 Contents

DoPDF

PDF

Q

T

软件开发人员通常渴望知道用户使用/喜爱他们软件的哪些部分。根据一项涵盖 4,000 名 Microsoft 工程师的调查,“软件产品的哪些部分(方面)最受客户使用和/或喜爱?”这个问题。在开发人员提出的前 145 个问题中排名第二[5]。这个问题需要开发人员分析对不同软件方面的偏好和意见。

用户评论是软件开发了解用户需求、偏好和抱怨的重要渠道[21], [31]。通过分析用户评论,开发人员可以评估他们的产品,确定用户的偏好[21],并改进软件维护和演化任务[33]。

然而,理解软件评论是非常具有挑战性和乏味的。首先,用户评论量太大,无法人工检查。开发人员每天都会收到成百上千条评论[10], [31]。鉴于大量评论,他们需要阅读并手动将评论分类为投诉或新功能请求[30]。这样的过程非常耗时且乏味。另一方面,用户评论的种类太多,需要加以区分[31]。它们可以是新功能请求、错误报告、表扬或投诉。不同类型的评论针对不同的任务和开发人员[30]。例如,赞扬评论可能对软件测试没有价值,但对产品评估可能是必不可少的。报告错误的评审对于需求分析并不重要,但对于软件测试可能至关重要。鉴于数百万条评论,开发人员必须首先手动对它们进行分类[30]。

提出了一些用于软件用户评论总结的工具。例如,陈等人。[10]通过分类技术过滤非信息性评论,并应用 Latent Dirichlet Allocation (LDA) [6]来总结信息性评论的主题。傅等。[15]过滤评级不一致的评论,这些评论的情绪与回归模型的评级不同。他们还应用 LDA 来总结剩余评论中的主题,并显示不同主题的评分趋势。雅各布等人。[21]通过语言规则过滤请求新特征的评论,并用LDA总结请求的关键词。这些工具总结了信息丰富且可靠的评论。然而,他们使用的 LDA 模型基于词袋假设,没有考虑句子结构和语义。这种假设对于表现出多种目的(例如,方面评估和功能请求)和情绪的软件评论可能是有问题的。由于这些工具混合了方面和意见,并且混合了与不同类别相关的主題,因此它们无法有效地衡量用户对各个方面的情绪。

为了解决这些限制,我们提出了 Software User Review Miner (SUR-Miner),这是一个可以总结用户对相应软件方面的情绪和意见的框架。SUR-Miner 没有将评论视为词袋,而是充分利用软件用户评论单调的结构和语义,直接根据预定义的句型从评论句子中解析出 aspect-opinion 对。然后,它分析每个评论句子的情绪,并将情绪与同一句子中的方面-意见对相关联。最后,它通过聚类具有相同方面的方面-意见对来总结软件方面。

我们根据 Swiftkey、Camera360、WeChat 和 Templerun2 等 17 个 Android 应用程序的近期用户评论,对 SUR-Miner 的性能进行了实证评估。我们通过 F1-score 衡量关键过程(即分类、方面意见提取和情感分析)的性能,这是文本挖掘文献 [14]、[38] 中常见的准确性衡量标准。结果表明,SUR-Miner 生成可靠的摘要,评论分类、方面意见提取和情感分析的平均 F1 分数分别为 0.75、0.85 和 0.80。SUR-Miner 的最终方面比最先进的技术更准确、更清晰,其 F1 得分为 0.81,高于 ReviewSpotlight (0.56) 和 Guzmans 的方法 (0.55)。

作为概念验证应用程序,我们设计了两个交互图,方面热图和方面趋势图,使用 SUR-Miner 的摘要来帮助开发人员掌握用户对每个软件方面的偏好和典型意见。相应应用程序开发者的反馈也令人鼓舞,88% 的受访者认为 SUR-Miner 的总结有用,表明 SUR-Miner 在实践中帮助开发者了解用户对不同方面的偏好。


总的来说,我们的研究做出了以下贡献:


- 1) 我们利用分类技术,在该技术中我们设计了文本特征来区分五个评论类别,例如错误报告和新功能请求。
- 2) 我们提出了一种基于模式的解析技术,可以解析复杂的应用评论句子,并提取方面和相应的意见。
- 3) 我们设计新颖的交互式可视化效果,为应用程序开发人员和管理人员高效地呈现摘要。
- 4) 我们对 SUR-Miner 进行了实证评估,以调查其实用性。


本文的其余部分安排如下。第二节介绍了相关工作。第三节介绍了我们框架的详细设计。第四节介绍了评估。第五节讨论验证的威胁,第六节总结本文。


第二部分。 相关工作

A. 应用评论过滤

应用评论过滤在软件工程社区中引起了越来越多的关注。陈等。[10]过滤非信息性评论并按重要性对用户评论进行排名。他们的框架训练分类器并将评论分为两类，即信息性和非信息性。傅等。[15]通过评论词汇表上的回归模型过滤评级不一致的评论（评论的情绪与其评级不同）。这些工具可以部分选择信息性评论。然而，他们没有明确定义在什么情况下评论可以提供信息，因为不同的开发人员需要不同类型的评论 [30]，[31]。为了进一步推进他们的工作，我们旨在区分不同的审查目的（类别）并从特定类别中选择审查以提取和总结软件方面。

Contents





Sorbo 等人最近的工作。[13]提出了一个类似的想法，根据他们的目的对开发电子邮件进行分类。他们还使用自然语言解析技术设计了一种分类方法。虽然他们的技术也可以应用于应用评论分类，但它不支持每个类别内的方面总结。

B. 从 App 评论中提取方面

方面抽取在软件工程中也得到了广泛的研究。陈等。[10]使用 LDA [6]提取评论主题。胡等。[19]提出了一种网络评论挖掘方法。他们的方法提取频繁词作为 aspect，并将相应的形容词词作为 opinions。Fu [15]解决了挖掘用户负面反馈的问题。他们应用 LDA 主题模型从负面反馈中挖掘主题并对每个版本的总结问题进行排序。加尔·维斯等人。[16]通过采用名为方面和情感统一模型（ASUM）[22]的主题模型来改变我的需求。他们还提取共同话题并呈现用户对这些话题的看法。

然而，他们的方法与我们的有很大不同。他们应用了基于词袋假设的频繁项挖掘或主题模型，而不考虑句子结构和语义。这意味着他们既不能区分评论类别（赞美、功能请求、错误报告、缺点），也不能区分方面和用户意见，这可能会导致不准确和混乱。例如，LDA 提取的主题词“prediction”可能意味着用户欣赏该预测特征，或者用户希望获得新的预测特征。在这种情况下，开发人员无法有效地解释主题。

Sarro 等人最近的工作。使用自然语言处理[34]从应用程序描述中提取特征。我们的工作与他们的不同之处在于我们从应用评论中提取特征。此外，我们旨在总结应用程序功能，而他们的目标是调查功能生命周期 [34]。

据我们所知，只有一项以前的工作与我们的工作密切相关。Guzman 和 Maaiej [17]提出提取软件特征并分析他们的情绪。我们的工作三个主要方面与他们不同。首先，我们的方法不仅旨在识别特征，还旨在区分特征评估和特征请求。其次，我们的方法可以识别复杂和新颖的特征，因为它使用语义模式解析评论句子，而他们的技术像传统方法一样基于频繁项目挖掘和主题模型。最后，我们提出了交互式可视化，以帮助应用程序管理员和开发人员掌握功能评估和情绪趋势。

C. 审查其他市场的挖矿

用户评论挖掘在其他市场（例如商品、电影）中也是一个有吸引力的话题。亚塔尼等人。[37]提出了一种名为 ReviewSpotlight 的评论摘要工具，它通过从评论句子中识别形容词-名词词对来提取方面-意见对。黄等。[20]采用了类似的想法并设计了 Revminer——一个用于总结餐厅评论的提取界面。尼科尔斯等人。[29]提出了 ReCloud，它使用 NLP 技术解析评论句子。庄等。[38]研究了电影评论摘要。他们的方法整合了多种知识，包括 WordNet、统计分析和电影知识。

然而，这些技术很难直接应用于应用评论。应用评论与其他市场上的评论有很大不同[10]，[15]。它们具有现有工具难以解析的不同词典和格式。ReviewSpot-light [37]通过提取形容词-名词词对来呈现词云。同样，RevMiner [20]使用引导算法提取词对。ReCloud [29]考虑了语义，它也呈现了一个词云，但具有反映 NLP 上下文的空间布局。然而，应用评论不能简单地用词云或词对来表示。例如：

情况1：“我喜欢我们可以改变主题这一事实”

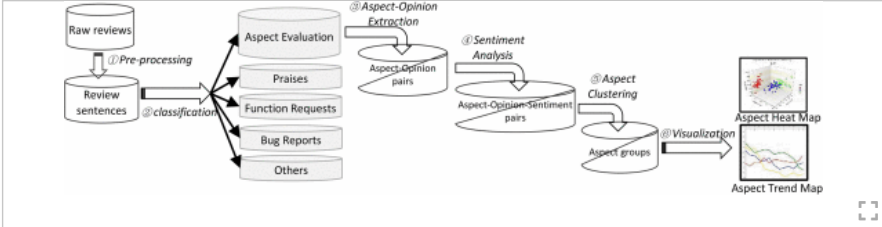


图. 1 :
拟议的 SUR 矿工框架概述

ReviewSpotlight 无法输出任何内容，因为没有形容词。RevMiner 和 ReCloud 可能会出现一些无意义的词对。相比之下，SUR-Miner 可以提供正确的配对（我们可以更改主题，爱），因为它考虑了语义和应用程序审查模式。此外，应用评论包含针对对不同开发者的多种目的 [31]。现有工具都无法区分此类类别。考虑以下情况

案例 2：“点击 \text{} 'ok' 按钮后的蓝屏很烦人。”
案例 3：“一个简单的用户界面会更好。”

Contents

PDF 从开发者的角度来看，它们只是错误报告和功能请求，不应被视为用户对“屏幕”和“UI”的意见。此类案例在应用评论中占很大比重[31]。虽然所有这些工具仍然输出单词对（云），例如（按钮，烦人）和（UI，简单），但 SUR-Miner 可以区分上述情况，因为它利用了分类技术。

第三部分。
苏尔矿工

本节介绍 SUR-Miner 的通用架构。

如图1所示，我们的框架将用户评论（包括文本和评级）作为输入，并输出对应用程序不同方面的主要意见和情绪。整个过程包括六个主要步骤：对于需要总结的原始评论，我们首先将它们拆分成句子（步骤 1）。然后，我们将每个句子分为五类，即方面评价、赞美、功能请求、错误报告和其他（步骤 2）。然后，我们只在aspect evaluation中选择句子分类并过滤掉其他类型的句子。然后，我们从“方面评价”句子集中提取方面和相应的意见和情绪（步骤 3-4）。由此产生的方面-意见-情感对被聚类并用两个交互式图表可视化（步骤 5-6）。下面详细解释每个步骤。

A. 第一步-预处理

原始用户评论需要预处理。它通常由多个具有不同目的的句子组成。例如，原始评论“用户界面很丑。我想要一个漂亮的用户界面”由两句话组成。第一句是对方面UI的评价，第二句是对方面UI的改进要求。他们有不同的目的和感受。因此，最好将这些句子分开进行分析。此外，用户评论有很多错别字和缩写，很难自动理解其含义。

为了解决这两个问题，我们使用 Stanford CoreNLP 工具[26]将原始评论文本拆分为句子。每个评论句子都带有时间戳和指定评级，与原始评论中的相同。我们还纠正了常见的拼写错误、缩写和重复，例如“U → you”、“coz → because”、“&→ and”、“Plz → Please”、“soooo → so”和“thx → thanks”。我们收集了 60 个这样的拼写错误和缩写，并用正则表达式²替换了它们。

B. 第 2 步 - 审查分类

正如在第一节中所讨论的，评论句子可能有不同的类别[31]。不同的类别针对不同的任务和开发人员[30]。开发人员手动对它们进行分类并选择合适的句子进行方面评估非常繁琐且耗时。在评论分类步骤中，我们的目标是自动分类和选择包含方面评价的评论句子。

我们定义了五个审查类别，包括方面评估、错误报告、功能请求、表扬和其他。帕加诺等。找到了 17 个类别（主题）的用户评论[31]。我们使用其分类法[31]中的前四个类别，并将其他次要类别合并到“其他”类别中。表 I说明了每个类别的定义和示例评论句子。

为了将评论句子分类为上述类别，我们采用了监督机器学习方法。我们首先收集历史评论句子，提取它们的文本特征，并根据表 I中的定义手动标记它们。然后，我们使用这些文本特征和标签训练分类器。最后，我们在新评论实例上执行分类器以预测它们的类别。

我们采用了著名的分类器Max Entropy，它在文本分类[24]、[28]方面具有出色的性能。在下文中，我们展示了我们为分类设计的文本特征。

1) 文本特征提取

我们提取了两个维度的文本特征：词典特征和结构特征。

词典对于表征评论类别很重要，因为不同的评论类别可能具有截然不同的词典。例如，“amazing”和“great”经常出现在赞美评论中，而“bug”和“fix”则是错误报告的代表词。我们选择字符 N-Gram和主词作为两个词典特征，因为它们反映了不同类别的词典。

表一五个评论类别的定义[31]

Category	Definition	Examples
Praise	Expressing emotions without specific reasons	Excellent! I love it! Amazing!
Aspect Evaluation	Expressing opinions for specific aspects	The UI is convenient. I like the prediction text.
Bug Report	Reporting bugs, glitches or problems	It always force closes when I click the "com" button.
Feature Request	Suggestions or new feature requests	It would be better if I could give opinion on it. It's a pity it doesn't support Chinese. I wish there was a "deny" button.
Others	Other categories that are defined in [31]	I've been playing it for three years

字符 N-Gram字符 N-Gram 是一种重要的词汇表示，是文本分类中常用的特征[8]、[9]、[18]、[23]、[25]。它还被发现在许多应用程序中有效，例如软件工程中的恶意代码检测[4]和

重复错误报告检测 [36]。一个句子的字符 N-Gram 特征是该句子标记中的所有 n 个连续字母。例如，句子“The UI is OK”的 3-Grams 是The、heU、eUI、UI、is、isO和sOK。我们使用 2-4 克进行分类。

主干词我们还建议将**主干词**作为词典特征。我们将**主干词**定义为语义依赖图[12]根部的词，这将在本节后面介绍。例如，“The graphics are amazing”这句话的主干词是“are”。

句子结构也可以反映文本特征，因为不同的评论类别可能具有不同的句法和语义。例如，对于 *aspect* 评价，用户倾向于使用描述性句法，如“The graphic (noun) is amazing (adjective)”，而对于 *feature request*，用户通常使用祈使句，如“please add more themes”和“It could最好有更多的主题 (名词)”。

我们利用三个结构特征：POS 标签、解析树和语义依赖图。

词性标记词性 (POS) [11]是文本中广泛使用的语法特征。它表示句子中每个词的属性。例如，句子“The user interface is beautiful”的 POS 标签在序列 [11]中是 DT-NN-NN-VBZ-JJ。在这里，单词 is 的词性标记是 VBZ，意思是 is 是一个第 3 人称现在单数的动词。我们使用 Stanford CoreNLP 工具 [3]、[26]生成 POS 标签，并将所有 POS 标签连接在一起作为文本特征。

解析树解析树是句子语法结构的典型表示[35]。它显示了一个句子是如何组成的。每个节点代表一个语法单元，其子节点是组成它的子单元。图 2说明了一个示例评论句子“The user interface is not very elegant”的解析树，它是由 Stanford Parser [3]生成的。每个节点中的标签表示一个 POS 标签。这棵树意味着句子 (ROOT) 由一个名词短语 (NP) 和一个子句 (S) 构成，其中名词短语由一个限定词 (DT) 和两个名词 (NN) 构成。

为了将解析树表示为平面文本特征，我们按广度优先顺序遍历树节点并选择前五个节点。我们将这五个节点的 POS 标签连接起来作为文本特征。例如，图 2中解析树的特征是“ROOT-NP-S-DT-NN-NN”。

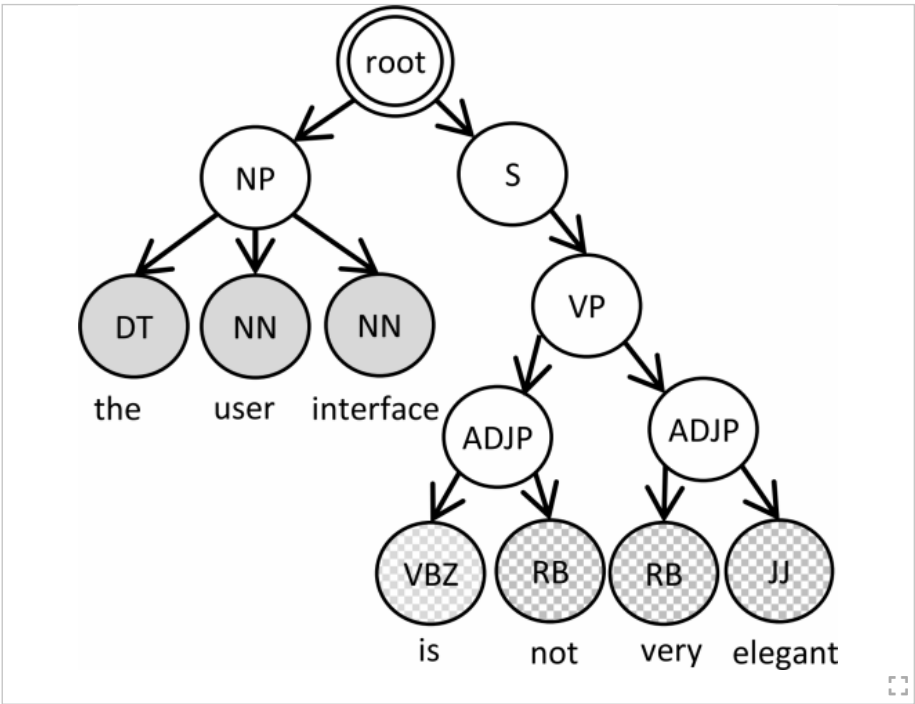


图 2：
句子的解析树：用户界面不是很优雅。

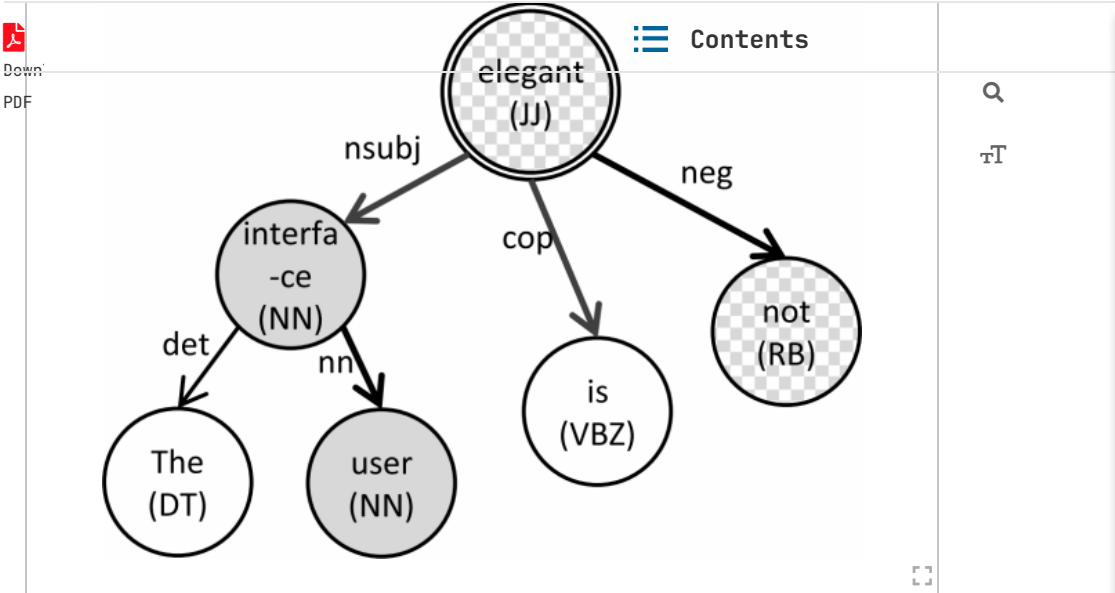


图 3：
句子的语义依赖图：用户界面不优雅。

语义依赖图 (SDG) 语义依赖图 (SDG) [12]展示了一个句子中单词之间的语义依赖。它是一个有向图 [12]。图中的节点代表单词和相应的 POS 标签。边表示单词之间的语义关系（例如，名词从属关系和形容词修饰语）。每个 SDG 都有一个没有传入边的根节点。图 3说明了由斯坦福解析器 [3]生成的示例评论句子“*用户界面不优雅*”的 SDG 。根节点是词*elegant*，它是一个形容词 (记为 JJ)。它有三个孩子：一个名词从属 (nsubj) 接口，copula (cop) *is* 和否定修饰符 (neg) *not*。子接口也有两个子接口：一个限定符 (det) *the* 和一个名词复合修饰符 (nn) *user*。

为了将 SDG 转换为平面文本特征，我们按广度优先顺序遍历其节点，然后在遍历中连接边和 POS 标签。我们忽略没有链接到根的叶子。例如，图 3中 SDG 的特征是“VBZ-nsubj-NN-cop-VBZ-neg-RB”。

C. Step 3-Aspect-Opinion 抽取

我们的下一个目标是总结用户对相应方面的意见。为此，我们需要识别表达方面的词和表达对这些方面的意见的词。在此步骤中，*SUR-Miner* 从分类在方面评价类别中的每个评论句子中提取方面意见对（即方面和意见词）。例如，评论句子“*The Prediction is accurate, but the auto-correct is annoying*”的aspect-opinion对是：(*prediction, accuracy*) 和 (*auto-correct, annoying*)。

一般来说，最先进的技术通过频繁的项目挖掘或通过用户评论视为词袋的主题模型来提取方面 [6]、[10]、[15]。这样的假设对于表现出多种目的和情绪的软件评论来说可能是有问题的。

正如一项实证研究表明的那样，针对不同目的的软件评论具有相当单调的模式[31]。因此，可以直接从句型中确定aspect-opinion pairs。基于这一假设，我们设计了一种基于模式的分析方法，该方法利用评论句子的句法和语义，直接从中分析方面和相应的观点。为此，我们首先应用 NLP 解析器来注释评论句子的语义依赖图 (SDG) [12]。然后，我们构建了一个基于模式的解析器来从 SDG 中提取 aspect-opinion 对。

1) 基于模式的解析

我们基于模式的解析器实现为一系列级联有限状态机[7]。解析器接受一个 SDG 并根据预定义的语义模板识别 aspect-opinion 对。

表二列出了我们使用的一些典型的语义模板。开头的两个字母（例如，JJ 和 NN）表示根的 POS 标记。以下圆括号中的单词（例如*have*和*like*）代表词根。根的子代列在方括号中作为边缘 POS 对。例如，第一行中的模板表示一个带有 JJ POS 标签的根节点和两个子节点：一个带有 NN POS 标签的名词 subjection (nsubj) 和一个带有 POS 标签 VBZ 的 copula (cop)。我们通过对评论句子中手动识别方面部分和意见部分来生成模板。我们随机选择了 2,000 个标记为*Aspect Evaluation* 的评论句子除了我们后来用于评估准确性的那些。首先，我们检查了所有这些句子并生成了它们的可持续发展目标。然后，我们将每个 SDG 与一个模板相关联，该模板指示 SDG 中方面部分和意见部分的位置。我们选择了所有与超过 10 个句子相关联的模板，以避免意外关联。我们确定了 26 个这样的模板来设计有限状态机³。

然后，给定一个新的 SDG 实例，解析器从根节点移动到所有其他节点，检查节点、边和相应的子节点，根据模板确定方面词和意见词。例如，给定图 3中的 SDG ，解析器检查根的 POS 标记。由

于它是一个形容词 (JJ) 匹配 表二中的第一和第二模板，它进一步检查它是否有三个孩子：名词 subjection (nsubj) 的 POS 标签为名词 (NN)，copula (cop) 的 POS 标签为 VBZ，否定修饰符 (neg) 的 POS 标签为RB。第二个模板匹配。然后，它检查第一个孩子是否有名词复合修饰符 (nn) 的孩子，其词性标记为名词 (NN)。由于第二个模板与样本 SDG 完全匹配，解析器将 nsubj-NN 节点接口及其子用户识别为 aspect 词，将 neg-RB 节点不连同根节点elegant识别为意见词。

D. Step 4-Aspect 情感分析

除了意见之外，量化总结用户对各个方面的感受也可能有助于掌握用户的偏好。用户的评分可以客观地提供这样的概括。然而，总体评级不能令人满意地表征用户对不同方面的偏好。例如，考虑一下评论“用户界面不错，但声音很糟糕”。评分为 2（满分 5）。用户显然喜欢方面的UI但不喜欢方面的声音。因此，两个方面的实际评分都不可能是2；UI可能是 3，声音可能是 1。

在第四步，我们对每个评论句子进行情感分析，并将情感与相应方面与用户评分和情感分析工具相关联。我们首先应用最先进的情感分析工具Deeply Moving [1]来分析每个评论句子的情感。Deeply Moving产生 0 到 4 等级的情绪，其中 4 表示强烈正面，0 表示强烈负面，2 表示中性。然后，为了提高准确性，我们通过用户评分（1 到 5）调整情绪。具体来说，如果整个评论的评级为 5（强烈正面），我们将 0 的情绪加 1。如果评级为 1（强烈负面），我们将 4 的情绪减 1。

比如下面的评论有两句话：The interface is beautiful。我不喜欢这个主题。这两个句子的情感分别为 4 和 0。如果评论的用户评分为 5，我们将第二句话的情绪调整为 1 (= 0 + 1)。如果用户评分为 1，我们将第一句话的情绪调整为 3 (= 4-1)。

E. 步骤 5-Aspect Clustering and Summarization

在这一步，我们将具有相同方面的方面-意见对分组，并为每个方面组总结情绪和典型意见。

为了对方面进行分组，我们首先为所有方面词挖掘频繁项，即提取的方面意见对中的方面词。然后，我们将 aspect-opinion 对与常见的频繁项（词）聚类。例如，假设auto correct是所有 aspect 词中的频繁项，如果有两个 aspect-opinion 对包含该项，它们将被聚类为一组。特别是，如果一对有两个或更多可以聚类到两个以上不同组中的频繁项目，我们将其聚类到项目或单词频率最高的组中。例如，一对（background color, nice）可以同时包含（background, beautiful）和（color, disgusting）。但是，如果我们已经知道 aspect background 的频率高于color的频率，我们会将第一对与第二对而不是第三对分组。如果两个 aspect-opinion 对中没有频繁项，当它们在 aspect 中有共同词时，我们将它们分组在一起。

表二依赖关系模板示例

Templates	Sample Sentence	Aspect Words	Opinion Words
JJ[nsubj-NN,cop-VBZ]	The UI is beautiful!	nsubj-NN	JJ
JJ[nsubj-NN[nn-NN],cop-VBZ,neg-RB]	The user interface is not elegant.	nn-NN + nsubj-NN	neg-RB + JJ
NN[amod-JJ]	nice UI!	NN	amod-JJ
VB[have-]nsubj-NN,nobj-NN]	The frame has nice UI!	nsubj-NN	have + nobj-NN
VB[like-]nsubj-NN]	I like the UI!	nobj-NN	like

对于每个组，我们选择一个组关键字作为该组中频率最高的词或项目。我们还将一个群体情绪计算为该群体中方面-意见对的平均调整情绪。

F. 步骤 6-可视化

我们设计了两个交互式图表，即Aspect Heat Map和Aspect Trend Map来说明摘要。

方面热图展示了用户关注的热门方面。它旨在帮助开发人员和管理人员掌握用户喜欢或不喜欢应用程序的哪些部分（方面）。图 4显示了方面热图的示例，其中每个圆圈表示一个方面。圆圈越大，表示该方面越受欢迎和喜欢。我们将圆圈的大小定义为size = log(#comments) + sentiment。横轴代表评论数，纵轴代表调整后的评分。因此，右上角的圆圈代表最受欢迎和最喜爱的方面，反之亦然。要深入了解方面组，开发人员可以单击每个方面（圆圈）以查看具有最高积极情绪和最高消极情绪的特定评论。对于每个评论，方面词都带有下划线，意见词以粗体显示。

Aspect Trend Map展示了随时间变化的情绪趋势。捕获用户反应对于开发人员选择功能并确定其优先级非常重要[16]，[34]。Aspect Trend Map旨在帮助开发人员评估他们最近的更改是否影响了用户的满意度。它还使开发人员能够估计和预测用户的偏好，以便他们可以在未来改进其产品的某些部分。图 5显示了一个图表示例，其中每条线表示一个流行方面的情绪趋势。横轴代表日期，纵轴代表用户情绪。

Aspect Heat Map 和 Aspect Trend Map 都可以在我们的项目网站 <http://www.cse.ust.hk/~xguaa/srminer/>上找到。

我们通过三个维度来评估我们的框架：有效性、比较性和有用性。为了评估有效性和优势，我们在文本挖掘文献中应用了常见的措施，并将结果与最先进的方法进行了比较。我们还进行开发人员调查以评估有用性。具体来说，我们的评估解决了以下研究问题：

表 III应用主题概览

Data Set	Category	Time Period	
Swiftkey	productivity	8.26.2014	– 9.10.2014
Camera360	photography	8.24.2014	– 9.8.2014
TempleRun2	game	8.30.2014	– 9.10.2014
WeChat	social network	9.5.2014	– 9.11.2014
KakaoTalk	communication	6.22.2014	– 9.12.2014
GooglePlayBooks	books	12.16.2014	– 3.18.2015
SpotifyMusic	music	3.8.2015	– 3.18.2015
YahooWeather	weather	1.30.2015	– 3.18.2015
GoogleMap	map	3.6.2015	– 3.20.2015
GoogleCalendar	productivity	2.4.2015	– 3.20.2015
ESPN	sports	9.19.2014	– 3.21.2015
TextPlus	social	12.16.2014	– 3.21.2015
Duolingo	education	3.5.2015	– 3.22.2015
ChaseMobile	finance	9.17.2014	– 3.23.2015
Medscape	medical	1.7.2013	– 3.23.2015
Yelp	food	12.8.2014	– 3.25.2015
IMDB	entertainment	10.13.2014	– 3.29.2015

- **RQ1** (有效性) : SUR-Miner 对评论进行分类、提取方面和意见以及分析应用评论的情绪的准确率如何？
- **RQ2** (比较) : SUR-Miner 与最先进的应用评论摘要技术相比如何？
- **RQ3** (有用性) : SUR-Miner 的摘要对开发人员有何用处？

A. 数据收集

我们从 Google Play 中选择了 17 个流行的 Android 应用程序，如 Swiftkey、Camera360、WeChat 和 Templerun2 作为我们的主题。这些应用涵盖了游戏、通讯、书籍和音乐等 16 个最受欢迎的类别。我们使用开源 Android 市场API [2]大致收集了 2014 年 8 月至 2015 年 3 月期间的评论。对于每条评论，我们都会收集其时间戳、评级、标题和内容。表三显示了主题的描述。

B. 有效性 (RQ1)

在本节中，我们将评估 SUR-Miner 在每个步骤中的有效性，即评论分类、方面意见提取和情感分析。

1) 审核分类

首先，我们在评论分类任务上评估 SUR-Miner。我们从每个数据集中抽取了 2,000 个评论句子，并将预测结果与黄金标准标签进行了比较。我们根据表 I 中的规则手动标记黄金标准类。为了减少标签偏差，两名研究人员分别将标签规则应用于 2,000 个评论句子。在第一次迭代中选择了共识标签。对于分歧，我们讨论并澄清了我们的标签规则并重新标记。第二次迭代导致两位研究人员之间达成 100% 的一致。

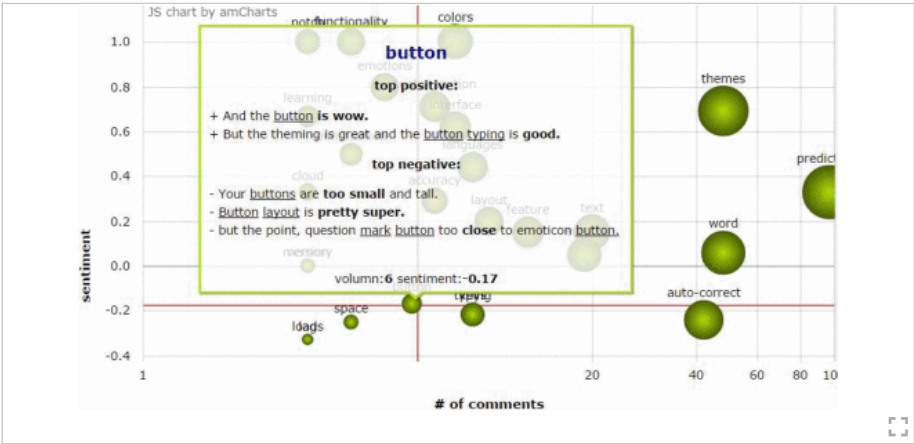


图 4 : aspect热力图演示

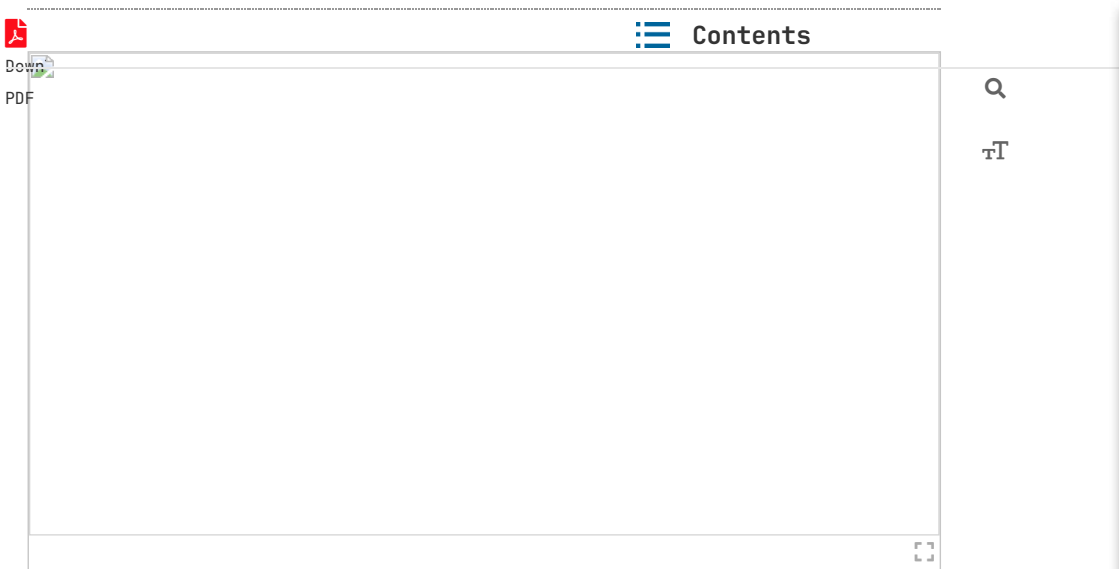


图 5 :
纵横趋势图演示

我们使用 F1-score 来衡量分类准确率。F1-score 在文本分类文献[16]、[38]中被广泛使用。定义如下

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \tag{1}$$

[查看源代码](#)

其中精度是正确分类为一个类的实例数（TP）与分类为该类的实例数（TP+PP）的比率。

$$precision = \frac{TP}{TP + PP} \tag{2}$$

[查看源代码](#)

召回率是正确分类为一个类的实例数（TP）与该类中的实例数（TP+FN）之比。

$$recall = \frac{TP}{TP + FN} \tag{3}$$

[查看源代码](#)

我们在数据集中执行了 5 次交叉验证[38] 100 次，每个文件夹包含 400 个评论句子。

表 IV显示了不同类别⁴的F1 分数。每列显示所有主题中评论类别的 F1 分数。最后一列是所有评论类别中每个主题的平均结果，最后一行是所有主题中每个评论类别的平均 F1 分数。如表中所示，分类性能合理（平均 F1 分数为 0.75）以及方面评估类别（平均 F1 分数为 0.74）。这意味着分类步骤可以准确地为不同的开发者提供不同类型的评论语句。特别是，它为 aspect 评估提供了可靠的评论句子。特定类别（例如“错误”）的 F1 分数在某些应用程序中并不好。我们手动检查了这些评论，发现这些应用收到了罕见的错误报告。极度不平衡的数据可能是这些异常值的主要原因。

表IV F1-所有科目的评论分类分数

PDF

Category	Evaluation	Praise	Request	Bug	Other	Overall
Swiftkey	0.72	0.87	0.78	0.58	0.86	0.74
Camera360	0.72	0.95	0.42	0.76	0.85	0.74
Templerun2	0.76	0.83	0.65	0.82	0.77	0.77
WeChat	0.70	0.93	0.50	0.76	0.89	0.76
KakaoTalk	0.76	0.96	0.66	0.47	0.91	0.75
Google Play Books	0.59	0.92	0.72	0.60	0.85	0.74
Spotify Music	0.68	0.94	0.54	0.57	0.87	0.72
Yahoo Weather	0.83	0.94	0.57	0.56	0.87	0.76
Google Map	0.73	0.88	0.60	0.76	0.84	0.76
Google Calendar	0.74	0.77	0.80	0.70	0.82	0.77
ESPN	0.77	0.80	0.57	0.77	0.83	0.75
TextPlus	0.65	0.94	0.41	0.72	0.88	0.72
Duolingo	0.79	0.95	0.67	0.50	0.88	0.76
Chasemobile	0.75	0.93	0.46	0.56	0.85	0.71
Medscape	0.84	0.94	0.63	0.71	0.88	0.8
Yelp	0.77	0.91	0.40	0.58	0.91	0.72
IMDB	0.71	0.90	0.60	0.64	0.84	0.74
Average	0.74	0.90	0.59	0.65	0.86	0.75

表 V F1-aspect-opinion 抽取分数

Data Set	Aspect	Opinion	Sentiment Positive	Sentiment Negative
Swiftkey	0.87	0.86	0.87	0.71
Camera360	0.87	0.87	0.89	0.53
Templerun2	0.95	0.93	0.91	0.79
WeChat	0.83	0.82	0.77	0.83
KakaoTalk	0.84	0.87	0.85	0.77
Google Play Books	0.84	0.86	0.82	0.82
Spotify Music	0.84	0.86	0.83	0.62
Yahoo Weather	0.90	0.63	0.89	0.77
Google Map	0.86	0.84	0.88	0.88
Google Calendar	0.79	0.82	0.78	0.85
ESPN	0.80	0.78	0.69	0.83
TextPlus	0.84	0.85	0.80	0.77
Duolingo	0.86	0.85	0.93	0.51
Chasemobile	0.84	0.88	0.87	0.77
Medscape	0.89	0.90	0.86	0.60
Yelp	0.84	0.87	0.89	0.82
IMDB	0.84	0.87	0.86	0.86
Average	0.85	0.84	0.85	0.75

2) 方面意见提取

为了评估 SUR-Miner 在 aspect-opinion 抽取方面的表现，我们按照与评论分类实验中相同的程序来检查 SUR-Miner 是否正确地从评论句子中提取 aspect 和相应的观点。对于每个主题，我们抽取了 2,000 个评论句子，并选择了方面评估类别中的句子。我们使用 F1-score 分别衡量 aspect 抽取和 opinion 抽取的准确性。特别地，等式 1-3 中的真阳性（TP）的数量是正确提取的方面或观点的数量；误报数（FP）是指错误提取的方面或观点的数量；假阴性（FN）的数量定义为未提取的方面或意见的数量。

结果显示在表 V 的前两列中。正如所指出的，方面提取和观点提取都具有合理的准确性，平均 F1 分数分别为 0.85 和 0.84⁴。结果表明，方面提取步骤提供了可靠的方面和意见。

3) 情感分析

为了评估情感分析步骤，我们还遵循与分类和方面提取阶段相同的程序。对于每个主题，我们抽取了 2,000 个评论句子并选择了方面评估类别中的句子，并将每个方面-意见对的情感与黄金标准情感标签进行了比较。为了简化估计，我们将情绪量表（0-4）分为两个极性，即积极（3-4）和消极（0-1）[32]，并根据它们的极性进行标记。我们像对评论分类所做的那样手动标记黄金标准情感。

我们使用 F1-score 来衡量每个情感类别的准确性。特别地，等式 1-3 中的真阳性（TP）数量定义为正确分类的情绪数量；误报数（FP）表示错误分类的情感数；假阴性（FN）的数量表示未归入该类别的情感数量。

结果示于表V的最后两栏中。如图所示，正面和负面情绪都具有可接受的准确性，平均 F1 分数分别为 0.85 和 0.75⁴。两者的平均 F1 分数均为 0.80。负面情绪在 Camera360 和 Duolingo 中表现相对较低的原因可能是这两个应用程序获得了更多的正面评价，因此情绪类别变得极不平衡。结果表明，情绪分析步骤产生了可靠的结果。

C. 比较 (RQ2)

我们的下一次评估旨在将 SUR-Miner 与最先进的技术在最终摘要方面进行比较。

1) 定量比较

我们首先将 SUR-Miner 用于方面提取的准确性与相关工作的准确性进行比较：ReviewSpotlight [37]和 Guzman 的方法 [17]。正如在第 II 节中讨论的，ReviewSpotlight 是一种通过识别名词-形容词对（第 II-C节）对一般产品进行评论总结的工具，而 Guzmans 的工具是与我们最相关的工作，它还从应用程序用户评论中提取方面（第 II 节）-B）。

我们通过模拟真实世界的使用场景来运行方面提取。对于每个主题，我们从原始数据集中随机选择 400 个所有类别的评论句子，但用于训练分类器的除外。首先，我们对这些句子进行评论分类。然后，我们对分类为方面评估的句子应用方面提取。我们将提取的方面与手动标记的黄金标准方面进行比较。我们使用 F1-score 来评估准确性，使用与第 IV-B2 节中相同的定义。

表六aspect 提取精度与相关工作的比较

表 VI 显示了所有科目中三种方法的平均 FI 分数。我们复制了 ReviewSpotlight 并将其应用于提取应用方面。Guzmans 方法的结果摘自他们的论文 [17]。正如我们所看到的, SUR-Miner 的 FI 分数是 0.81, 明显高于 ReviewSpotlight (0.56) 和 Guzman 的工具 (0.55)。

为了调查这些结果的原因，我们手动检查了 ReviewSpotlight 的结果。我们发现，在不区分评论类别的情况下，它倾向于在其他类别（例如方面请求和错误报告）中提取方面进行评论。例如，考虑评论“我讨厌你不能使用离线词典”，它需要一个新的方面 *离线词典*。ReviewSpotlight 只是输出（*dictionary, offline*）这是没有意义的，而 SUR-Miner 可以从方面评估中过滤这样的评论，因为它谈论的是一个不存在的方面。

这些相关方法的另一个缺点是，它们无法识别复杂的短语，因为它们只是将频繁出现的项目或名词-形容词对视为方面。例如，对于评论“*Also, love the way it auto ads reminders*”，ReviewSpotlight 简单地输出 (*ads, auto*) 而 SUR-Miner 输出 (*the way it auto ads reminders, love*)。

同样有趣的是，即使分类和提取阶段都有错误，但将它们结合起来并不会导致更差的准确性。分类步骤的 F1 分数为 0.74。aspect 提取步骤的 F1 得分为 0.85 (第 1 节)。然而，当从分类阶段的输出中提取 aspects 时，最终的 F1-score 为 0.81，甚至比分类阶段的还要高。通过手动检查提取的方面，我们发现虽然一些评论在分类阶段被错误分类，但方面提取阶段仍然可以“重新纠正”它们，因为我们的语义模式可能无法解析错误分类的评论。例如，考虑错误分类的评论“没有公共交通导航！”这需要一个新的方面，但在分类阶段被错误分类为方面评估。尽管如此，SUR-Miner 仍然无法识别任何方面，因为没有语义模式来解析此评论。

2) 定性比较

LDA 等主题模型被大多数最先进的应用评论摘要工具广泛使用[10]、[15]、[17]。为了研究 SUR-Miner 相对于这些基于主题的技术的优势，我们定性地比较了 SUR-Miner 提取的方面与主题模型提取的主题。

表 VIII 将我们提取的前五个方面与 AR-Miner (应用 EMNB-LDA 主题模型的最先进的评论摘要工具) [10] 在 Swiftkey 主题中的前五个主题进行了比较。我们和 AR-Miner 在同一时期从 Google Play 收集数据。我们有两个观察结果：1) SUR-Miner 可以区分不同的审查目的。例如，SUR-Miner 提取的意见是除了一些噪音之外的方面评价，而 LDA (AR-Miner) 的顶级词是杂项。例如，如果管理者想知道用户对 aspect 预测的评价，SUR-Miner 可以提供用户的评价，如excellent、accurate、hate而 AR-Miner (LDA) 无法提供此类信息；2) SUR-Miner 可以区分用户的情绪，而 AR-Miner (LDA) 不能。例如，管理者和开发人员可以通过 SUR-Miner 发现正面和负面情绪，但无法判断用户是否喜欢 LDA 预测的方面。

表 VIII 主题 (通过 LDA) 和方面 (通过 SUR-miner) 的比较

Questions	Strongly Disagree	Disagree	Neither	Agree	Strongly Agree	Total
Q1. Do you think Figure 1 "Aspect Heat Map" is useful to understand users preferences for aspects?	0	1	0	7	8	22
Q2. Do you think Figure 2. "Aspect Trend Map" helps developers to understand the users preferences trend over time?	0	1	2	5	8	22

总的来说，SUR-Miner 在区分评论目的和情绪方面比 LDA 模型产生更清晰的摘要。

SUR-Miner produces much more accurate and clearer summaries than state-of-the-art methods.

D. 有用性 (RQ3)

由于有用性评估可能是主观的，我们咨询了开发人员来评估 SUR-Miner 的有用性。我们将 SUR-Miner 应用于 Swiftkey、Camera360、WeChat 和 Templerun2 等 17 款热门安卓应用的最新用户评论。我们在我们的网站上以演示的形式展示了可视化摘要，并向开发人员提出了表 VII 中所示的问题。这两个问题分别与两个图相关。我们为他们每个人提供了五个选项 (5 个非常同意，4 个同意，3 个都不是，2 个不同意和 1 个强烈不同意)。还为每个问题列出了每个选项的选择数量。

我们向选定应用程序的开发者发送了邀请邮件，将我们的网站发布到 Google+ 的 Android 开发者社区，并邀请了三星、腾讯和百度等 IT 企业的开发者进行反馈。

开发人员对我们的 SUR-Miner 表现出了极大的兴趣。如表 VII 所示，在收到的所有 32 个答案中，有 28 个 (88%) 同意我们的工具可以帮助开发人员。只有两个人持保守意见 (6.3%)，两个人 (6.3%) 不同意。图 6 显示了开发者反馈的箱线图统计。

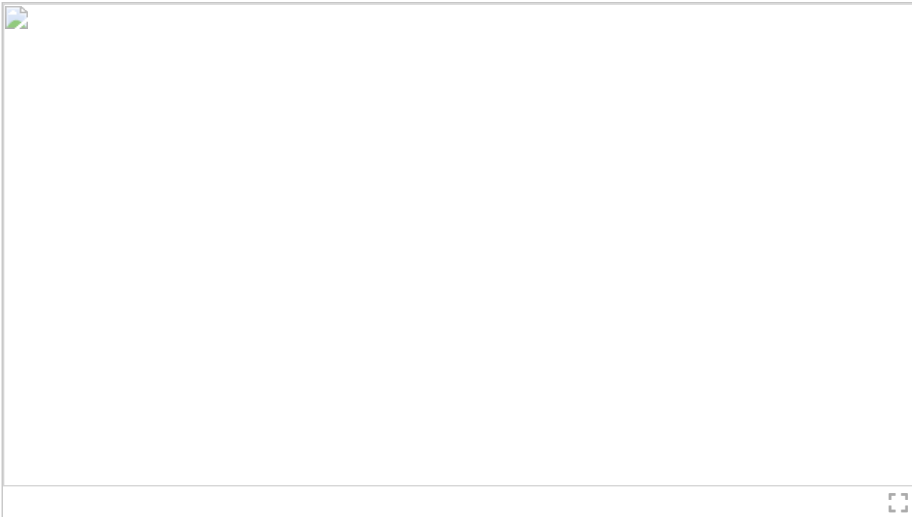


图 6 :
开发人员对实用性的评分

Contents

PDF



表 VII 开发者调查中的问题及结果

(a) Topics and typical words by AR-Miner (LDA) [10]. The first row lists the top 5 topics. The following 4 rows list the top 5 words for each topic

Topics	theme	Chinese	jelly bean	predict	space
Keywords	more	languag	bean	word	space
	theme	chines	jelli	predict	period
	wish	need	galaxi	text	email
	love	wait	note	complet	enter
	custom	user	keyboard	auto	insert

(b) Aspects and opinions extracted by SUR-Miner. The first row lists top five aspects with most comments. The following three rows show opinions with top positive sentiments while the last two rows show opinions with top negative sentiments

Aspects	predictions.	auto-correct	words	theme	key
Opinions	amazing	flawless	love	great	best
	excellent	good	like	love	like
	amazing	amazing	like	over top	
	accurate	stubborn	a pain	ugly	breaker
	hate	nightmare	not-need	just	obnoxious

我们将评分的答案从 1 量化到 5。每个方框显示一个问题的答案。正如结果所示，这两个问题的答案的平均评分都远高于 3。这意味着开发人员总体上认可 SUR-Miner 的实用性。

此外，我们还收到了来自开发人员的以下令人鼓舞的评论：

“这是一个伟大的项目。可视化数据让我印象深刻！” “我认为，如果可能的话，我们愿意与这些研究人员合作。我真的很喜欢你们的情绪分类器的性能。” “所提供的可视化信息是了解产品优缺点的一种非常清晰的方式。分析大规模的用户评论需要大量的人力。这样的项目让产品的理解和迭代速度更快。”

这些评论表明开发者很欣赏我们的工具可以帮助掌握用户对不同方面的意见。

Developers feedback indicates our SUR-Miner helps developers grasp users' opinions and sentiments in practice.

第五节
有效性的威胁

我们已经确定了以下有效性威胁：

主题都是免费的 Android 应用程序。本文调查的所有项目都是免费的 Android 应用程序。因此，它们可能不代表收费应用程序和其他市场（例如 AppStore）中的应用程序[27]。商业应用程序可能有不同的审查模式。将来，我们将通过调查用户对商业应用程序和其他市场应用程序的评论来减轻这种威胁。

地面实况标签由两个人判断。由于黄金标准标签需要大量人力，因此在我们的实验中仅由两个人进行判断。他们可能会偏离真正的应用程序开发人员。为了减轻这种威胁，我们向开发人员展示了最终结果，并确保他们对准确性感到满意。未来，我们将通过邀请更多的开发者进行标注来进一步降低这种威胁。

第六节。
结论

我们提出了 SUR-Miner 用于有效和自动的用户评论摘要。SUR-Miner 的摘要为开发人员的重要问题“您的应用程序的哪些部分受到用户喜爱”提供了理想的答案。

Contents

我们的评估结果表明，SUR-Miner 提供了可靠的结果，评论分类、方面意见提取和情感分析的平均 FI 分数分别为 0.75、0.85 和 0.80。SUR-Miner 的最终方面比最先进的技术更准确、更清晰，FI 得分为 0.81，高于 ReviewSpotLight (0.56) 和 Guzmans 的方法 (0.55)。应用程序开发人员的反馈也非常鼓舞人心，88% 的开发人员回答都认同 SUR-Miner 的实用性。

以后我们会总结其他评论类别，比如功能请求。此外，我们将提出技术来总结其他软件文本数据，例如代码注释和错误报告。


作者	▼
人物	▼
参考	▼
引文	▼
关键字	▼
指标	▼
脚注	▼

更多类似产品

改进的基于 CNN 的数据挖掘特征提取
2022 年计算、通信和应用信息学进展国际会议 (ACCAI)
出版时间：2022

在数据挖掘中使用特征提取和特征选择技术进行心脏病分类的有效框架
2016 年工程、技术和科学新兴趋势国际会议 (ICETETS)
出版时间：2016

展示更多


Download
PDF

Contents

Q

T

- IEEE 个人帐户

更改用户名/密码
- 购买详情

付款方式

查看购买的文件
- 档案信息

通讯偏好

职业与教育

技术兴趣
- 需要帮忙？

美国和加拿大：+1 800 678 4333

全球：+1 732 981 0060

联系与支持
- 跟随

f in

关于 IEEE Xplore | 联系我们 | 帮助 | 辅助功能 | 使用条款 | 非歧视政策 | IEEE 道德报告 | 网站地图 | IEEE 隐私政策
作为一个非营利组织，IEEE 是世界上最大的技术专业组织，致力于为人造福推进技术。

© 版权所有 2023 IEEE - 保留所有权利。

- IEEE Account
- » Change Username/Password

» Update Address
- Purchase Details
- » Payment Options

» Order History

» View Purchased Documents
- Profile Information
- » Communications Preferences

» Profession and Education

» Technical Interests
- Need Help?
- » US & Canada: +1 800 678 4333

» Worldwide: +1 732 981 0060

» Contact & Support

PDF

