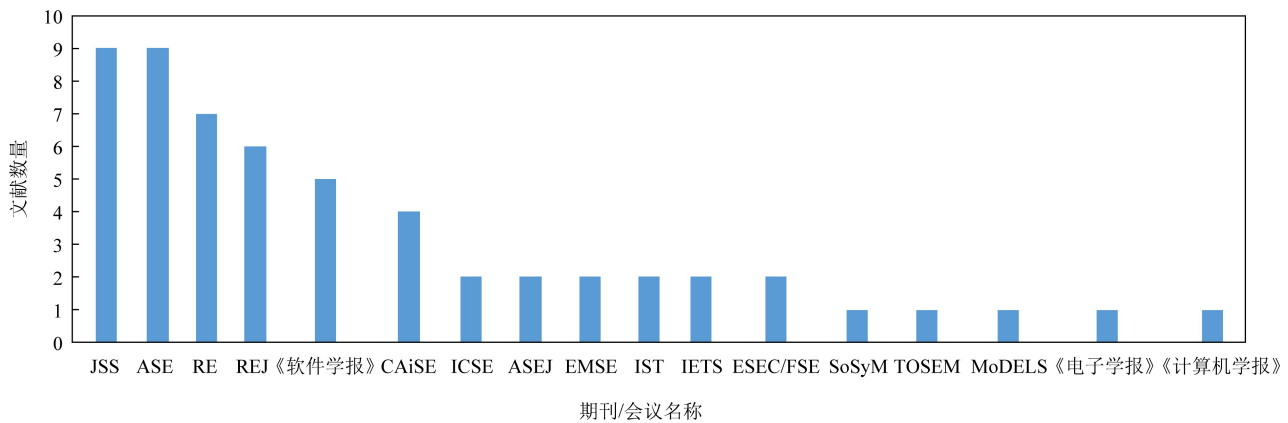


Fig. 2 Literature distribution

图 2 文献分布

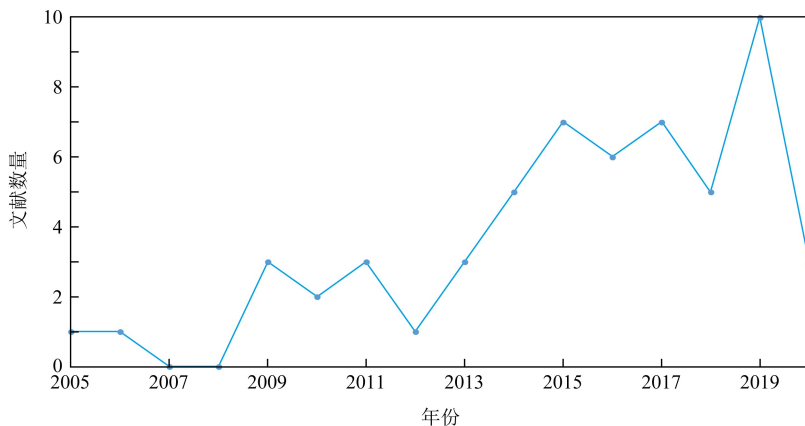


Fig. 3 Statistics of publication years

图 3 文献发表年限统计

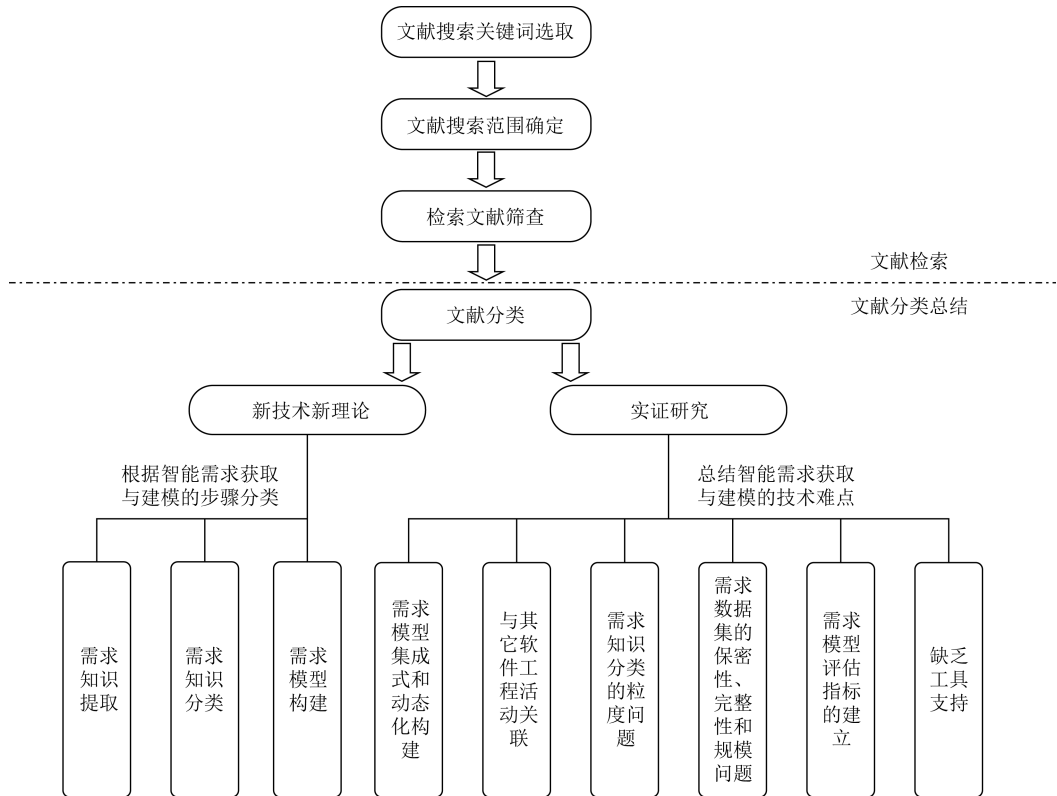


Fig. 4 Overview framework

图 4 综述框架

指标,最后归纳出目前智能需求获取与建模在集成式和动态化模型构建、与其他软件工程活动关联、需求知识分类的粒度、数据集构建、评估指标的建立和缺乏工具支持等关键问题与挑战。

1 智能需求获取与建模的过程框架

本节我们主要介绍智能需求获取与建模的一般过程框架。通过对相关文献的总结,我们将智能需求获取与建模过程归纳为 3 个步骤,如图 5 所示。

1) 需求知识提取^[28]。通过自然语言处理、机器

学习、深度学习等技术从项目文档、记录、用户评论和用户请求中提取相应的需求知识,作为后续模型构建的基本元素。

2) 需求知识分类^[38-39]。采用支持向量机(support vector machine, SVM)方法、长短期记忆网络(long short-term memory, LSTM)等技术对不同种类的需求知识进行分类,帮助开发人员更好地改善系统,明确系统所需完成的目标,为模型构建提供帮助。

3) 需求模型构建^[30]。根据已提取或已分类的需求知识元素,分析需求关注点,结合 Markov 链、贝叶斯网络、基于规则的模型构建等算法自动化地构建模型。

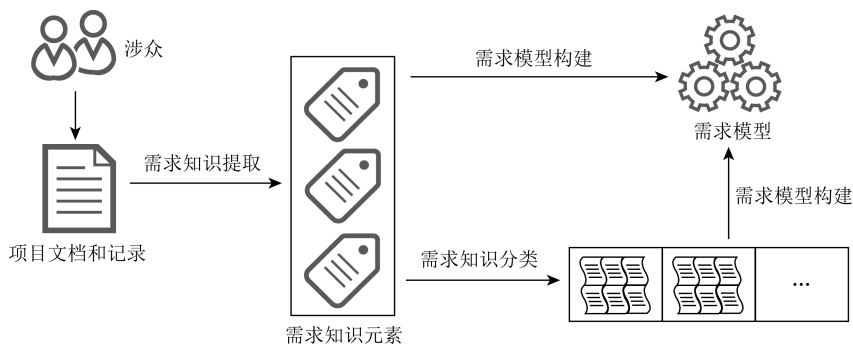


Fig. 5 General steps for intelligent requirements elicitation and modeling

图 5 智能需求获取与建模的一般步骤

根据我们的调研和总结,在需求知识的智能提取和分类方面,早期应用的技术以自然语言处理和信息检索技术为主,包括文本预处理、术语权重和相似度计算等方法,后期采用的较多方法是将机器学习与自然语言处理技术结合,近2年来深度学习技术也被尝试着应用到需求知识的提取和分类中。在需求模型的智能构建方面,早期以基于规则和依赖分析的模型构建方法为主,近年来基于数据挖掘和机器学习的方法逐渐兴起,并取得了不俗的成果。本文分别介绍了上述3个步骤所采用的技术和方法。

2 需求知识提取

软件需求明确定义了软件应该“做什么”的问题,通过访谈、问卷等方法从用户获得并记录的需求文本大多采用自然语言描述,采用人工方法从大量的自然语言描述文档中提取需求知识将耗费巨大的时间和人力成本。因此,越来越多的需求工程研究人员致力于将LDA(latent Dirichlet allocation)^[40], BiLSTM(bi-directional long short-term memory)^[26]和CNN(convolutional neural networks)^[41]等人工智能技术应用于需求知识元素的提取,不但大幅降低提取时间成本和提高提取效率,而且提高了知识提取的准确性。下面将重点介绍这些技术在需求知识提取步骤的应用,根据提取方法的种类,我们将其大致分为2类:(半)自动化的自然语言处理方法和基于人工智能的需求知识提取方法。前者偏向对需求文本做语法、语义层面的分析和计算;后者虽然也会采用文本分析方法,但是更侧重于采用机器学习和深度学习技术从文本中挖掘有用的需求信息。

2.1 (半)自动化的自然语言处理方法

目前需求描述和记录方式大多为用例、用户故事、功能列表等由短文本构成的结构化形式^[42-43],已有研究表明^[44],采用自然语言处理(natural language processing, NLP)技术对需求文本进行分析、处理,挖掘需求文档中包含的知识元素具有广泛的研究前景。

与自动化技术相比,半自动化技术在提取中需要人工干涉,过程相对繁琐,例如通过添加人工确认需求或微调需求的步骤,以提高提取的准确率。此外,如果提取的需求是针对某个特有的领域场景,则需要人工对某些偏好属性进行设定。自动化技术则实现了需求文本处理全过程的自动化,提取效率得到极大的提升。然而,提取的效果会由于样本需求的大小不同而呈现不稳定性,并且存在准确率低于召

回率的结果。因此,在需求的初步处理阶段,对于提取较为简单的实体、对象、动作等元素,自动化技术更为适合,因为可减少人工成本的投入,并获得较好的成果。而在获取不同需求之间的关系时,半自动化技术更为合适。

2.1.1 文本预处理

文本预处理是早期用来进行需求知识提取的方法,是指对采用自然语言描述的需求文本进行语法和语义分析,包括分词^[44]、词干提取^[45]、停用词删除^[46]等方法。

利锦标等人^[28]利用NLP技术从需求文本中自动提取类图元素,但该方法缺乏工具支持。Vu等人^[45]采用自定义的词干提取算法,从原始用户评论中提取意见关键词,正确率达到97.9%。Elbendak等人^[47]基于文本预处理开发了一个Class-Gen工具用来提取类和对象,以生成初始UML(unified modeling language)类模型。Rago等人^[48]为了提取用例描述中的语义信息,开发了一款REAssistant的自动化文本分析工具,并为用例生成不同的注释。

文本预处理方法只能用来提取显式的基本需求知识,即与需求知识相关的单词、短语或语句,如对象或主体对应名词,动作或关联对应动词,功能或目标对应动宾短语,但是却难以分析不同需求之间的关系。此外,文本预处理方法很难识别出不必要或冗余的需求知识,无法处理由此产生的信息噪声问题。随着需求文本的规模扩大,复杂句式出现的频率大大增加,对于表述很差或粒度较大的需求文本,若仅采用文本预处理的需求提取方法,提取效果将受到严重影响。

2.1.2 术语权重

为了获取需求文本中的某一术语对该文本的重要性,信息检索技术里的术语权重方法逐渐在需求知识提取中得到应用。术语权重是指根据术语在文档中出现的频率为其分配不同的权重。Rago等人^[49]自定义了一种新的加权术语频率的计算方法,实现了对非功能需求的提取,正确率达到86%~90%。术语权重方法在一定程度上弥补了文本预处理方法无法处理的信息缺陷,但在提取过程中严重依赖语料库,且难以提供词的位置信息,降低了需求知识的提取精度。

2.1.3 相似度度量

针对术语权重在处理需求文本上的不足,需求工程研究者们后期又提出了相似度度量方法。该类方法通过计算不同需求语句的相似度,以便提取出

所需要的知识元素,如重复的需求、具有二义性的需求等.常见的相似度计算方法包括余弦相似度、Jaccard 相似度,以及概念相似度、String 相似度、Bi-gram 相似度等^[50-52].为了更好地提取需求知识,相似度度量通常与文本预处理、术语权重方法组合使用.Jaccard 相似度一般用于计算集合之间的相似度,适用于需求文本中包含大量重复性的词语,而余弦相似度用于计算 2 个单词之间的相似度,适用于单词重复性较少的需求文本.

针对需求文本中存在的歧义性需求,Rauf 等人^[53]利用了 ET(extraction tools)工具,从需求文本中提取出结构模型元素,包括实体、属性以及多重性.Vu 等人^[54]在基于关键词的方法后,又采用余弦相似度和 Jaccard 度量方法从应用程序评论中提取用户观点,准确率和召回率得到提升.Ferrari 等人^[55]基于特定领域的语言模型,提出了通过余弦相似度计算方法提取不同领域之间的歧义术语,并根据歧义度对其进行排序的方法.

Table 1 (Semi) Automated Natural Language Processing Approaches

表 1 (半)自动化的自然语言处理方法

类别	技术/方法	工具	需求来源	需求知识	需求模型	需求类型	文献
文本预处理	数据清洗、分词、词性标注、词性还原		需求文档	动作、活动、任务			[28]
		MARK	用户评论	用户的视角关键词		FR	[45]
		Class-Gen	自然语言需求规约	对象和类	类模型		[47]
		REAssistant	用例文本	横切关注点			[48]
术语权重	本体和加权术语	QAMiner	用例规约	质量属性		NFR	[49]
相似度度量	Bi-gram 相似度	ET	需求文档	逻辑结构			[53]
	余弦相似度和 Jaccard 相似度	MARK	移动应用评论	用户观点		FR	[54]
	余弦相似度		需求文档	歧义术语			[55]

表 1 列举了目前 NLP 技术在需求知识提取方面的应用,包括 NLP 技术分类、NLP 技术、工具、需求来源、提取的需求知识元素、构建的需求模型、关注的需求类型和对应的参考文献.表 1 中 FR (functional requirements) 代表功能需求, NFR (non-functional requirements) 代表非功能需求.

2.1.4 开源的 NLP 支撑工具

需求知识提取阶段,研究者们通常会借助开源的 NLP 支撑工具对需求文本进行语法分析和处理,提高需求知识元素的提取效率,极大地节约了人力成本.常用的开源工具包括 Stanford NLP 工具^[56]、GATE^[57] 和 Open NLP 工具^[58] 等.

Stanford NLP 工具^[56]由斯坦福大学开发,是被应用最广泛的文本处理工具.基于 Stanford NLP 工具,Sagar 等人^[59]通过利用该工具和 WordNet,从功能需求规格中自动提取类及其关系,以便自动构建出概念模型;其他研究人员也采用了 Stanford NLP 工具从需求文档、用户手册等提取功能需求^[60-62]. Han 等人^[63]提出一种从需求文本的约束描述中自动提取过程模型元素的方法,将提取出的活动及其

语义关系转换为声明性约束.GATE^[57]是谢菲尔大学自然语言处理小组研究并开发的免费文本分析平台.采用该工具,Morales-Ramirez 等人^[64]对应用商店的评论、开源软件的邮件列表和用户论坛中的文本消息进行自动化分析和知识提取,提取出系统的功能特征;AlHroob 等人^[65]则从用例中提取参与者和动作.Open NLP 是一个专门用于处理自然语言的机器学习工具集.基于 Open NLP,Rago 等人^[66]开发出 ReqAligner 方法,帮助分析人员以自动化的方式提取出用例中重复的功能需求,提高需求的可读性,便于后期工作量估算、功能优先级划分和维护等工作的进行.HanLP 是一种自然语言分析工具,王春晖等人^[67]利用这个工具帮助提取用户故事场景.上述 NLP 支撑工具针对的需求文本格式和语言以及提取的需求知识各有不同,研究人员可根据实际需要选择适合自己项目的工具进行需求文本的自然语言处理.

表 2 列举了开源的 NLP 工具在需求知识提取方面的应用,其中 FR 代表功能性需求, NFR 代表非功能需求.

Table 2 Open Sourced NLP Supporting Tools

表 2 开源的 NLP 支撑工具

支撑工具	需求来源	需求知识	需求模型	需求类型	文献	
Stanford NLP	用户评论	用户的视角关键词			[45]	
	移动应用评论	用户观点			[54]	
			类、属性、方法和关联	概念模型		[59]
	需求文档	与软件特征相关的信息				[62]
			动作和约束	过程模型		[63]
	用例规约	对象和消息		顺序模型	FR	[60]
GATE	需求规约	领域类	领域模型		[61]	
	在线反馈	特征			[64]	
	需求文档	参与者和动作			[65]	
Open NLP		模糊需求			[21]	
	需求文档	重复需求			[66]	
	用例规约	质量属性		NFR	[49]	
HanLP	用户故事	用户故事关系		FR	[67]	

2.2 基于人工智能的需求知识提取方法

自动化或半自动化自然语言处理方法通常仅能提取出显式的需求知识,对于语句中所包含的隐式需求则很难获取.鉴于机器学习和深度学习在挖掘隐藏信息上的优势,近年来越来越多的研究者们尝试着采用人工智能技术提取需求知识,包括较早尝试的贝叶斯网络^[68]、LDA^[40],以及最新被应用的BiLSTM^[26]、CNN^[41]等.

2.2.1 基于机器学习的需求知识提取方法

采用机器学习方法提取需求知识过程与普通机器学习任务类似,通常包含 4 个步骤:

1) 文本的自然语言处理.对包含需求知识的文本进行自然语言处理,包括分词、词性标注、关键词提取等,必要时进行进一步的缺失值处理、离散化等数据处理,使得算法的效果和性能得到显著提高.

2) 文本特征提取.利用熵、Word2Vec, TF-IDF 等相关技术,筛选出数据中不同需求知识元素的显著特征,也可手动添加新数据特征.

3) 提取模型的构建和训练.选择合适的机器学习模型,采用训练集数据对模型进行训练,不断调优,使得模型达到最优.

4) 需求知识提取.将测试集数据输入到机器学习模型,完成相关需求知识的提取.

在功能性需求知识的提取方面,主题建模、SVM 和主动学习是比较常用的智能技术.Bhowmik 等人^[69]、Chen 等人^[70]和 Li 等人^[71]基于主题建模技术分别从需求文本和用户反馈意见中提取创新性

需求.但如果需求文本或 e-mail 信息的描述多样化,例如包含一些程序方法或类名,则会阻碍主题的构建.LDA 是广泛应用的主题建模技术,但其对过短的需求文本进行主题提取时效果不佳,通常需将多个短文本组合成一个长文本进行主题建模.Rodeghero 等人^[72]分别采用了逻辑回归和 SVM,从用户与开发人员之间的对话记录中智能提取与用户故事相关的信息,并发现逻辑回归在所评价的各类指标(如准确率、召回率)下的综合效果最好.Arora 等人^[73]为了提取并过滤冗余的需求,采用主动学习方法对已提取领域模型元素的相关性和冗余性进行迭代学习得到反馈信息,最后过滤掉冗余的需求.主动学习是一个循环的过程,通过查询最有用的未标记样本,交由专家进行标记,再用查询到的样本训练模型提高模型的精确度.主动学习通常依赖于专家经验或领域经验,并且需要构建合适的标注候选集.Sardinha 等人^[74]基于贝叶斯学习方法分别提取出不同方面需求间的冲突,并开发了 EA-Analyzer 作为支持工具.在非功能需求知识的提取方面,贝叶斯学习方法^[75]被证明是效果较好的技术.然而,上述机器学习方法需要依赖大量的特征工程和领域专业知识,且对于不同领域的适应性较差,在数据集异常的情况下提取精度会受较大影响,需要进行异常值处理,同时容易产生过拟合或者欠拟合现象.

2.2.2 基于深度学习的需求知识提取方法

为了降低特征工程对提取精确度上的影响,近

2年来,深度学习技术被尝试应用于需求知识提取中.采用深度学习提取需求知识通常包含4步:文本的自然语言处理、文本特征提取、神经网络模型构建与训练、需求知识提取.该过程与机器学习过程类似,不同地方在于神经网络模型的构建与训练.目前对于基于深度学习的需求知识提取方法研究,学术界尚处于初探阶段.

BiLSTM和CNN是应用较为广泛的神经网络模型.CNN运行速度很快,可以在较简单的网络结构下快速、准确地实现任务,BiLSTM可以通过时间序列的方式更好地捕捉需求文本的双向语义依赖.在智能需求知识提取中,Pudlitz等人^[76]使用自训练

的基于BiLSTM和CNN的命名实体识别模型从需求规格说明书中提取对象的状态.Saraiva等人^[26]基于BiLSTM从文本和多媒体数据中自动识别、提取涉众关于功能需求的意见和情感.Qian等人^[41]提出了一种层次神经网络方法,将提取任务形式化为多粒度文本问题,对多粒度的过程模型信息进行提取.然而,由于其卷积核大小的限制,CNN目前只能提取局部特征,而BiLSTM由于时间序列性导致其结构较为复杂,效率低,网络性能因此大受影响,并且在长距离依赖问题上表现较差.

表3列举了目前基于人工智能的需求知识提取方法、分类和工具等.

Table 3 Intelligent Requirements Knowledge Extraction Approaches

表3 智能需求知识提取方法

类别	技术/方法	工具	需求来源	需求知识	需求模型	需求类型	文献
机器学习	LDA		用户意见	创新性需求		FR/NFR	[69]
			移动应用评论	关于特征的意见			[70]
		GATE	用户手册和项目报告	功能需求			[71]
	逻辑回归/SVM		会话记录	用户故事信息		FR	[72]
	主动学习	Extractor	需求文档	冗余需求	领域模型		[73]
	贝叶斯学习方法		EA-Analyzer	需求文档	需求冲突		
		HeRA	安全性需求文档	安全性需求		NFR	[75]
深度学习	POS-AttWD-BLSTM-CRF		需求文档和多媒体数据	观点、情感、情绪			[26]
	CNN		需求文档	动作和控制流	过程模型	FR	[41]
	BiLSTM-CNN			状态			[76]

2.3 其他方法

除了2.1节和2.2节的自然语言处理技术和机器学习等技术,还有研究学者们应用其他自动化技术进行需求知识提取,包括:基于树结构的提取方法和基于抽取规则的提取方法等.刘伟等人^[77]为了从Web网页中提取出用户评论,提出一种基于深度加权的树相似性算法和DOM树方法,该方法在一定程度上解决了由不同用户撰写评论的风格不统一带来的知识提取问题.李天颖等人^[78]基于自定义的抽取规则,从需求文本中提取出有用的关系信息,在实验的结果上充分体现了其优势,同时,该抽取关系的方法易扩展,应用广泛.胡甜媛等人^[79]为了在用户评论中挖掘有价值的使用反馈,对评论种子进行定义,并通过抽取规则挖掘出与之类似的有用用户反馈,该方法适用于用户评论数据量大、更新快速的语料库.但目前的方法面对复杂的用户评论时,在完备性上略显不足.

3 需求知识分类

需求知识分类可以提高对构建某类模型所需元素的关注度,减少建模工作量,提高需求建模的效率.根据在分类过程中所使用的技术,按照时间顺序,我们将其分为基于信息检索的分类方法、基于机器学习的分类方法和基于深度学习的分类方法.

3.1 基于信息检索的需求知识分类方法

Cleland-Huang等人^[80]提出了一种基于信息检索的方法,对非功能需求进行自动检测和分类.该方法首先从设计文档中识别出非功能需求的指标术语,然后通过计算术语与需求关键词之间的相似度对非功能需求进行分类.贾一获等人^[81]提出了一种基于语义距离和特征词相似度的计算方法,将非功能性需求语句自动划分为安全性、可用性、性能、可靠性、可维护性这5类,便于后期针对不同种类

非功能需求的架构设计.基于信息检索的方法减少了需求分析人员的工作量,但此类方法适用于需求较为明确的文本,且在很多场景下仍需要人为定义与需求相关的关键词.

3.2 基于机器学习的需求知识分类方法

为了克服传统信息检索技术通过匹配关键词导致的分类精度不高的问题,研究者们对机器学习在需求知识分类上的应用进行了探索.基于机器学习的需求知识分类方法与大多采用有监督或半监督的文本分类方法类似^[82],主要步骤包括:文本的自然语言处理、特征提取、分类模型设计与训练、需求知识分类.此外,也有研究人员采用 SMOTE(synthetic minority oversampling technique)等算法、数据增强等方法对于分类样本数据不平衡、不充足进行处理.

针对安全性需求的分类问题,Riaz 等人^[83]根据安全目标,采用基于 K 近邻算法(K -nearest neighbor, KNN)、朴素贝叶斯等机器学习技术从项目文档中针对与安全需求相关的信息进行自动化识别和分类,但该方法仅应用于健康领域,在其他领域的应用效果未知.

Kurtanović 等人^[84]针对数据挑战,提出一种基于 SVM 和词汇特征的有监督机器学习方法,对功能需求和非功能需求进行智能分类.该方法在手动特征选择时得到的准确率和召回率较高,但是进行自动化特征选择时准确率则会有明显的下降.Li 等人^[85]将信息检索技术与多种分类器如 KNN、朴素贝叶斯进行集成,对论坛中的用户请求文本进行功能需求与非功能需求分类,通过主动学习减少了人工标注的成本,提高了准确率和召回率,但是该方法暂时无法处理较为复杂的用户请求.Khan 等人^[86]同样采用 TF-IDF 与逻辑回归、SVM 等机器学习算法结合的方法,对 Reddit 论坛中的用户评论进行二阶段分类,从中识别出功能需求、正面评论、负面评论和问题.

针对移动应用程序 APP 中用户评论的分类问题,Vu 等人^[45]采用 K -means 和相似度算法从移动应用程序中将用户的用户关键词划分为内聚性更强的子集,该方法的准确率达到 83.11%.陈琪等人^[87]采用 SVM 构建了 APP 用户评论的多标签分类模型,将评论进一步分为功能问题、内容问题、响应时间等,并采用了代价敏感学习方法解决了不平衡数据问题.该方法虽然有部分涉及到需求知识分类,但其主要目的还是为了主题挖掘和评论分析.

针对 NFR 的分类问题,Casamayor 等人^[88]提出了一种基于朴素贝叶斯和 EM(expectation maximization)算法的半监督学习方法,虽然降低了有监督学习中的人工参与成本,但准确率仍有待提高.为了更好地研究不同机器学习算法在 NFR 分类上的效果,Jha 等人^[89]设计了不同的组合方法,分别基于贝叶斯、SVM 和基于字典的分类算法,自动捕获并分类用户评论中的 NFR 类别,结果发现 SVM 方法的性能相比贝叶斯较优,而基于字典的分类算法的准确性很大程度上取决于与 NFR 类别匹配的术语选择是否准确.

相比信息检索方法,机器学习方法对需求进行分类时在保证较高精确度的同时能够减少人力成本^[90-91],如果采用的是半监督或无监督学习方法,需要标记的训练数据会大大减少,学习效率更高.但是机器学习方法在模型的学习过程中,对于不同领域的相同任务其特征工程可能大不相同,导致其效果通常与适用场景紧密相关,尤其在目前需求分类尤其是功能需求分类缺少公开数据集的情况下,很难验证上述方法在不同领域的有效性.

3.3 基于深度学习的需求知识分类方法

随着人工智能技术的发展,能够捕捉特征的上下文、对向量化数据具有高可计算性的深度学习技术进入了研究人员的视线,并被迅速应用到需求知识分类上.基于深度学习的需求知识分类方法在特征提取阶段可以采用 Word2Vec 和 BERT 等技术将需求文本映射到向量空间,再采用 LSTM, CNN 等神经网络进行训练调优,最后获取所需的分类类别.

Guzman 等人^[92]针对手机应用程序下的评论,对比了多个机器学习和深度学习方法,将应用程序评论划分为与软件演化相关的类别,结果表明采用神经网络的需求分类方法表现最优,直接证明了深度学习技术在需求工程中的应用具有广阔前景.Rahman 等人^[93]通过 Word2Vec 将文档向量化,然后输入到神经网络模型 LSTM, GRU(gated recurrent unit)和 CNN 中进行了分类,比较发现 LSTM 更好地满足了用户对软件开发高质量的期望.相比于传统的机器学习方法,基于深度学习的分类方法准确度较高,对具有噪声的数据也能保持较好的鲁棒性和容错性.但是深度学习方法存在可解释性不足的问题,且需要大量数据支持.

表 4 列举了在信息检索、机器学习和深度学习方面的智能需求分类方法.

Table 4 Intelligent Requirements Knowledge Classification Approaches

表 4 智能需求知识分类方法

类别	技术/方法	分类的需求	文献
信息检索	TF-IDF	可用性、可维护性、性能、安全性、易用性、合法性和可操作需求等	[80]
	语义距离和相似度计算	安全性、可用性、性能、可靠性和可维护性	[81]
	K-means	聚类后的用户关键词	[45]
	KNN	保密性、完整性、身份识别和认证、可用性、可问责性和隐私需求等	[83]
机器学习	SVM	易用性、安全性、性能和可操作需求等	[84]
		功能需求、响应时间、资源、网络等	[87]
	KNN/SVM/朴素贝叶斯	容量、安全性、可靠性、性能、生命周期、易用性和系统接口	[85]
	朴素贝叶斯/随机森林/SVM/逻辑回归	特征、支持观点、反对观点、问题	[86]
	EM算法-朴素贝叶斯	功能需求、可用性、可维护性、合法性需求等	[88]
	贝叶斯/SVM	可靠性、性能、可支持性和易用性需求	[89]
	朴素贝叶斯/决策树/MaxEnt	漏洞报告、特征请求、用户体验、等级评定	[90]
	层次聚类/划分聚类算法	安全性、性能、可访问性、准确性、可移植性、安全性、合法性、可靠性和可用性	[91]
深度学习	LSTM	执行者、动作	[41]
	神经网络	漏洞报告、特征优势、特征弱势、用户请求、用户赞扬、用户抱怨、使用场景	[92]
	LSTM/GRU/CNN	可用性、合法性、用户观感、可维护性、可操作性、性能、可扩展性、安全性、易用性、容错性、可移植性	[93]

4 需求模型构建

需求模型构建是将抽象化的需求知识进行重组和结构化的过程.目前需求模型构建主要采用的2种自动化技术:基于规则的模型构建方法和基于机器学习/数据挖掘的模型构建方法.

4.1 基于规则的需求模型构建方法

基于规则的需求模型构建方法是较为早期使用的一种自动化模型构建方法,主要通过构建自定义的模型生成规则,帮助模型构建.

Arora 等人^[94]提出了一种整合已有文献中的提取规则方法,并对这些规则进行扩展后形成新的模型生成规则,自动化生成领域模型.利锦标等人^[28]采用自定义的语义框架,自动构建类图模型,并添加智能问卷系统,实现对模型的丰富和完善.Sagar 等人^[59]设计了类、关系、属性和操作相关的规则,根据已经提取的知识元素生成构建概念模型.针对目标模型和用例模型,Nguyen 等人^[95]开发了一种新颖的基于规则的方法,可以从自然语言需求文档中自动生成目标和用例模型,解决了人工识别目标和用例的问题.

研究人员也开发了一些支持工具对规则进行定义或封装. Deeptimahanti 等人^[96]开发了 UMGAR 工具,定义了一组语法重构规则,将复杂需求转换为

简单需求,同时生成 UML 用例模型和类模型等. Lucassen 等人^[97]开发了 Visual Narrator 工具,该工具对用户故事进行句法分析提取知识元素后,封装了 11 种启发式规则用于定义实体及其关系,可以自动化生成概念模型. Miranda 等人^[98]开发了 LUCAMTOOL 工具并定义了一种新的领域特定语言 LUCAM,规范了用例文本的描述格式,自动化生成用例图、类图和序列图.此外,TRAM 工具^[99]、RADAR 工具^[100]可以分别将文本需求转换成分析类模型和目标模型.这些工具的应用极大提高了需求模型构建的效率,但是在准确度上仍有待提高.

基于规则的需求模型构建方法在生成不同的需求模型时,大多需要人为设计术语与模型元素之间的映射关系和算法,严重受制于规则设计的完整性和准确性,且在应用于复杂场景时,效率和准确性有待提高.

4.2 基于机器学习/数据挖掘技术的模型构建方法

目前需求模型构建采用的机器学习或数据挖掘技术主要包括 SVM^[101]、贝叶斯网络^[102]、Markov 模型^[103]和聚类方法^[104].

1) SVM

Kumar 等人^[32]使用与线性、多项式和径向基函数内核函数相关联的最小二乘支持向量机 (least squares support vector machine, LSSVM) 学习方法构建可靠性模型,从而实现缺陷预测,节省维护

资源,提高了用户满意度.但该预测模型仅对故障类别小于阈值的项目表现较优.

2) Markov 模型

Markov 模型被广泛应用于非功能需求的预测模型中.基于 Markov 模型,Mathur 等人^[105]提出了一个基于分层队列的性能模型,准确预测了系统吞吐量的变化.Perez-Palacin 等人^[106]针对高可变和突发性的负载,对系统非功能需求进行自动化建模,构建出性能模型,准确模拟了工作负载的变化趋势.但该模型基于多维 Markov 链的求解,存在状态空间爆炸问题.

3) 贝叶斯网络

Si 等人^[30]针对网构软件的可靠性进行评估,提出了一种基于贝叶斯网络(Bayesian network, BN)的可信软件可靠性评估模型,分析了网构软件的结构,为其建立了可靠性评价系统.

4) 聚类方法

Chen 等人^[16]提出了一种基于互连性聚类算法的特征模型构建方法,分析了单独需求间的关系,并采用特征聚类的方法生成了特征模型,然而该方法仍需要人工干预.Masoud 等人^[107]提出了一种基于聚类的模型来解决类职责的分配问题,通过对收集到的数据进行特征提取,使用层次聚类方法对类的职责进行聚类,最后设置类之间的关联生成类图模型.该模型采用了基于图的交互式方法能够更好地显示需求的层次结构.

上述机器学习或数据挖掘技术帮助需求分析人员解决了需求模型构建过程中的冗余和低效问题,极大地提高了智能模型构建的效率.

表 5 对需求模型构建方法进行了汇总,主要包括使用的技术或方法、构建的需求模型和需求类型等方面.

Table 5 Intelligent Requirement Models Construction Approaches

表 5 智能需求模型构建方法

类别	技术/方法	需求模型	需求类型	文献
基于规则	语义框架	类图		[28]
	规则和依赖分析	概念模型		[59]
		领域模型		[94]
	GUEST	目标模型和用例模型		[95]
	UMGAR	用例模型和类模型	FR	[96]
	Visual Narrator	概念模型		[97]
	LUCAM	用例图、类图和顺序图		[98]
	TRAM	类模型		[99]
	RADAR	特征模型		[100]
		SLINK, CLINK, UPGMA	特征模型	
机器学习/数据挖掘	贝叶斯网络	可靠性模型		[30]
	SVM		NFR	[32]
	Markov 链	性能模型		[105]
	自定义聚类算法	类模型	FR	[107]

5 评估方法

在智能需求获取与建模的工作中,研究者需要对提出的模型进行案例研究和实验验证,并通过一系列不同的评估指标对模型的性能和有效性进行评估.

5.1 案例研究

案例研究(case study)是指将研究成果应用到

某一特定的个体、单位、案例或主题从而验证研究方法的过程.对于本文,案例研究是指针对智能需求获取与建模方法的偏差行为进行深入研究的过程,旨在了解这些方法的有效性.案例研究在目前的智能需求获取与建模评估方法中使用较多^[16,41-42,59-60].

Kumar 等人^[32]为了验证其构建的可靠性模型,在 30 个 Java 开源项目上进行实验,发现其预测模型的效果较为优异.Rago 等人^[48]为了验证其开发的 REAssistant 工具的语义分析性能,对 3 个需求规

范进行了案例研究.Cleland-Huang 等人^[80]为了评估非功能性需求的自动检测和分类方法,对西门子物流系统的需求文档进行了案例研究.Lucassen 等人^[97]为了验证其开发的工具,将其应用于现实项目中的 4 个数据集以评估方法的可行性和准确性.案例研究方法能够较好地证明智能需求获取与建模新方法在实际项目开发中的应用价值,但是却难以判断新方法相比其他方法的优越性.

5.2 实验验证方法

5.2.1 交叉验证方法

交叉验证是机器学习领域常用的模型验证方法^[25]. K 折交叉验证^[108]是交叉验证技术的基本形式,在此技术中,数据会首先被划分为 K 个子数据集. K 折交叉验证使得每一个样本数据都既能被用作训练数据,也能被用作测试数据,可以有效地避免过学习以及欠学习状态的发生,最后得到的结果也比较具有说服力,然而交叉验证方法的准确性和稳定性在很大程度上取决于 K 的取值大小,通常 K 取值 5, 10, 20 不等^[41].

贾一荻等人^[81]组合使用不同的文本特征和分类模型,并采用 5 折交叉验证的方法对组合进行测试,即将全部数据集平均划分为 5 份,每次取其中的 4 份用于训练,1 份用于测试,提高预测精度,减少预测偏差.Kumar 等人^[32]使用 20 折交叉验证比较模型,将数据集分为 20 个子集,使用准确性和 F -measure 指标比较缺陷预测模型的性能.

5.2.2 对比验证方法

对比验证方法是基于相同数据集,将新提出的方法(技术、算法、模型等)与其他已有方法的实验结果进行对比,以验证新方法的有效性.对比验证是基于机器学习和深度学习的智能需求提取和分类方法中最常用的验证方法^[26,41,75,92-93].例如 Riaz 等人^[83]为了验证其所提出的组合算法对安全需求的提取效果,将由朴素贝叶斯、SMO 和 KNN 组成的组合算法分别与这 3 种算法进行对比实验.通常,研究者们在做案例研究或者实验时会组合使用交叉验证和对比验证,以便更准确地评估新方法.

5.3 评估指标

为了量化评估智能需求获取与分类方法的有效性,研究者们提出了不同的评估指标.这一节将详细阐述在该领域研究中常用的评估指标,包括准确率($Precision$)、召回率($Recall$)、 F -measure、正确率($Accuracy$)等指标.这些指标主要被应用在智能需

求知识提取和分类方法的评估中^[59,65,80-81].下面以需求知识分类的应用场景为例,解释这些指标的定义及其计算方法.

准确率($Precision$),是指被正确检索的样本数与被检索到样本总数之比.具体计算为

$$Precision = \frac{TP}{TP + FP}, \quad (1)$$

其中, TP (true positive)表示被分类模型正确划分到该类别下的需求个数, FP (false positive)表示错误地被划分到该类别下的需求个数.

召回率($Recall$),是指被正确检索的样本数与应当被检索到的样本数之比:

$$Recall = \frac{TP}{TP + FN}, \quad (2)$$

其中, FN (false negative)表示本该被划分到该类别却没有被分类模型划分到该类别下的需求个数.

F -measure 是结合了准确率和召回率的综合性性能指标,是准确率和召回率的调和平均, β 是参数,其中 $F1$ -measure 是最常用的形式($\beta=1$):

$$F_{\beta}\text{-measure} = \frac{(\beta^2 + 1) \times Precision \times Recall}{\beta^2 Precision + Recall}. \quad (3)$$

正确率($Accuracy$),是指分类正确的样本数与样本总数之比.具体计算为

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}, \quad (4)$$

其中, TN (true negative)表示被分类模型正确地没有归类到此类别的需求个数.

特异度($Specificity$)是指所有负例(没有归类到此类别的需求)中被分对的比例:

$$Specificity = \frac{TN}{TN + FP}. \quad (5)$$

超规格度($Over-specification$)是指在实际得到的样本中自包含了不必要但是正确的样本数占应得到的样本数的比例.

$$Over-specification = \frac{N_{\text{extra}}}{TP + FN}, \quad (6)$$

其中, N_{extra} 指划分到该类别下正确却额外的需求数.

表 6 总结了检索到的论文中涉及评估指标的文献.表 7 选取了一些代表性工作,展示它们在各个评估指标上的具体表现,表 7 中展示的各个指标数据为文献所提方法在数据集上的平均结果或最好结果.表 7 中 CRS 表示休斯顿大学的课程注册系统;HWS 表示兰卡斯特大学的健康监护系统;MSLite 表示轻量级管理系统;DLibraCRM 表示一个用于在线分类和销售与书籍相关的商品网络系统;MobileNews

Table 6 Summary of Evaluation Metrics

表 6 评估指标总结

文献	Precision	Recall	F-measure	Accuracy	Over-specification	Specificity
[21,28,48,66,72,81,90]	✓	✓				
[80]	✓	✓				✓
[47,59,99]	✓	✓			✓	
[49,73]	✓	✓		✓		
[32]			✓	✓		
[41,45,74]				✓		
[107]			✓			
[26,62-65,70,75-78,84,86,89,91-92]	✓	✓	✓			
[83,88,93,95]	✓	✓	✓	✓		

注：“✓”表示文献采用了该指标来评估方法。

Table 7 Comparison of Various Indicators of Intelligent Requirements Elicitation and Modeling Methods

表 7 智能需求获取与建模方法的各项指标对比

类别	数据源	Precision/%	Recall/%	F-measure/%	Accuracy/%	Specificity/%	Over-specification/%	文献
	SemEval2014 数据集	89.81 ^①	86.00 ^①	87.86 ^①				[26]
	CRS, HWS, MSLite 数据集	70 ^②	80 ^②					[48]
	CRS, HWS 数据集	62.5 ^①	66.6 ^①		86.5 ^①			[49]
	DLibraCRM, MobileNews, WebJSARA, HWS 数据集	63 ^①	86 ^①					[66]
	用例描述	85 ^②	90 ^②				45 ^②	[47]
	多信息文本	100 ^①	100 ^①					[53]
需求知识 提取	场景				63 ^②			[55]
	文本约束描述	77 ^②	72 ^②	74 ^②				[63]
	自然语言需求文档	56 ^①	100 ^①	71 ^①				[65]
	工业界案例	98 ^①	43 ^①					[73]
	工业界需求文档				92.97 ^①			[74]
	工业界早期需求文本	91.07 ^②	84.72 ^②	86.60 ^②				[78]
	会议记录	70.8 ^①	18.3 ^①					[72]
	需求规约	88 ^①	48 ^①	62 ^①				[76]
	工业界需求文档	43.5 ^②	79.9 ^②			97.9 ^①		[80]
	实验数据集和工业数据集	70 ^②	90 ^②					[81]
	NFR 数据集	92 ^①	86 ^①	88 ^①				[84]
需求知识 分类					89.72 ^②			[45]
	应用评论	70 ^②	86 ^②	82 ^②				[89]
		74 ^②	59 ^②	64 ^②				[92]
	Reddit 论坛	78 ^①	94 ^①	85 ^①				[86]
	PROMISE 需求库	88 ^②	54 ^②	66 ^②	100 ^①			[88]
		71.7 ^②	71.5 ^②	70 ^②	71.5 ^②			[93]
	PROMISE 需求库			100 ^①	100 ^①			[32]
需求模型 构建	ATM, EFP 和课程注册 案例数据集	94.44 ^①	100 ^①				41.18 ^①	[59]
	案例和 PROMISE 需求库	86 ^②	86 ^②		88 ^②			[95]
	图书馆的场景描述	60 ^②	71 ^②			55 ^②		[99]

注：①表示所选数据集上的最好结果；②表示多个数据集上的平均结果。

表示一个用于将最新公告发布到移动设备的新闻提要系统;WebJSARA 表示具有发布、搜索、上传和下载功能的信息管理系统.在智能需求获取与建模过程中,为了能综合体现方法的有效性,研究者们通常会综合考虑多个指标.

6 智能需求获取与建模的科学问题与技术难点

结合人工智能技术的需求获取与建模近年来获得了大量研究者的关注,取得了很多突出的进展,但仍存在一些亟待解决的关键问题.本节将从科学问题、技术难点和工程实践 3 个方面阐述目前该领域存在的问题.

6.1 科学问题

6.1.1 需求模型的统一性和多样性

现有半自动化或者自动化需求建模方法大多只面向单一的模型,例如概念模型^[59]、过程模型^[63]、非功能需求模型^[75]等,对于构建统一的、多样化的功能需求模型、非功能需求模型以及它们之间的多元关系却鲜有研究.即使在构建某个单一模型时,其需求文档中的细化知识和隐性需求也很少被挖掘^[109],例如构建领域类模型时,领域类之间的聚合、组合和继承关系、关联的多重性以及类方法则很少涉及.另一方面,需求模型之间的关系复杂,包括不同类型的功能需求模型之间的关系(如领域类模型中的方法可能与目标模型的目标以及用例图里的用例直接相关)、功能需求与非功能需求之间的关系(如领域类模型的某个类、目标模型中软目标可能与性能模型中某性能指标直接相关).上述需求模型之间的内在联系在智能需求建模时极易被忽略.此外,中文需求与英文需求在自然语言处理上有很大的不同,然而,目前大多数需求建模技术尤其需求知识提取仍然针对的是英文文档和葡萄牙语文档^[73,82],对于采用中文描述的需求进行智能获取和建模的研究非常少^[81].因此,需要展开更全面、更深入的研究,构建统一的、多样化的需求模型及其多元关系.

6.1.2 需求模型的准确性评估

目前通过信息检索、机器学习或者深度学习等方法构建的需求模型,尚未有统一的、全面的指标评估其准确性.现有的评估方法包括案例研究^[48-49]、交叉验证^[81]和对比验证^[93],上述方法大多与人工构建的或已有的需求模型进行比对,最后采用准确率、召回率、 F -measure 等常规指标等进行评估.但这些

评估方法仍然存在的问题有:1)过于依赖人工经验,导致模型的准确性判断具有主观性^[68];2)无法采用统一的标准对不同的需求模型进行对比,如具有静态特征的目标模型需要比较目标的覆盖率^[17,95],但对于具有动态特征的过程模型和序列图^[19,41,63],除了比较模型元素的覆盖率之外,还需对模型元素之间的时序逻辑进行判断,这就导致不同类型的模型准确性评估方法大不相同.因此,需要展开关于不同类型需求模型的准确性评估方法研究.

6.1.3 智能需求获取与建模方法的实用性

通过信息检索、机器学习和深度学习等方法构建的需求模型中包含的需求知识在完整性上略显不足,通常需要人工对一些细节和隐性知识进行增加和修改,同时还要删除一些冗余的知识元素^[16].另一方面,如何提高智能需求获取与建模方法的效率也是一大难题^[28,85-86].提高智能需求获取与建模的效率应综合考虑需求知识提取、需求知识分类和需求模型构建的效率.RNN,CNN 和 LSTM 等深度学习技术的应用虽然能够提高模型构建的自动化程度和准确性,但也带来了性能的损耗^[84-85].上述原因都会导致研究工作中提出的方法或技术在应用到大规模的工程项目时出现实用性问题.因此,如何将现有大部分研究工作复制到工业背景下,是需要研究突破的方向.

6.2 技术难点

6.2.1 需求模型集成式和动态化构建

不同种类需求模型采用的建模语言所基于的语法和语义各不相同,单独构建某个需求模型时语法和语义并不会成为问题,但是在对不同需求模型之间的关系进行智能识别和建模时,不同的语法和语义则会阻碍建模过程.如何基于需求文档,采用机器学习或深度学习技术对不同需求模型之间的关系进行识别、分类和定义,是在进行智能需求建模时必须解决的难题.此外,对于具有动态特征需求模型的智能构建,例如过程模型和状态模型^[19,41,63],目前的研究工作有所缺失.其原因在于,相比静态模型通常只需提取对象、动作、对象之间的关系^[28,95,99],动态需求模型还需要提取动作之间的时序逻辑.该问题在单个句子的知识提取时较易解决,但是在提取跨句子的时序逻辑时会变得非常困难,尤其是面对没有任何格式限制的需求描述文本^[95].因此,构建集成式需求模型时既需要关注功能需求和非功能需求,也需要关注静态需求和动态需求.

当前很多软件都部署在开放的、异构的网络中,

如信息物理融合系统、自适应系统等,其系统所处的上下文环境会经常发生变化,需求模型在构建后,由于用户需求和上下文环境的变化会导致初始构建的需求模型并不能满足后续设计和开发的要求,因此如何进行动态化的需求建模^[105-106],根据上下文环境和需求文本的变更记录智能识别出需要修改的模型元素并完成需求模型的自动演化,也是当前迫切需要解决的问题。

6.2.2 其他软件工程活动关联

需求工程作为软件工程的起始阶段,其所包含的诸如需求获取、需求建模、需求分析等活动的主要目的是为后期软件工程活动(如设计、实现、追踪、测试)奠定基础,而目前智能需求获取与建模所涉及的方法大多只专注于需求工程^[23-26,38-39,43-44,46,48-49,53,55,59-61,64,66-67],对于与其他软件工程活动的关联缺少深入的挖掘和分析,这就导致提取和分类好的需求知识以及构建的需求模型与后续软件工程活动有所脱节,影响了智能需求建模相关技术在实际项目中的应用,如何深入挖掘智能需求获取与建模相关技术、方法、产出与其他软件工程活动如架构设计、详细设计、系统实现、代码追踪、自动化测试的关系,从而反向优化需求知识提取、需求知识分类和需求模型构建的方法和过程,是当前需要解决的关键问题。

6.2.3 需求知识分类的粒度问题

需求知识分类也是实际应用面临的一个技术难点,现有需求知识分类的粒度较粗,虽然能为需求优先级划分、需求管理提供帮助,却难以满足后期构建精准模型的需要,例如,目前在进行非功能需求知识分类时,大多只关注非功能需求类别(如性能、安全性、可靠性等)的识别,对于每种非功能需求分解后的属性或指标(如并发量、响应时间等)却没有形成更进一步的分类和映射,这就导致在进行多层次的需求建模(如目标模型、特征模型)时无法实现精化的过程。

6.2.4 需求数据集的保密性、完整性和规模问题

目前已公开的数据集包括 PROMISE 的软件需求库和分类库^[110]、SemEval-2014 的需求库^[111]、休斯顿大学的课程注册系统(CRS)需求文档集^[112]、兰卡斯特大学的健康监护系统(HWS)需求文档集^[113]等,如何全面地构建数据集是智能需求建模中的技术难点之一,原因在于:1)数据集不公开^[76,93],一方面,数据的保密性导致其他研究者无法直接获得数据集,限制了对于更复杂应用场景的实验;另一方面,数据的保密性^[72]也导致案例研究时所采用的数

据集缺乏代表性,故在方法的准确性和实用性验证上略显不足,2)数据集不完整^[64],通常表现在数据集只包含了需求描述,缺少与之对应的需求知识元素、需求分类类别和需求模型数据,例如在 CRS, HWS 和 SemEval-2014 需求库都存在类似的问题,导致研究人员在应用机器学习和深度学习技术时缺乏训练集,3)数据集规模有限^[26,34,80],Rahman 等人^[93]认为在实验中缺乏大型的公开数据集,限制了对更复杂场景的实验,大多数数据集的规模只适用于学术研究,极少能真正应用到工业案例上,很多涉及机器学习和深度学习技术的文献提到了现有数据集存在的上述问题,并希望在未来的工作中将提出的新技术新方法应用于更大规模的工业实践项目中。

6.2.5 需求模型评估指标的建立

评估指标的全面化和标准化对需求模型的智能构建有着至关重要的影响,评估指标的建立难点在于标准数据集的缺失和需求模型的多样性^[41,55,94,107],目前软件工程领域尚无专门针对需求建模的标准数据集,包括需求文本集和与之对应的需求模型集,标准训练集和测试集的缺失使得研究人员在评估自己的方法时,只能通过各自的渠道获取不同的数据集^[82-85],最后根据领域专家自建标准或其主观评价计算出评估结果^[62,97],从而导致评估结果的主观性加大,影响对于方法的客观判断,而需求模型的多样性^[41,94,97]则决定了无法采用单一的方法对所有类型的模型进行评估,因此,根据不同需求模型设计不同的模型评估算法,同时建立领域内的训练集和测试集,成为智能需求建模评估中的一大难题。

6.2.6 缺乏智能需求获取与建模工具支持

现有研究表明,智能需求工程的研究工作中大多缺乏可视化的工具支持^[26,41,76],已经开发出的工具包括 REAssistant^[48], EA-Analyzer^[74], Visual Narrator^[97]等,然而这些工具有些仅支持需求知识提取这一步骤,有些则支持的需求模型种类非常有限^[48,74],由于用户在描述需求时可能采用不同格式的描述方法,如用例^[49]、用户故事^[72]等,在设计和使用工具时考虑需求文档的异构性或系统的应用领域^[66]是必须要解决的问题,此外,如何设计出同时涵盖需求知识提取、需求知识分类和需求模型构建 3 阶段的工具集仍然需要进一步研究。

6.3 解决思路

针对 6.1 节和 6.2 节所总结的科学问题和技术难点,本节阐述这些问题的可能解决思路和未来发展趋势。

6.3.1 基于深度学习的需求模型构建研究

近年来,多项研究证明,深度学习技术在应用到自然语言处理和分类问题上可以取得更好的效果^[26,93]。在目前基于深度学习的自然语言处理技术中,与传统的神经网络 CNN, LSTM 不同的是, EMLO^[114], BERT^[27], XLNet^[115]等都是根据具体的上下文关系来判断当前单词或文本的语义。这些新兴的深度学习技术不仅可以在跨句子场景下学习关于时间声明的复杂约束,克服目前动态模型构建的困难,还可以较好地解决需求频繁变更的问题,即使变更的范围很小,上述技术仍然可以捕捉到其变动的语义信息,准确获取变更后的需求文本信息。对于上述已预训练好的模型,其本身就能较好地做到泛化性,能够适合不同语料库和不同语义的需求模型。对于深度学习中经常出现的数据不足问题,在有监督的学习方法中可以考虑使用数据增强的方法来扩展不足的数据。同时半监督学习也是一个很好的方法,因此使用新的神经网络模型并与已有的信息检索、文本分析技术进行结合将是智能需求建模的方向之一。对于需求工程领域训练集的小数据特征问题,可采用基于 transformer 的预训练模型, transformer 拥有更强的抽取特征能力,预训练模型可以利用大量的外部语料信息,使模型的效果更好。

6.3.2 智能需求工程对其他软件工程活动的影响研究

智能需求获取与建模方法和产出与其他软件工程活动紧密相关。需求知识提取和分类除了能为模型构建提供建模元素,还可为后续的代码追踪、代码补全、API 推荐等一系列软件开发活动提供帮助。例如,代码注释或者代码本身的语义信息非常丰富,通常表示了与之提供的功能、服务或与之对应的实体、对象等,因此,将现有的基于人工智能技术的代码追踪^[116]、代码补全^[117]、API 智能推荐技术^[118]与基于文本分析、机器学习、深度学习的需求知识提取和分类方法相结合,利用新的应用场景帮助选择和优化需求文本的特征,能够将智能需求获取与建模推向一个新的阶段。

此外,需求知识提取和分类可以提高大规模需求优先级排序的效果和效率。例如,从用户评论中提取的需求通过 TF-IDF, SVM, BiLSTM, CNN 等方法进行不同等级的重要性分类,再聚类,然后结合已有的推荐算法对需求优先级进行自动化排序,为实际项目开发中遇到的任务分解和任务排序提供新的解决思路。

通过智能方法构建的需求模型能够指导软件设计,而软件系统架构设计和详细设计也能对需求模型构建产生一定的影响。通过归纳总结多个领域不同类型的系统的常用设计风格、设计模式、设计策略和设计模型,建立系统设计相关数据集,采用机器学习或深度学习方法提取其对应的功能需求和非功能需求特征,指导需求建模阶段的模型选择,从而弥补智能需求工程与软件设计之间的鸿沟。

6.3.3 智能分类和模型构建的精细化方法研究

安全性需求是当前关注较多的非功能需求,专门针对安全性需求的智能提取、分类和模型构建已有一些研究。除安全性需求外,还有很多非功能需求需要得到进一步的挖掘,如性能、可靠性、易用性等。首先需要对已公开非功能需求的数据集进行改进。由研究人员通过调查问卷等方式调研实际项目中出现频率较高的非功能需求种类及其属性,参考安全性需求的已有研究成果,形成新的划分粒度更细的非功能需求数据集。然后采用目前效果较好的 SVM, LSTM, CNN 等对其进行分类,再结合能实现具有精化语义的建模语言如 NFR 框架、KAOS 等方法,实现非功能需求模型的自动化分解,并将非功能需求属性与这些建模语言中的元素进行映射,形成非功能需求知识智能提取、分类和建模过程的统一。

6.3.4 需求数据集构建方法研究

智能需求获取与建模中所使用的数据集被分为训练集和测试集,其中训练集应包含大量的采用自然语言描述的需求文档以及由领域专家已经获取并经过多人验证的需求知识元素、需求分类类别和需求模型,并且每个需求文档对应的模型类型应包含如特征模型、目标模型等多种模型。为更有效地进行智能建模,解决上述在需求建模过程中存在的问题迫在眉睫,可以考虑的解决思路有:1)针对目前数据集的不公开问题,对已经经过多次实验验证的开放数据集按照应用场景和领域进行归类、整理、扩展,分别建立有标注和无标注的训练集,以便研究者采用不同的机器学习和深度学习方法进行研究,包括有监督、无监督或半监督的学习方法,并在互联网上共享数据集,这能在一定程度上解决由于数据不公开导致的问题;2)针对数据集不完整的问题,为已公开的需求文本数据集构建相应需求模型、需求类别,形成标准的需求文本与需求模型、需求类别映射的数据集;3)针对数据集的规模问题,可通过学术界提供研究方法、工业提供数据集以供实践,两者展开合作,以扩大数据的实用性和方法的可信性。对数据

集构建的一系列处理,将会减小由数据集缺陷对实验造成的影响,进一步增强智能需求建模方法的可行性和可验证性。

6.3.5 基于图论的自动化模型评估方法研究

近年来,在图论领域,已有研究者提出自动化测试图形化模型准确性的算法^[119],首先将源模型和目标模型进行抽象,通过计算抽象后的源模型与目标模型的编辑距离,获得可被量化的准确性度量结果。这些图论算法可以为智能需求建模方法的有效性评估提供解决思路。同时,针对不同的需求模型,可为其构建不同的模型评估策略并设计相应的评估算法。在评估指标方面,综合已有的评估指标并进行细化,考虑需求建模的不同阶段,提出有针对性的分阶段的评估指标。

6.3.6 需求建模数据集的自动化标注方法研究

在实际项目中,面对大规模的工程数据,实现智能需求建模仍然面临许多技术难题,解决这些难题从4方面开展研究工作:1)研究大规模需求文本和需求类别、需求模型数据集的自动化标注技术,尽早实现大规模标准数据集的构建;2)引入更先进的机器学习和深度学习技术,设计并开发出适合在项目实践中所使用的智能需求获取与建模方法;3)建立智能需求获取与建模方法有效性以及需求模型准确性的评估方法,推进智能需求建模向更统一、更实用的方向发展;4)开发更先进、更快速的智能需求获取与建模工具,支持不同格式、不同语言的需求描述文本。

7 总 结

近年来,人工智能技术由于其广泛的应用性,在各个领域均取得了重大研究成果。本文围绕人工智能技术在需求获取与建模上的应用,采取综述的方式,从需求知识提取、需求知识分类、需求模型构建3个方面,梳理并总结了当前的研究进展。本文归纳了当前智能需求获取与建模的关键问题和技术难点及未来展望。根据系统的综述表明:1)智能需求获取与建模的相关研究数量正在增加;2)NLP是需求知识提取中常用到的技术;3)LDA、朴素贝叶斯、SVM、CNN、LSTM、GRU等智能技术已被广泛应用于需求知识分类上,并取得显著进展^[120];4)在需求模型构建方面,目前大多数方法仍然集中在传统的基于规则的模型生成方法和基于机器学习的方法,研究者们尚未探索将更先进的深度学习技术应用到模型

的自动化生成上;5)大多数需求知识提取和需求建模研究中关注的是功能需求。随着应用技术的增加,越来越多的文献开始研究非功能需求的挖掘和分类问题。

参 考 文 献

- [1] Harman M. The role of artificial intelligence in software engineering [C] //Proc of the 1st Int Workshop on Realizing AI Synergies in Software Engineering. Piscataway, NJ: IEEE, 2012: 1-6
- [2] Jia Runying, Li Jing, Wang Gang, et al. Optimization and choice of hard drive failure prediction models based on Adaboost and genetic algorithm [J]. Journal of Computer Research and Development, 2014, (Suppl1): 148-154 (in Chinese)
(贾润莹, 李静, 王刚, 等. 基于 Adaboost 和遗传算法的硬盘故障预测模型优化及选择 [J]. 计算机研究与发展, 2014, (增刊1): 148-154)
- [3] Feldt R, de Oliveira Neto F G, Torkar R. Ways of applying artificial intelligence in software engineering [C] //Proc of the 6th Int Workshop on Realizing Artificial Intelligence Synergies in Software Engineering. Piscataway, NJ: IEEE, 2018: 35-41
- [4] Xie Tao. Intelligent software engineering: Synergy between AI and software engineering [C] //Proc of the 4th Int Symp on Dependable Software Engineering: Theories, Tools, and Applications. Berlin: Springer, 2018: 3-7
- [5] Xie Tao. Intelligent software engineering: Synergy between AI and software engineering [C] //Proc of the 11th Innovations in Software Engineering Conf. New York: ACM, 2018: 1-1
- [6] Pohl K. Requirements Engineering Fundamentals: A Study Guide for the Certified [M]. Santa Barbara: Rocky Nook, 2015
- [7] Nuseibeh B, Easterbrook S. Requirements engineering: A roadmap [C] //Proc of the Conf on the Future of Software Engineering. New York: ACM, 2000: 35-46
- [8] Maiden N, Ncube C. Acquiring requirements for commercial Off-The-Shelf package selection [J]. IEEE Software, 1998, 15(2): 46-56
- [9] Davis A. Operational prototyping: A new development approach [J]. Software, 1992, 9(5): 70-78
- [10] Maiden N. CREWS-SAVRE: Scenarios for acquiring and validating requirements [J]. Automated Software Engineering, 1996, 11(3): 419-446
- [11] Loucopoulos P, Kavakli E. Enterprise modelling and the teleological approach to requirements engineering [J]. The International Journal of Intelligent and Cooperative Information Systems, 1995, 4(1): 45-79

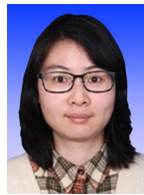
- [12] Carte A S, Hundhausen C D, Adesope O. The normalized programming state model: Predicting student performance in computing courses based on programming behavior [C] // Proc of the 11th Annual Int Conf on Int Computing Education Research, New York: ACM, 2015: 141-150
- [13] Chung L, Nixon B A, Yu E, et al. Non-Functional Requirements in Software Engineering [M]. Boston: Kluwer Academic Publishers, 2000
- [14] Lian Xiaoli, Rahimi M, Cleland-Huang J, et al. Mining requirements knowledge from collections of domain documents [C] // Proc of the 24th Int Requirements Engineering Conf. Piscataway, NJ: IEEE, 2016: 156-165
- [15] El-Attar M. Evaluating and empirically improving the visual syntax of use case diagrams [J]. Journal of Systems and Software, 2019, 156(6): 136-163
- [16] Chen Kun, Zhang Wei, Zhao Haiyan, et al. An approach to constructing feature models based on requirements clustering [C] // Proc of the 13th IEEE Int Conf on Requirements Engineering. Piscataway, NJ: IEEE, 2005: 31-40
- [17] Abrahão S, Insfran E, González-Ladrón-de-Guevara F, et al. Assessing the effectiveness of goal-oriented modeling languages: A family of experiments [J]. Information and Software Technology, 2019, 116(8): 106-171
- [18] Jin Zhi, Liu Lin. Towards automatic problem decomposition: An ontology-based approach [C] // Proc of the 2006 Int Workshop on Advances and Applications of Problem Frames. New York: ACM, 2006: 41-48
- [19] Zhang Xuan, Wang Xu, Li Tong, et al. Correctness control and detection of aspect-oriented business process modeling [J]. Chinese Journal of Computers, 2018, 41(3): 521-544 (in Chinese)
(张璇, 王旭, 李彤, 等. 面向方面业务过程建模的正确性控制与检测[J]. 计算机学报, 2018, 41(3): 521-544)
- [20] Neill C J, Laplante P A. Requirements engineering: The state of the practice [J]. IEEE, 2003, 20(6): 40-45
- [21] Cruz B D, Jayaraman B, Dwarakanath A, et al. Detecting vague words & phrases in requirements documents in a multilingual environment [C] // Proc of the 25th Int Requirements Engineering Conf. Piscataway, NJ: IEEE, 2017: 233-242
- [22] Kassab M, Neill C, Laplante P. State of practice in requirements engineering: Contemporary data [J]. Innovations in System and Software Engineering, 2014, 10(4): 235-241
- [23] Ferrari A, Spagnolo G O, Dell'Orletta F. Mining commonalities and variabilities from natural language documents [C] // Proc of the 17th Int Software Product Line Conf. New York: ACM, 2013: 116
- [24] Boutkova E, Houdek F. Semi-automatic identification of features in requirement specifications [C] // Proc of the 19th Int Requirements Engineering Conf. Piscataway, NJ: IEEE, 2011: 313-318
- [25] Mitchell T, Buchanan B, Dejong G, et al. Machine learning [J]. Annual Review of Computer Science, 1990, 4(1): 417-433
- [26] Saraiva F Z R, Silva T L C D, Macêdo J A F D. Aspect term extraction using deep learning model with minimal feature engineering [C] // Proc of the Int Conf on Advanced Information Systems Engineering. Berlin: Springer, 2020: 185-198
- [27] Lin Zhouhan, Feng Minwei, Santos C N D, et al. A structured self-attentive sentence embedding [EB/OL]. (2017-03-09) [2020-07-15]. <https://arxiv.org/abs/1703.03130>
- [28] Li Jinbiao, Li Tong, Liu Lin. Chinese requirements analysis method based on class diagram semantic framework [J]. Acta Electronica Sinica, 2011, 39(Suppl1): 94-98 (in Chinese)
(利锦标, 李童, 刘琳. 基于类图语义框架的中文需求分析方法[J]. 电子学报, 2011, 39(增刊1): 94-98)
- [29] Maalej W, Nabil H. Bug report, feature request, or simply praise? On automatically classifying app reviews [C] // Proc of the 23rd Int Requirements Engineering Conf. Piscataway, NJ: IEEE, 2015: 116-125
- [30] Si Guannan, Xu Jing, Yang Jufeng, et al. An evaluation model for dependability of Internet-scale software on basis of Bayesian networks and trustworthiness [J]. Journal of Systems and Software, 2014, 89(3): 63-75
- [31] Woodcock H. The next step in requirements management: Artificial intelligence [C] // Proc of the 27th Int Requirements Engineering Conf. Piscataway, NJ: IEEE, 2019: 5-5
- [32] Kumar L, Sripada S, Sureka A, et al. Effective fault prediction model developed using least square support vector machine (LSSVM) [J]. Journal of Systems and Software, 2018, 137(3): 686-712
- [33] Yang Zhuoqun, Li Zhi, Jin Zhi, et al. A systematic literature review of requirements modeling and analysis for self-adaptive systems [C] // Proc of Int Working Conf on Requirements Engineering; Foundation for Software Quality. Berlin: Springer, 2014: 55-71
- [34] Li Yuanbang, Peng Rong, Ji Jingjing, et al. Systematic literature review of empirical studies on context aware requirement elicitation and modeling [J]. Journal of Software, 2018, 29(2): 320-339 (in Chinese)
(栗元邦, 彭蓉, 季晶晶, 等. 经验研究中情景感知需求获取与建模系统文献综述[J]. 软件学报, 2018, 29(2): 320-339)
- [35] Perkusich M, Silva L C E, Costa A, et al. Intelligent software engineering in the context of agile software development: A systematic literature review [J]. Information and Software Technology, 2020, 119(2020): 1-75
- [36] Gea J M C D, Nicolás J, Alemán J L F, et al. Requirements engineering tools: Capabilities, survey and assessment [J]. Information and Software Technology, 2012, 54(10): 1142-1157
- [37] Kitchenham B A, Pretorius R, Budgen D, et al. Systematic literature reviews in software engineering—A tertiary study [J]. Information and Software Technology, 2010, 52(8): 792-805
- [38] Dalpiaz F, Dell'Anna D, Aydemir F B, et al. Requirements classification with interpretable machine learning and dependency parsing [C] // Proc of the 27th Int Requirements Engineering Conf. Piscataway, NJ: IEEE, 2019: 142-152

- [39] Parra E, Dimou C, Llorens J, et al. A methodology for the classification of quality of requirements using machine learning techniques [J]. *Information and Software Technology*, 2015, 67(11): 180-195
- [40] Zibran M F. On the effectiveness of labeled latent Dirichlet allocation in automatic bug-report categorization [C] //Proc of the 38th Int Conf on Software Engineering Companion. New York: ACM, 2016: 713-715
- [41] Qian Chen, Wen Lijie, Kumar A, et al. An approach for process model extraction by multi-grained text classification [C] //Proc of the Int Conf on Advanced Information Systems Engineering. Berlin: Springer, 2020: 268-282
- [42] Denger C, Berry D M, Kamsties E. Higher quality requirements specifications through natural language patterns [C] //Proc of the 2003 Symp on Security and Privacy. Piscataway, NJ: IEEE, 2003: 80-90
- [43] Niu N, Easterbrook S. Extracting and modeling product line functional requirements [C] //Proc of the 16th IEEE Int Requirements Engineering Conf. Piscataway, NJ: IEEE, 2008: 155-164
- [44] Bakar N H, Kasirun Z M, Salleh N. Feature extraction approaches from natural language requirements for reuse in software product lines: A systematic literature review [J]. *Journal of Systems and Software*, 2015, 106(8): 132-149
- [45] Vu P M, Nguyen T T, Pham H V. Mining user opinions in mobile app reviews: A keyword-based approach (T) [C] //Proc of the 30th IEEE/ACM Int Conf on Automated Software Engineering. Piscataway, NJ: IEEE, 2015: 749-759
- [46] Hariri N, Castro-Herrera C, Mirakhorli M, et al. Supporting domain analysis through mining and recommending features from online product listings [J]. *IEEE Transactions on Software Engineering*, 2013, 39(12): 1736-1752
- [47] Elbendak M, Vickers P, Rossiter N. Parsed use case descriptions as a basis for object-oriented class model generation [J]. *Journal of Systems and Software*, 2011, 84(2): 1209-1223
- [48] Rago, A, Marcos C A, Diaz-Pace J A. Assisting requirements analysts to find latent concerns with REAssistant [J]. *Automated Software Engineering*, 2016, 23(2): 219-252
- [49] Rago A, Marcos C A, Diaz-Pace A. Uncovering quality-attribute concerns in use case specifications via early aspect mining [J]. *Requirements Engineering*, 2013, 18(1): 67-84
- [50] Wu Zhibiao, Palmer M. Verbs semantics and lexical selection [C] //Proc of the 32nd Annual Meeting on Association for Computational Linguistics. New York: ACM, 1994: 133-138
- [51] Lin Dekang. An information-theoretic definition of similarity [C] //Proc of the 15th Int Conf on Machine Learning. New York: ACM, 1998: 296-304
- [52] Angell R C, Freund G E, Willett P. Automatic spelling correction using a trigram similarity measure [J]. *Information Processing and Management*, 1983, 19(4): 255-261
- [53] Rauf R, Antkiewicz M, Czarnecki K. Logical structure extraction from software requirements documents [C] //Proc of the 19th Int Requirements Engineering Conf. Piscataway, NJ: IEEE, 2011: 101-110
- [54] Vu P M, Pham V H, Nguyen T T, et al. Phrase-based extraction of user opinions in mobile app reviews [C] //Proc of the 31st IEEE/ACM Int Conf Automated Software Engineering. New York: ACM, 2016: 726-731
- [55] Ferrari A, Esuli A. An NLP approach for cross-domain ambiguity detection in requirements engineering [J]. *Automated Software Engineering*, 2019, 26(9): 559-598
- [56] Stanford NLP Natural Language Processing Group. Stanford NLP [EB/OL]. 2004 [2020-07-13]. <http://nlp.stanford.edu/software/index.html>
- [57] The University of Sheffield. GATE: General architecture for text engineering [EB/OL]. 1995 [2020-07-13]. <https://gate.ac.uk/sale/tao/splitap7.html>
- [58] Apache. Open NLP [EB/OL]. (2010-09-23) [2020-07-13]. <http://opennlp.sourceforge.net/projects.html>
- [59] Sagar V B R V, Abirami S. Conceptual modeling of natural language functional requirements [J]. *Journal of Systems and Software*, 2014, 88(2): 25-41
- [60] Thakur J S, Gupta A. AnModeler: A tool for generating domain models from textual specifications [C] //Proc of the 31st IEEE/ACM Int Conf on Automated Software Engineering. New York: ACM, 2016: 828-833
- [61] Thakur J S, Gupta A. Identifying domain elements from textual specifications [C] //Proc of the 31st IEEE/ACM Int Conf on Automated Software Engineering. New York: ACM, 2016: 566-577
- [62] Quirchmayr T, Paech B, Kohl R, et al. Semi-automatic rule-based domain terminology and software feature-relevant information extraction from natural language user manuals [J]. *Empirical Software Engineering*, 2018, 23(12): 3630-3683
- [63] Han V D A, Ciccio C D, Leopold H, et al. Extracting declarative process models from natural language [C] //Proc of the Int Conf on Advanced Information Systems Engineering. Berlin: Springer, 2019: 365-382
- [64] Morales-Ramirez I, Kifetew F M, Perini A. Analysis of online discussions in support of requirements discovery [C] //Proc of the Int Conf on Advanced Information Systems Engineering. Berlin: Springer, 2017: 159-174
- [65] AlHroob A, Imam A T, Al-Heisa R. The use of artificial neural networks for extracting actions and actors from requirements document [J]. *Information and Software Technology*, 2018, 101(4): 1-15
- [66] Rago A, Marcos A C, Diaz-Pace J A. Identifying duplicate functionality in textual use cases by aligning semantic actions [J]. *Software and Systems Modeling*, 2016, 15(8): 579-603
- [67] Wang Chunhui, Jin Zhi, Zhao Haiyan, et al. Human-assisted elicitation and evolution of user stories with scenarios [J]. *Journal of Software*, 2019, 30(10): 3186-3205 (in Chinese)

- (王春晖, 金芝, 赵海燕, 等. 人机协作的用户故事场景提取与迭代演进[J]. 软件学报, 2019, 30(10): 3186-3205)
- [68] del Águila I M, del Sagrado J. Bayesian networks for enhancement of requirements engineering: A literature review [J]. *Requirements Engineering*, 2016, 21(4): 461-480
- [69] Bhowmik T, Niu Nan, Savolainen J, et al. Leveraging topic modeling and part-of-speech tagging to support combinational creativity in requirements engineering [J]. *Requirements Engineering*, 2015, 20(3): 253-280
- [70] Chen Ning, Lin Jialiu, Hoi S C H, et al. AR-miner: Mining informative reviews for developers from mobile app marketplace [C] //Proc of the 36th Int Conf on Software Engineering. New York: ACM, 2014: 767-778
- [71] Li Yang, Guzman E, Tsiamoura K, et al. Automated requirements extraction for scientific software [J]. *Journal of Systems and Software*, 2015, 51(5): 582-591
- [72] Rodeghero P, Jiang Siyuan, Armaly A, et al. Detecting user story information in developer-client conversations to generate extractive summaries [C] //Proc of the 39th Int Conf on Software Engineering. Piscataway, NJ: IEEE, 2017: 49-59
- [73] Arora C, Sabetzadeh M, Nejati S, et al. An active learning approach for improving the accuracy of automated domain model extraction [J]. *ACM Transactions on Software Engineering and Methodology*, 2019, 28(1): 1-34
- [74] Sardinha A, Chitchyan R, Weston N P, et al. EA-Analyzer: Automating conflict detection in aspect-oriented requirements [C] //Proc of the 2009 IEEE/ACM Int Conf on Automated Software Engineering. Piscataway, NJ: IEEE, 2009: 530-534
- [75] Schneider K, Knauss E, Houmb S H, et al. Enhancing security requirements engineering by organizational learning [J]. *Requirements Engineering*, 2012, 17(1): 35-56
- [76] Pudlitz F, Brokhausen F, Vogelsang A. Extraction of system states from natural language requirements [C] //Proc of the 27th Int Requirements Engineering Conf. Piscataway, NJ: IEEE, 2019: 211-222
- [77] Liu Wei, Yan Hualiang, Xiao Jianguo, et al. Solution for automatic Web review extraction [J]. *Journal of Software*, 2010, 21(12): 3220-3236 (in Chinese)
(刘伟, 严华梁, 肖建国, 等. 一种 Web 评论自动抽取方法 [J]. 软件学报, 2010, 21(12): 3220-3236)
- [78] Li Tianying, Liu Lin, Zhao Dewang, et al. Eliciting relations from requirements text based on dependency analysis [J]. *Chinese Journal of Computers*, 2013, 36(1): 54-62 (in Chinese)
(李天颖, 刘麟, 赵德旺, 等. 一种基于依存文法的需求文本策略依赖关系抽取方法 [J]. 计算机学报, 2013, 36(1): 54-62)
- [79] Hu Tianyuan, Jiang Ying. Mining of user's comments reflecting usage feedback for app software [J]. *Journal of Software*, 2019, 30(10): 3168-3185 (in Chinese)
- (胡甜媛, 姜琪. 体现使用反馈的 APP 软件用户评论挖掘 [J]. 软件学报, 2019, 30(10): 3168-3185)
- [80] Cleland-Huang J, Settini R, Zou Xuchang, et al. The detection and classification of non-functional requirements with application to early aspects [C] //Proc of the 14th IEEE Int Requirements Engineering Conf. Piscataway, NJ: IEEE, 2006: 39-48
- [81] Jia Yidi, Liu Lin. Research on recognition and classification of Chinese non-functional requirements description [J]. *Journal of Software*, 2019, 30(10): 3115-3126 (in Chinese)
(贾一荻, 刘麟. 中文非功能需求描述的识别与分类方法研究 [J]. 软件学报, 2019, 30(10): 3115-3126)
- [82] Souza E, Vitória D, Castro D, et al. Characterizing opinion mining: A systematic mapping study of the Portuguese language [C] //Proc of the Computational Processing of the Portuguese Language. Berlin: Springer, 2016: 122-127
- [83] Riaz M, King J, Slankas J, et al. Hidden in plain sight: Automatically identifying security requirements from natural language artifacts [C] //Proc of the 22nd Int Requirements Engineering Conf. Piscataway, NJ: IEEE, 2014: 183-192
- [84] Kurtanović Z, Maalej W. Automatically classifying functional and non-functional requirements using supervised machine learning [C] //Proc of the 25th Int Requirements Engineering Conf. Piscataway, NJ: IEEE, 2017: 490-495
- [85] Li Chuanyi, Huang Liguo, Ge Jidong, et al. Automatically classifying user requests in crowdsourcing requirements engineering [J]. *Journal of Systems and Software*, 2018, 138(12): 108-123
- [86] Khan J A, Liu Lin, Wen Lijie. Requirements knowledge acquisition from online user forums [J]. *IET Software*, 2020, 14(3): 242-253
- [87] Chen Qi, Zhang Li, Jiang Jing, et al. Review analysis method based on support vector machine and latent Dirichlet allocation [J]. *Journal of Software*, 2019, 30(5): 1547-1560 (in Chinese)
(陈琪, 张莉, 蒋竞, 等. 一种基于支持向量机和主题模型的评论分析方法 [J]. 软件学报, 2019, 30(5): 1547-1560)
- [88] Casamayor A, Godoy D, Campo M R. Identification of non-functional requirements in textual specifications: A semi-supervised learning approach [J]. *Information and Software Technology*, 2010, 52(4): 436-445
- [89] Jha N, Mahmoud A. Mining non-functional requirements from app store reviews [J]. *Empirical Software Engineering*, 2019, 24(12): 3659-3695
- [90] Mahmoud A, Williams G. Detecting, classifying, and tracing non-functional software requirements [J]. *Requirements Engineering*, 2016, 21(3): 357-381
- [91] Maalej W, Kurtanovic Z, Nabil H, et al. On the automatic classification of app reviews [J]. *Requirements Engineering*, 2016, 21(3): 311-331

- [92] Guzman E, El-Haliby M, Bruegge B. Ensemble methods for app review classification: An approach for software evolution (N)[C] //Proc of the 30th IEEE/ACM Int Conf on Automated Software Engineering. Piscataway, NJ: IEEE, 2015: 771-776
- [93] Rahman Md A, Haque Md Al, Tawhid Md N A, et al. Classifying non-functional requirements using RNN variants for quality software development [C] //Proc of the 3rd ACM SIGSOFT Int Workshop on Machine Learning Techniques for Software Quality Evaluation. New York: ACM, 2019: 25-30
- [94] Arora C, Sabetzadeh M, Briand L C, et al. Extracting domain models from natural-language requirements: Approach and industrial evaluation [C] //Proc of the 19th Int Conf on Model Driven Engineering Languages and Systems. New York: ACM, 2016: 250-260
- [95] Nguyen H T, Grundy J C, Mohamed A. Rule-based extraction of goal-use case models from text [C] //Proc of the 10th Joint Meeting on Foundations of Software Engineering. New York: ACM, 2015: 591-601
- [96] Deeptimahanti D K, Ali Babar M. An automated tool for generating UML models from natural language requirements [C] //Proc of the 24th IEEE/ACM Int Conf on Automated Software Engineering. Piscataway, NJ: IEEE, 2009: 680-682
- [97] Lucassen G, Robeer M, Dalpiaz F, et al. Extracting conceptual models from user stories with visual narrator [J]. Requirements Engineering, 2017, 22(3): 339-358
- [98] Miranda M A, Ribeiro M G, Marques-Neto H T, et al. Domain-specific language for automatic generation of UML models [J]. IET Software, 2018, 12(2): 129-135
- [99] Letsholo K, Zhao Liping, Chioasca E. TRAM: A tool for transforming textual requirements into analysis models [C] //Proc of the 28th IEEE/ACM Int Conf on Automated Software Engineering Silicon Valley. Piscataway, NJ: IEEE, 2013: 738-741
- [100] Busari S A. Towards search-based modelling and analysis of requirements and architecture decisions [C] //Proc of the 32nd IEEE/ACM Int Conf on Automated Software Engineering. Piscataway, NJ: IEEE, 2017: 1026-1029
- [101] Geng Xiang, Gu Bin, Li Xiang, et al. Scalable semi-supervised SVM via triply stochastic gradients [C] //Proc of the 28th Int Joint Conf on Artificial Intelligence. New York: ACM, 2019: 2364-2370
- [102] Mandel T, Liu Yunen, Brunskill E, et al. Efficient Bayesian clustering for reinforcement learning [C] //Proc of the Int Joint Conf on Artificial Intelligence. New York: ACM, 2016: 1830-1838
- [103] Nguyen T T, Pham H V, Vu P M, et al. Recommending API usages for mobile apps with hidden Markov model [C] //Proc of the 30th IEEE/ACM Int Conf on Automated Software Engineering. Piscataway, NJ: IEEE, 2015: 795-800
- [104] Liu Ming, Chen Lei, Liu Bingquan, et al. VRCA: A clustering algorithm for massive amount of texts [C] //Proc of the Int Joint Conf on Artificial Intelligence. New York: ACM, 2015: 2355-2361
- [105] Mathur V, Apte V. An overhead and resource contention aware analytical model for overloaded Web servers [J]. Journal of Systems and Software, 2009, 82(1): 39-55
- [106] Perez-Palacin D, Mirandola R, Merseguer J. Accurate modeling and efficient QoS analysis of scalable adaptive systems under bursty workload [J]. Journal of Systems and Software, 2017, 130(5): 24-41
- [107] Masoud H, Jalili S. A clustering-based model for class responsibility assignment problem in object-oriented analysis [J]. Journal of Systems and Software, 2014, 93(7): 110-131
- [108] Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection [C] //Proc of the Int Joint Conf on Artificial Intelligence. New York: ACM, 1995: 1137-1145
- [109] Sharma V S, Ramnani R R, Sengupta S. A framework for identifying and analyzing non-functional requirements from text [C] //Proc of the 4th Int Workshop on the Twin Peaks of Requirements and Architecture. New York: ACM, 2014: 1-8
- [110] DePaul University. Openscience tera-promise software requirement [EB/OL]. 2004 [2020-07-13]. <https://terapromise.csc.ncsu.edu/#!/#repo/view/head/requirements/nfr>
- [111] Athens University of Economics and Business, Athena Research Center, University of York, et al. SemEval-2014 [EB/OL]. (2014-12-05) [2020-07-13]. <http://alt.qcri.org/semeval2014/task4/index.php?id=data-and-tools>
- [112] University of Houston. CRS: Course registration system [EB/OL]. 1999 [2020-07-13]. https://sce.uhcl.edu/helm/RUP_course_example/courseregistrationproject/
- [113] Lancaster University. HWS: Health monitoring system [EB/OL]. 1986 [2020-07-13]. http://www.comp.lancs.ac.uk/*greenwop/tao/
- [114] Peters M, Neumann M, Iyyer M, et al. Deep contextualized word representations [C] //Proc of the 2018 Conf of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Stroudsburg, PA: Association for Computational Linguistics, 2018: 2227-2237
- [115] Yang Zhilin, Dai Zihang, Yang Yiming, et al. XLNet: Generalized autoregressive pretraining for language understanding [C] //Proc of the Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 2019: 5754-5764
- [116] Guo J, Monaikul N, Cleland-Huang J. Trace links explained: An automated approach for generating rationales [C] //Proc of the 23rd Int Requirements Engineering Conf. Piscataway, NJ: IEEE, 2015: 202-207

- [117] Hellendoorn V J, Proksch S, Gall H C, et al. When code completion fails: A case study on real-world completions [C] //Proc of the 41st Int Conf on Software Engineering. Piscataway, NJ: IEEE, 2019: 960-970
- [118] Cai Liang, Wang Haoye, Huang Qiao, et al. BIKER: A tool for bi-information source based API method recommendation [C] //Proc of the 27th ACM Joint Meeting on European Software Engineering Conf and Symp on the Foundations of Software Engineering. Piscataway, NJ: IEEE, 2019: 1075-1079
- [119] Habib A, Pradel M. Is this class thread-safe? inferring documentation using graph-based learning [C] //Proc of the 33rd ACM/IEEE Int Conf on Automated Software Engineering. New York: ACM, 2018: 41-52
- [120] Abad Z S H, Karras O, Ghazi P, et al. What works better? A study of classifying requirements [C] //Proc of the 25th Int Requirements Engineering Conf. Piscataway, NJ: IEEE, 2017: 496-501



Wang Ye, born in 1986. PhD, associate professor, master supervisor. Member of CCF. Her main research interests include software engineering, machine learning, and service computing.

汪 焯, 1986 年生. 博士, 副教授, 硕士生导师, CCF 会员. 主要研究方向为软件工程、机器学习和服务计算等.



Chen Junwu, born in 1995. Master candidate. Student member of CCF. His main research interests include software engineering, machine learning, and service computing.

陈骏武, 1995 年生. 硕士研究生, CCF 学生会会员. 主要研究方向为软件工程、机器学习和服务计算等.



Xia Xin, born in 1986. PhD. Lecturer. ARC DECRA Fellow. His main research interests include data mining, software engineering, and machine learning.

夏 鑫, 1986 年生. 博士, 讲师, ARC DECRA 研究员. 主要研究方向为数据挖掘、软件工程和机器学习等.



Jiang Bo, born in 1970. PhD, professor. Member of the IEEE, ACM and CCF. Her main research interests include software engineering, machine learning, and service computing.

姜 波, 1970 年生. 博士, 教授. IEEE, ACM, CCF 会员. 主要研究方向为软件工程、机器学习和服务计算等.