计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.200100022

基于主题模型的 Ubuntu 操作系统缺陷报告的分类及分析

周 凯 任 怡 汪 哲 管剑波 张 芳 赵言亢
国防科技大学计算机学院 长沙 410073
(alidechengbao@163.com)

摘 要 软件缺陷(Bug)是造成系统失效的主要原因之一,为了更好地开发软件与修复软件失效,需要对缺陷的分布等特征有更好的理解。Ubuntu 是一款得到广泛应用的开源软件,也是 Linux 操作系统当前在全球最成功的发行版之一。利用缺陷报告来发掘软件缺陷特征,对缺陷进行合理分类并分析操作系统常见缺陷的分布规律及特点,对于基于 Ubuntu 的国产混源操作系统开发、测试及维护过程中的代码质量分析及提升具有重要参考价值。首先,获取 Launchpad 上 32805 份 Ubuntu 操作系统的缺陷报告。然后,采用主题模型分析 Ubuntu 上常见的缺陷,并结合操作系统的组成特点将其分为内核相关异常、桌面环境异常、网络相关异常、硬件驱动相关异常以及上层应用及开发环境相关异常。进一步,利用 F1 值对分类结果进行评估,结果表明缺陷分类具有较好的准确率。最后,通过分析缺陷报告统计结果得到 Ubuntu 操作系统的近期缺陷的一般分布规律和特点,同时通过缺陷报告的分类结果,得到了有助于进一步认知 Ubuntu 操作系统缺陷的相关发现和结论。

关键词:Ubuntu 操作系统;LDA 模型;缺陷分类;缺陷报告分析
中图法分类号 TP311

Classification and Analysis of Ubuntu Bug Reports Based on Topic Model

ZHOU Kai,REN Yi,WANG Zhe,GUAN Jian-bo,ZHANG Fang and ZHAO Yan-kang
College of Computer,National University of Defense Technology,Changsha 410073,China

Abstract Software bug is the main cause of system failure. Better understanding of bug characteristics is needed to develop software and repairing failure. Ubuntu is one of the most successful distributions of the Linux operating system and also a popular open-source software platform in the world. Using bug reports to discover software bug characteristics,analyze and classify reasonably common bugs of the operating system,has important guiding value for the bug analysis during the development,testing and maintenance of the domestic mixed source operating system based on Ubuntu. Firstly,32805 bug reports are downloaded from launchpad through crawler. Though analyzing the common bug of Ubuntu by using topic mode,bug are divided into 5 categories:kernel related,desktop environment,network,hardware driver related anomaly and the abnormal system management based on Ubuntu operating system composition and experience. Next,the results of the classification are evaluated through F1 value. Finally,the general distribution rules and characteristics of the recent bugs in the Ubuntu operating system are obtained by analyzing the statistical results of the bug reports. At the same time,through further analysis of the classification results,relevant findings and conclusions that help to further understand the bugs of Ubuntu operating system are obtained.

Keywords Ubuntu operating system,Latent dirichlet allocation model,Bug classification,Bug report analysis

1 引言

由于软件开发的复杂性,缺陷是不可避免的。对于大规模软件而言,由于其代码量庞大,在软件开发和维护过程中,验证代码的正确性以及进行缺陷定位、修复等活动往往存在较大难度。以 Linux 通用操作系统为例,至 2019 年,仅内核代码已超过 2500 万行[1],且代码之间依赖关系复杂,而传统的静态分析等代码检查工具可以处理的软件规模有限。另一方面,开源软件重用已成为软件开发的有效途径之一。2018 年,知名开源软件分析公司 Blackduck 对超过 1100 个商业代码库中的匿名数据进行了开源代码的分析和审计,结果显示,96%的被扫描代码中存在开源组件[2]。因此,通过分析开源

到稿日期:2020-01-05 返修日期:2020-05-30 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:国家自然科学基金(61872444);国家核高基重大专项(2017ZX01038104-002)
This work was supported by the National Natural Science Foundation of China(61872444) and National High-End Generic Chips and Basic Software Project of China(2017ZX01038104-002).
通信作者:任怡(renyi@nudt.edu.cn)

社区缺陷管理系统中的缺陷报告，发掘并获取代码缺陷分布规律、种类等缺陷特征，有助于提高对缺陷分布规律和现象的认知，也为进一步建立操作系统缺陷知识库提供了分类基础和相关知识积累，对于辅助缺陷定位、修复等工作具有重要参考价值和意义。

国产操作系统开发中往往存在定位缺陷困难、某些缺陷的软硬件根源界限不易界定等问题。为了更好地理解缺陷的特点及其分布规律等，本文以 Ubuntu 近两年的缺陷报告数据为研究对象，运用主题模型技术发掘 Ubuntu 操作系统的常见缺陷，以期获取对基于 Ubuntu 的国产混源操作系统缺陷分析具有指导价值的相关信息和结论。同时，对操作系统的缺陷进行统计分析和分类研究，有助于开发人员提升对其正在开发的系统的缺陷认知，并且可以为将来开发类似系统的人员提供进一步的指导。

为了有效地保证软件的质量，许多项目使用缺陷报告来收集和记录开发人员、测试人员和用户报告的缺陷[3]。文献[4-9]致力于研究重复或者相似缺陷报告的检测。其中，Vincent 等[9]对 Debian 和 Ubuntu 的缺陷仓库数据进行了实证研究，利用已有重复缺陷报告检测技术发现了两个发行版本之间的重复缺陷报告，并评估了开发人员为此损耗的时间。还有一些研究利用缺陷报告来发掘软件缺陷特征[10-12]。Li 等[10]研究了开源软件 Mozilla 和 Apache HTTP Server 的缺陷报告，手工分析了 362 个运行时缺陷并对其进行了分类，最后用机器学习文本分类技术对约 29 000 个缺陷进行自动分类，以验证分类的准确性。Tan 等[11]对 Linux kernel，Mozilla 及 Apache 进行了实证研究，指出了诸如语义缺陷是造成软件失效的主要原因、安全类缺陷比例正在增加等一系列对开源软件开发者和使用者具有指导意义的问题。上述工作主要围绕开源软件的缺陷特征进行研究，目前对操作系统缺陷的研究较少。文献[12]利用主题模型技术对 Ubuntu 的缺陷报告进行了分析，将 Ubuntu 常见缺陷初步分为维护、图形用户接口及运行时 3 类，并参照 Ubuntu 软件中心的组织层次对包含缺陷数量最多的前 100 个软件包的缺陷做了进一步细分。与文献[12]关注公共领域的缺陷特征不同，本文在缺陷分类中结合了国产操作系统的开发经验及对 Linux 操作系统组成的认知，分类过程与结果参考操作系统的组成，并对分类结果进行了细分，有助于操作系统的开发及维护人员定位缺陷位置。

本文以 Ubuntu 近期的缺陷报告为研究对象，结合课题组在国产操作系统研发及应用过程中对系统功能及软件包组成的较为深入的了解和认知，重点对 Ubuntu 近两年的缺陷进行了分析，以主题模型作为主要分析辅助工具，参照操作系统内核及核外功能组成，将已有缺陷分为内核相关异常、桌面环境相关异常、网络相关异常、硬件驱动相关异常以及上层应用及开发环境相关异常，并利用 F1 值评估分类效果；同时通过统计分析缺陷报告发现 Ubuntu 操作系统近期的缺陷分布规律；为了更好地理解每个类别的缺陷组成和特点，又将每个类别展开进行了细分及进一步分析。上述工作对国产操作系统的缺陷认知与修复具有重要的指导意义。

2 缺陷报告及数据采集

2.1 缺陷报告

Ubuntu 是一个构建在 Linux 内核之上的开源操作系统。Launchpad 是一个开源软件项目协作和托管平台，集成了约 43 000 个项目，近 180 万份缺陷报告。Ubuntu 使用 Launchpad 来报告、记录和管理它的缺陷。缺陷报告是描述软件缺陷的软件文档，由开发人员、测试人员或最终用户提交。表 1 列出了常见的缺陷报告形式的主要字段。

表 1 缺陷报告部分字段描述

Table 1 Part of field description of bug report

字段	描述
Title	缺陷报告名称
Description	关于缺陷的详细描述
Important	重要性(缺陷分配的优先级)
Affects	缺陷涉及到的软件包
Status	缺陷的生命周期状态
Heat	热度(缺陷受关注程度)

在缺陷报告中，Status 用于描述缺陷报告的生命周期状态。图 1 是 Launchpad 中的缺陷生命周期状态转换图。其中，New 是一份缺陷报告生命周期的开始；如果被标为 Incomplete 则表示该报告没有提供对问题的清晰的描述，需要缺陷的报告者额外增加对该缺陷的描述；Triaged 则意味着工程师有足够的信息来修复该缺陷；Won't Fix 表示尽管工程师理解该报告及其背后的原因，但其出于某些原因明确决定不修复它；Opinion 表示尽管该缺陷已经属于关闭状态，但仍然可以自由地在该缺陷报告的讨论区讨论[13]；Confirmed 状态表示缺陷报告包含足够多的信息可以使这个缺陷复现；In progress 表示该缺陷正在被修复。

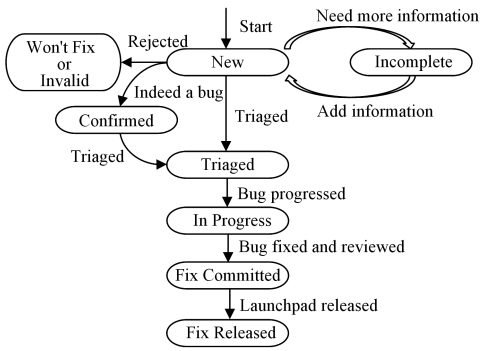

图 1 Launchpad 中缺陷报告的生命周期

Fig. 1 Life-cycle of bug report in Launchpad

2.2 缺陷报告数据的采集

本文以 Ubuntu 近期的缺陷报告为研究对象。为此，我们编写了一个 Web 爬虫程序来遍历和存储 Launchpad 上 2016 年 8 月到 2019 年 5 月期间 Ubuntu 的缺陷报告[14]。

Launchpad 系统中一项缺陷报告中的缺陷可以同时影响多个软件包并在不同软件包中具有不同的状态和重要性。对于这样的缺陷报告，本文将其划分为多份缺陷报告，即一份缺

陷报告中的缺陷只对应一个软件包。通过这样的方式，最终得到 32805 份缺陷报告。这些缺陷报告可以分为两类：一类缺陷报告仅涉及 Ubuntu 系统，如缺陷报告 # 1050358，其影响到 Ubuntu 项目中 3 个软件包，但只涉及 Ubuntu 项目；另一类缺陷报告会同时涉及 Ubuntu 项目与 Ubuntu 项目之外的其他项目，如缺陷报告 # 194904，其不仅影响到 Ubuntu 项目，同时还涉及到 gedit 项目。本文通过信息检索技术，删除涉及非 Ubuntu 项目的缺陷报告，最终保留第一类和第二类中 Ubuntu 项目相关的缺陷报告，得到实验用的 32641 份报告。

3　基于 LDA 的常见缺陷分析

3.1　主题模型

主题模型在机器学习和自然语言处理等领域中是一种在一系列文档中发现抽象主题的统计模型。主题模型通过词项在文档级的共现信息中抽取出语义相关的主题集合，并能将词项空间中的文档变换到主题空间，得到文档在低维空间的表达。潜在狄利克雷分布（Latent Dirichlet Allocation，LDA）模型[15]是一种广泛使用的主题模型。LDA 是一个文本-主题-词的三层贝叶斯产生式模型，采用了词袋（Bag of Words）的方法，这种方法将每一个文本集合视为一个词频向量，从而将文本信息转化为易于建模的数字信息。直观来讲，如果一份缺陷报告有一个或者多个主题，那么一些特定词语会更频繁地出现，如“icon”和“display”等与显示相关的词会更频繁地出现在有关显示异常的缺陷报告中。因此，本文使用 LDA 从大规模的缺陷报告中提取 Ubuntu 操作系统的主要缺陷并将其分类。

3.2　基于 LDA 的缺陷分析框架设计

图 2 给出了基于 LDA 常见缺陷分析的总体框架，它接收缺陷报告的自然语言文本部分（缺陷报告的 Title 与 Description）作为输入数据集。该框架首先对数据进行预处理，提取文本中的单词。然后，在数据集上构建 LDA 模型，得到 n 个主题。对于每个主题，提取该主题中的前 k 个关键词（k 值可以预先设置，k 值越大，实验产生的结果信息越多，主题就越发散；k 值越小，主题越收敛。通过多次实验，设置较为合理的 k 值）。最后，选择能够反映主题含义的典型主题并结合已有知识对缺陷报告进行分类汇总。

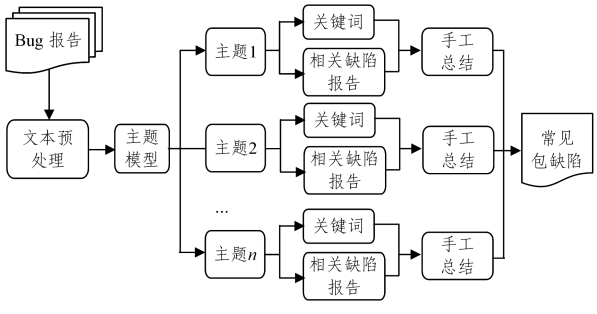

图 2　基于 LDA 缺陷分析的总体框架

Fig. 2　Overall framework of our approach

将预处理分为 3 部分：分词（只保留英文单词）、去除停用词（即去掉文中对主题无关的词语，如“is”“what”“we”等与主题无关的词，这里使用的是 Snowball 项目[16]的停词表）、提取词干（即将不同时态的词归并为一个，本文使用的是 Porter Stemmer 算法）。为减小手工总结可能带来的噪音影响，由实验室团队的多名不同成员独立进行分析，并归纳汇总。

3.3　主题数的选取依据

一致性度量（Coherence Measure），即根据主题的可理解性进行评估，应用于由主题模型计算出的主题质量的评估。其中，U_Mass Coherence[17]是一种基于单词共现信息的度量方法；C_V Coherence 被证明相比其他广泛使用的主题一致性度量方法有更优的表现。本文使用一致性度量模型（Coherence Model）中的 C_V Coherence 和 U_Mass Coherence 来评价主题模型的主题数[18]，两个指标的数值越大，主题模型的效果就越好[18]。如图 3 和图 4 所示，横轴代表实验中选取的不同主题数，纵轴代表随着主题数的变化，U_Mass Coherence 和 C_V Coherence 的变化趋势。评估实验所用数据集与上文实验数据集相同，结果显示选取主题数为 10 会有更优的效果。因此，本文实验最终选定的主题数为 10。

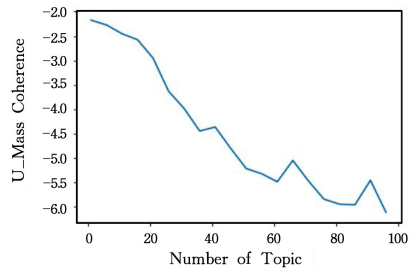

图 3　不同主题数下的 U_Mass Coherence

Fig. 3　U_Mass Coherence with difference topic number

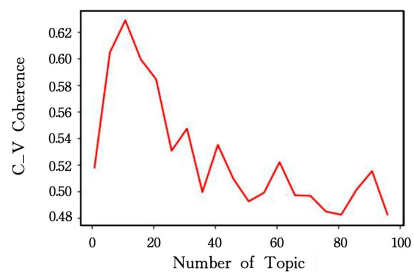

图 4　不同主题数下的 C_V Coherence

Fig. 4　C_V Coherence with difference topic number

3.4　基于 LDA 的 Ubuntu 缺陷分类

使用主题模型最终得到 n 个主题（$n=10$），以及每个主题下的 k 个关键词（$k=10$）。通过查看关键词以及与主题相关的缺陷报告，将主题进行汇总。

表 2 列出了训练后的一个主题分布的部分结果。其中，右边列表示主题分布及该主题概率最高的前 10 个关键词与其概率。可以看到，在第 1 个主题中，window，display，screen，menu 等都是与显示相关的词，可以归纳出这个主题下的缺陷报告所描述的是关于桌面环境显示的缺陷（由于使用 Stemmer 算法，因此部分结果不是完整的词语）。

表 2 主题分布的部分结果示例

Table 2 Example of topic distribution results

主题序号	关键词及概率
0	'0.041 * "window"+0.033 * "display"+0.030 * "app"+0.028 * "menu"+0.026 * "show"+0.025 * "support"+0.025 * "icon"+0.022 * "open"+0.020 * "applic"+0.016 * "screen"'
1	'0.140 * "crash"+0.055 * "file"+0.037 * "sigsegv"+0.026 * "instal"+0.016 * "miss"+0.015 * "control"+0.015 * "directori"+0.013 * "sigabrt"+0.012 * "mount"+0.011 * "partit"'
2	'0.055 * "fail"+0.048 * "kernel"+0.037 * "error"+0.021 * "size"+0.021 * "modul"+0.020 * "build"+0.016 * "invalid"+0.015 * "image"+0.015 * "password"+0.015 * "corrupt"'
3	'0.032 * "test"+0.031 * "fail"+0.031 * "sound"+0.023 * "kernel"+0.022 * "driver"+0.021 * "broken"+0.019 * "hang"+0.017 * "suspend"+0.016 * "reboot"+0.016 * "dep8"'
4	'0.059 * "start"+0.039 * "depend"+0.028 * "remove"+0.026 * "freez"+0.024 * "file"+0.021 * "video"+0.019 * "key"+0.017 * "sigabrt"+0.016 * "server"+0.015 * "log"'
5	'0.089 * "configure"+0.082 * "package"+0.067 * "fail"+0.058 * "update"+0.038 * "state"+0.036 * "bad"+0.035 * "attempt"+0.033 * "inconsist"+0.032 * "depend"+0.032 * "reinstal"'

结合课题组在国产操作系统研发及应用过程中对系统功能及软件包组成的了解和认知，本文提出面向修复的缺陷分类准则。已有的工作[12]将 Ubuntu 缺陷分为桌面环境、运行时与维护时 3 类。这样的分类方式会造成缺陷分类有较大的重叠，即一些缺陷可以归结到两个以上的类别，同时运行时、维护时这两个类别覆盖的技术范畴较为笼统，在分配缺陷时不能直接分配给其相关领域的工程师，对缺陷修复的指导价值有待提高。例如，表 3 所列的 Bug # 1726402（数字对应 Launchpad 中缺陷报告的唯一 ID）描述的是谷歌浏览器图标缺失问题，从分配缺陷的效率考虑，本文将类似的缺陷归为桌面环境异常相关，有助于缺陷分配人员快速将缺陷分配给领域内的工程师。

Linux 自底向上可分为 4 个层次：内核、基础运行环境、桌面环境、应用软件。文献[11]将 Linux 内核缺陷按操作系统组件分为驱动、内核、网络、文件系统及体系结构。

表 3 缺陷分类例子

Table 3 Example of bug classification

缺陷报告 ID	缺陷报告 Title	已有分类[12]	本文分类
# 1720510	GUI session crashes on boot	GUI/Runtime	桌面环境
# 353053	package gxine 0.5.903-4ubuntu1 failed to install/upgrade	Runtime/Maintenance	上层应用及开发环境

如表 4 所列，本文结合已有认知与修复分配原则，提出将 Ubuntu 操作系统的缺陷分为以下 5 类：内核异常、桌面环境异常、网络异常、硬件及驱动异常、上层应用及开发环境异常。

表 4 Ubuntu 缺陷分类及其关键词

Table 4 Bug classification and corresponding keywords

类别	描述	关键词	典型缺陷报告 ID
内核	与内核异常有关	kernel termin process call	# 1635597
桌面环境	与图形显示异常有关	gnome screen display menu icon show	# 1720510
网络	与网络配置和环境异常有关	wifi failed network server	# 1829838
硬件驱动	与硬件驱动异常有关	driver sound disk keyboard nvidia	# 1827446
上层应用及开发环境	基础应用、三方应用异常等	update install depend release package	# 353053

4 实验结果评估与分析

4.1 实验结果评估

本文使用 F-measure(F1)对分类结果进行效果评估。其中，准确率用于检验分类效果的准确性，召回率用于检查分类效果的完整性，F1 值作为准确率和召回率的调和平均可以对二者进行整体评价。本文使用这三者作为评价指标，具体公式如下：

$$P=\frac{T_+}{T_+ + F_+} \tag{1}$$

$$R=\frac{T_+}{T_+ + F_-} \tag{2}$$

$$F1=\frac{2PR}{R+P} \tag{3}$$

其中，T_+ 表示成功分类到主题的缺陷报告数量，F_+ 表示错误分类到该主题的缺陷报告数量，F_- 表示本应该分类到该主题却被错误分类到其他主题的报告数量。随机抽取 500 个缺陷报告样本，验证其基于 LDA 的分类结果并与实际分类结果进行对比，得到相应的 P，R，$F1$ 值，重复实验 5 次，取各次结果的平均值得到最终结果。

表 5 列出了基于 LDA 与操作系统组成的分类效果的评估。内核相关、桌面环境相关以及硬件相关缺陷的 $F1$ 值都超过了 0.8，表明分类效果较好。

同时，分类为桌面环境的 P 值较低（P 值为 0.583），其主要原因在于其他类型的缺陷可能会导致桌面环境的异常，最终这些缺陷报告会有较大的概率被归为桌面环境异常。

表 5 分类评估

Table 5 Evaluation of classification

分类	P	R	$F1$
内核	0.846	0.852	0.849
桌面环境	0.583	0.824	0.683
网络	0.875	0.777	0.823
硬件驱动	0.857	0.752	0.801
上层应用及开发环境	0.897	0.656	0.759

4.2 缺陷分布的一般规律及特点

检查缺陷以确定它们的状态通常是一项庞大而又必不可少的软件开发任务[19]。对缺陷分布规律和现象的分析，有助于为建立操作系统缺陷知识库提供分类基础和相关知识积

累。本文从重要性、状态、热度 3 个方面对 Ubuntu 缺陷报告数据进行了统计。

(1)不同重要性的缺陷分布

如图 5 所示，Undecided 类别的缺陷报告的数量最多，占总数的 76.78%，这与缺陷被发现的时间相对较短有关。Critical 类别的缺陷报告数量非常少，大约占所有缺陷的 0.66%，占已确认的缺陷报告(即去掉 Undecided 部分)的 2.63%。

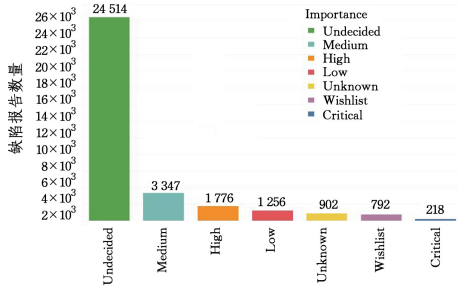

图 5　依据不同 Importance 的缺陷报告数量分布

Fig. 5　Number of bug reports with different Importance

(2)不同状态的缺陷分布

如图 6 所示，在 Ubuntu 缺陷生命周期中，缺陷报告数量最多的前 5 个状态是 New，Confirmed，FixReleased，Triaged，Incomplete。其中，待进一步处理的缺陷 Undecided，In progress，Confirmed，Triaged 共占总数的 88.89%，说明大多数缺陷报告尚未得到及时处理，缺陷的处理能力需要进一步提高。

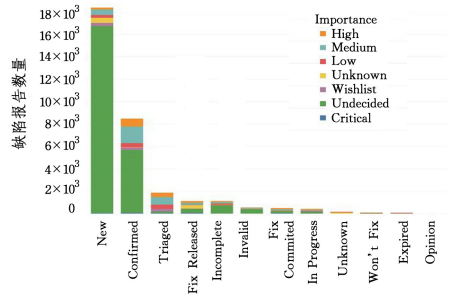

图 6　依据不同重要性的 Status 的缺陷报告数量

Fig. 6　Distribution of Status with different importance

(3)不同热度的缺陷分布

热度是 Launchpad 中用于反映缺陷的受关注程度。如图 7 所示，横轴代表不同的热度，其中热度为 6 的缺陷报告数量最多，为 16579 份，其次为热度 8 和热度 10，分别有 5072 份和 2909 份。当热度超过 34 后，每个热度值的缺陷报告的数量都普遍小于 100(除了热度 44 的缺陷报告数量为 118，热度 56 的缺陷报告数量为 101)。

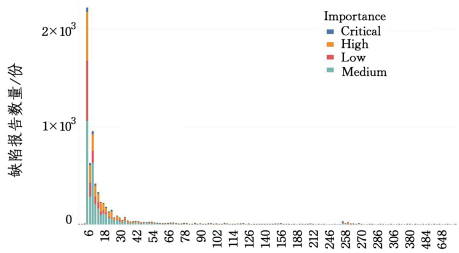

图 7　依据不同重要性的 Heat 的缺陷报告数量

Fig. 7　Distribution of heat with different important

在实验数据集中，Ubuntu 中一个缺陷平均影响到 1.2 个软件包，共计 6201 个 Ubuntu 软件包受到影响，其中有 3749 个软件包只受到 1 次影响，有 5767 个软件包受到影响的次数小于 11。图 8 给出了受影响次数超过 40 的软件包的分布，即存在缺陷较多的软件包的分布。

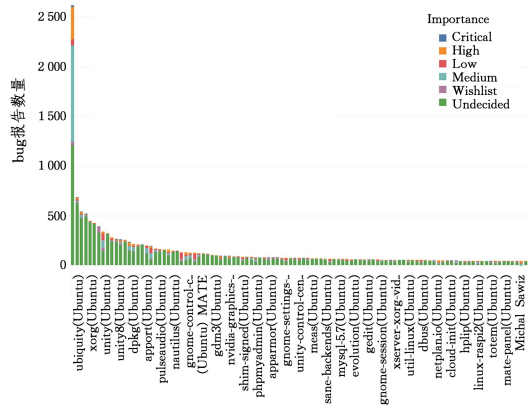

图 8　缺陷数量大于 40 的软件包

Fig. 8　Software package with more than 40 bugs

基于 Ubuntu 操作系统缺陷报告的初步统计分析，可以得出以下结论：

(1)状态为 New 的缺陷报告占总数的 56.08%，而重要性为 Undecided 的状态占总数的 74.7%。通过查阅缺陷报告发现：相当部分的缺陷报告中，缺陷的生命周期状态发生了变更，但其重要性并没有改变，如 Bug＃1610828，＃1718254 等。因此，Launchpad 应该采取一些规范的方法来应对这一问题，如利用脚本使得缺陷报告的状态与重要性可以同步发生变更等。

(2)Ubuntu 操作系统中缺陷报告热度跨度大，最小的热度值为 2，最大的热度值为 1506。在重要性为 Critical 的缺陷报告中，热度超过 56 的报告占总数的 15.36%。这表明 Heat 也许并不能作为缺陷重要程度的参考度量。建议在 Heat 的算法中适当增加一些权重，以期减小热度的跨度并用来更好地衡量缺陷的重要程度。

对操作系统缺陷数据分布规律的统计，可以辅助以 Linux 为内核的开源操作系统开发人员以及操作系统开源社区维护人员更好地认识操作系统的常见缺陷、规避类似缺陷、了解已有缺陷分类，同时也对缺陷管理社区的优化给出了建议，为未来的系统软件缺陷研究工作提供了进一步的指引。

4.3　基于主题模型的缺陷分析

图 9 给出了基于主题模型的缺陷分类及其比例。其中，上层应用及开发环境相关缺陷占全部缺陷的 41%，其次是桌面环境和内核相关的缺陷，分别占 27%和 17%。

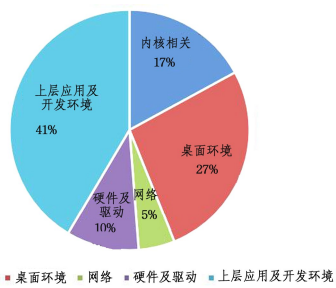

图 9　基于主题模型的缺陷分类

Fig. 9　Classification of bug based on topic model

为了更好地理解操作系统的缺陷分类，下面对每个缺陷类别做了进一步分析，针对每个小类别给出了缺陷报告示例。

(1)Kernel 相关的缺陷

1)操作系统内核本身的缺陷，约占当前类别的 9.94%。这类缺陷主要发生在 mm 目录(内存单元代码)、fs 目录(内核支持的文件系统代码)以及 driver 目录(硬件驱动程序代码)。

-kernel BUG at fs/ext4/inode.c:2679!

2)内核相关的系统异常，约占当前类别的 18.61%。

-failed to load kdump kernel

3)内核相关兼容性异常，占当前类别的 32.6%。主要包括内核与硬件驱动的兼容异常、内核与软件包的兼容异常。

-Ubuntu 16.04 with kernel 4.4.0-148 hangs on boot on a ThinkPad L460 while 4.4.0-145 doesn't

可以看出，在内核缺陷中相当一部分为内核与其他软硬件的兼容性异常，因此开发者应该提供更多的操作系统升级时的兼容性保证。同时，内核的缺陷经常会造成系统的崩溃、中止、悬停等一系列严重的问题。

(2)桌面环境相关缺陷主要包括 3 类

1)桌面环境软件包相关的缺陷，占当前类别的 37.16%。在 Ubuntu 操作系统中主要表现为 Gnome 相关的缺陷。

-gnome-control-center not opening in Ubuntu 19.04

2)锁屏相关，即缺陷的发生与锁屏有关，约占 6.01%。

-Lockscreen doesn't turn off the screen

3)图形接口服务器相关的缺陷，约占 4.55%。

-Wayland session freezes

这部分缺陷通常会通过桌面环境的图标缺失、图像显示不全等直观的方式表现出来，会给使用者带来不良的交互体验。其中，由桌面管理软件包缺陷造成的影响占了相当大的一部分，因此诸如软件包的二次开发过程的测试工作等应该得到更多关注。

(3)网络类别相关的缺陷

1)网络管理软件包相关缺陷，如 Network-manager，占当前类别的 17.66%。

-Network-Manager Incorrect settings during Installation

2)VPN 相关缺陷，约占 14%。

-bionic:VPN not working at all

网络类别的缺陷，造成的影响通常较小，如导致连不上网等一些并不会严重影响系统性能的问题。

(4)硬件驱动相关的缺陷

1)显卡驱动相关缺陷，如 Nvidia 驱动，占当前类别的 19.72%。

-Nvidia driver does not build on Xenial LTS 4.15

2)关于 USB 的缺陷，占 10.14%。

-HP Thunderbolt 3 Dock (90W):USB does not work

3)声卡相关缺陷，占 11.89%。

-PCI/internal sound card not detected

4)网卡驱动有关缺陷，占 2.06%。具体表现为网络协议驱动相关缺陷，因此与网络类别区分开。

-b44 Ethernet driver null pointer during initialization

在硬件驱动相关的缺陷中，与 Nvidia 相关的驱动最多，其次为声卡驱动与 USB 驱动相关的缺陷，网卡驱动缺陷在当前类别中的占比不是很高。

(5)上层应用及开发环境相关的缺陷

1)数据库相关缺陷，约占当前类别的 5.17%。

-mysql_ssl_rsa_setup generates server-key.peminacessible by mysqld

2)程序开发软件包相关异常，占 27.92%。在 Ubuntu 中表现为 php，java，python 等编程语言软件包相关缺陷。

-Python venv does not copy the activate scripts

3)办公软件相关缺陷报告，约占 6.59%。

-LibreOffice Impress 6.2.2.2 Embedded Video Problem

在被分析的 Ubuntu 缺陷报告中，上层应用及开发环境相关的缺陷、桌面环境相关的缺陷比例最高，因此操作系统软件包开发者对这两类缺陷应给予更多关注。与此同时，兼容性异常，包括系统与软件之间、开源软件包之间的相互适配问题，在 Ubuntu 操作系统中是存在于各个类别中的比较常见的问题。这与 Mohamed 的工作[20]结论相符合，他们的架构分析得出的结论表明，内核模块本身并不是内核缺陷的主要制造者。实际上，模块之间的相互依赖和交互是系统中关键缺陷的重要来源。因此，在国产操作系统未来的研发工作中，需要进一步关注兼容 API、应用兼容性评估方法及工具、应用平滑迁移、服务无缝更新或升级等，同时软件包的开发接口、软件包的依赖问题也需要进一步规范。

结束语 本文对 Ubuntu 操作系统共计 32805 份缺陷报告进行了实证研究，提出了一种基于 LDA 并结合操作系统组成的 Ubuntu 操作系统缺陷分类方法，将 Launchpad 中的缺陷分为 5 类，分别是内核异常、桌面环境异常、网络相关、硬件驱动相关异常以及上层应用及开发环境相关异常。本文对分类效果进行了评估，并对每个缺陷类别做了进一步细分，观察到 Ubuntu 操作系统的一些缺陷分布规律及特点，这些发现和结论对未来操作系统的缺陷研究工作具有重要的指导意义。

主题模型对大规模文本进行聚类，实现了文本信息的抽象化分析，但由于不同缺陷报告往往由不同的人员完成，其描述方式和语言风格各异，分类效果有一定局限性。未来，我们将进一步深化目前的工作，如将研究的缺陷报告的时间跨度拉大，选取更多的数据；同时，将进一步研究大规模的系统软件缺陷报告，运用更多的机器学习或者 NLP 的方法来进一步发现其中的规律，以便更好地发现系统软件的缺陷规律。

参 考 文 献

[1] GitStatus-linux [EB/OL]. (2018-09-15) [2019-10-05]. https://phoronix.com/misc/linux-20180915/index.html.

[2] Synopsys. 2019 Open Source Security and Risk Analysis (OSSRA) Report[OL]. (2019-04-02). https://www.synopsys.com/content/dam/synopsys/sig-assets/reports/rep-ossra-19.pdf.

[3] LI H, GAO G, CHEN R, et al. The Influence Ranking for Testers in Bug Tracking Systems[J]. International Journal of Soft-

ware Engineering & Knowledge Engineering，2019，29(1)：93-113.

[4] DANG Y，WU R，ZHANG H，et al. Rebucket：a method for clustering duplicate crash reports based on call stacksimilarity [C]// Proceedings of the ACM/IEEE International Conference on Software Engineering. Zurich，2012：1084-1093.

[5] BETTENBURG N，PREMRAJ R，ZIMMERMANNT，et al. Duplicate bug reports considered harmful... really? [C]// Proceedings of the IEEE International Conference on Software Maintenance. Beijing，2008：337-345.

[6] CAVALCANTI Y，MOTA SILVEIRA NETO P，LUCRDIO D，et al. The bug report duplication problem：an exploratory study [J]. Software Quality Journal，2013，21：39-66.

[7] CHEN M，HU D Y，WANG T，et al. Using Document Embedding Techniques for Similar Bug ReportsRecommendation [C]// 9th IEEE International Conference on Software Engineering and Service Science (ICSESS 2018). Beijing，2018.

[8] HU D Y，CHEN M，WANG T，et al. Recommending Similar Bug Reports：A Novel Approach Using Document Embedding Model[C]// APSEC. 2018：725-726.

[9] BOISSELLE V，ADAMS B. The impact of cross-distribution bug duplicates，empirical study on Debian and Ubuntu[C]// IEEE International Working Conference on Source Code Analysis & Manipulation. IEEE，2015.

[10] LI Z，TAN L，WANG X，et al. Have Things Changed Now? An Empirical Study of Bug Characteristics in Modern Open Source Software[C]// Workshop on Architectural & System Support for Improving Software Dependability. DBLP，2006：25-33.

[11] TAN L，LIU C，LI Z，et al. Bug characteristics in open source software[J]. Empirical Software Engineering，2014，19(6)：1665-1705.

[12] REN X，HUANG Q，XIA X，et al. Characterizing Common and Domain-Specific Package Bugs：A Case Study on Ubuntu[C]// 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC). IEEE，2018：426-431.

[13] Bug life cycle [EB/OL]. (2009-03-03) [2019-10-05]. https://dev. Launchpad. net/BugTriage/Draft?#Bug_life_cycle.

[14] Launchpad [EB/OL]. [2019-12-20]. https://bugs. launchpad-. net/ubuntu.

[15] BLEI D M，NG A Y，LATENT M I. Dirichlet allocation[J]. Journal of MachineLearning Research，2003，3：993-1022.

[16] Snowball [EB/OL]. (2001-11) [2019-10-06]. http://snowball. tartarus. org/algorithms/english/stop. txt.

[17] MIMNO D M，WALLACH H M，TALLEYE M，et al. Optimizing Semantic Coherence in Topic Models[C]// Proceedings of the Conference on Empirical Methods in Natural Language Processing(EMNLP '11). 2011：262-272.

[18] ROEDER M，BOTH A，HINNEBURG A. Exploring the space of topic coherence measures[C]// Proceedings of the Eighth ACM International Conference on Web Search and Data Mining. 2015：399-408.

[19] GUO S，CHEN R，LI H，et al. Identify Severity Bug Report with Distribution Imbalance by CR-SMOTE and ELM[J]. International Journal of Software Engineering and Knowledge Engineering，2019，29(2)：139-175.

[20] AHMED M F，GOKHALE S S. Linux bugs：Life cycle，resolution and architectural analysis[J]. Information and Software Technology，2009，51(11)：1618-1627.

ZHOU Kai，born in 1996，postgraduate，is a member of China Computer Federation. His main research interests include system software and software engineering.

REN Yi，born in 1977，Ph. D，researcher. Her main research interests include operating system，cloud computing and virtualization，distributed computing，large-scale mixed-source software code feature analysis and so on.