2015 年第 30 届 IEEE/ACM 自动化软件工程国际会议

“你的应用程序的哪些部分受到用户的喜爱?”1”

顾晓东和金成勋

计算机科学与工程系

香港科技大学，香港{xguaa，hunkim}@cse.ust.hk

*摘要——最近，Begel 等人发现，软件开发者最常问的一个问题是“软件的哪些部分被用户使用/喜爱”。***用户评论为解决这个问题提供了一个有效的渠道。然而，大多数现有的评论总结工具将评论视为词袋(即混合评论类别)，并且仅限于提取软件方面和用户偏好。**

我们提出了一种新颖的评论摘要框架，SUR- Miner。它没有使用词袋假设，而是将评论分为五类，并从句子中提取方面，其中包括使用基于模式的解析器对方面进行评估。然后，SUR- Miner 使用两个交互式图表将总结可视化。我们对 17 个流行应用程序的评估表明，SUR-Miner 比最先进的技术更准确、更清晰地总结了各个方面，平均 f1得分为 0.81，显著高于 ReviewSpotlight(0.56)和 Guzmans 的方法(0.55)。来自开发者的反馈显示，88%的开发者同意 SUR-Miner 总结的有用性。

*Index Terms-Review Summarization;***用户反馈;情绪分析;数据挖掘**

我的介绍。

通常软件开发人员都渴望知道他们的软件的哪些部分被用户使用/喜爱。根据一项针对 4000 名微软工程师的调查，*“软件产品的哪些部分(方面)最受客户使用和喜爱?”*的问题在[5]上开发者最常问的 145 个问题中排名第二。这个问题要求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者了解用户需求、偏好和抱怨[21]，[31]的重要渠道。通过分析用户评论，开发人员可以评估自己的产品，识别用户的偏好[21]，改进软件维护和演进任务[33]。

然而，理解软件评论是非常具有挑战性和繁琐的。首先，用户评论的数量太大，无法手工检查。开发者每天会收到成百上千条评论[10]、[31]。鉴于评论数量庞大，他们需要阅读并手动将评论分类为投诉或新功能请求[30]。这样的过程极其耗时和繁琐。另一方面，用户评论的种类太多，需要区分[31]。它们可以是新功能请求、bug 报告、表扬，也可以是抱怨。不同类型的评论针对不同的任务和开发人员[30]。例如，一个赞扬的评论可能对软件测试没有价值，但可能有价值

对产品评估至关重要。报告 bug 的评审对需求分析并不重要，但对软件测试却至关重要。考虑到数百万条评审，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汇总的工具。例如，Chen 等人[10]通过分类技术过滤非信息性评论，并应用潜在狄利 let 分配(LDA)[6]来总结信息性评论的主题。Fu 等人[15]通过回归模型过滤评级不一致的评论，这些评论的情绪不同于它们的评级。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级趋势。Iacob 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 模型是基于一个词袋假设，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，方面评估和特征请求)和情感的软件审查可能会有问题。由于这些工具混合了方面和观点，并且混合了与不同类别相关的主题，因此它们不能有效地衡量用户对每个方面的情绪。

为了解决这些限制，我们提出了软件用户评论挖掘器(Software User Review Miner, SUR-Miner)，这是一个可以总结用户对相应软件方面的看法和意见的框架。SUR- Miner 并没有把评论当作一袋单词来处理，而是充分利用了软件用户评论的单调结构和语义，根据预定义的句式直接从评论句子中解析出方面-意见对。然后，它分析每个评论句子的情感，并将情感与同一句子中的方面意见对关联起来。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。

我们在最近对 Swiftkey、Camera360、微信和 Templerun2 等 17 款安卓应用的用户评论中对 SUR-Miner 的性能进行了实证评估。我们通过 F1-score 来衡量关键过程(即分类、方面意见提取和情感分析)的性能，F1-score 是文本挖掘文献[14]，[38]中常见的精度度量。结果表明，SUR-Miner 产生了可靠的摘要，评论分类、方面意见提取和情感分析的平均 f1 得分分别为 0.75、0.85 和 0.80。与最先进的技术相比，SUR-Miner 的最终方面明显更准确、更清晰，f1 得分为 0.81，高于 ReviewSpotlight(0.56)和 Guzmans 的方法(0.55)。

1这个问题来自于 Begel 等人在微软[5]的一项研究

978-1-5090-0025-8/15 $31.00©2015 ieee doi 10.1109/ ase.2015.57

作为一个概念验证应用程序，我们设计了两个交互式图表，方面热图和方面趋势图，使用 SUR-Miner 的总结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。来自相应应用开发者的反馈也令人鼓舞，88%的受访者认为 su - miner 的总结是有用的，这表明 su - miner 可以帮助开发者在实践中了解用户对不同方面的偏好。

总体而言，我们的研究做出了以下贡献:1)

我们利用了一种分类技术，其中我们

2) 设计文本特征来区分五种审查类别，如 bug 报告和新功能请求。我们提出了一种基于模式的解析技术，该技术可以解析复杂的应用评论句子，并提取出方面和相应的意见。

3) 我们设计了新颖的交互式可视化，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。

本文的其余部分组织如下。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。第四节给出了评估。第五节讨论验证面临的威胁，第六节对论文进行总结。

2 相关工作

A.*应用评论过滤*

App 评论过滤在软件工程界引起了越来越多的关注。Chen 等人[10]对非信息性评论进行过滤，并根据重要性对用户评论进行排序。他们的框架训练了一个分类器，并将评论分为两类，即信息性和非信息性。Fu 等人[15]通过对评论词汇的回归模型过滤评级不一致的评论(情感与其评级不同的评论)。这些工具可以部分地选择信息丰富的评论。然而，由于不同的开发人员需要不同类型的评论[30]，[31]，它们并没有明确定义在什么情况下评论是信息性的。为了进一步开展他们的工作，我们的目标是区分不同的评审目的(类别)，并从特定类别中选择评审，以提取和总结软件方面。

Sorbo 等人最近的工作[13]提出了一个类似的想法，根据其目的对开发邮件进行分类。他们还设计了一种使用自然语言解析技术的分类方法。虽然他们的技术也可以应用于应用程序评论分类，但它不支持每个类别内的方面摘要。

B.*从应用评论中提取方面*

Aspect extraction 在软件工程中也得到了广泛的研究。Chen 等人[10]使用 LDA[6]提取评论的主题。Hu 等人[19]提出了一种 web 评论挖掘的方法。他们的方法提取频繁的词作为方面，链接相应的形容词词作为意见。Fu[15]解决了挖掘用户负面反馈的问题。他们应用 LDA 主题模型从负反馈中挖掘主题，并对每个版本的总结问题进行排序。Galvis 等人[16]通过采用一种名为 Aspect and Sentiment Unification model (ASUM)[22]的主题模型来挖掘需求变化。他们还提取常见主题，并呈现用户对这些主题的意见。

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(表扬、功能请求、bug 报告、缺点)，也不能区分方面和用户意见，这可能导致不准确和混乱。例如，由 LDA 提取的主题词“预测”可能意味着用户欣赏预测功能，或者用户希望获得新的预测功能。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，我们从应用程序评论中提取特征。此外，我们的目标是总结应用程序的功能，而他们的目标是调查功能生命周期[34]。

据我们所知，之前只有一项工作与我们的工作密切相关。Guzman 和 Maalej[17]提出了提取软件特征并分析其情绪的方法。我们的工作与他们的不同主要体现在三个方面。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的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人员和开发人员掌握特征评估和情感趋势。

C. *Review Mining in Other marketplace*

用户评论挖掘在其他市场(如商品、电影)也是一个有吸引力的话题。Yatani 等人[37]提出了一个评论摘要工具 ReviewSpotlight，它通过识别评论句子中的形容词-名词词对来提取方面-意见对。Huang 等人[20]采用了类似的思路，设计了 Revminer——一个用于总结餐厅评论的提取界面。Nichols 等人[29]提出了 ReCloud，它使用 NLP 技术解析评论句子。Zhuang 等人[38]研究了电影评论摘要。他们的方法集成了包括 WordNet、统计分析和电影知识在内的多种知识。

然而，这些技术很难直接应用到应用程序评论中。应用评论与其他市场[10]、[15]上的评论有很大不同。它们有不同的词汇和格式，现有的工具几乎无法解析。ReviewSpot- light[37]通过提取形容词-名词词对呈现了一个词云。同样，RevMiner[20]使用 bootstrapping 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:

案例 1:*“我喜欢我们可以改变主题的事实。”*

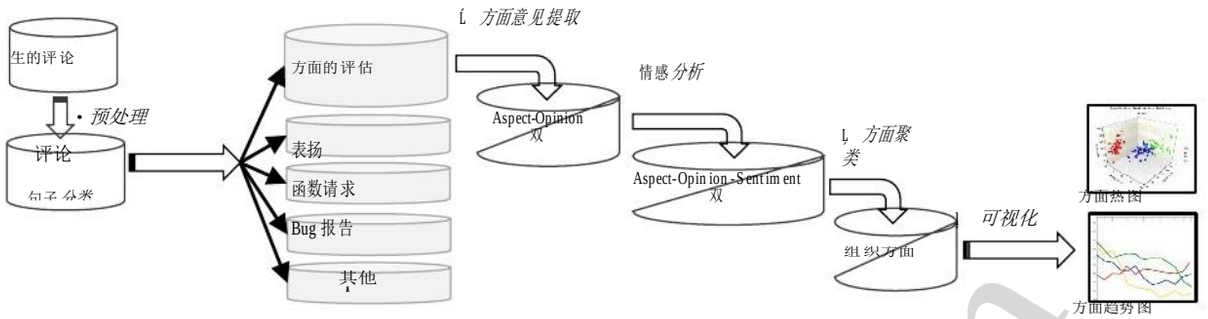


图 1:提出的 SUR-Miner 框架概述

ReviewSpotlight 不能输出任何东西，因为没有形容词。RevMiner 和 ReCloud 可能会呈现一些无意义的单词对。相比之下，SUR-Miner 可以呈现正确的对词*?我们可以改变主题，爱?*因为它考虑了语义和应用审查模式。此外，应用评论包含针对不同开发者的多种目的[31]。现有的工具都无法区分这样的类别。考虑以下情况

案例 2:*"点击'确定'按钮后蓝屏很烦人。"*

案例 3:*"简单的 UI 会更好。"*

从开发者的角度来看，它们只是一个 bug 报告和一个功能请求，不应该被认为是用户对"屏幕"和"UI"的意见。这样的案例在应用评论[31]中占了很大的比例。而所有这些工具仍然输出词对(云)如*?按钮，恼人?*和 *UI，简单?*，surminer 可以区分上述情况，因为它利用了分类技术。

3SUR-MINER

本节介绍 SUR- Miner 的通用架构。

如图 1 所示，我们的框架将包括文本和评分在内的用户评论作为输入，并输出对应用程序不同方面的主要意见和情绪。整个过程由六个主要步骤组成:对于需要总结的原始评论，我们首先将其分成句子(步骤 1)，然后将每个句子分为五类，即方面评价、*表扬*、*功能*请求、*bug 报告等(步骤 2)，然后，我们只选择方面评价类别中的*句子。然后，我们从"方面评价"的句子集合中提取方面和相应的意见和情感(步骤 3-4)。得到的方面-意见-情感对被聚类，并通过两个交互图表可视化(步骤 5- 6)。下面将详细解释每个步骤。

A.步骤 1 -预处理

原始用户评论需要预处理。它通常由一个以上的句子组成，目的不同。例如，一个原始的评论"*UI 是丑陋的。我想要一个漂亮的 UI*"由两句话组成。第一句话是对一个方面 UI 的评价，第二句是对方面 UI 的改进请求。它们有不同的目的和情感。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略语使得自动理解意思变得困难。

为了解决这两个问题，我们使用斯坦福 CoreNLP 工具[26]将原始评论文本分割成句子。每个评论句子都有时间戳，并分配了评级，与原始评论相同。我们还会纠正常见的拼写错误、缩写和重复，比如"U→you"、"coz→because"、"&→and"、"Plz→Please"、"soooo→so"和"thx→thanks"。我们收集了 60 个这样的错字和缩略词，并用正则表达式[2]代替。

B.步骤 2 - Review Classification

如第一部分所讨论的，复习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]。对于开发人员来说，手动对它们进行分类并选择合适的句子进行方面评估是非常繁琐和耗时的。在评论分类步骤中，我们的目标是自动对包含方面评价的评论句子进行分类和选择。

我们定义了五个评审类别，包括方面评价、bug *报告*、*特性请求*、*表扬等*。Pagano 等人发现了用户评论[31]的 17 个类别(主题)。我们使用他们分类法[31]中的前四个类别，并将其他次要类别合并为"其他"类别。表 1 说明了每个类别的定义和示例回顾句。

为了将复习句分类到上述类别中，我们采用了有监督的机器学习方法。我们首先收集历史评论句子，提取其文本特征，并根据表 1 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，我们在新的评论实例上执行分类器来预测它们的类别。

我们采用了一个著名的分类器，Max *Entropy，它在文本分类[24]，[28]上有很好的*表现。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:我们提取了文本特征的两个维度:词汇特征和结构特征。

由于不同的评论类别可能具有显著不同的词汇，因此词汇对于表征评论类别非常重要。例如，会出现"amazing"和"great"

[2]完整的错别字列表在 http://www.cse.ust.hk/~xguaa/srminer/appendix。超文本标记语言

表一:五个评审类别的定义

类别	定义	例子
赞美	没有具体原因地表达情感	太好了!我喜欢!神奇的!
方面的评估	对具体方面发表意见	UI 方便。 我喜欢预测文字。
错误报告	报告 bug、小故障或问题	当我点击“。com”按钮时，它总是强制关闭。
功能要求	建议或新功能请求	如果我能给点意见就更好了。 很遗憾它不支持中文。 真希望有个“拒绝”按钮。
其他人	[31]中定义的其他类别	我已经玩了三年了

频繁在*好评评论*中，而“bug”和“fix”是 bug *报告的代表性词汇。我们*选择字符 *N- Gram 和 trunk word* 作为两个*词汇特征*，因为它们反映了不同类别的词汇。

字符 N-Gram 是一种重要的词法表示，是文本分类[8]、[9]、[18]、[23]、[25]中常用的一种特征。在软件工程中的恶意代码检测[4]和重复 bug 报告检测[36]等许多应用中也被发现是有效的。一个句子的字符 n - gram 特征是该句子的令牌中所有连续的 n 个字母。例如，“the UI is OK”这句话的 3- gram 是 the、heU、*eUI*、*UIi*、*Iis*、*isO 和 sOK*。我们用 2-4 克进行分类。

主干词我们也提出*主干词*作为一个词汇特征。我们将主干词定义*为语义依赖图[12]的词根*，这将在本节后面介绍。例如，句子“the graphics are amazing”的主干词是“are”。

句子结构也可以反映文本特征，因为不同的复习类别可能有不同的语法和语义。例如，对于*方面评价*，用户倾向于使用描述性语法，如“*图形(名词)是惊人的(形容词)*”，而对于*功能请求*，用户经常使用祈使句，如“*请添加更多的主题*”和“*它可能会更好有更多的主题(名词)*”。

我们利用了三个结构特征:POS 标签、解析树和语义依赖图。

POS 标签词性(POS)[11]是文本中广泛使用的语法特征。它表示句子中每个单词的属性。例如，句子“The user interface is beautiful”的 POS 标签为 DT-NN-NN-VBZ-JJ，顺序为[11]。在这里，is 这个词的 POS 标签*是* VBZ，意思*是 is 是*第三人称现在时单数的动词。我们使用斯坦福 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[35]语法结构的典型表示。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，它的子节点是由它组成的子单元。图 2 展示了一个由 Stanford Parser[3]生成的样例复习句“The user interface is*not very elegant*”的解析树。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)组成，其中名词短语由一个限定词(DT)和两个名词(NN)组成。

为了将解析树表示为平面文本特征，我们

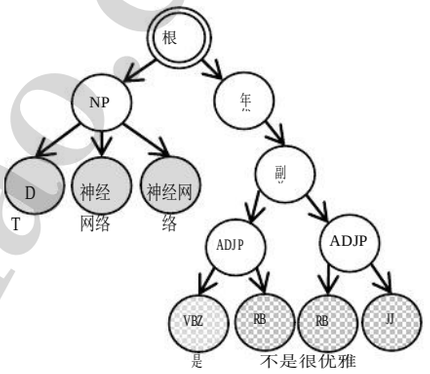


图 2:句子的解析树:*用户界面不是很优雅。*

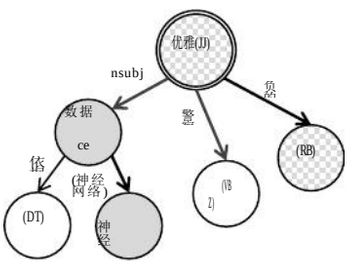


图 3:句子的语义依赖图:*用户界面不够优雅。*

按广度一阶遍历树节点，选择前 5 个节点。我们将这 5 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”。

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关系。它是一个有向图[12]。图中的节点表示单词和相应的 POS 标签。边表示词之间的语义关系(例如，名词从属关系和形容词修饰语)。每个 SDG 都有一个根节点，该节点没有传入边。图 3 展示了一个样本复习句“The*user interface is not elegant”的 SDG，它*是由 Stanford Parser[3]生成的。根节点是单词 elegant，*它是*一个形容词(标注为 JJ)。它有三个子节点:一个名词从属词(nsubj)接口，*一个连词*(cop) is 和一个否定*修饰语*(neg) not。子接口*也*有两个子接口:*限定词*(det)和名词复合修饰语*(nn)*用户。

为了将 SDG转换为平面文本特征，我们遍历其

节点的宽度优先顺序，然后在遍历中连接边和 POS 标签。我们忽略那些没有链接到根节点的叶节点。例如，图 3 中 SDG 的特征是“vbz - nsubject - nn -cop- vbz - negg - rb”。

C.步骤3 - Aspect-Opinion Extraction

我们的下一个目标是总结用户对相应方面的意见。要做到这一点，我们需要识别表达方面的词语和表达对这些方面的意见的词语。在这一步中，SUR-Miner 从每个分类在方面*评价类别*中的评论句子中提取方面-意见对(即方面和意见词)。例如，对于评论句子“the Prediction is accurate,*but the auto-correct is annoying*”，*得到的方面意见对是*:? *Prediction*, accuracy*?和?auto-correct*, annoying*?*

一般来说，最先进的技术通过频繁项目挖掘或主题模型提取方面，该模型将用户评论视为单词[6]，[10]，[15]的包。对于表现出多种目的和情感的软件评论，这样的假设可能是有问题的。

正如一项实证研究表明的那样，针对不同的目的，软件评审具有相当单调的模式[31]。因此，直接从句式中确定方面-意见对是可能的。基于这一假设，我们设计了一种基于模式的解析方法，利用复习句的语法和语义，直接从复习句中解析出方面和相应的意见对。要做到这一点，我们首先应用 NLP 解析器为复习句子标注语义依赖图(SDG)[12]。然后，我们构建一个基于模式的解析器，从 SDG 中提取方面-意见对。

1)基于模式的解析:我们基于模式的解析器被实现为一系列级联有限状态机[7]。解析器接受 SDG，并根据预定义的语义模板识别方面-意见对。

表 II 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)代表词根的 POS 标签。下面圆括号中的单词(例如 *have* 和 *like*)代表词根。词根的子词列在方括号中，作为 edge-POS 对。例如，第一行的模板表示一个 POS 标签为 JJ 的根节点和两个子节点:一个名词从属(nsubj)的 POS 标签为 NN，一个 copula (cop)的 POS 标签为 VBZ。我们通过从回顾句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 2000 个评论句子，标记为 *Aspect Evaluation，除了*我们后来用于评估准确性的那些。首先，我们遍历了所有这些句子并生成了它们的 sdg。然后，我们将每个可持续发展目标与一个模板关联起来，该模板表示可持续发展目标中方面部分和意见部分的位置。为了避免意外关联，我们选择了所有与 10 个以上句子相关的模板。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，解析器从根到所有其他节点，检查节点、边和相应的子节点，以确定方面和意见文字根据模板。例如，给定图 3 中的 SDG，解析器检查词根的 POS 标签。由于它是一个形容词(JJ)，匹配表 II 中的第一个和第二个模板，因此它进一步检查它是否有三个子词:一个名词从属词(nsubj)，其 POS 标签为名词(NN)，一个联结词(cop)，其 POS 标签为 VBZ，一个否定修饰语(negg)，其 POS 标签为 RB。第二个模板是匹配的。然后，它检查第一个子模板是否有名词复合修饰语(nn)的子模板，其 POS 标签为名词(nn)。由于第二个模板与样本 SDG 是绝对匹配的，因此解析器将 ntheme - nn 节点接口*及其子用户*识别为方面词，*而*将不与根节点优雅的- rb *节点*识别为意见词。

D.第4步-方面情感分析

除了意见之外，对用户对每个方面的感受进行定量总结，可能也有助于把握用户的偏好。用户的评分可以客观地提供这样的总结。然而，总体评分并不能令人满意地描述用户对不同方面的偏好。例如，考虑这样的评论:“*UI 很好，但声音很糟糕。*”，评分为 2 分(满分 5 分)，用户显然喜欢方面的 *UI*，但不喜欢方面的*声音*。因此，这两个方面的实际评分不可能都是 2;*UI*可能是 3 分，*声音可能是 1 分*。

在第四步，我们对每个评论句子应用情感分析，并将情感与用户评分和情感分析工具的相应方面关联起来。我们首先应用最先进的情感分析工具 *deep Moving[1]来分析*每个评论句子的情感。The deep Moving *产生 0 到 4 级的情绪*，其中 4 表示强烈积极，0 表示强烈消极，2 表示中性。然后，为了提高准确性，我们根据用户评分(1 到 5)来调整情绪。具体来说，如果整个评论的评分是 5(强烈正面)，我们在 0 的情绪上加 1。如果评分为 1(强烈负面)，我们将 4 的情绪减去 1。

例如，下面的评论有两句话:*界面很漂亮。我不喜欢主题。*两句话的情绪分别是 4 分和 0 分。如果用户对评论的评分是 5，我们将第二句话的情感调整为 1(= 0+1)。如果用户评分为 1，我们将第一句的情感调整为 3(= 4-1)。

E.步骤5 -方面聚类和总结

在这一步中，我们将具有相同方面的方面-意见对分组，并总结每个方面组的情绪和典型意见。

为了对方面进行分组，我们首先挖掘所有方面词的频繁项，即抽取的方面-意见对中的方面词。然后，我们用常见的频繁项(词)聚类方面意见对。例如，假设自动*纠正是所有方面词中的一个频繁*项，如果有两个方面意见对包含这个项目，它们将被聚类成一组。特别是，如果一对有两个或两个以上的频繁项目，可以聚类到两个以上不同的组中，我们将其聚类到项目或单词频率最高的组中。比如一对?背景*颜色*，*好看吗*?都能分组吗

[3] 完整的模板列表在 http://www.cse.ust.hk/ xguaa/srminer/ appendix.html

表二:依赖关系模板的例子

模板	样的句子	方面的话	意见的话
JJ (nsubj-NN, cop-VBZ)	UI 很美!	nsubj-NN	JJ
JJ [nsubj-NN [nn-NN], cop-VBZ neg-RB]	用户界面不够优雅。	nn-NN + nsubject -nn	- rb + JJ
神经网络 (amod-JJ)	漂亮的 UI。	神经网络	amod-JJ
VB(已经)[nsubj-NN nobj-NN]	框架有不错的 UI!	nsubj-NN	have + nobj-NN
VB(像)[nsubj(我),nobj-NN]	我喜欢它的 UI!	nobj-NN	就像

*?背景,漂亮吗?*和*?颜色，恶心?*但是，如果我们已经知道 aspect *背景*的频率高于 *color*，我们会将第一对与第二对分组，而不是将第三对分组。如果两个方面-意见对中没有频繁项，当它们在各自方面有常用词时，我们将它们分组在一起。

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单词或项目。我们还将*群体情绪*计算为该组中方面-意见对的平均调整情绪。

F.第 6 步-可视化

我们设计了两个交互式图表，即*方面热图*和*方面趋势图*，以说明总结。

*方面热图*展示了用户关心的流行方面。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 展示了一个方面热图的例子，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。我们将圈的大小定义为 size =*log*(#*comments*) + *sentiment*。横轴表示评论数，纵轴表示调整后的评分。因此，右上方的圆圈代表最受欢迎和喜爱的方面，反之亦然。为了深入了解一个方面组，开发者可以点击每个方面(圈)，查看正面和负面情绪最高的具体评论。对于每条评论，方面词都用下划线标注，意见词用粗体标注。

*方面趋势图*展示了随着时间推移的情绪趋势。捕捉用户反应对于开发者选择和优先考虑[16]，[34]功能非常重要。*方面趋势图*旨在帮助开发人员评估他们最近的变化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部分产品。图 5 展示了该图表的一个示例，每条线表示一个受欢迎方面的情绪趋势。横轴表示日期，纵轴表示用户情绪。

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/**上获得。

Iv.实证评价

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，我们应用了文本挖掘文献中的常用测量方法，并将结果与最先进的方法进行了比较。我们还进行了开发人员调查来评估

表三:App 主题概述

数据集	类别	时间		
Swiftkey	生产力	8.26.2014	...	9.10.2014
Camera360	摄影	8.24.2014	...	9.8.2014
Templerun2	游戏	8.30.2014	...	9.10.2014
微信	社交网络	9.5.2014	...	9.11.2014
KakaoTalk	沟通	6.22.2014	...	9.12.2014
GooglePlayBooks	书	12.16.2014	...	3.18.2015
SpotifyMusic	音乐	3.8.2015	...	3.18.2015
YahooWeather	天气	1.30.2015	...	3.18.2015
GoogleMap	地图	3.6.2015	...	3.20.2015
GoogleCalendar	生产力	2.4.2015	...	3.20.2015
ESPN	体育	9.19.2014	...	3.21.2015
TextPlus	社会	12.16.2014	...	3.21.2015
Duolingo	教育	3.5.2015	...	3.22.2015
Chasemobile	金融	9.17.2014	...	3.23.2015
起到了推动作用	医疗	1.7.2013	...	3.23.2015
Yelp	食物	12.8.2014	...	3.25.2015
IMDB	娱乐	10.13.2014	...	3.29.2015

的实用性。具体来说，我们的评估解决了以下研究问题:

- **RQ1**(有效性):SUR-Miner 如何有效地对评论进行分类，提取方面和意见，并分析应用评论的情绪？
- **RQ2**(比较):与最先进的应用评论总结技术相比，SUR-Miner 如何？

•**RQ3**(实用性):SUR-Miner 的总结对开发人员有何用处?

A.数据收集

我们从谷歌 Play 中选择 Swiftkey、Camera360、微信、Templerun2 等 17 款热门安卓应用作为我们的研究对象。这些 app 涵盖了游戏、通讯、书籍、音乐等 16 个最受欢迎的类别。我们大致在 2014 年 8 月至 2015 年 3 月期间使用开源 Android 市场 API[2]收集了这些评论。对于每条评论，我们都会收集它的时间戳、评分、标题和内容。表 III 显示了受试者的描述。

B.有效性(RQ1)

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

*1)评论分类:首先，我们在评论*分类任务上对 SUR-Miner 进行评估。我们从每个数据集中抽取了 2000 个评论句子，并将预测结果与金标准标签进行了比较。我们根据表 1 中的规则手动标注黄金标准类。为了减少标注偏差，两位研究人员分别将标注规则应用于 2000 个评论句子。共识的标签

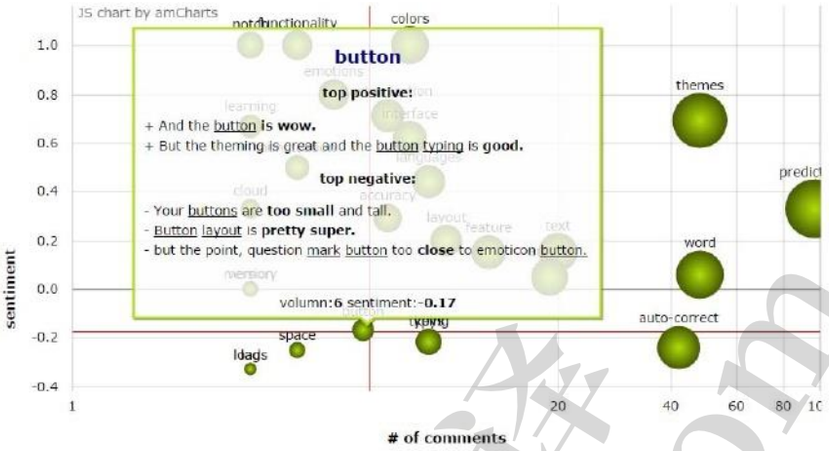


图 4:方面热图的演示

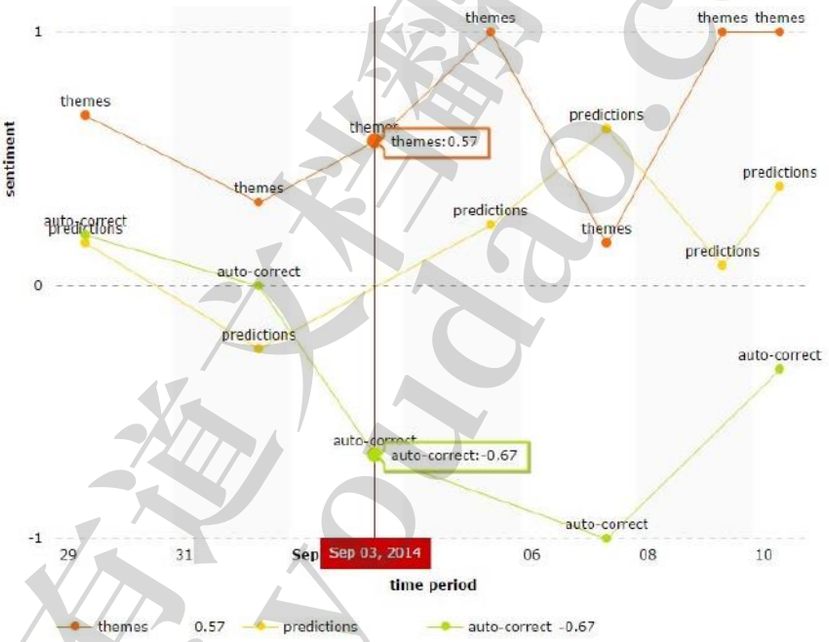


图 5:方面趋势图演示

在第一次迭代中被选中。对于分歧，我们讨论并澄清了我们的标注规则，并再次重新标注。第二次迭代的结果是两位研究者 100%的一致。

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

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \quad (1)$$

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+FP)之比。

$$precision = \frac{TP}{TP + FP} \quad (2)$$

召回率是正确分类为一个类的实例数(TP)与该类的实例数(TP+FN)之比。

$$recall = \frac{TP}{TP + FN} \quad (3)$$

我们在数据集中执行了 100 次五重交叉验证[38]，每个文件夹包含 400 个评论句子。

表 IV 显示了不同类别的 f1 分数 [4]。每一列显示了一个复习类别在所有科目中的 f1 分数。最后一列是所有复习类别中每个科目的成绩的平均值，最后一行是所有科目中每个复习类别的 F1-分数的平均值。如表所示，分类性能是合理的(带有

[4]包括精密度和召回率在内的完整结果见 http://www.cse.ust.hk/~xguaa/srminer/appendix.html

表 IV:所有受试者的评价分类 f1 得分

类别	评价表扬		请求	错误	其他	整体
Swiftkey	0.72	0.87	0.78	0.58	0.86	0.76
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
微信	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
GooglePlayBooks	0.59	0.92	0.72	0.60	0.85	0.74
SpotifyMusic	0.68	0.94	0.54	0.57	0.87	0.72
YahooWeather	0.83	0.94	0.57	0.56	0.87	0.76
GoogleMap	0.73	0.88	0.60	0.76	0.84	0.76
GoogleCalendar	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
起到了推动作用	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
平均	0.74	0.90	0.59	0.65	0.86	0.75

表五:方面意见提取的 f1 得分

数据集	方面	的意见	情绪积极	情绪负面的
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
微信	0.83	0.82	0.77	0.83
KakaoTalk	0.84	0.87	0.85	0.77
GooglePlayBooks	0.84	0.86	0.82	0.82
SpotifyMusic	0.84	0.86	0.83	0.62
YahooWeather	0.90	0.63	0.89	0.77
GoogleMap	0.86	0.84	0.88	0.88
GoogleCalendar	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
起到了推动作用	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
平均	0.85	0.84	0.85	0.75

平均 f1 得分为 0.75)，以及*方面评价*类别(平均 f1 得分为 0.74)。这意味着分类步骤可以准确地为不同的开发人员提供不同类型的评论句子。特别是，它为方面评价提供了可靠的评论句。在某些应用中，“bug”等特定类别的 f1 得分并不好。我们手动检查了这些评论，发现这些应用收到了罕见的 bug 报告。极度不平衡的数据可能是这些异常值的主要原因。

2)方面意见提取:为了评估 su - miner 在方面意见提取上的表现，我们遵循与评论分类实验相同的步骤，检查 su - miner 是否正确地从评论句子中提取了方面和相应的意见。对于每个主题，我们都进行了抽样

2000 个评论句子，并选择了那些在*方面评价*类别。我们使用 F1-score 分别衡量方面提取和意见提取的准确性。特别是公式 1-中真阳性(TP)的数量

3 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;假阴性数(FN)定义为未被提取的方面或意见的数量。

结果显示在表 V的前两列中。

如表所示，方面提取和意见提取都具有合理的准确性，平均 f1 得分分别 [4] 为 0.85 和 0.84。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，我们也遵循与分类和方面提取阶段相同的程序。对于每个主题，我们抽样了 2000 个评论句子，并选择了“方面评价”*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比较*。为了简化估计，我们将情绪量表(0- 4)分为两个极性，即正面(3-4)和负面(0- 1)[32]，并根据其极性对其进行标记。我们手动标记黄金标准情绪，就像我们对评论分类所做的那样。

我们使用 F1-score 来衡量每个情感类别的准确性。特别地，公式 1-3 中的真阳性数(TP)被定义为正确分类的情绪的数量;假阳性数(FP)表示错误分类的情绪数;假阴性数(FN)表示未分类在该类别中的情绪数量。

结果显示在表 v 的最后两列中。如所示，积极和消极情绪都具有可接受的准确性，平均 f1 得分分别 [4] 为 0.85 和 0.75。两者的平均 f1 得分均为 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获得了更多的正面评价，以至于情绪类别变得极其不平衡。结果表明，情绪分析步骤产生了可靠的结果。**吗 ??**

surm - miner 提供可靠的审查结果

分类、观点提取和情感分析，平均 f1 得分为 0.75，

*分别为 0.85 和 0.80。*吗？

C.比较(RQ2)

我们的下一个评估旨在将 SUR-Miner 与最先进的技术在最终总结方面进行比较。

1)定量比较:我们首先将 SUR-Miner 在方面提取方面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]。如第 II 节所述，ReviewSpotlight 是一个通过识别名词-形容词对(Section II- c)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，它也从应用程序用户评论中提取方面(Section II- 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除了训练分类器之外的所有类别中随机选择 400 个评论句子。首先，我们对这些句子进行复习分类。然后，我们对分类为 aspect Evaluation 的句子应用 aspect 提取。我们将提取的方面与人工标记的黄金标准方面进行比较。我们使用 F1-score 来评估使用章节 IV-B2 中相同定义的准确性。

表六:方面提取精度与相关作品的比较

度规	SUR-Miner	ReviewSpotlight	古斯曼 和 Maalej [17]
F1-score	0.81	0.56	0.55

表六显示了三种方法在所有科目中的平均 f1 分数。我们复制了 ReviewSpotlight，并将其应用于提取 app 方面。Guzmans 方法的结果摘自他们的论文[17]。我们可以看到，su - miner的 f1 得分为 0.81，显著高于 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，在不区分审查类别的情况下，它倾向于在其他类别(如方面请求和 *bug 报告*)*中为审查提取方面*。例如，考虑评论*“我讨厌你不能使用离线字典”*，这需要一个新的方面*离线字典*。ReviewSpotlight 只是输出*?dictionary, offline?*这是没有意义的，而 su - 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，它们不能识别复杂的短语，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 su - miner则输出*?它自动广告提醒的方式，爱?*

同样有趣的是，尽管分类和提取阶段都有错误，但将它们结合起来并不会导致准确性变差。分类步骤的 f1 得分为 0.74。方面提取步骤的 F1-得分为 0.85(章节 IV-B)。然而，当从分类阶段的输出中提取方面时，最终的 f1 -得分为 0.81，甚至大于分类阶段。通过手动检查提取的方面，我们发现尽管在分类阶段有一些评论被错误分类，但方面提取阶段仍然可以“重新纠正”它们，因为错误分类的评论可能无法被我们的语义模式解析。例如，考虑一个错误分类的评论“没有公共交通导航!”，它需要一个新的方面，但在分类阶段被错误地分类为方面评价。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这篇评论。

2)定性比较:LDA 等主题模型被大多数最先进的应用程序评论汇总工具[10]，[15]，[17]广泛使用。为了研究 SUR-Miner相对于这些基于主题的技术的优势，我们对 SUR-Miner 提取的方面与主题模型提取的主题进行了定性比较。

表 VIII 将我们提取的前五个方面与 AR-Miner(一种应用 EMNB-LDA 主题模型的最先进的评论总结工具)[10]在 Swiftkey 主题中提取的前五个*主题进行了比较*。我们从谷歌 Play 中收集了与 AR-Miner 相同时期的数据。我们有两个观察结果:1)ar - miner可以区分不同的评论目的。例如，由 su - Miner提取的意见是除一些噪声外的方面评价，而 LDA (AR-Miner)提取的 top words则是杂项。例如，如果一个经理会

表八:主题(LDA)和方面(SUR- Miner)的比较

(a) AR-Miner (LDA)[10]的主题和典型词汇。第一行列出了排名前 5 位的话题。接下来的 4 行列出了每个主题的前 5 个单词

主题	主题	中国人	优柔寡断的人	预测	空间
关键字	更多的	门外语	豆	词	空间
	主题	中国	jelli	预测	期
	希望	需要	galaxi	文本	电子邮件
	爱	等待	请注意	完整	输入
	自定义	用户	键盘	汽车	插入

(b) SUR-Miner 摘录的方面和意见。第一行列出评论最多的前 5 个方面。接下来的三行显示了最积极的观点，而最后两行显示了最消极的观点

方面	自动纠错的预测……话说. .主题…				关键
意见	令人惊异的	完美的	爱	伟大的	最好的
	优秀的	好	就像	爱	就像
	令人惊异的	令人惊异的	就像	在顶部	
	准确的	顽固的	一种痛苦	丑陋的	断路器
	讨厌	噩梦不需要		只是	讨厌的

想知道用户对方面*预测*的评价，SUR- Miner 可以提供*优秀*、*准确*、*讨厌*等用户意见，而 AR-Miner (LDA)无法提供此类信息;2) SUR-Miner 可以区分用户的情绪，AR- Miner (LDA)不能。例如，管理人员和开发人员可以通过 su - miner找到积极和消极的情绪，但无法判断用户是否喜欢或不喜欢 LDA 的方面预测。

总的来说，与 LDA 模型相比，su - miner在区分评论目的和情绪方面产生了更清晰的总结。

吗？

与最先进的方法相比，SUR-Miner产生更准确、更清晰的摘要。

吗？

D.实用性(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 应用到 Swiftkey、Camera360、微信和 Templerun2 等 17 款热门安卓应用的最新用户评价中。我们在网站上展示了可视化的总结作为演示，并向开发人员提出了表 VII 中所示的问题。这两个问题分别与这两张图有关。我们为他们每个人提供了 5 个选项(5 个非常同意，4 个同意，3 个都不同意，2 个不同意，1 个非常不同意)。我们还列出了每个问题的每个选项的选项数。

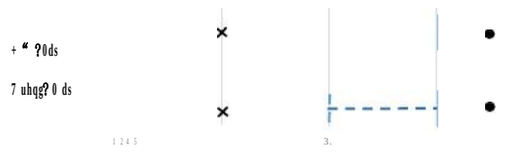
我们向入选应用的开发者发送了邀请邮件，并在谷歌+的安卓开发者社区发布了我们的网站，同时还邀请了三星、腾讯、百度等 IT 公司的开发者进行反馈。

开发者们对我们的 SUR-Miner 表现出了极大的兴趣。如表七所示，在收到的所有 32 个答案中，有 28 个(88%)同意

我们的工具对开发人员有帮助。只有两个持保守意见(6.3%)，两个持反对意见(6.3%)。图 6 为开发者反馈的箱形图统计。

表七:开发者调查中的问题和结果

问题	强烈不同意	不同意	既不	同意	强烈同意	总计
Q1。你觉得图 1“方面热图”对了解用户有用吗对方面的偏好?	0	1	0	7	8	22
Q2。你觉得图 2。“方面趋势图”帮助开发者了解用户偏好随时间变化的趋势?	0	1	2	5	8	22



6 wurqjo \?GLVDJUHH 6 wurqjo \?DJUHH 图 6:开发者对实用性的评价

我们将评分的答案量化，从 1到 5。每个方框显示一个问题的答案。结果表明，这两个问题的答案的平均评分远远大于 3分。这意味着开发人员总体上同意 SUR-Miner 的有用性。

此外，我们还收到了来自开发者的以下鼓舞人心的评论:

“这是一个伟大的项目。可视化数据给我留下了深刻的印象!”

“我想如果可能的话，我们愿意与这些研究人员合作。我真的很喜欢你的情感分类器的表现。”

“提供的可视化信息是一种非常清晰的方式来了解产品，包括优点和缺点。分析大规模的用户评论需要大量的人力。这样的项目使得产品的理解和迭代速度更快。”

这些评论表明，开发人员很欣赏我们的工具，以帮助掌握用户对不同方面的意见。

吗??

开发者反馈表明，我们的SUR-Miner在实践中帮助开发者掌握用户的意见和情绪。

吗??

五、有效性威胁

我们确定了以下对效度的威胁:

实验对象都是免费的安卓应用。本文调查的所有项目均为免费的 Android 应用。因此，它们可能不能代表收费应用和其他市场(例如 AppStore)中的应用[27]。商业应用可能有不同的评价模式。未来，我们将通过调查商业应用和其他市场应用的用户评论来缓解这一威胁。

地面真相标签由两个人来评判。由于黄金标准标签需要大量的人力，所以在我们的实验中，它们只由两个人来判断。他们可能会对真正的应用程序开发人员产生偏见。为了减轻这种威胁，我们向开发者展示了最终结果，并确保他们对准确性感到满意。在未来，我们将通过邀请更多的开发人员进行标注来进一步降低这一威胁。

六。结论

我们提出了 SUR-Miner 用于有效和自动的用户评论汇总。来自 SUR-Miner 的总结为开发人员提供了一个理想的答案，回答了*“你的应用程序的哪些部分是用户喜欢*的”这个重要问题。

我们的评估结果显示，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 f1 得分分别为 0.75、0.85 和 0.80。SUR- Miner 的最终方面明显比最先进的技术更准确、更清晰，f1 得分为 0.81，高于 ReviewSpotlight(0.56)和 Guzmans 的方法(0.55)。来自应用开发者的反馈也非常鼓舞人心，88%的开发者回答都认同 SUR-Miner 的有用性。

在未来，我们会总结其他的评论类别，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，如代码注释和 bug 报告。

参考文献

[1] 深深感动:用于情感分析的深度学习。http://www-nlp.stanford.edu/sentiment/。检索于 2014 年 6 月 02 日。

[2] android 市场的开源 api。https://code.google.com/p/android-market-api/。检索于 2014 年 9 月 1 日。

[3] 斯坦福解析器:一个统计解析器。http://nlp.stanford.edu/software/lex-parser.shtml。检索于 2014 年 6 月 02 日。

[10] T. Abou-Assaleh, N. Cercone, V.Ke + selj, R. Sweidan。基于 N-gram 的新恶意代码检测。*计算机软件与应用会议，2004。COMPSAC 2004。第28 届国际年会论文集*，第 2 卷，41-42 页，2004 年。

[5] A. Begel 和 T. Zimmermann。分析这个!软件工程领域数据科学家的 145 个问题。见*《第 36 届国际软件工程会议论文集》*，2014 年 12-23 页。

[6] D. M. Blei, A. Y. Ng 和 M. I. Jordan。潜在狄利克雷分配。*机器学习研究*，3:993-1022,2003。

[7] B. K. Boguraev。走向词汇衔接的有限状态分析。*《第三届* NLP 有限状态方法国际会议论文集》，2000 年。

[8] P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra，和 J. C. Lai。基于类的自然语言 n-gram 模型。*计算语言学*，18(4):467-479,1992。

[9] W. B. Cavnar, J. M. Trenkle，等。基于 n -gram 的文本分类。*Ann Arbor MI*, 48113(2): 161-175, 1994。

[10]陈宁，林俊杰，郝世昌，肖晓霞，张斌。Ar-miner:来自移动应用市场的开发者挖掘信息评论。*第 36 届国际软件工程会议论文集*，767-778 页，2014。

[11] D. Das 和 S. Petrov。基于双语图投影的无监督词性标注。*《计算语言学协会第 49 届年会论文集:人类语言技术》第 1 卷*，第 600-609 页，2011 年。

[12] M.-C。De Marneffe, B. MacCartney, C. D. Manning，等。从短语结构解析生成类型化依赖解析。*《Proceedings of LREC》*，第 6 卷，449-454 页，2006 年。

[13] A. Di Sorbo, S. Panichella, C. Visaggio, M. Di Penta, G. Canfora，和 H. Gall。开发邮件内容分析器:开发者讨论中的意图挖掘。在*第 30 届自动化软件工程国际会议(ASE 2015)上。林肯，内布拉斯加州*，2015。

[14]丁旭，刘 b，余 p.s.。一种基于词典的整体意见挖掘方法。*2008 年网络搜索和数据挖掘国际会议论文集*，页 231 - 240,2008。

[15] B. Fu, J. Lin, L. Li, C. Faloutsos, J. Hong, N. Sadeh 。为什么人们讨厌你的应用:在移动应用商店中理解用户反馈。《第 *19 届ACM SIGKDD 知识发现与数据挖掘国际会议论文集》*，第 1276-1284 页，2013。

[b] L. V.加尔维斯·卡雷·诺和 K.温布思。用户评论分析:软件需求演化的一种方法。*《2013 年软件工程国际会议论文集》*，第 582 - 591 页，2013 年。

[17] E. Guzman 和 W. Maalej。用户觉得这个功能怎么样?对应用评论进行细粒度的情感分析。在需求*工程会议(RE)，2014 年 IEEE 第 22 届国际会议，页 153-162,2014*。

[18] J. Houvardas 和 E. Stamatatos。作者身份识别的 N-gram 特征选择。见《人工*智能:方法论*、系统和应用》，第 77-86 页。施普林格,2006 年。

[19]胡明，刘斌。挖掘和总结客户评论。*第十届 ACM SIGKDD 知识发现与数据挖掘国际会议论文集*，第 168-177 页，2004 年。

[20] J. Huang, O. Etzioni, L. Zettlemoyer, K. Clark，和 C. Lee。Revminer:用于智能手机上浏览评论的提取界面。在*第 25 届 ACM 年度用户界面软件与技术研讨会*论文集中，第 3-12 页，2012。

[21] C. Iacob 和 R. Harrison。从在线评论中检索和分析移动应用程序的功能请求。In *Mining Software Repositories (MSR)，2013 年第10 届IEEE 工作会议*，第 41-44 页，2013。

[22] Y. Jo 和 A. H. Oh。面向在线评论分析的面向和情感统一模型。在*第四届ACM 网络搜索和数据挖掘国际会议论文集中，页 815-824,2011*。

[23] V. Keselj, F. Peng, N. Cercone, and C. Thomas。基于 n -gram 的作者简介，用于作者归属。《*太平洋计算语言学协会会议*论文集》，PACLING，第3卷，255-264 页，2003 年。

[24] R. Konig, R. Renner，和 C. Schaffner。最小和最大熵的运算意义。*信息理论，IEEE 学报*，55(9):4337-4347,2009。

[b] S. Lahiri 和 R. Mihalcea。使用 n-gram 和单词网络特征进行母语识别。《*第八届 NLP 创新应用于构建教育应用研讨会论文集》，第 251-259 页*，2013 年。

[26] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and d . McClosky。斯坦福大学 CoreNLP 自然语言处理工具包。*《计算语言学协会第 52 届年会论文集:系统演示》*，第 55-60 页，2014。

[27] W. Martin, M. Harman, Y. Jia, F. Sarro, and Y. Zhang。应用商店挖掘的应用抽样问题。In *Mining Software Repositories (MSR)，第十二届 IEEE 工作会议*，2015。

[28] A. McCallum, D. Freitag，和 F. C. Pereira。用于信息提取和分割的最大熵马尔可夫模型。*《第十七届机器学习国际会议论文集》*，591-598 页，2000 年。

[29] J. Nichols, M. Zhou, H. Yang, J.-H.。康，孙晓辉。分析社交媒体上从目标陌生人那里征集的信息质量。*《2013 年计算机支持的合作工作会议论文集》*，967-976 页，2013 年。

[30] D. Pagano 和 B. Bruegge。软件进化实践中的用户参与:一个案例研究。*《2013 年国际软件工程会议论文集》*，953-962 页，2013。

[31] D. Pagano and W. Maalej。appstore 中的用户反馈:一项实证研究。载于*需求工程会议(RE)，2013 年第 21 届 IEEE 国际*会议，125-134 页，2013。

[32] B. Pang 和 L. Lee。观点挖掘与情绪分析。*信息检索的基础与趋势*，2(1-2):1 - 135,2008。

[10] S. Panichella, A. Di Sorbo, E. Guzman, C. Visaggio, G. Canfora, H. Gall。如何改进我的应用程序?对用户评论进行分类，用于软件维护和进化。*第 31 届软件维护与进化国际会议(ICSME 2015)论文集。德国不莱梅*。

[34] F. Sarro, A. A. Al-Subaihin, M. Harman, Y. Jia, W. Martin, and Y. Zhang。在应用商店中传播、迁移、保留和消亡的功能生命周期。In *Requirements Engineering Conference (RE)，2015 IEEE 第23 届国际*会议，2015。

[35] R. Socher, J. Bauer, C. D. Manning，和 A. Y. Ng。使用组合向量语法进行解析。摘自 2013 年 *ACL 会议论文集*。

[36] A. Sureka 和 P. Jalote。使用基于字符 n-gram 的特征检测重复的 bug 报告。*软件工程会议(APSEC)，2010 年第 17 届亚太地区，366-374 页，2010*。

[37] K. Yatani, M. Novati, A. Trusty，和 K. N. Truong。评论聚光灯:一个使用形容词-名词词对总结用户生成评论的用户界面。*《SIGCHI 会议论文集:计算系统中的人为因素》*，第 1541-1550 页，2011 年。

[38]庄莉，井芳，x.y。朱。影评挖掘与总结。*《第 15 届 ACM 信息与知识管理国际会议论文集》*，第 43-50 页，2006 年。