DOI:10.7544/issn1000-1239.2021.20200756
58(4):763-776,2021

基于版本更新日志的移动应用演化趋势自动分析

钟仁毅[1] 王 翀[1] 梁 鹏[1] 罗 忠[2]
[1](武汉大学计算机学院 武汉 430072)
[2](海军工程大学舰船与海洋学院 武汉 430033)
(xiyezry@whu.edu.cn)

Automatic Trend Analysis of Mobile App Updates Based on App Changelogs

Zhong Renyi[1], Wang Chong[1], Liang Peng[1], and Luo Zhong[2]
[1](*School of Computer Science, Wuhan University, Wuhan* 430072)
[2](*Department of Naval Architecture Engineering, Naval University of Engineering, Wuhan* 430033)

Abstract Data-driven analysis on the development, maintenance, and evolution has recently become an area of active research. However, little is known to treat app changelogs as the input to explore the types of requirements that app developers pay the most attention when releasing an app, as well as trend of app development and updates. This paper reports the results of an exploratory study in which we analyze the requirements and buzzwords that dominate the changes of apps, according to a set of 6527 changes collected from 60 apps from three categories in the Apple App Store: "Travel", "Social Networking" and "Books". First, the performance of three supervised machine learning algorithms is evaluated to find the most suitable classifiers for the automatic classification of app changelogs. Furthermore, based on the classification results of app changelogs, characteristics and trends of app updates are revealed from two perspectives, i.e., the requirement type that app changelog items mention and the hot words in app changelog items that are labeled as a certain requirement type. The results are valuable for researchers and practitioners to have a comprehensive understanding on the current app stores from RE perspective.

Key words requirements engineering; non-functional requirement; release planning; changelogs; app store; empirical study

摘 要 数据驱动的移动应用开发、维护和演化分析正成为移动应用领域的研究热点.然而,鲜少有研究以移动应用的版本更新记录为对象,从需求类型的角度探索开发者在发布移动应用时的偏好以及移动应用的开发和更新趋势.为此,以苹果 App Store 中社交、旅游和阅读 3 种类别 60 个应用的 6 527 条版本更新记录条目为数据集,验证并评估了监督式机器学习算法对移动应用版本更新记录自动分类的可行性和有效性;进一步,基于最优的监督式机器学习算法对版本更新记录自动分类的结果,从需求类型和更新热点 2 个方面对移动应用的演化特点进行了分析,展示了苹果 App Store 中 3 种类别的移动应用在近 5 年的更新趋势,以帮助研究者与实践者从需求工程的角度了解当前移动应用市场的现状和变化动态.

收稿日期:2020-09-16;**修回日期**:2020-12-21
基金项目:国家重点研发计划项目(2018YFB1003800);国家自然科学基金项目(61702378,61972292,62032016)
This work was supported by the National Key Research and Development Program of China (2018YFB1003800) and the National Natural Science Foundation of China (61702378, 61972292, 62032016).
通信作者:王翀(cwang@whu.edu.cn)

关键词 需求工程;非功能需求;发布计划;更新日志;移动应用市场;实证研究

中图法分类号 TP311.5

近年来,移动应用程序数量的增长越来越快.截至2020年第1季度,Android应用商店Google Play为用户提供的移动应用数量已达256万个,苹果应用商店Apple App Store也为iOS用户提供了185万个应用供其下载和使用[1].为了满足用户不断变化的新需求并在激烈的市场竞争中保持优势,无论新开发的移动应用还是已有移动应用的新版本,其发布数量均在不断增长,为开发者及时了解移动应用的重要更新、探索移动应用的演化趋势以及把握移动应用的市场动态带来了挑战.

移动应用的版本更新日志是由移动应用开发者在移动应用商店中以标准格式提供的短文本,其发布时间和周期与移动应用的新版本完全一致,描述了移动应用当前版本的主要或重要更新,表征了移动应用演化的部分特点.目前,许多研究者已将移动应用的用户评论[2-4]或不同版本的移动应用源码[5]用于移动应用的维护和演化分析.尽管用户评论和版本更新记录都是移动应用商店中公开可爬取的文本信息,但后者所包含的无用信息更少,使用的词汇和术语相对规范且无二义性,更易于通过人工或自动的方式处理和分析.与移动应用历史版本的安装包或源码相比,版本更新记录中提供的更新内容更加直观且易于获取,更便于开发人员快速了解移动应用的发展趋势和市场动态.因此,移动应用的版本更新记录作为一种公开的数据源,可用于快速分析移动应用的演化特点,了解移动应用的发展趋势.

目前,已有部分研究将移动应用的版本更新记录用于移动应用的分析.例如,文献[6]使用移动应用的版本更新日志来识别和确定应用评论中提出的问题.Hassan等人[7]对主流安卓应用的紧急更新(emergency updates)进行了实证研究,而该研究对版本更新记录的分析仅用于确定它们是否为安卓应用的紧急更新提供了有用信息.与该研究的不同之处在于,本文的探索性研究仅将移动应用的版本更新记录作为研究对象和实验数据,拟从需求工程的角度探索移动应用更新的特点和趋势.本文的前期研究已对3000条人工标注的版本更新条目进行了初步分析,从功能和非功能需求的角度探索了移动应用的演化特点[8].然而,前期研究仅基于有限数量的版本更新日志及其人工分类标签.在前期研究基础上,本文将从2个方面设计并完成基于版本更新日志的移动应用演化趋势分析研究:1)探讨使用监督式机器学习算法从版本更新日志中自动识别不同类型的需求的可行性;2)使用规模更大的数据集对移动应用的演化趋势进行分析.具体来说,本文的主要贡献有3个方面:

1) 为了减少从版本更新日志中识别出不同类型需求所需的时间和人力成本,引入监督式机器学习算法,用以在数量庞大的移动应用版本更新日志中自动预测特定类型的需求.

2) 选择朴素贝叶斯(Naïve Bayes, NB)、随机森林(random forest, RF)和J48这3种监督式机器学习算法作为候选的版本更新日志自动分类算法,并设计了一系列实验对上述3种算法的性能进行评估,找出最适合对版本更新记录进行自动分类的机器学习算法.

3) 使用最适合对版本更新记录进行自动分类的监督式机器学习算法,完成版本更新日志中特定需求类型的自动预测和版本更新条目的自动分类,并基于自动分类结果分析移动应用的演化趋势.

1 相关工作

在发布工程(release engineering)领域,Web及桌面软件的发布规划研究和工程设计已较为成熟,而面向移动应用的发布工程研究则在近几年才受到软件工程和移动应用领域的关注[9-10].

目前,许多研究主要使用软件的用户评论和源码作为实验和分析的数据源.例如,通过对移动应用的用户评论进行分析以探索其对需求获取和软件演化的重要性[3-4,11],或者对用户评论进行语义分析进行项目推荐[12].特别是,在移动应用领域,仅有少量实证研究从开发人员的角度使用并分析软件的版本更新日志.其中,Nayebi等人开展的调查研究探索了移动应用开发人员采用何种方法对版本进行组织,以及在此过程中应用了哪些发布策略[10].文献[10]的作者发现,50%参与该调查研究的开发人员为移动应用制定了明确的发布策略.Mcilroy等人[13]对30个移动应用类别中排名前10 713个移动应用的更新频率进行了实证分析.其研究结果指出,14%更新频繁的移动应用排名较高,但其中45%的移动应用未向用户提供关于更新的任何有效信息.

Hassan 等人在文献[7]中分析了 Google Play 中 1000 个移动应用的紧急更新,用于确定应用的更新模式及其对用户体验的影响.研究结果表明,移动应用的大部分紧急更新都是由简单错误造成的,开发人员应避免这类紧急更新以改善用户体验.Nayebi 等人[14]在 2018 年进行的另一项实证研究对开发人员如何考虑在移动应用中删除和添加功能进行了探索.这一研究主要基于 Lehman 的软件演化定律,即:程序的功能必须随时间的推移而增加,以保持用户满意度.然而,Nayebi 的这项研究发现,在移动应用领域中,开发人员认为"删除已有功能与增加新功能同样重要,或比增加新功能更加重要"[14].

在对从移动应用商店收集的数据进行自动分类方面,许多研究已经成功地将监督机器学习算法应用于应用评论中的需求识别和分类[7,15-18].特别是,文献[15]和文献[17]从需求工程的角度对用户评论进行了处理,致力于将移动应用的评论自动分类为功能需求(functional requirements, FR)和特定类型的非功能需求(non-functional requirements, NFR).与移动应用的用户评论相比,版本更新日志由开发人员提供,所包含的无用信息相对较少.这一差异使得在用户评论自动分类上表现最优的分类器[7-8]可能在版本更新记录自动分类上无法表现出最优的性能.因此,需要在移动应用版本更新日志的数据集上评估监督式机器学习算法的分类效果,便于以半自动的方式为移动应用的演化趋势分析提供服务.

2 研究设计

2.1 研究问题

本文设计了 2 个研究问题(research question, RQ),从移动应用版本更新日志所属的需求类型这一角度,探索如何根据版本更新日志对移动应用的更新趋势进行分析.

问题 1.哪些技术或方法可有效用于移动应用版本更新日志的自动分类?

问题 1 原理.在文献[15,19]中,多种监督式机器学习算法已被有效用于对移动应用的用户评论进行自动分类.故理论上,同属于短文本的移动应用更新日志也可使用监督式学习算法进行自动分类,以减少人工分类的时间开销.该研究问题旨在对已有研究中分类效果较优的机器学习算法进行性能评估,探索哪种算法能更准确地对移动应用的版本更新日志进行自动分类,进而为基于版本更新日志的移动应用演化趋势自动分析提供具有优势的技术支持.

问题 2.从需求类型的角度对移动应用的版本更新日志进行分析,其结果揭示了移动应用演化的哪些特点?

作为移动应用演化的表现形态之一,版本更新日志是由开发人员在移动应用库中发布的一组短文本,通常描述了移动应用当前版本中最重要的变化.为了保持并进一步增大移动应用的市场占有率,开发人员往往对版本更新日志中的内容进行了精挑细选,以有效表征该应用的重要和关键改进.本文将通过 2 个子研究问题对移动应用的演化特点进行分析:

问题 2.1.根据版本更新日志,开发者在更新移动应用时着重考虑的是哪种类型的需求?

问题 2.1 原理.该研究问题有助于分析哪种需求类型(功能需求、非功能需求或某种特定的非功能需求)主导了移动应用的更新,进而探索这类需求类型在不同移动应用类别中的关注度差异及其随时间的变化趋势.

问题 2.2.从版本更新日志所属需求类型的角度来看,移动应用演化的热点有哪些?这些更新热点随时间有何变化?

问题 2.2 原理.该研究问题进一步从不同需求类型的角度抽取移动应用更新的热点,有助于分析移动应用更新热点的变化趋势.

2.2 数据采集

本文所使用的数据来源于本文作者在前期研究[8]中收集和使用的移动应用版本更新日志,数据采集时间为 2019 年 3 月.在数据采集任务正式开始前,作者对主流的移动应用商店中不同类别移动应用的版本更新记录进行了预调研,并根据预调研的结果制定了 2 个数据采集策略:1)考虑移动应用版本更新记录采集的便捷性与可行性,选择苹果 App Store 作为数据采集源;2)选择社交(Social Networking)、旅游(Travel)和阅读(Books)这 3 个类别移动应用的版本更新记录作为待收集和分析的对象.第 2 个采集策略选择 3 个移动应用的原因为:1)根据苹果 App Store 应用市场中各类别应用数量排行[20],游戏类应用占比最多(21.86%),新闻类占比最少(1.83%).过高或过低的样本占比可能对移动应用演化趋势分析产生较大影响,故选择移动应用数量占比在 2%~6%这个区间范围的应用类别较为合适.2)所选类别的移动应用版本更新频次不能过低,版本更新日志的内容应较为丰富且有价值,有助于基于版本更新日志对移动应用演化趋势开展探索性

分析.需要说明的是,匹配上述 2 个采集策略的移动应用类别较多,作者为开展本文的探索性研究随机选择出的 3 个类别为社交(Social Networking)、旅游(Travel)和阅读(Books),样本数据有一定的多样性,但仅能用于部分探索移动应用演化的特点和趋势,不能代表整个移动应用市场的发展动态.

前期研究[8]在进行正式的数据采集时,首先浏览了苹果 App Store 六个地区(亚洲、北美洲、欧洲、南美洲、非洲和大洋洲)的移动应用商店中社交、旅游和阅读这 3 个类别移动应用的版本更新日志[8].其次,分别爬取了在每个地区 3 个类别中排名前 10 的免费移动应用的版本更新日志;接着,排除了重复、更新日志的数量低于 30 以及近 3 年无版本更新日志的移动应用;最终,保留了 120 个移动应用(社交类 40 个、旅游类 50 个、阅读类 30 个)的 8 647 个版本更新日志.由于一个版本更新日志通常包含多条更新条目,上述 120 个移动应用发布的更新条目共计 17 024 条.

在已收集的版本更新日志中,"本次更新"和"近期更新"的内容往往有较大程度的重复.为消除冗余,本文制定了筛选原则:若本次更新中的版本更新条目与"近期更新"中的更新条目相同,则保留"本次更新"中的更新条目.将这一筛选原则应用于 120 个移动应用的 17 024 条版本更新条目,可得到8 325 条不重复的版本更新条目组成的初始数据集.

为了平衡不同类别移动应用所提供的版本更新条目的数量,作者从初始数据集中选择了 60 个移动应用(在社交、旅游、阅读类别中分别随机选择 20 个移动应用)的版本更新条目,构建了本文研究所需的数据集.同时,考虑版本更新日志的稀疏性和部分缺失,剔除了 2013 年之前和 2019 年之后发布的版本更新条目.最终,得到的数据集由 60 个应用的 6 527 条版本更新条目组成,其发布时间区间为 2014 年至 2018 年.其中,旅游类的版本更新条目为 2 782 条,社交类的为 2 666 条,阅读类的为 1 079 条.

2.3 数据集划分

根据前期研究的经验,在有限的时间内人工标记和分析 6 527 条版本更新条目是一项非常耗时的任务.因此,我们从 6 527 条版本更新条目中选择了 3 000 条(社交、旅游、阅读类分别随机选择 1 000 条)进行人工标注和分类,用以训练和评估监督式分类算法;而其余的 3 527 条将用评估结果中最有效的监督式分类算法进行自动分类.

2.4 数据标注

为提高人工标注的准确性,本文采用了 Krippendorff 等人[21]提出的先验编码(priori coding)方法进行人工标注,使定性分析的结果更加丰富[22].本文的人工标注过程由 2 名计算机专业的本科生(含第 1 作者)完成;标注结果的一致性校验由标注者与第 2 作者和第 3 作者共同完成,标注结果的正确性校验由第 2 作者和第 3 作者共同完成.具体的标注过程如下:

首先,作者与 2 位标注者一起对拟采用的分类标准(ISO 25010[23])进行了解释与讨论.接着,2 名标注者对 120 条(社交、旅游、阅读类分别随机选择 40 条)版本更新条目进行了独立的人工预标注,分别耗时 30 min 和 35 min,2 人标注结果的内部一致性[17]为 63%.2 名标注者与第 2 作者和第 3 作者对 37%不一致的标注结果进行了讨论并达成一致后,制定了一个人工标注指南,以帮助标注者提高人工分类的准确性.之后,重新选择了 500 条更新条目进行第 2 轮试标注.经统计,第 2 轮试标注的内部一致性为 73.8%.2 名标注者与第 2 作者和第 3 作者对第 2 轮中不一致的人工分类结果进行讨论并达成一致后,对人工标注指南进行了更新.最终,2 位标注者依据该指南对 3 000 条更新条目进行了标注并达成一致意见.为降低人工标注的准确性对后续版本更新日志自动分类结果准确性的影响,第 2 作者和第 3 作者分别从数据集中随机抽取了 5%进行标注结果的正确性校验,以确保人工标注的质量.

在人工标注的过程中,绝大多数移动应用的版本更新条目只提及 1 种需求类型,但仍有少量(34/3 000)更新条目的内容涉及多种需求类型.为确保 1 条更新条目仅包含 1 种需求类型的标签以便对监督式分类器进行训练,作者将这 34 条更新条目进行了手动拆分,故人工标注的数据集最终由 3 034 条移动应用的版本更新条目构成.

本文采用的分类标准 ISO 25010 定义了 8 种类型的 NFR:功能适用性(Functional Suitability)、性能效率(Performance Efficiency)、兼容性(Compatibility)、可用性(Usability)、可靠性(Reliability)、安全性(Security)、可维护性(Maintainability)和可移植性(Portability).在对 3 034 条版本更新条目进行人工标注时,未发现提及"功能适用性"和"可维护性"这 2 类非功能需求的更新条目.因此,本文仅考虑 ISO 25010 中定义的其余 6 种 NFR 类型和 FR 共 7 种与需求类型相关的类别.此外,本文还定义了

类别“其他”以归类不能分为以上 7 种类型的更新条目.例如,对用户支持的感谢以及与需求无关的更新条目(“keep the good work”或“please continue to pay attention”),或是未指明具体缺陷仅声明“对部分问题进行了修复”(“Bug fixed”)的版本更新条目.表 1 列举了本文考虑的 8 种类型(即“性能效率”“兼容性”“可用性”“可靠性”“安全性”“可移植性”“功能需求”和“其他”)及其相应示例.

Table 1 Exemplary App Changes of Eight Types

表 1 8 种类型的更新记录示例

需求类型	版本更新条目示例
性能效率	Establishing video session tasks less time.
兼容性	General compatibility improvements for iPhone 6 and 6 s users.
可用性	Read magazines in a more friendly interface format.
可靠性	Email notification work again.
安全性	Password Manager support on login for that easy yet secure access.
可移植性	Support for 3D Touch on iPhone 6 and 6 s.
功能需求	We have added a password function you can set a password to enter the program.
其他	Thank you for your use, please continue to pay attention. Bug fixes in previous version.

2.5 数据预处理

本文使用 Weka 3.7 对原始数据进行预处理,具体流程为:首先,将人工标注的 3 034 条版本更新条目(.csv 格式)导入 Weka 完成数据的规范化,使用 Weka 中的 *NominalToSting* 函数将初始文件数据类型从 Nominal 转换为 String.其次,使用 Weka 中的 *StringToWordVector* 函数进行文本向量化,生成词向量.其中,使用 Snowball 进行词干提取,使用 MultiStopwords 停用词表进行去停用词处理,选择 WordTokenizer 作为分词器.同时,将 TFTransform 和 IDFTransform 设置为 True,使用 TF-IDF 进行特征词提取.

2.6 分类器的选择、训练及评估

基于已有相关研究和前期研究的实验结果,本文拟对比 NB,RF 和 J48 这 3 种监督式机器学习算法在自动分类移动应用版本更新日志上的性能.3 种分类器的选择依据为:1)文献[24]指出,NB 是一种被广泛应用的机器学习算法.同时,相关研究[15,19,25]的实验结果显示,在对移动应用用户评论这类短文本进行自动分类时,采用 NB 算法可以得到更准确的分类结果.2)若在文本向量化时采用 TF-IDF 进行用户评论特征词的提取,使用 J48 对用户评论进行自动分类的效果略优于 NB[15].3)在我们的前期研究中[8,26],应用 RF 对版本更新日志进行自动分类可得到最优的分类效果.

为评估 3 种算法的性能,本文实验采用十折交叉检验以减少分类器对版本更新条目中需求类型进行识别和分类的过拟合.此外,本文采用准确率(*Precision*)、召回率(*Recall*)和 F 值($F1$)来评估上述 3 种分类算法的分类效果.各评估指标加权平均值为

$$Weighted\_Average(Precision_i)=\frac{\sum\limits_{i\subseteq type}Precision_i\times Number_i}{\sum\limits_{i\subseteq type}Number_i},\tag{1}$$

$$Weighted\_Average(Recall_i)=\frac{\sum\limits_{i\subseteq type}Recall_i\times Number_i}{\sum\limits_{i\subseteq type}Number_i},\tag{2}$$

$$Weighted\_Average(F1_i)=\frac{\sum\limits_{i\subseteq type}F1_i\times Number_i}{\sum\limits_{i\subseteq type}Number_i},\tag{3}$$

其中,$Precision_i$,$Recall_i$,$F1_i$ 分别表示与需求类型 i 相关的版本更新条目自动分类的准确率、召回率、F 值,$Number_i$ 表示隶属于该类型的版本更新日志条目数量.

2.7 热词热度计算

针对问题 2.2,本文用版本更新条目中的热词表示版本更新的热点,用热词热度来表示版本更新记录中热点的受关注度.热词热度的计算公式为

$$DegreeOfConcerns_i=\frac{\sum\limits_{j=1}^{n}OccurrenceW_{ij}}{Number_i},\tag{4}$$

其中,$DegreeOfConcerns_i$ 表示与需求类型 i 相关的版本更新记录的热词热度,$Occurrence\ W_{ij}$ 表示需求类型为 i 的版本更新记录中第 j 个特征词 W_{ij} 的出现次数,$j\in\{1,2,\cdots,n\}$(n 为需求类型 i 的版本更新记录中热词数目),$Number_i$ 表示与需求类型 i 相关的版本更新条目数.

3 实验与结果

3.1 版本更新日志自动分类的算法评估(问题 1)

针对问题 1,本节将根据应用 3 种监督式机器学习算法得到的分类结果的 F 值,对不同分类算法在版本更新日志自动分类中的性能进行评估,以确定

最适合对版本更新日志进行自动分类的算法,探索基于版本更新日志进行移动应用演化趋势自动分析的可行性.

3.1.1 实验准备

首先,根据 2.4 节定义的人工标注过程完成 3 034 条版本更新条目的人工标注,其标注结果如图 1 所示.其中,1 170 条(占比约为 38.56%)涉及 FR,1 519 条(占比约为 50.07%)涉及 NFR,345 条(占比约为 11.37%)未涉及 FR 或 NFR.进一步,在 1 519 条标记为 NFR 的更新条目中,提及"可用性"这一非功能属性的版本更新条目最多(953/1 519),占人工标记总条数的 62.74%;提及"安全性"和"兼容性"的版本更新条目最少,分别为 13 条和 40 条,占人工标记总条数的 0.40%和 1.32%.

考虑监督式自动分类方法,如果待分类样本集的数量过小,无法对分类器进行有效和准确地训练.鉴于标注需求类别为"兼容性"和"安全性"的版本更新条目数量较少,我们将上述 2 个类别的更新条目并入"其他"类别,以便更好地训练分类器.也就是说,在本文的版本更新记录自动分类实验中,3 种监督式自动分类方法仅用于将移动应用的版本更新条目分为 6 种类型:"功能需求""性能效率""可用性""可靠性""可移植性"和"其他".

本节实验的实验环境配置如下:CPU 为 Intel 9100f,GPU 为 AMD RX580,内存为 16 GB 的便携计算机,操作系统版本为 Windows 10 1909,实验软件为 Weka3.7.

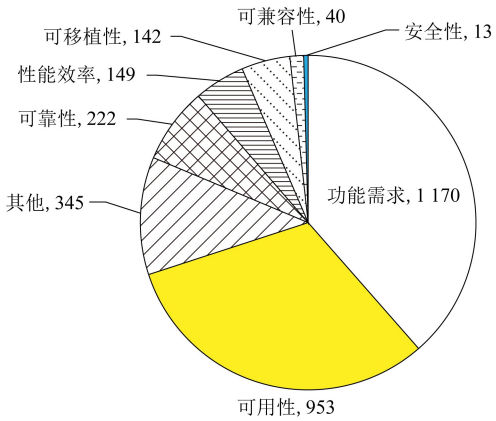

Fig. 1 Summary of 3 034 manually labeled app changes over eight types of requirements

图 1 3 034 条人工标注更新条目在需求类型中的分布情况

3.1.2 3 种算法的分类结果对比

本节将对比 NB,RF 和 J48 这 3 种分类算法在版本更新条目自动分类中的性能.实验数据为 3.1.1 节中 3 034 条人工标注的版本更新条目.

表 2 展示了上述 3 种算法对 3 034 条版本更新条目分别进行二分类(binary classifier, BC)和多分类(multi classifier, MC)时的准确率、召回率和 F 值.综合实验结果表明,对任意一种需求类型而言,3 种机器学习算法的二分类效果均明显优于多分类.同时,从分类算法的角度可得出结论:NB 和 RF 分别在二分类和多分类任务中性能最优.

Table 2 Performance of the Three Evaluated Machine Learning Algorithms on App Changes

表 2 3 种监督式机器学习算法的算法性能对比表

分类器种类	指标	功能需求	性能效率	可用性	可靠性	可移植性	其他	加权平均值
NB(BC)	准确率	0.744	0.976	0.821	0.973	0.964	0.927	0.840
	召回率	0.748	0.974	0.819	0.973	0.956	0.924	0.840
	F 值	0.745	0.975	0.806	0.973	0.959	0.925	**0.835**
NB(MC)	准确率	0.618	0.662	0.793	0.766	0.475	0.431	0.652
	召回率	0.779	0.691	0.548	0.752	0.725	0.312	0.636
	F 值	0.690	0.644	0.648	0.759	0.574	0.362	0.631
RF(BC)	准确率	0.746	0.972	0.817	0.970	0.963	0.886	0.834
	召回率	0.749	0.972	0.816	0.971	0.967	0.902	0.837
	F 值	0.784	0.972	0.813	0.971	0.963	0.887	0.831
RF(MC)	准确率	0.683	0.782	0.756	0.803	0.577	0.396	0.677
	召回率	0.766	0.624	0.601	0.698	0.500	0.528	0.659
	F 值	0.722	0.694	0.670	0.747	0.536	0.453	**0.662**

Continued (Table 2)

分类器种类	指标	功能需求	性能效率	可用性	可靠性	可移植性	其他	加权平均值
J48(BC)	准确率	0.734	0.959	0.766	0.957	0.963	0.860	0.808
	召回率	0.738	0.963	0.763	0.960	0.967	0.889	0.837
	F 值	0.730	0.960	0.732	0.956	0.962	0.858	0.831
J48(MC)	准确率	0.681	0.617	0.544	0.804	0.548	0.374	0.666
	召回率	0.741	0.679	0.627	0.739	0.444	0.583	0.638
	F 值	0.710	0.668	0.503	0.770	0.490	0.455	0.643

注:黑体数字表示最优值.

此外,对版本更新日志中 FR 这一类型进行自动分类时,使用 NB(BC)和 RF(MC)得到的 *F* 值较为接近;而对"性能效率""可移植性"和"其他"这 3 种类型而言,使用 NB(BC)自动分类的结果明显优于 RF(MC).

进一步,图 2 展示了应用 NB(BC)和 RF(MC)对版本更新日志进行自动分类时,不同需求类型更新条目的占比对自动分类性能评估指标 *F* 值的影响.由图 2 可知,与 FR 和"可用性"相比,"性能效率""可靠性""可移植性"和"其他"这 4 种类型的更新条目数占比较低,但使用 NB(BC)进行自动分类得到的 *F* 值较高.

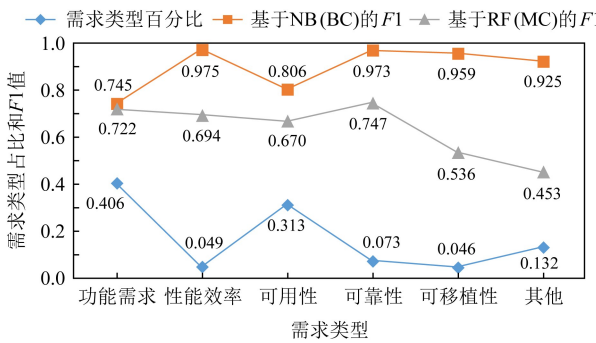

Fig. 2 Influence of the percentages of each type in app changelogs on the *F*1 of NB(BC) and RF(MC)

图 2 不同类型版本更新日志数量占比对 NB(BC)/RF(MC)自动分类结果 *F* 值的影响

类似地,对类型为"性能效率"和"可靠性"这 2 类占比较低的版本更新条目来说,RF(MC)的分类正确率较高.不同的是,"可移植性"和"其他"这 2 类更新条目数量的低占比,其对应的 RF(MC)分类正确性也随之降低.

3.1.3 版本更新条目的自动分类结果

表 3 和表 4 展示了应用 RF(MC)对数据集中 6 527 条版本更新条目进行自动分类的结果.

从表 3 中可以发现,总体上,FR 相关(47.0%)的版本更新日志多于与 NFR 相关(44.4%).但对于阅读类移动应用而言,其版本更新记录中与 FR 相关(43.6%)的更新条目数略少于与 NFR 相关(48.1%)的版本更新日志.

Table 3 Distribution of App Changelogs Over App Categories (FR vs NFR) Using RF(MC)

表 3 基于 RF(MC)的版本更新记录自动分类统计结果(功能需求与非功能需求)

应用类别	功能需求	非功能需求	其他	总计
旅游类	1 293	1 280	209	2 782
社交类	1 306	1 104	256	2 666
阅读类	471	520	88	1 079
总计	3 070	2 904	553	6 527

Table 4 Distribution of App Changelogs Over App Categories (4 NFR Types) Using RF(MC)

表 4 基于 RF(MC)的版本更新记录自动分类统计结果(4 种非功能需求)

应用类别	可用性	可靠性	性能效率	可移植性
旅游类	867	202	104	107
社交类	765	132	109	98
阅读类	306	118	59	37
总计	1 938	452	272	242

表 4 进一步展示了 3 类移动应用在 4 个 NFR 类型的详细分布.其中,"可用性"在 NFR 中占据主要地位.其次,"可用性"和"可靠性"的占比与 3.1.1 节中人工分类结果中的占比近似(前者为 82%,后者为 77%).再者,在社交类移动应用的发布的版本更新记录中,"可靠性"相关版本更新条目的占比明显少于旅行类和阅读类,而"性能效率"相关版本更新日志占比则略多于其他 2 个移动应用类别.

问题 1 结论:

1) NB,RF 和 J48 这 3 种监督式机器学习算法均可用来识别和预测版本更新日志的需求类型.

2) 应用上述 3 种算法预测版本更新条目的需求类型并进行自动分类时,BC 比 MC 的结果更准确.

3) 在使用 BC 时,NB 的性能优于 RF 和 J48;在使用 MC 时,RF 的性能优于 NB 和 J48.因此,如果仅需对某一特定需求类型进行自动分类和分析,可采用 NB(BC)算法,见 3.2.2 节;否则,则可使用 RF(MC),见 3.2.1 节.

4) 对于某些需求类型(如"可靠性"),RF(MC)分类结果的 F 值与数据集中提及该类型的版本更新条目占比成反比;而对其他的需求类型(如 FR)而言,RF(MC)分类结果的 F 值与数据集中提及该类型的版本更新条目占比成正比.

3.2 移动应用的演化特点(问题 2)

3.2.1 需求类型视角下移动应用的演化特点(问题 2.1)

基于 3.1.3 节中应用 RF(MC)得到的 6 527 条移动应用版本更新条目的自动分类结果,本节从更新条目涉及的需求类型这一角度,对移动应用的演化趋势展开如下分析:

从总体上看,6 527 条更新条目中的 289 条发布于 2014 年,780 条发布于 2015 年,1 305 条发布于 2016 年,1 780 条发布于 2017 年,2 373 条发布于 2018 年.由此可见,近年来,移动应用版本更新日志中更新条目的数量持续增加.进一步,图 3 展示了旅游、社交和阅读类移动应用的版本更新条目数量随时间的变化趋势,即:旅游类移动应用的版本更新条目数量增长最快,社交类次之,阅读类移动应用更新条目数量的增长则十分缓慢.

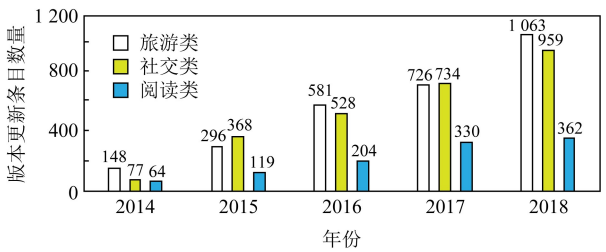

Fig. 3 Distribution of app changes over three app categories and years

图 3 版本更新日志数量随类别及年份的变化

图 4 展示了 FR 和 4 种 NFR 需求类型相关的移动应用版本更新条目数随时间变化的情况.近 5 年来,FR 和"可用性"相关的更新条目数量增速较快,而"可靠性""性能效率"和"可移植性"类型的更新条目数量则增速缓慢.

进一步,图 5 显示了旅游、社交和阅读类移动应用中 FR 相关的版本更新条目数随时间的变化情况.与阅读类移动应用相比,旅游和社交类移动应用

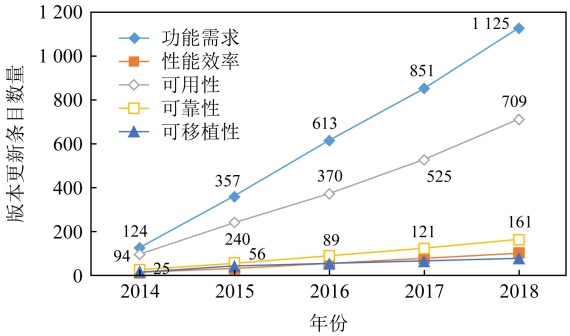

Fig. 4 Trends of different requirements types in app changelogs

图 4 不同需求类型的版本更新日志数随年份的变化

在演化更新时,涉及 FR 的版本更新条目数持续增长,且增速较快.这一现象说明,旅游和社交类移动应用在版本更新时更侧重引入新的功能.

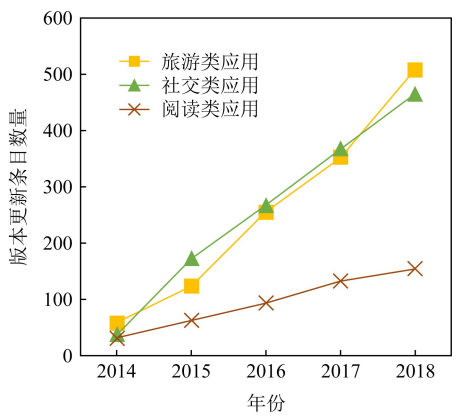

Fig. 5 Trends of FR over the three app categories

图 5 FR 类型版本更新日志数随年份的变化

图 6 显示了 3 个类别移动应用中与"性能效率"相关的版本更新条目数量随时间的变化情况,即与

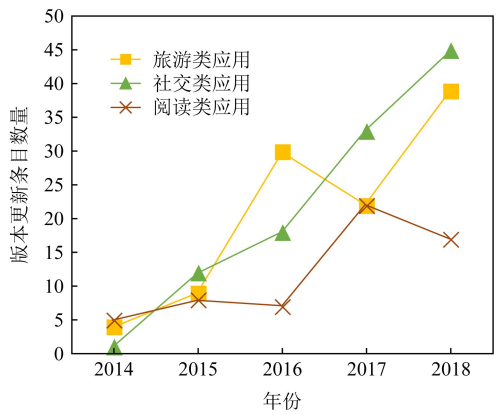

Fig. 6 Trends of performance requirements over the three app categories

图 6 "性能效率"类型版本更新日志数随年份的变化

其他 2 个类别的移动应用相比,社交类移动应用的版本更新记录中此需求类型被提及的频率更高;而除 2016 年和 2018 年外,阅读类和旅游类频率基本一致.

图 7 显示了 3 个类别移动应用中,与“可用性”相关的版本更新条目数量随时间的变化情况.与阅读类移动应用中收集到的更新日志相比,旅游类和社交类的移动应用均对“可用性”给予了更为持续的关注.

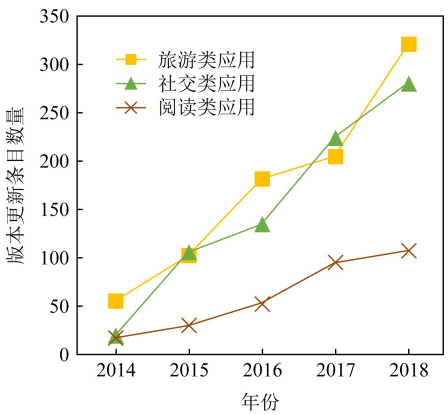

Fig. 7 Trends of usability requirements over the three app categories

图 7 “可用性”类型版本更新日志数随年份的变化

图 8 显示,近 5 年,这 3 类移动应用的版本更新日志对“可靠性”的关注度均有所提升.而与其他 2 个应用类别相比,旅游类应用更注重与可靠性相关的改进.

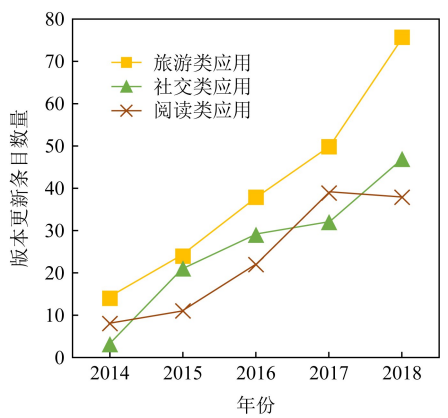

Fig. 8 Trends of reliability requirements over the three app categories

图 8 “可靠性”类型版本更新日志数随年份的变化

图 9 展示了移动应用中“可移植性”类型的版本更新条目数量随时间的变化趋势.自 2016 年以来,该需求类型在社交类应用中被频繁提及.而在过去 3 年,旅游类应用在更新时对“可移植性”的关注度基本保持不变.

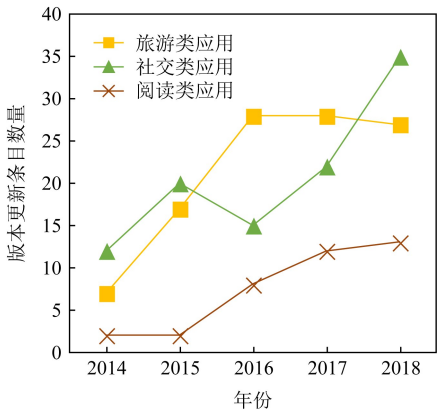

Fig. 9 Trends of portability requirements over the three app categories

图 9 “可移植性”类型版本更新数日志随年份的变化

问题 2.1 结论:

1) 近 5 年来,移动应用的更新日志数目在不断增加.相较于其他 2 类应用,阅读类应用更新日志数目增长速度较慢.

2) FR 和“可用性”是移动应用版本更新日志中增长最快的 2 个需求类别.不同类型需求相关的版本更新日志数量在不同应用类别上的增长情况并不相同.例如,对于“可靠性”而言,3 个类别移动应用的演化趋势相似;而对于 FR 和“可用性”来说,旅游和社交类移动应用发布的相关版本更新条目数的增速远快于阅读类.

3.2.2 热词视角下移动应用的演化特点(问题 2.2)

从热词的角度对移动应用的演化特点进行分析,需要针对不同的需求类型分别提取出对应版本更新记录中的热词.基于 3.1.2 节的结论,本节将使用 NB(BC)对移动应用的版本更新记录进行自动分类,并基于分类结果对移动应用演化热词的特点进行分析.

表 5 中比较了 FR 和 4 种 NFR 类型的热词.其中第 2 列中的特征词包含在 2.4 节中提及的标注指南中,第 3 列的热词则是在 5 类版本更新日志中分别应用 Weka 中 TF-IDF 自动生成的特征词的子集,本文抽取了排名前 10 的特征词并去除部分无意义的词后组成了热词集合.5 种需求类型对应的热词之间无重复,即 5 种需求类型对应的 5 组热词集合之间没有交集.然而,对任一种需求类型而言,表 5 第 2 列中的特征词和第 3 列中热词集合存在交集,即人工标注指南中 75.8%(25/33)的特征词与自动分类结果中 54.3%(25/46)的热词重合.

Table 5 Hot Words for FR and Four Types of NFRs

表 5 FR 和 4 种 NFR 相关的热词

需求类型	标注指南中的特征词	基于 TF-IDF 自动抽取的热词
功能需求	**new**, **add**, **option**, **able**, **can**, **ability**, **available**, **enable**	**new**, **add**, **option**, theme, **able**, **can**, **ability**, **available**, now, **enable**
性能效率	**performance**, **fast**, efficiency, **speed**, **quick**, battery, reduced, smaller, **low**, memory	**performance**, **fast**, **speed**, **quick**, run, less, time, **low**, screen
可用性	**more**, **easier**, **better**, **clearer**, **smoother**, enhance, **improve**, optimized, upgraded	**more**, **better**, **easier**, **smoother**, **clearer**, experience, **improve**, update, UI
可靠性	**crash**, **back**, **correctly**, **again**	**crash**, **back**, stability, stable, support, fix, reliability, **again**, **correctly**, fight
可移植性	**support**, **ios**	**support**, **ios**, ipad, iphone, retina, you, optimization, button

注:黑体词表示自动分类结果与人工标注指南中重复的热词.

接下来,将进一步分析不同需求类型的版本更新记录中所包含热词随时间的变化情况.根据 3.1.1 节中不同需求类型的版本更新条目在数据集中的占比,本文将以 FR 和 NFR 中的"可用性"这 2 个占比较高的需求类型为例,展示移动应用版本更新记录中热词及其热度的变化,从热词的角度对移动应用的演化特点进行分析.

基于 NB(BC)的自动分类结果,2014～2018 年,与 FR 相关的版本更新条目数为 2 312 条,与"可用性"相关的版本更新条目数为 1 249 条.据统计,表 5 行 1 列 3 中列出的 10 个 FR 相关热词在 2 312 条 FR 类型更新条目中出现的总频次为 2 889 次,FR 相关热词的综合热度为 1.250,其计算公式见式(2).图 10 进一步展示了这 10 个 FR 相关热词在 2 312 条 FR 类型更新条目中的出现频次及其随时间的变化趋势.可以看出,在提及 FR 的版本更新日志中,出现次数最多的 3 个热词是"now"(799 次)、"can"(672 次)和"new"(536 次),且其出现次数逐年增多;而"theme""enable"和"ability"等热词在版本更新日志中出现次数较少,在某些年份的出现次数甚至为 0,故这几个热词随时间的变化趋势无明显规律.

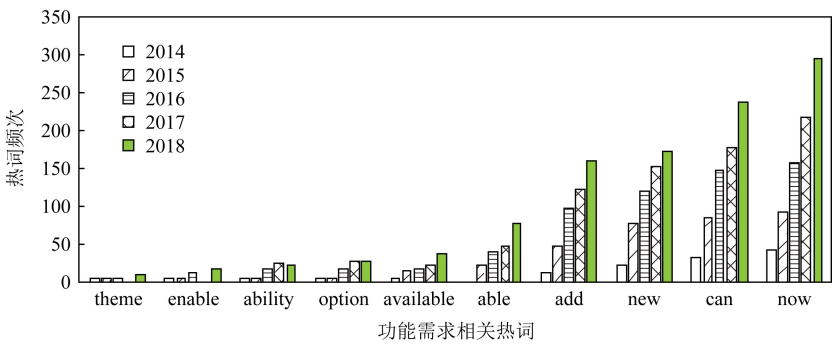

Fig. 10 Occurrence of hot words in app changes over years (FR)

图 10 FR 相关热词的频次随年份的变化

类似地,表 5 行 3 列 3 中列出的 9 个"可用性"相关热词在 1 249 条版本更新条目中出现的总频次为 1 609 次,依据式(2)计算得到的相关热词综合热度为 1.288.图 11 展示了这 9 个"可用性"相关热词在 1 938 条"可用性"类型更新条目中的出现频次及其随时间的变化趋势.可以看出,在与"可用性"相关的版本更新记录中,出现总次数最多的 3 个热词是"improve"(515 次)、"more"(296 次)和"update"(221 次).然而,从热词出现次数的增速来看,在 2017 至 2018 年间,"improve"和"easier"的增幅较快,而"better"和"update"的增幅较缓.

图 12 展示了 FR 和"可用性"相关热词的综合热度随时间的变化情况.在移动应用的版本更新日志中,与 FR 相关热词的综合热度在 2016 年达到峰值后开始下降;相反,与"可用性"相关热词的综合热度在 2016 年后呈现出上升趋势.

问题 2.2 结论:

1) 标注指南中人工定义的特征词与自动提取的特征词重复率较高,且在标注指南中所指定的特征词大部分都包含在采用 TF-IDF 提取的特征词中.

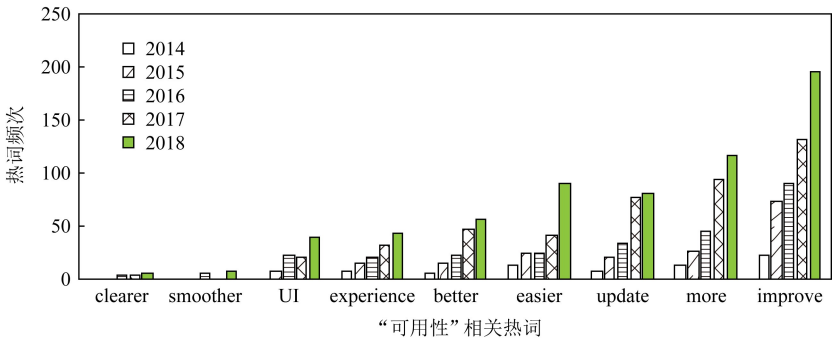

Fig. 11 Occurrence of hot words in app changes over years (Usability)

图 11 “可用性”相关热词的频次随年份的变化

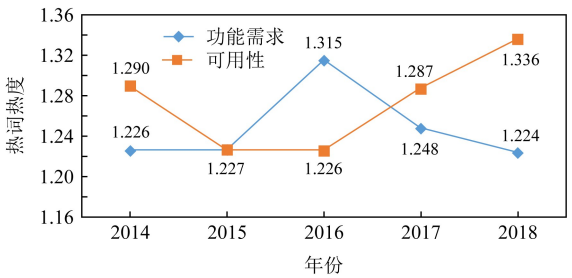

Fig. 12 The degree of concerns of hot words (FR and Usability)

图 12 FR 与“可用性”相关热词的综合热度变化趋势

2) 考虑采用 TF-IDF 从 FR 和“可用性”相关更新条目中提取出的热词,其出现频次均逐年增多.此外,不同需求类别的版本更新日志所包含热词的综合热度随时间的变化趋势不尽相同.

4 实验结果分析

4.1 3 种监督式分类算法的性能分析(问题 1)

3.1 节中的问题 1 的结论 1)~3)表明,对版本更新记录的自动分类而言,使用 RF(MC)的分类结果最优.这一结论与文献[15,19]中的实验结果并不一致,即对用户评论进行自动分类的最优分类器并不适用于对移动应用版本的更新日志进行自动分类.可能的原因是,用户评论和版本更新记录这 2 类移动应用数据的数据质量不同,而某些分类器的性能会部分受到数据集质量的影响.此外,在自动分类版本更新日志时,BC 明显优于 MC,而此结论与文献[15]的结论完全一致.

3.1 节的问题 1 的结论 4)表明,不同需求类型在版本更新日志中的百分比并不总是与 RF(MC)的 F 值成正比.例如,“性能效率”类型的版本更新日志在数据集中占比较低,但对该需求类型的更新日志进行自动分类时分类的正确性却较高.究其原因,可能是对“性能效率”这一需求类型来说,分类器从训练集中提取的文本特征词对识别该需求类型的版本更新日志更加敏感.

4.2 移动应用的演化特点及趋势分析(问题 2)

移动应用版本更新条目的数量变化揭示,随时间的推移,版本更新条目的数目持续增多.原因可能是,移动应用的用户数量在增加,用户的个性化需求也越来越多,移动应用需要通过更频繁的更新、更丰富的更新内容以快速满足用户需求,以达到维持现有用户和吸引潜在用户的目的.

从移动应用所属类别的角度来探索移动应用的演化趋势时,作者发现:相对于社交和阅读类移动应用,旅游类应用发布的版本更新条目更多.原因可能是旅游类应用通常与其他类别的移动应用关系有更紧密的关联,故需要发布更多数量的版本更新条目以与其他应用进行协调.

根据与各需求类型相关的版本更新条目数量的占比和变化情况可以发现:从整体上看,FR 相关的版本更新条目数量比 NFR 相关的增长更快,原因可能是用户不断地期望有新功能对应用进行扩展.然而,该结论因移动应用的类型不同而有所差异:对于阅读类移动应用而言,NFR 类型更新的数量大于 FR 类型的更新数量;而对于社交类和旅游类移动应用而言,FR 类型的版本更新数量略大于 NFR 类型的更新数量.原因可能是,阅读类应用的功能相对专一,与其他类别应用的交互较少,故更侧重于改善用户体验.

根据不同需求类型版本更新日志中的热词及其出现次数的变换情况可以发现:首先,由于在标注指南中指定的特征词大部分都包含在采用 TF-IDF

提取的特征词中,基于 NB(BC)的半自动方法可极大地减少制定人工标注指南的时间和人力成本.其次,不同类型(如 3.3.2 节中的 FR 和“可用性”)的版本更新日志所包含热词的出现频次随时间的变化趋势不同.一方面,同一类型版本更新日志中不同热词出现次数随时间的变化趋势表明了开发者在描述移动应用版本更新内容时的用词偏好,可用于指导版本更新日志内容的撰写以最大程度吸引用户的关注.另一方面,考虑不同类型热词的综合热度,“可用性”类热词热度的攀升以及 FR 类热词热度的降低说明质量属性正逐渐成为开发者发布版本更新日志的重要关注点.

5 局限性

本文的研究结果由于 5 个方面的原因存在一定的局限性.

1) 数据集所涉及的移动应用类别仅有 3 类.作为探索性研究,这 3 种移动应用类别的选择策略(见 2.2 节)在一定程度上保证了样本的均衡性,但 3.3 节的实验结果和结论仅能展现移动应用市场中部分类别的移动应用的演化趋势,不能完整准确地代表移动应用市场的整体特点和变化.文献[16]曾指出,如果选择和收集苹果 App Store 中其他类别应用的版本更新记录进行实验,其结果或许有所不同,故扩大移动应用数据集的类别将是本文下一步的研究工作之一.此外,所有的版本更新日志均来源于苹果 App Store.从其他移动应用市场(如 Google Play)中收集移动应用的版本更新记录进行处理和分析,会极大增强本文结论的通用性.

2) 本研究中作为监督式机器学习算法训练集的 3 034 条版本更新条目是由 2 位计算机专业本科生分析和标注.为了得到高质量的标注并以此为基础进行分析,作者采取的措施有:在人工标注之前,2 位标注者从本科的专业课程中学习了需求工程和软件工程的知识体系.考虑标注的过程,本文进行了 2 轮预标注,以帮助 2 位标注者对不同类型需求(特别是 NFR)的含义和理解达成共识.如 2.4 节所述,两轮预标注的一致性分别为第 1 轮 63%,第 2 轮 73%.该过程可以帮助确定并完善人工标注指南,以便进一步对其他版本更新日志进行标注.同时,第 2 作者和第 3 作者不仅参与了 2 轮预标注过程中不一致意见的讨论和决策,还通过随机抽取少量样本的方式对 3 034 条版本更新条目的人工标注结果进行了正确性检验,以保证分析过程和结论的质量.

3) 本文 3.1 节的所有实验均在 Weka 中完成.Weka 是用 Java 语言编写的机器学习工具套件.如果用不同的语言(如 Python)对 3 种分类算法进行实现,则版本更新日志的自动分类结果可能有所不同.因此,如何保证不同语言的算法实现所得到分类结果的一致性还有待进一步研究与分析.

4) 版本更新记录是一种易获取且数据质量较高的公开数据源,从开发者的角度展示了移动应用更新的主要特点.理论上,可以为研究和分析移动应用的演化趋势提供高质量的数据支持.然而在实验数据的收集和分析过程中,作者发现,部分趋于成熟的移动应用在一定时间段内发布的多个版本更新记录不仅具有相同的内容,且其内容多为无意义的信息,这对基于版本更新记录的移动应用演化趋势分析提出了挑战.而对于版本更新记录详尽的移动应用而言,使用本文提供的方法能够在较短时间内了解和掌握其演化的特点和趋势.鉴于此,作者拟进一步研究,在基于版本更新记录分析移动应用演化趋势中移动应用类型、应用综合评分等因素对分析结果的影响.

5) 本文 3.2 节中对移动应用演化特点的分析和研究分别基于应用 RF(MC)和 NB(BC)对版本更新条目进行自动分类的结果.鉴于上述人工标注过程存在的问题以及上述 2 种分类算法准确率(详见表 2),基于该自动分类结果得到的移动应用演化趋势未必能反映出移动应用的真实更新趋势.如果改进分类器以得到更好的分类效果,或对 RF(MC)/NB(BC)生成的需求类型进行人工复核,移动应用演化趋势分析的准确性将得到进一步提高.

6 总结和展望

本文从需求工程的角度,研究和分析了苹果 App Store 中 3 个类别 60 个移动应用的版本更新日志及其内容,发现与功能需求和非功能需求相关的更新条目数量接近.

本研究涉及旅游、社交和阅读这 3 个类别的移动应用,探讨了如何利用监督式机器学习算法促进版本更新日志的自动分类,以降低人工标注的成本,帮助移动应用开发者快速了解和分析特定类别的移动应用更新的趋势.

下一步计划扩大研究范围,引入其他类别的移动应用(如苹果 App Store 和 Google Play 中健康和

教育类应用)并考虑中国移动应用市场(如华为应用市场和小米应用市场),以提高对整个移动应用市场整体特点和演化趋势分析的准确性以及研究结果的普适性.同时,还计划向本文所分析的移动应用的开发人员进行调研,以探索本研究结果的正确性和有效性及其对移动应用发布工程的影响.再者,除了采用有监督的分类方法外,还将基于人工标注数据所得到的标注指南,探索相应的启发式分类算法,以提高版本更新日志的自动分类效果.

参 考 文 献

[1] Statista. Number of apps available in leading app stores as of 1st quarter 2020 [EB/OL]. 2020 (2020-04)[2020-05-16]. https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/

[2] 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, 姜瑛. 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)

[3] Meng Xiangwu, Wang Fan, Shi Yancui, et al. Mobile user requirements acquisition techniques and their applications [J]. Journal of Software, 2014, 25(3): 439-456 (in Chinese)
(孟祥武, 王凡, 史艳翠, 等. 移动用户需求获取技术及其应用[J]. 软件学报, 2014, 25(3): 439-456)

[4] Jiang Wei, Zhang Li, Dai Yi, et al, Analyzing helpfulness of online reviews for user requirements elicitation [J]. Chinese Journal of Computers, 2013, 36(1): 119-131 (in Chinese)
(姜巍, 张莉, 戴翼, 等. 面向用户需求获取的在线评论有用性分析[J]. 计算机学报, 2013, 36(1): 119-131)

[5] Nguyen D C, Derr E, Backes M, et al. Short text, large effect: Measuring the impact of user reviews on Android app security & privacy [C] //Proc of 2019 IEEE Symp on Security and Privacy (SP). Piscataway, NJ: IEEE, 2019: 555-569

[6] Gao Cuiyun, Zeng Jichuan, Lyu M R, et al. Online app review analysis for identifying emerging issues [C] //Proc of the 40th Int Conf on Software Engineering (ICSE'18). New York: ACM, 2018: 48-58

[7] Hassan S, Shang Weiyi, Hassan A E. An empirical study of emergency updates for top Android mobile apps [J]. Empirical Software Engineering, 2017, 22(1): 36-44

[8] Wang Chong, Li Ju, Liang Peng, et al. Developers' eyes on the changes of apps: An exploratory study on app changelogs [C] //Proc of the 3rd Int Workshop on Crowd-Based Requirements Engineering (CrowdRE). Piscataway, NJ: IEEE, 2019: 207-212

[9] Nagappan M, Shihab E. Future trends in software engineering research for mobile apps [C] //Proc of the 23rd IEEE Int Conf on Software Analysis, Evolution, and Reengineering (SANER'16). Piscataway, NJ: IEEE, 2016: 21-32

[10] Nayebi M, Adams B, Ruhe G. Release practices for mobile apps—what do users and developers think? [C] //Proc of the 23rd IEEE Int Conf on Software Analysis, Evolution, and Reengineering (SANER'16). Piscataway, NJ: IEEE, 2016: 552-562

[11] Wang Chong, Wang Tao, Liang Peng, et al. Augmenting app review with app changelogs: An approach for app review classification [C] //Proc of the 31st Int Conf on Software Engineering and Knowledge Engineering (SEKE). Pittsburgh, PA: KSI Research Inc, 2019: 398-403

[12] Chen Jiaying, Yu Jiong, Yang Xingyao. A feature extraction based recommender algorithm fusing semantic analysis [J]. Journal of Computer Research and Development, 2020, 57(3): 562-575 (in Chinese)
(陈嘉颖, 于炯, 杨兴耀. 一种融合语义分析特征提取的推荐算法[J]. 计算机研究与发展, 2020, 57(3): 562-575)

[13] Mcilroy S, Ali N, Hassan A E. Fresh apps: An empirical study of frequently-updated mobile apps in the Google play store [J]. Empirical Software Engineering, 2016, 21(3): 1346-1370

[14] Nayebi M, Kuznetsov K, Chen P, et al. Anatomy of functionality deletion: An exploratory study on mobile apps [C] //Proc of the 15th Int Conf on Mining Software Repositories (MSR'18). New York: ACM, 2018: 243-253

[15] Maalej W, Kurtanovic Z, Nabil H, et al. On the automatic classification of app reviews [J]. Requirements Engineering, 2016, 21(3): 311-331

[16] McCann T. The Art of the App Store: The Business of Apple Development [M]. Hoboken, NJ: John Wiley & Sons, 2011

[17] Zhao Xinshu, Liu Jun, Deng Ke. Assumption behind intercoder reliability indices [J]. Annals of the International Communication Association, 2013, 36(1): 419-480

[18] 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)

[19] Lu Mengmeng, Liang Peng. Automatic classification of non-functional requirements from augmented app user reviews [C] //Proc of the 21st Int Conf on Evaluation and Assessment in Software Engineering (EASE'17). New York: ACM, 2017: 344-353

[20] Statista. Most popular Apple APP Store categories in August 2020, by share of available apps [EB/OL]. (2020-08)[2020-09-13]. https://www.statista.com/statistics/270291/popular-categories-in-the-app-store/

[21] Krippendorff K. Content Analysis: An Introduction to Its Methodology [M]. 2nd ed. Thousand Oaks, CA: Sage Publications, 2004

[22] Stemler S. An overview of content analysis [J]. Practical Assessment, Research, and Evaluation, 2001, 17(7): 137-146

[23] ISO. ISO/IEC 25010, Systems and software engineering—Systems and software Quality Requirements and Evaluation (SQuaRE)-System and software quality models [S]. Geneva, Switzerland: ISO, 2011

[24] Bird S, Klein E, Loper E. Natural Language Processing with Python [M]. Sebastopol, CA: O'Reilly Media, 2009

[25] Guzman E, Muhammad E H, Bruegge B. Ensemble methods for app review classification: An approach for software evolution [C] //Proc of the 30th IEEE/ACM Int Conf on Automated Software Engineering (ASE'15). Piscataway, NJ: IEEE, 2015: 771-776

[26] Li Ju. Analysis on the trends of mobile app updates using machine learning [D]. Wuhan: Wuhan University, 2019 (in Chinese)
(李桔. 基于机器学习的移动应用演化趋势分析[D]. 武汉: 武汉大学, 2019)

Zhong Renyi, born in 2000. Undergraduate student at the School of Computer Science, Wuhan University. Student member of CCF. His main research interests include requirements engineering and machine learning.

钟仁毅,2000 年生.武汉大学计算机学院本科生,CCF 学生会员.主要研究方向为需求工程与机器学习.

Wang Chong, born in 1981. PhD. Lecturer at the School of Computer Science, Wuhan University. Member of CCF, IEEE, and ACM. Her main research interests include requirements engineering and service-oriented software engineering.

王 翀,1981 年生.博士,武汉大学计算机学院讲师.CCF,IEEE 和 ACM 会员.主要研究方向为需求工程与面向服务的软件工程.

Liang Peng, born in 1978. PhD. Professor and PhD supervisor at the School of Computer Science, Wuhan University. Member of CCF, IEEE, and ACM. His main research interests include software architecture and requirements engineering.

梁 鹏,1978 年生.博士,武汉大学计算机学院教授,博士生导师.CCF,IEEE 和 ACM 会员.主要研究方向为软件体系结构和需求工程.

Luo Zhong, born in 1982. PhD. Associate professor at the Department of Naval Architecture Engineering, Naval University of Engineering. His main research interests include data-driven science and information engineering.

罗 忠,1982 年生.博士,海军工程大学舰船与海洋学院副教授.主要研究方向为数据驱动科学和信息工程.