系统与软件学报(2018):125-146

内容列表可在 ScienceDirect 查阅

The Journal of Systems & Software

期刊主页:www.elsevier.com/locate/jss

SentiStrength-SE:利用领域特异性来改进软件工程文本中的情感分析

Md Rakibul Islam, Minhaz F. Zibran

新奥尔良大学(University of New Orleans),美国洛杉矶新奥尔良

A 是我的母校,也是我的母校

关键词:情感情感软件
工程实证研究自动化
领域词典

摘要

软件工程文本工件中的自动情感分析长期以来一直受到那些少数可用工具的不准确性的困扰。我们进行了深入的定性研究,以确定造成如此低准确率的困难。然后通过构建领域字典和适当的启发式方法仔细解决大多数暴露的困难。然后在 SentiStrength-SE 中实现这些特定领域的技术,SentiStrength-SE 是我们为改进文本情感分析而开发的工具,专为软件工程领域的应用而设计。

使用由 5,600 个手动注释的 JIRA 问题评论组成的基准数据集,我们对我们的工具进行定性和定量评估。我们还分别评估了 SentiStrength-SE 的各个主要组件(即领域字典和启发式)的贡献。经验评估证实,在我们的 SentiStrength-se 中利用的领域特异性使其在检测软件工程文本中的情感方面大大优于现有的领域独立工具/工具包(SentiStrength, NLTK 和 Stanford NLP)。

1.介绍

情绪是人性不可分割的一部分,它影响着人们的活动和互动,因此情绪会影响任务质量、生产力、创造力、群体关系和工作满意度(Choudhury and Counts, 2013)。软件开发高度依赖于人的努力和互动,更容易受到实践者情绪的影响。因此,很好地理解开发人员的情绪及其影响因素可以用于有效的协作、任务分配(Dewan, 2015),并设计提高工作满意度的措施,从而提高生产力和项目的成功。

过去已经进行了几项研究,以了解人在软件开发和工程中的作用。其中一些早期的研究解决了员工何时以及为什么会受到情绪的影响(Choudhury 和 Counts, 2013;Guzman et al., 2014;Guzman and Bruegge, 2013;Pletea et al., 2014;Tourani et al., 2014),而其他一些工作则解决了如何(Graziotin et al., 2013;Islam and Zibran, 2016a;2016 b;Lesiuk, 2005;Mäntylä 等人,2016;Murgia 等人,2014; “2013;2016)情绪影响员工在工作中的表现。

试图捕捉开发人员的情绪通过传统的方法,如访谈、调查(Wrobel, 2013)和生物测量(McDuff et al., 2012)。对于依赖于地理分布的团队设置和自愿贡献(例如,开源项目)的项目来说,用传统方法捕捉情感更具挑战性(Guzman et al., 2014;Destefanis et al., 2015)。此外,涉及直接观察和与开发人员交互的传统方法往往会阻碍他们的自然工作流程。因此,为了补充或补充这些传统方法,最近的尝试是从软件工程文本工件(如 issue 注释)中检测情感(Guzman et al., 2014;Pletea et al., 2014;Islam and Zibran, 2016a;2016 b;Mäntylä 等人,2016;Ortu 等人,2015;Calefato 和 Lanubile, 2016;Chowdhury and Hindle, 2016),邮件内容(Tourani et al., 2014;Garcia et al., 2013),以及论坛帖子(Guzman and Bruegge, 2013;Novielli et al., 2014)。

为了从软件工程领域的文本工件中自动提取情感,使用了三种工具(即 SentiStrength(Thelwall 等人,2012),NLTK(自然语言工具箱)(NLTK, 0000)和 Stanford NLP(Socher 等人,2013b)),而发现使用 SentiStrength 占主导地位(Jongeling 等人,2015;Novielli, 0000)。然而,软件工程研究(Pletea et al., 2014;Tourani et al., 2014;Islam and Zibran, 2016a;Calefato and Lanubile, 2016;乔杜里和辛德尔,2016;

* 相应的作者。

电子邮件:mislam2@uno.edu (M.R. Islam), zibran@cs.uno.edu (M.F. Zibran)

https://doi.org/10.1016/j.jss.2018.08.030

2017 年 7 月 3 日收稿;2018 年 6 月 18 日收到修改稿;2018 年 8 月 10 日接受,2018 年 8 月 11 日在线提供

Jongeling et al.， 2015;Novielli 等，2015;Tourani 和 Adams, 2016)涉及情感分析的研究反复报告了对这些情感分析工具在检测纯文本内容的情感极性(即消极、积极和中立)时的准确性的担忧。例如，当应用于软件工程领域时，据报道，SentiStrength 和 NLTK 在识别积极情绪方面的准确率分别只有 29.56%和 52.17%，在检测消极情绪方面的准确率甚至更低，分别为 13.18%和 23.45% (Tourani et al.， 2014;Jongeling et al.， 2015)。

这些情感分析工具是使用来自非技术社交网络媒体(例如 twitter 帖子、论坛帖子、电影评论)的数据开发和训练的，当在软件工程等技术领域操作时，它们的准确性大大降低，这主要是由于经常使用的技术术语的含义在领域特定的变化。尽管这种领域依赖关系被认为是文本内容中自动情感分析的一般困难，但我们需要更深入地了解这种领域依赖关系为什么以及如何影响工具的性能，以及我们如何减轻它们。事实上，软件工程界需要一个更准确的自动情感分析工具(Pletea et al.， 2014;Tourani et al.， 2014;Islam and Zibran, 2016b;Calefato and Lanubile, 2016;乔杜里和辛德尔，2016;Novielli et al.， 2015;Ortu等人，2016b;Sinha et al.， 2016)。在这方面，本文做出了三大贡献:

- 使用大型基准数据集，我们进行了深入的探索性研究，以揭示软件等技术领域文本内容自动情感分析的困难
- 工程。我们开发了一个专门针对软件工程文本的领域词典。据我们所知，这是第一个特定于领域的词典

我们针对 SentiStrength-SE、工程领域的软件提出了分析技术词典和实现。我们开发了一个原型工具，用于改进软件工程文本内容中的情感分析。该工具也在网上免费提供(sentistrength - se, 0000)。SentiStrength-SE 是第一个专门为软件工程文本设计的特定领域情感分析工具。

我们没有从头开始构建工具，而是在 SentiStrength(Thelwall 等人， 2012)的基础上开发了 SentiStrength- se，迄今为止，SentiStrength 是软件工程中使用最广泛的自动情感分析工具(Islam 和 Zibran, 2017b)。通过与原始的 SentiStrength (Thelwall et al.， 2012)、NLTK 和斯坦福 NLP 在软件工程领域的操作进行定量比较，我们发现我们的特定领域 SentiStrength- se 显著优于那些独立于领域的工具/工具包。我们还单独评估了我们的 SentiStrength-SE 在软件工程文本情感分析中的各个主要组件(即领域字典和启发式)的贡献。我们的评估表明，对于软件工程文本，特定领域的情感分析技术在准确检测情感方面表现得更好。我们进一步对我们的工具进行定性评估。基于探索性研究和定性评估，我们概述了在软件工程领域进一步改进自动化情感分析的计划。

本文是对我们最近工作的重要延伸(Islam and Zibran,2017b)。本文通过对软件工程文本中自动情感分析的困难进行更深入的分析，提出了新的证据和见解。更详细地描述了 SentiStrength-SE 开发中应用的技术。除了之前发表的比较外，还通过更深入的定性分析和与 NLTK 和斯坦福 NLP 的直接比较，大大扩展了对该工具的实证评估与原始的 SentiStrength 进行比较。我们对 SentiStrength-SE 的各个主要组件(即领域字典和启发式)进行了单独的评估。根据显著性统计检验对定量比较进行了验证。

提纲:论文的其余部分组织如下。第 2 节描述了一项定性实证研究，揭示了软件工程中自动化情感分析面临的挑战。在第 3 节中，我们介绍了我们通过解决已识别的困难而开发的原型工具 SentiStrength-SE。第 4 节，介绍了我们的工具的定量和定性评估。在第 4.9 节中，我们讨论了对经验评估有效性的威胁。在第 5 节中，我们讨论了进一步改进的范围和未来的研究方向。第 6 节讨论了相关工作。最后，第 7 节对本文进行总结。

2.情感分析难点的探索性研究

为了探索文本中自动情感检测的困难，我们围绕 Java 版本的 SentiStrength(Thelwall et al.， 2012)进行了定性分析。这个 Java 版本是 SentiStrength 的最新版本，而严格用于 Windows 平台的旧版本仍然可用。如前所述，SentiS- strength 是软件工程界最广泛采用的最先进的情感分析工具。选择这种特殊工具的原因在第 6 节中进一步证明。

英语词典认为“emotion”和“sentiment”是同义词，因此这两个词在实践中经常被使用。虽然这两者之间可能存在微妙的差异，但在描述这项工作时，我们认为它们是同义词。我们形式化地认为，除了主观性之外，人类的表达可以有两个可感知的维度:情感极性和情感强度。情感极性表示表达的积极性、消极性或中性，而情感强度则捕获情感/感性表达的强度，情感分析工具通常在数字情感分数中报告这种表达。

2.1.基准数据

在我们的工作中，我们使用了一个“金标准”数据集(Ortu et al.，2016b;Gold Standard Dataset Labeled with Manually Annotated Emotions, 0000)，该数据集由从 JIRA 问题跟踪系统中提取的 5992 条问题评论组成。整个数据集分为三组，分别命名为 Group-1、Group-2 和 Group-3，分别包含 392,1600 和 4000 条问题评论。5992 条问题评论中的每一条都由 n 个不同的人类评判员(Ortu et al.， 2016b)手动解释，并使用这些评论中发现的情感表达进行注释。对于 Group-1, n = 4，而对于 Group-2 和 Group-3, n = 3。这是软件工程领域唯一公开可用的此类数据集(Ortu et al.， 2016b;Islam and Zibran, 2017b)。

封闭布景?的情绪表达在数据集中问题评论的注释中使用，其中?={喜悦，爱，惊喜，愤怒，悲伤，恐惧}。人类评分员根据他们是否在评论中发现情感表达来标记每个问题评论。在形式上，

吗?𝑗,𝑟(?)=1，如果情绪?在?中被 rj 发现。
0,否则。

表 1 显示了一个来自数据集的问题评论的人工注释示例。

2.2.情感表达到情感两极

情绪表达喜悦和爱传达积极的情感极性，而愤怒、悲伤和恐惧表达消极极性。在某些情况下，惊讶的表达在极性上可以是积极的，表示为

表 1

四名人类评价员对问题评论的注释示例

发布评论(评论 ID-53257):感谢补丁;米歇尔。

稍加修改后应用。

人类	情绪(?)					
评级机构(右 j)	快乐	爱	惊喜	愤怒	悲伤	恐惧
Rater-1(右 1)	1	1	0	0	0	0
Rater-2(右 2)	0	0	0	0	0	0
Rater-3(右 3)	1	0	0	0	0	0
Rater-4(右 4)	1	0	0	0	0	0

解释:评分者 1 在评论中发现了“喜悦”和“爱”,而评分者 3 和评分者 4 只发现了“爱”的存在,而评分者 2 没有发现任何情感表达式。

惊喜+,而其他情况可以表达负面的惊喜,表示为 *惊喜*−。因此,基准数据集中标注了惊喜表达的问题评论,需要根据其传达的情感极性进一步区分。因此,我们让三个额外的人类(计算机科学研究生)评判员重新解释每条评论,他们独立地确定每条评论中惊讶表达的极性。

如果三个评级中的两个确定了其中的消极(或积极)极性,我们将评论中的惊讶表达视为消极(或积极)极化。我们在基准数据集中发现了 79 条问题评论,这些评论都用惊喜表达进行了注释。其中 20 条表达了正极性的惊讶,其余 59 条表达了负极性的惊讶。

那我们就分了?的情绪表达分为两个互不关联的集合,分别是?+ ={*喜悦*、*爱*、*惊喜*+}和?− ={*愤怒*、*悲伤*、*恐惧*、*惊喜*−}。因此,?+ 只包含积极的情感表达和?只包含消极的情感表达。其他研究也使用了类似的方法(Jongeling et al., 2015;2017),根据情绪表达的极性对其进行分类。

2.3.从人类评级数据集计算情感分数

对于“金标准”数据集中的每个问题评论,我们使用人类评分者评估的极性标签来计算情感极性。对于一个问题评论?我们使用 Eq. 1 和 Eq. 2 为 n 个评分者 rj(其中 1≤j≤n)中的每一个计算一对 ρcrj, ηcrj of 值:

$$\rho_c^{r_j} = \begin{cases} 1, & \text{if } \sum_{\mathcal{E}_i \in \mathcal{E}_+} \mathcal{F}_{\mathcal{E}_i}^{r_j}(C) > 0 \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

$$\eta_c^{r_j} = \begin{cases} 1, & \text{if } \sum_{\mathcal{E}_i \in \mathcal{E}_-} \mathcal{F}_{\mathcal{E}_i}^{r_j}(C) > 0 \\ 0, & \text{otherwise.} \end{cases} \quad (2$$

因此,如果评分者发现评论中存在任何积极的情感表达?,则 ρj crj = 1,*否则* ρcrj = 0。同样,*如果*评论中发现了*任何*消极的情感表达,则 ηcrj=1,否则 ηcrj=0。

issue 评论?在情感极性中被认为是中性的,如果我们得到至少 *n*− 1(即多数)评级者的 ρcrj, ηcrj对,其中 ρcrj = 0 和 ηcr= 0。如果评论不是中性的,那么我们确定该问题评论的积极和消极情感极性。要做到这一点,使用下面的公式,我们计算人类评分者的数量,?+(?)谁在评论中发现了积极的情绪?以及在评论中发现负面情绪的评分者的数量?− (?)?。

$$\mathcal{R}_+(C) = \sum_{j=1}^{n} \rho_c^{r_j} \quad \text{and} \quad \mathcal{R}_-(C) = \sum_{j=1}^{n} \eta_c^{r_j} \quad (3)$$

议题评论?如果至少有 *n*− 1 个人类评判员在消息中发现了积极的情绪,则被认为表现出积极的情绪。

类似地,如果至少有 *n*− 1 个评分者在其中发现了负面情绪,我们就认为一条评论具有负面情绪。最后,我们计算一个问题评论的情感极性?作为一对 ρch, ηch,使用 Eq. 4 和 Eq. 5。

$$\rho_c^h = \begin{cases} 0, & \text{if } C \text{is neutral} \\ +1, & \text{if } \mathcal{R}_+(C) \geq n-1 \\ -1, & \text{otherwise.} \end{cases} \quad (4)$$

$$\eta_c^h = \begin{cases} 0, & \text{if } C \text{is neutral} \\ +1, & \text{if } \mathcal{R}_-(C) \geq n-1 \\ -1, & \text{otherwise.} \end{cases} \quad (5$$

由此, ρch= 1,仅当评论?具有正面情绪,*且*仅当评论包含负面情绪时 ηch = 1。注意,一个给定的评论可以同时表现出正面和负面情绪。当评论的对 ρch, ηch *出现*为⟨ *0,0*⟩ 时,评论被认为是情感中立的。如果至少 *n* - 1 个人类评分者(即大多数)不能就评论的任何特定情感极性达成一致,则从我们的研究中丢弃一个问题评论。我们在 Group-2 数据集中发现了 33 条这样的评论,这些评论被排除在我们的研究之外。在另一项研究中,也采用了类似的方法来确定评论的情绪(Jongeling et al., 2015)。

2.3.1.计算情感极性的说明性例子

考虑表 1 中的问题评论。对于这个问题评论,我们为所有四个评价者(即 *n* = 4)计算对 ρcrj, ηcrj。对于四个评价者中只有一个(第二个评价者),我们得到对为⟨ 0,0⟩ ,评论不被认为是中性的。因此,我们计算?+(C)和?− *(C)*的值,它们分别是 3 和 0。?+*(C)*为 3 满足?+(C)≥n− *1* 的条件。因此, ρch = 1,即表 1 中的评论具有积极情绪。对于同一条评论?(C)<n− 1,故 ηch=− 1,表示该评论*没有*负面情绪。

2.4.使用 SentiStrength 进行情感检测

我们应用 SentiStrength 来确定在“黄金标准”数据集的第 1 组的问题评论中表达的情绪。使用 SentiStrength 对给定文本(例如, issue 评论)进行情感分析?计算一对整数⟨ ρc, ηc⟩ ,其中+ 1≤ρ ≤c +5 和- 5≤ ηc ≤- 1。在这里,cρ 和 cη 分别表示给定文本的正面和负面情感分数?。给定的文本?如果 cρ > +1,被认为有积极的情绪。类似地,当 η <− 1 时,c文本被认为包含负情感。此外,当文本的情感得分为 1, − 1 时,文本被认为是情感中立的。

因此,对于 cissue 评论的情感分数 c的⟨ ρ , η⟩ 对?由 SentiStrength 计算,我们计算另一对整数 ρct, ηct 如下:

$$\rho_c^t = \begin{cases} 1, & \text{if } \rho_c > +1. \\ 0, & \text{otherwise.} \end{cases} \quad \leftarrow 1。\frac{\eta_{ct}}{\text{和}\rho:2)} 1,\text{如果 } \eta_c \text{丨} 0,\text{否则。} (6)$$

这里, ρct = 1 表示 issue comment ?有正面情绪,且 ηct = 1 暗示 issue *评论*?有负面情绪。

我们应用 SentiStrength 计算“黄金标准”数据集的 Group-1 部分中每个问题评论的情感得分,然后对于每个问题评论?,我们计算对 ρct, ηct,它代表情感极性得分?。

2.5. 分析与发现

对于 392 条评论中的每一条?在 Group-1 中,我们比较了从 SentiStrength 产生的情感极性分数 ρct, ηct 和使用我们的方法计算的分数 ρch, ηch

2.3 节。我们一共找到 151 条评论，其中从 SentiStrength 得到的《τ，ηct 分数与 ρch，ηch 不匹配。这 》

这意味着对于这 151 条问题评论，SentiStrength 的情绪计算可能是不正确的。

在对 SentiStrength 的情感检测算法有了扎实的理解之后，我们仔细研究了所有这 151 条问题评论，以确定导致 SentiStrength 在文本内容中识别情感的原因/困难。我们确定了 12 个这样的困难。在讨论这些困难之前，我们首先简要描述一下 SentiStrength 内部工作机制的亮点，为读者提供必要的背景和背景。

2.5.1.洞察 SentiStrength 的内部算法

SentiStrength 是一个基于词典的分类器，它也使用额外的(非词汇的)语言信息和规则来检测用英语写的纯文本中的情感(Thelwall 等人，2012)。SentiS- strength 维护了一个包含多个单词和短语列表的字典，作为其关键字典，用于计算文本中的情感。在这些列表中，情感词列表、助推器词列表、短语列表和否定词列表在情感计算中起着至关重要的作用。除否定词表外，所有这些表中的条目都是预先用情感分数签名的。第四个表中的否定词用于当一个词位于文本中的否定词之后时，该词的情感极性反转。

对于一个输入句子，SentiStrength 从句子中提取单个单词，并在情感词列表中搜索每个单个单词，以检索相应的情感分数。在助推器词列表中进行类似的搜索，以增强或削弱感伤分数。短语列表用来区分一组词作为常用短语。当这样的短语被识别出来时，短语的情感得分会覆盖构成短语的单个单词的情感得分。表 2 中的示例阐明了 SentiStrength 如何依赖于列表字典来计算纯文本中的情感分数。

2.5.2.软件工程中自动情感分析的难点

表 3 给出了我们发现 SentiStrength 被人工调查中发现的 12 个困难所误导的次数。从表 3 中可以明显看出，单词的领域特定含义是导致 SentiStrength 词法方法准确性较低的所有困难中最普遍的。然而，并不是所有的困难都是特定于软件工程领域的，而是一些困难一般会影响情感分析(包括软件工程)，而一些困难实际上是 SentiStrength 工具的特定限制。表 3 中最右边的一列表明了所识别的困难的范围。我们现在用说明性的例子来描述 12 个难点。

(D1)单词的特定领域含义:在技术领域，文本工件包括许多技术术语，这些术语在字典含义方面具有两极，但在其技术上下文中并没有真正表达任何情感。例如，单词“Super”、“Support”、“Value”和“Resolve”是已知具有积极情绪的英语单词，而“Dead”、“Block”、“Default”和“Error”则是已知具有积极情绪的

表 3

误导情绪分析的困难频率

困难		频率(%)	范围*
D1:	单词的特定领域含义	123 (60.00)	当种子
D2:	的意义的上下文敏感变化单词	35 (17.07)	凹陷
D3:	对字母“X”的误读	12 (05.85)	当种子
D4:	复制粘贴内容中的感伤词语(例如:代码)	12 (05.85)	当种子
D5:	处理否定的困难	08 (03.90)	凹陷
D6:	字典里找不到感伤的词	02 (00.97)	凹陷
D7:	拼写错误会误导情感分析	02 (00.97)	凹陷
D8:	考虑重复的数字字符	01 (00.49)	风场
D9:	专有名词检测错误	01 (00.49)	风场
D10:	疑问句中的感伤词	01 (00.49)	风场
这里:	难以处理反讽和讽刺	01 (00.49)	凹陷
D12:	难以察觉微妙的情感表达	07 (03.41)	凹陷

*这里，SEDS = Software Engineering Domain Specific,SAG= Sentiment Analysis in General, SST = Specific to the SentiStrength Tool。

有负面情绪，但这些词在软件开发工件中都没有真正承载任何情绪。

由于 SentiStrength 最初是为用普通英语编写的非技术文本开发和训练的，它将这些单词识别为情感词汇，这在软件工程等技术领域的上下文中是不正确的。在以下来自“黄金标准”数据集的评论中，SentiStrength 将“错误”视为负面情感词，并将“支持”和“刷新”检测为积极情感词。因此，它为评论分配了积极和消极的情感分数，尽管评论是情感中立的。”WODEN-86 可能修复了这个问题，它在 http 位置模板中引入了**对花括号**语法的支持。这个 JIRA 现在可以关闭了。这个测试用例现在通过了…现在 **Woden 在**这个测试用例上报告了 12 个错误，在 r480113 中生成了结果。我会刷新 W3C 报告的。”(评论 **ID: 18059)**

(D2)单词含义的上下文敏感变化:除了单词的特定领域含义外，在自然语言中，一些单词根据使用的上下文具有多种含义。例如，“喜欢”这个词在“我喜欢你”这样的句子中使用时，表达的是积极的情绪。另一方面，在“乔治·华盛顿说过，我想成为一名水手”这句话中，这个词却没有表达任何情感。同样，SentiStrength 将“Please”这个词识别为积极的情感词，尽管我们发现这个词在训练数据集中被用作中性词来表达请求。例如，在下面的评论中，“请”这个词并没有表达任何情感。

， 1.2 分支更新。大卫;1.2 测试版在一周左右发布的时候**请**下载试用一下…(评论 ID: 4223)

再说一遍，那些被认为天生感性的词往往会这样

表 2

字典列表在 SentiStrength 计算文本中的情感分数中的作用

样本	发送。分数		字典	解释
句子			正在使用的清单	“好”这个词的感伤得分被预赋为 02;所以句子被赋正分 02。
这是一个很好的特写。	2	1	伤感的话	由于助推词“very”被用在感伤词之前，所以这句话被打了 03 分。
这是一个非常好的特点。	3.	1	辅助词	由于之前使用了否定词“not”，这里感伤词的情感极性被颠倒了

这不是一个好功能。	1	1	否定	
				情感词。
这是杀手锏。	2	1	短语	“killer feature”是词典中得分为 02 分的短语。虽然“kill”这个词带有贬义情绪，它的效果被这个短语的感伤分数所掩盖。

Not 携带情绪时用来表达可能性和不确定性。区分这类词的上下文敏感意义对于文本中的自动情感分析来说是一个很大的挑战，而 SentiStrength 的词法方法在这方面也存在不足。

例如，在下面的问题评论中，情感词“Nice”只是用来表达关于改变某事的可能性，但 SentiStrength 错误地计算了消息中的积极情绪。

“你想要的改变**会很好**;但根本不可能。表单数据……Jakarta FileUpload 图书馆。(评论 ID: 51837)

同样，在评论中，感情词“误用”用在条件句中，不表达任何情感，但 SentiStrength 另有解释。

“加了几个小点……**如果**有人发现文档格式有任何**错误**……(评论 ID: 2463)

(D3)对字母“X”的误解:在非正式的计算机中介聊天中，字母“X”经常被用来表示“Kiss”的一个动作，这是一种积极的情绪，因此被记录在 SentiStrength 的词典中。然而，在技术领域，字母“x”经常被用作通配符。例如，序列' 1.4.;下面注释中的 X 用来表示一个版本/发布的集合。

“集成在 Apache Wicket **1.4 中。x**……(评论 ID: 20748)

由于 SentiStrength 使用点(.)作为分隔符将文本分割成句子，因此“x”被认为是一个单字句子，并被错误地解释为表达了积极的情绪。

(D4)复制粘贴内容(例如，代码)中的煽情词:在提交时，开发人员经常在他们的问题注释中复制粘贴代码片段、堆栈跟踪、url 和驼峰大小写词(例如，变量名)。这种复制粘贴的内容通常包括变量名等形式的感伤词，这些感伤词并不传达提交者的任何情感，但 SentiStrength 检测到这些感伤词，并错误地将这些情感与问题注释和提交者联系在一起。考虑下面的问题注释，它包含了一个复制粘贴的堆栈跟踪。

“…Stack: [main] **FATAL**…表示。x-艾伦。模板.ElemTemplateElement。**resolve**PrefixTables…(评论 ID: 9485)

“Fatal”和“Resolve”这两个词(骆驼格单词“resolvePrefixTables”的一部分)，在 SentiStrength 的字典里分别是积极和消极的情感词汇。因此，SentiStrength 在问题评论中检测到积极和消极的情绪，但堆栈跟踪内容肯定不代表开发人员/提交者的情绪。

(D5)处理否定的困难:对于自动情感检测来说，识别在情感词之前是否存在任何否定词是至关重要的，因为否定词颠倒了情感词的极性。例如，“我心情不好”这句话就相当于“我心情不好”。当肯定词“好”的否定不能等同于否定词“坏”时，那么情感极性的检测就出错了。SentiStrength 的默认配置使其只有在否定词被直接放置在情感词之前时才能检测到情感词的否定。在所有其他情况下，SentiStrength 都无法正确检测到否定词，并且经常检测到与文本中表达的完全相反的情绪。在我们的调查过程中，我们发现了大量案例，其中 SentiStrength 被问题评论中存在的否定的复杂结构变化所误导。

例如，在以下两条评论中，SentiStrength 不能正确地检测到否定，它被用在单词“坏”(在第一条评论中)和“好”(在第二条评论中)之前，因此错误地分类了这些评论的情绪。

“我**没有看到**任何不好的行为。我正在使用 open ssh 进行测试。我使用....打开 ssh 断开连接;”(评论 ID: 6688)

“3.0.0 已发布;关闭……黄麻我没改——**别以为**这是个好主意;Esp 也会影响.....安德鲁，你能看看这个吗?(评论 ID: 1725)

此外，我们发现 SentiStrength 无法识别诸如“haven’t”、“havent”、“hasn’t”、“hasnt”、“shouldn’t”、“shouldn’t”和“not”等否定的缩写形式，因为这些术语没有被收录在字典中。

(D6)字典中缺少感伤词:由于 SentiStrength 的词汇方法在很大程度上依赖于它的单词列表字典(如第 2.5.1 节**所述**)，当文本中使用的感伤词在字典中不存在时，该工具经常无法检测到某些文本中的情感。例如，下面两个评论中的单词“Apology”和“Oops”表达了负面情绪，但由于这些单词不包括在它的字典中，因此 SentiStrength 无法检测到它们。

“…这确实不是问题。我很**抱歉**……(评论 ID: 20729)

“**哎呀**;发行评论里有错误的票号……”

(评论 ID: 36376)

(D7)拼写错误误导情绪分析:拼写错误的单词在非正式文本中很常见，作者经常故意拼错单词来表达强烈的情绪。例如，拼写错误的单词“Happy”比拼写正确的单词“Happy”更能表达快乐。虽然 SentiStrength 可以从这些拼写错误的感伤词中检测出一些这样强烈的情感，但它的能力仅限于那些故意拼写错误，即在感伤词中出现某些字母的重复。大多数其他类型的(无意的)情感单词拼写错误导致 SentiStrength 无法在字典中找到这些单词，从而导致对情感的错误计算。例如，在一个问题评论(评论 ID: 11978)中，“不幸的”一词被拼错为“不幸的”，而在另一个评论(评论 ID: 927)中，“我将”被写为“ill”。在这两个评论中，SentiStrength 的检测情绪都被发现是错误的。

(D8)重复的字符被认为是感性的:如前所述，SentiStrength 通过考虑故意拼写错误的带有重复字母的感性单词来检测更高强度的情感。为了达到同样的目的，该工具也采用了同样的策略，它考虑了重复的字符故意输入的单词，而这些单词本身并不一定是感性的。如果有人写“I am googoing to watch movie”而不是“I am going to watch movie”，那么前一句就被认为是积极的多愁善感，因为重复了三次字母“O”，强调了“going”这个词。

然而，这种策略在处理一些数值时也会误导 SentiStrength。例如，在下面的评论中，SentiStrength 错误地将数字“20001113”识别为遇到数字“0”和“1”重复的积极情感词。

“请参阅 bug 5694 查看…**20001113** /introduction.html…带有测试用例的 Zip 文件(java 源代码和 XML 文档)你使用延迟 DOM 吗?2. 你能试着在 Xerces2 beta4(或者 CVS 中的最新代码)上运行它吗?能否提供一个样例文件?谢谢你！(评论 ID: 6447)

(D9)专有名词检测错误:一个专有名词可以

被认为情绪中立是正确的。当一个单词位于句子的中间或结尾时，SentiStrength 会将以大写字母开头的单词检测为专有名词。不幸的是，语法规则在非正式文本中经常被忽略，因此，放在句子中间或结尾的感伤词通常以大写字母开头，这导致 SentiStrength 错误地忽略了这些感伤词中的情感。下面的问题评论就是这种情况的一个例子，以大写字母开头的感伤词“Sorry”被放在句子中间，而 SentiStrength 错误地认为“Sorry”是一个中性专有名词。

> “酷。谢谢你考虑我的 bug 报告!…关于 bug 的标题;在描述中;我写了**很抱歉**票名模糊。我不想对这个问题进行臆断……为密码工作。(评论 ID: 76385)

不过，Windows 老版本的 SentiStrength 并没有这个缺点。

(D10)疑问句中的感伤词:通常，疑问句(即疑问句)中的否定感伤词要么不表达任何情绪，要么至少削弱了情绪的强度(Thelwall et al.， 2012)。然而，我们已经发现了一些例子，在这些例子中，SentiStrength 无法正确解释这类疑问句的情感极性。例如，SentiStrength 在下面的评论中错误地识别了负面情绪，尽管评论中仅仅包含了一个问题，没有表达负面情绪，正如人类评分者所指出的那样。

> “…是我提交错了还是重复了?……”

(评论 ID: 24246)

(D11)处理反讽和讽刺的困难:在用自然语言写的文本中自动解释反讽是非常具有挑战性的，而且 SentiStrength 也经常无法从表达反讽和讽刺的文本中检测到情感(Thelwall et al.， 2012)。例如，由于存在积极的情感词汇“亲爱的上帝!，在下面的评论中，SentiStrength 检测到了句子中的积极情绪，尽管评论发布者以讽刺的方式使用它，只表达了消极情绪。

> “其他的先例还可以;据我所知……**“打鼾声”亲爱的上帝!**你的意思是这里的意图是……我得承认，我只是看到了这种模式，就妄下结论了;根本没有检查代码。但你只是让工作变得更难了……?”

(评论 ID: 61559)

(D12)难以察觉微妙的情感表达:用自然语言写的文本可以不使用任何语言表达情感固有的感伤词语。SentiStrength 的词法方法无法在这样的文本中识别情感，因为它高度依赖单词列表的字典，并且无法正确捕获句子结构和语义。考虑一下下面的问题评论，尽管没有情感词汇，但它被三位人类评分者贴上了负面情绪的标签。不出所料，SentiStrength 错误地将其解读为情感中立的文字。

> “布莱恩;我理解你所说的和 XSLT 中关于“序列化”而不是“缩进”的规范。就像我之前说的;缩进只是我们很容易看到 XML 文档的结构和数据。Xalan 的输出就不容易看到了。最后一次发射;我认为非空白字符的例子与缩进没有关系。非空白字符一定不能被剥离;但空白字符可以被剥离。问候;Tetsuya 吉田。(评论 ID: 10134)

3.利用自动情绪分析

我们解决了第 2.5.2 节所述的探索性研究中发现的挑战，并开发了一个专门为软件工程领域的应用而设计的工具。我们称我们的工具 SentiStrength-SE，它建立在原始 SentiS- strength 之上。我们现在描述了我们如何减轻在开发 SentiStrength-SE 时发现的困难，以改进软件工程中文本工件的情感分析。

3.1.为软件工程创建新的领域词典

如表 3 所述，我们的探索性研究(第 2.5.2 节)发现，特定领域的挑战(困难 D,D, D)是软件工程文本中情感分析的最大 1阻碍 63因素。因此，可以通过采用特定领域的词典来提高情感分析的准确性(Godboleet al.， 2007;Huand Liu, 2004;Qiu et al.， 2009)。因此，我们首先为软件工程文本创建一个领域词典，以解决领域困难的问题。

图 1 描述了为软件工程文本开发领域词典所采取的步骤/行动。我们收集了(Islam and Zibran, 2016b)工作中使用的大型数据集。该数据集由 49 万条提交消息组成，这些消息来自 GitHub 上的 50 个开源项目。使用斯坦福 NLP 工具(StanfordCoreNLP, last access: June 2018)，我们提取了提交消息中所有单词的一组词法化形式，记为 M_w。

为了从集合 M_w 中识别情感词，我们利用

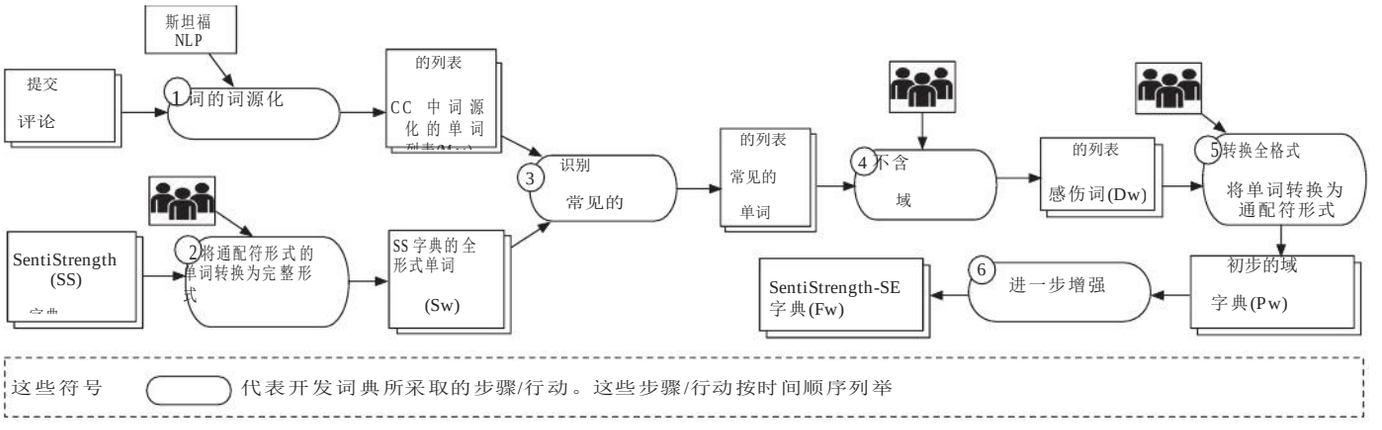

图 1所示。为 SentiStrength-SE 创建域字典的步骤

SentiStrength 现有的词典。我们选择 SentiStrength 的现有词典作为我们新词典的基础，因为在最近的一项研究(Islam and Zibran, 2017a)中，发现 SentiStrength 的词典构建方法优于其他方法(AFINN(Nielsen, 2011)、MPQA(Wilson 等人，2009)和 VADER(Hutto 和 Gilbert, 2014))，用于创建软件工程领域特定的词典。我们在 SentiStrength 的原始字典中识别那些具有通配形式(即以符号*作为后缀的单词)的单词，并使用 AFINN (Nielsen, 2011)字典将它们转换为相应的词素化形式。例如，在 SentiStrength 字典中，条目“Amaz*”被转换为单词“Amaze”、“Amazing”、“Amazes”和“Amazing”，因为这些词在 AFINN 字典中被发现是与该特定条目对应的情感词。使用 AFINN 词典主要有两个原因:(i)该词典与 SentiStrength 词典非常相似，因为两者都使用相同的数字尺度来表达单词的情感极性，(ii) AFINN 词典也被广泛用于许多其他研究(Riloff 等人，2013;Gan and Yu,2015;Koto and Adriani, 2015;Islam and Zibran, 2017a)。如果在 AFINN 字典中找不到任何通配符形式的单词，我们会用自己的智慧将该单词转换为其词素化形式。因此，通过将所有短形式的单词转换为完整形式，并将它们与 SentiS-强度字典中的剩余单词结合，我们得到一组单词 s。然后，我们区分出一组 ₙC= M∩s的单词，*该 ₙ集合 ₙ 包含 7ₙ 16个单词 ₙ*，它们代表 ₙ了与软件工程领域相关的初始情感词汇。

我们认识到，这 716 个单词中的一些只是软件工程领域特定的技术术语，在软件工程上下文中没有表达任何情感，否则在非技术领域(如社交网络)解释时，它们会表达情感。还有其他一些词，如“减少”、“消除”和“不足”，这些词不太可能在软件工程领域表达情感。因此，我们聘请了三位人类评判员(枚举为 A、B、C)来独立识别 C中的这些非情感领域词。这三位 ₙ人类评判员中的每一位都是具有至少三年软件开发经验的计算机科学研究生。如果一个词在他/她看来极不可能在软件工程领域解释时表达任何情感，那么一个人类评判员就会将这个词注释为中性的。

在表 4 中，我们给出了人类评分员不同意配对的情况下的情感明智百分比。我们还根据 Fleiss-ᴋ (Fleiss, 1971)值测量了评分者之间的一致程度。得到的 Fleiss-ᴋ 值 0.739 表明独立评分者之间存在实质性的一致。

当三个评分者中有两个认为这个词是中性的，我们就认为这个词是中性的领域词。因此，216 个词被识别为中性领域词，我们将其从集合 ₙ中排除，从而产生剩下 500 个词的另一个集合 ₙ。在一些研究中也建议对特定领域的单词进行这种中和(Pleteaet al., 2014;Tourani et al., 2014;Islam and Zibran, 2016a;2016b)中的文献。

接下来，我们通过将 d 中的单词恢复为通配 ₙ符形式(如果可用)来调整它们，以符合 SentiStrength 的原始字典。这组新的单词被称为初步领域词典(Pₙ)，它有 167 个正极性单词和 310 个负极性单词。这一初步词典根据

表 4

评分者对情绪解释的分歧

情感极性	人类评分员之间的分歧		
	A、B	B、C	C一个
积极的	11.81%	19.68%	17.32%
负	08.62%	10.19%	09.41%
中性	18.13%	11.81%	15.69%

描述下面创建 SentiStrength-SE 字典(Fₙ)。

3.1.1.对初步域字典的进一步增强

根据我们在第 2 节中描述的探索性研究中的观察，我们进一步扩展了新开发的初步词典。

扩展新的情感词和否定:在我们的探索性研究中，我们发现了几个非正式的情感词，如“Woops”，“Uhh”，“Oops”和“zzz”，这些词没有包括在原始词典中。正式词“sorry”也没有出现在词典中。我们在我们的 SentiStrength- SE 词典中添加了所有这些缺失的单词，作为具有适当情感极性的情感术语，从而减轻了难度 ₆D。

此外，我们还在字典中添加了 2.5.2 节中关于难度的讨论中提到的缺失的否定 ₅词的缩写。

从字典中丢弃字母“X”:我们从 SentiStrength-SE 的领域字典中排除了字母“X”，以避免词汇情感分析从第 2.5.2 节中描述的难度 Das₃ 中解脱出来。

3.2.在情感计算中包含启发式

虽然新领域词典的创建是软件工程文本中自动情感分析的重要一步，但我们意识到情感检测的计算也需要改进。因此，在实现我们特定于领域的 SentiStrength-SE 时，我们在计算中加入了许多启发式方法，我们将在下面描述。

3.2.1.增加语境感，尽量减少歧义

回想一下，在创建我们最初的领域词典时，我们根据三个独立的人类评判员的判断中和了 216 个单词。然而，单词的中和并不总是合适的。例如，在软件工程领域，“Fault”这个词通常表示程序错误，表达中立的情绪。然而，同样的单词也可以传达负面情绪，就像下面的评论中发现的那样。

> “As WING……**我的错**:假期过后不能生育……我可能会补充这一点;too”，“Fault”这个词表达了评论者的负面情绪。(评论 ID: 4694)

同样，“喜欢”这个词如果被用作“我喜欢”、“我们喜欢”、“他喜欢”和“他们喜欢”，则表达了积极的情绪。在大多数其他情况下，“Like”这个词被用作介词或从属连词，这个词可以安全地被认为是情感中立的。例如，下面的评论使用了“Like”这个词而没有表达任何情感。

> “在我看来**像是**用户问题……(评论 ID: 40844)

从上面的例子中我们可以观察到，216 个中和词中的一些实际上可以表达情感，当它们前面有指代一个人或一群人的代词时，例如，“我”，“我们”，“我的”，“他”，“她”，“你”和所有格代词，如“我的”和“你的”。在 SentiStrength-SE 中考虑了这些上下文信息，以适当地处理软件工程领域中这些词的上下文使用，以尽量减少困难 dᵢ 和 d。这些词的完整 ₂列表在 SentiStrength-SE 字典文件“ModifiedTermsLookupTable”中给出，这些词也经过三位评分者的审查。请注意，为了确定句子中单词的词性(POS)，我们应用了斯坦福 POS 标注器(Stanford CoreNLP，最后一次访问:2018 年 6 月)。

3.2.2.使中和剂生效

我们从探索性研究(如第 2 节所述)中观察到，如果一个词前面有任何中和词，如“Would”，

“可能”、“应该”和“可能”。例如，在“如果测试能尽快完成就好了”这句话中，积极的情感词“好”并没有表达任何被前面的词“会”所中和的情感。我们在 SentiStrength-SE 中添加了一种方法，使其能够正确检测句子中这种中和词的使用，从而在情感检测中更加准确。这有助于最小化前面描述的难度 2d。

3.2.3.一个预处理阶段的集成

为了尽量减少困难 D4, D, D7 和 D9 8(如第 2.5.2 节所述)，我们将预处理阶段作为其组成部分包含到 SentiStrength- SE 中。在计算任何给定的输入文本之前，SentiStrength-SE 应用这个预处理阶段来过滤掉数字字符和某些复制粘贴的内容，如代码片段、url 和堆栈跟踪。为了定位文本中的代码片段、URL 和堆栈跟踪，我们使用了类似于 Bettenburg 等人(2011)提出的方法的正则表达式。此外，还包括一个拼写检查器(Jazzy- The Java Open Source Spell Checker, 0000)来处理 Din 识别和纠正 7 拼写错误的英语单词的困难。在代码片段中标识符名称的近似识别方面，拼写检查也对我们基于正则表达式的方法进行了补充。

为了特别减轻难度，预处理 9 阶段还将注释的所有字母转换为小写字母。然而，将所有字母转换为小写也可能导致专有名词(如开发人员和系统的名称)的检测失败，这也很重要，正如 Din 第 9 2.5.2 节难度描述中所讨论的那样。从我们的探索性研究中，我们观察到，开发人员通常会在评论中紧接着一些称呼词(如“Dear”、“Hi”、“Hello”、“Hello”)或在字符“@”之后提到他们同事的名字。因此，除了将所有字母转换为小写字母外，预处理阶段还会丢弃这些单词，这些单词会立即放在任何这些称呼词或字符“@”之后。此外，SentiS- strength - se 保持了灵活性，允许用户指示工具将任何特定的单词视为情感中立，以防一个固有的情感词必须被识别为专有名词，例如，处理将情感词用作系统名称的情况。

3.2.4.参数配置，以便更好地处理否定

我们小心地为我们的 SentiStrength-SE 工具设置了一些默认的配置参数，如图 2 所示。SentiStrength- se 的这个默认配置与原来的 SentiStrength 有所不同。特别是，为了减轻 Din 处理否定的难度，在 SentiStrength-SE 中，图 2 中用黑色矩形标记的否定配置 5 参数被设置为 5，这使得工具能够在更大的接近范围内检测否定，允许否定和情感词之间的 0 到 5 个中间词，这在之前的研究中也被建议过(Hu 和 Liu, 2004)。

4.SentiStrength-SE 的实证评价

在做出设计和调优决策以开发 SentiStrength-SE 的同时，我们仍然小心翼翼地考虑在一个领域应用特定的启发式改进可能会导致另一个标准的性能下降的副作用。我们围绕七个研究问题，在几个阶段对特定领域的技术和特定领域的 SentiStrength-SE 的准确性进行了实证评估。

数据集:为了对我们的 SentiStrength-SE 进行实证评估，我们使用了第 2.1 节中介绍的“黄金标准”数据集的第 2 组和第 3 组中的 5,600 个问题评论。关于这些问题评论的情感极性的基本事实是根据第 2.3 节中描述的人类评分员的手动评估来确定的。

在进行评估之前，我们在表 5 中展示了 Group-2 和 Group-3 数据集的文本特征，这表明两个数据集的特征没有实质性差异。

指标:情感分析的准确性是根据三种情感极性(即积极、消极和中立)中的每一种计算的精度、召回率和 f 分来衡量的。给定一组文本内容 S，特定情感极性 e 的精度(p)、召回率(r)和 f 分数(ɪ)计算如下:

$$\wp = \frac{|S_e \cap S'_e|}{|S'_e|}, \qquad \Re = \frac{|S_e \cap S'_e|}{|S_e|}, \qquad \text{ɪ} = \frac{2 \times \wp \times \Re}{\wp + \Re}$$

其中?e 表示具有情感极性 e 的文本集，而? ’e 表示(通过工具)检测到具有情感极性 e 的文本集。

显著性统计检验:我们应用非参数 McNemar 检验(Dietterich,1998)来验证两种工具(例如?和?)获得的结果差异的统计显著性。a b 由于非参数检验不需要数据的正态分布，所以这个检验很适合我们的目的。我们对 2 × 2 列联表进行 McNemar 检验，该列联表由工具和工具获得的结果推导而来。a b 这种列联表的结构如表 6 所示。

令，Fb a and f 分别用?和?表示错误分类评论的集合。a b 在列联表(表 6)中，n 表示同时被?和?错 00 分类的问题评论的数量(即;a b ， $n_{00} = F_a \cap F$)，n 表示被?但未被?错分类的评论 01 数(即 b b ;a ， $n_{01} = F_b - F$)， n 表示被?而不是被?错分类的评论数(即;a 10 a b ， $n_{01} = F_a - F$)，n 表示被两个工具 b 正确分类的评论数量 11 。设，S 表示根据基本真理正确分类的所有问题评论的集合。因此，n= S− (F∪F).如果 n> n，则观察到工具?优于工具?。否则，如果 n> n，则认为?优于工具?。如果从 bMcNe- mar 检验中获得 a 的 p 值小于 b 预先指定的显著 11 性水平 α，则认为这种观察到的优势具有 a 统计显著性。10 01a 01 10 在我们的工作中，我们设置 α = 0.001，这是一个在文献中广泛使用的合理设置。

4.1.使用基准数据集进行正面比较

我们将我们的软件工程领域特定的 SentiStrength- se 与原始的 SentiStrength (Thelwall 等人，2012)工具以及另外两个工具包 NLTK (NLTK, 0000)和 StanfordNLP (Socher 等人，2013)进行比较。据我们所知，这些是过去唯一尝试用于软件工程文本情感分析的领域独立工具/工具包(Pletea et al.， 2014;Ortu et al.， 2015;Novielli 等，2015;Rahman et al.， 2015)。特别地，我们解决了以下研究问题:

RQ1:在软件工程文本中，我们的领域特定的 sentistrength- se 是否优于现有的用于情感分析的领域独立工具?

我们编写了一个 Python 脚本来导入 NLTK 情感分析包(Sinha, 2016;NLTK, 0000)并在文本上运行，以确定这些文本的情感极性。NLTK 决定文本的积极、消极和中立的概率。此外，它还提供了一个复合值 Cv，取值范围在-1 到+1 之间。如果 $C_v > 0$，则文本包含积极情绪，如果 $C_v < 0$，则文本将具有消极情绪。否则，当 $C_v = 0$ 时，文本被认为是情感中立的。另一项研究(Sinha, 2016)也遵循了类似的程序，使用 NLTK 来确定文本的情感。

我们使用斯坦福 NLP 工具的 JAR 开发了一个 Java 程序，将其应用于文本以确定其情感极性。对于文本，斯坦福 NLP 提供了 0 到 4 之间的情感得分 v S，其中 $0 \leqslant S_v \leqslant 1$ 表示消极情绪，$3 \leqslant S_v \leqslant 4$ 表示积极情绪，$S_v = 2$ 表示文本的中性情绪(Socher et al.， 2013a)。

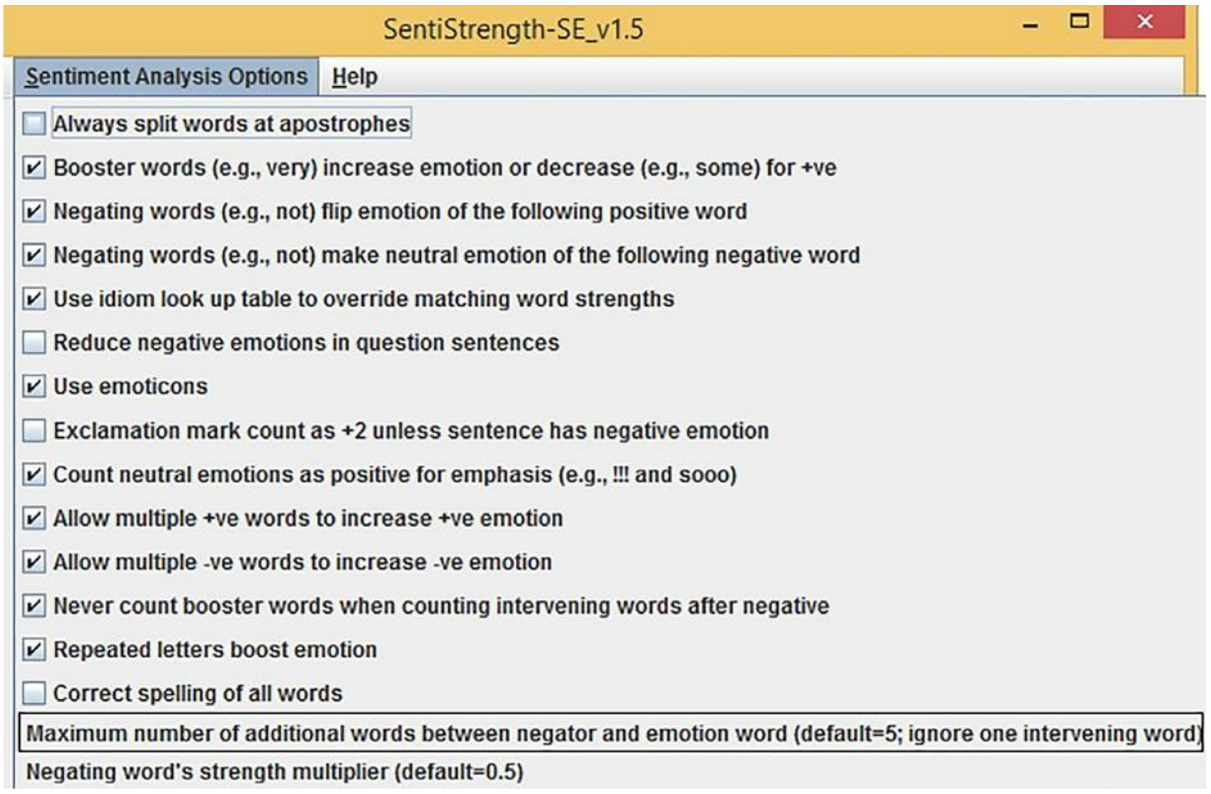

图 2所示。我们的 SentiStrength-SE 中参数的默认配置

表 5

数据集中单词的文本特征

数据集	不同单词的数量	复杂性因子(词汇密度)	平均数句子的句子长度(字)		每个评论的句子数
第二组	5295	21.00%	5671	8.23	3.54
第三组	5527	- 24.30%	4000	- 8.26	- 1.00

表 6 工具?a和?McNemar 检验的 2 × 2 权变矩阵结构

和错误分类的评论数量?a	n00	n01	有多少评论被?而不是ь被?
#条评论被?而不是 a ?	n10	n11	被?和?a 正确分类的评论的数量

我们分别在“黄金标准”数据集的 Group-2 和 Group-3 部分上操作每个选定的工具和我们的 SentiStrength-SE。回想一下，Group-2 和 Group-3 数据集分别包含 1600 条和 4000 条问题评论。对于三种情感极性(即积极、消极和中立)中的每一种，我们将工具的结果与基本事实进行比较，并分别计算所有工具相对于每个数据集的精度、召回率和 f 分。表 7 给出了所有工具在检测积极、消极和中性情绪方面的精度(p)、召回率(r)和 F-score(Ⅎ)，以及所有这三种情感极性的平均值。最高的度量值以粗体突出显示。

请注意，对于 Group-2 数据集，与其他工具相比，我们的 SentiStrength-SE 始终达到最高的精度、召回率和 f 分。

对于 Group-3 数据集，SentiStrength-SE 在检测负面情绪方面达到了最高的精度和 f 分，并且它达到了

表 7

四个工具/工具包性能的正面对比

数据	生梯-总共	满足。	生梯-Strength-SE	生梯-强度	NLTK	斯坦福大学 NLP
组 2	积极的	℘	**88.86%**	74.48%	69.47%	79.77%
		ℛ	**98.81%**	**98.81%**	81.55%	71.67%
		Ⅎ	**93.57%**	84.93%	75.0%	75.50%
	负	℘	**53.42%**	28.22%	40.46%	13.28%
		ℛ	**97.66%**	**97.66%**	54.69%	88.28%
		Ⅎ	**69.06%**	43.78%	46.51%	23.08%
	中性	℘	**98.14%**	96.83%	69.53%	63.70%
		ℛ	**83.00%**	52.42%	50.86%	25.57%
		Ⅎ	**89.94%**	68.01%	58.75%	36.49%
第三组	积极的	℘	41.80%	31.69%	20.32%	**69.47%**
		ℛ	82.04%	**87.79%**	86.33%	81.55%
		Ⅎ	55.39%	46.58%	32.89%	**75.03%**
	负	℘	**68.61%**	47.61%	50.65%	40.46%
		ℛ	71.00%	**78.40%**	70.24%	54.69%
		Ⅎ	**69.78%**	59.25%	58.86%	46.51%
	中性	℘	90.64%	**91.28%**	91.17%	69.53%
		ℛ	**80.05%**	56.16%	45.78%	50.86%
		Ⅎ	**85.02%**	69.54%	60.96%	58.74%
整体平均准确率		℘	**73.58%**	61.69%	56.93%	56.04%
		ℛ	**85.43%**	78.54%	64.91%	62.10%
		Ⅎ	**79.06%**	62.02%	55.50%	52.56%

在中性情绪检测中召回率和 f 值最高。在那些少数情况下，SentiStrength-SE 没有达到最好的结果，它仍然是第二好或稍微接近最好的。在 Group-3 数据集中积极情绪的检测中，satnlp 达到了最高的精度和召回率，其中我们的 SentiStrength- SE 产生了第二好的结果。同样，原始的 SentiStrength 在 Group-3 数据集的负面情绪检测中实现了最高的召回率，同样我们的 SentiStrength- se 获得了第二好的结果。中性情绪的最高准确率(原始 SentiStrength 达到 91.28%)仅为 0.64%

表 8

原始 SentiStrength 和 SentiStrength- se 对比 McNemar 测试的权变矩阵

被错误分类的评论#条和?$_a$	748	365	错误分类的评论数 By ?$_b$ 但不 By ?
#条评论被错误分类为?$_a$但不是 by?	**1527 年**	2924 年	正确评论#被?和?$_a$

这里，?$_a$ = original SentiStrength 和?$_b$ = SentiStrength- se

高于我们的 SentiStrength-SE。

因此，如果我们考虑整体平均精度，如表格底部所示，很明显，我们的 SentiStrength- SE 表现最好，其次是原始的 SentiStrength 和 NLTK。请注意，我们的 SentiStrength- se 的整体精度、召回率和 f 分都大大高于表现第二好的工具(即原始的 SentiStrength)。

为了验证观察到的 SentiStrength- se 与原始 SentiStrength 之间的性能差异是否具有统计学意义，我们对这两个工具的结果进行了 McNemar 测试。对于 Group-2 和 Group-3 这两个数据集，我们根据 $_{11}$ $_{10}$ $_{01}$表 6 中描述的规格计算 n、n、nandoon4。在表 8 中，我们给出了为 McNe- mar 测试计算的权变矩阵。我们观察到列联 $_b$ 表中?(SentiStrength-SE)的优越性为 n> n。根据从测试中获得的 p 值(p = 2.2 × 10^{-16}, p <$_{01}$ α)，观察到的 SentiStrength-SE 的优越性能差异具有统计学意义。基于这些观察结果和统计检验，我们现在推导出研究问题 RQ1 的答案如下:

对 RQ1 的回答:在软件工程文本的情感检测中，我们的领域特定的 SentiStrength- sesi 显著优于领域独立的 NLTK、斯坦福 NLP 和原始的 SentiStrength。

4.2.与人类评分者意见分歧的比较

回想一下，“黄金标准”数据集中的问题评论是用独立的人类评级员识别的情绪进行注释的。在一些问题评论中，人类评级员对情感的识别存在分歧。虽然人类对某些问题评论中的情绪存在分歧，但自动化工具很可能也会产生不同的结果，导致不同的精度和召回率。

因此，我们调查了人类评分者之间注释的一致和分歧在多大程度上导致了前一节中描述的工具正面比较结果的偏差。特别是，我们想要验证当考虑到评分者的协议和分歧时，我们的领域特定的 SentiStrength-SE 是否仍然优于其他领域独立的工具。在这里，我们解决了以下研究问题:

RQ2:当考虑到人类评分者之间的一致和分歧时，sentistrength - seli 的准确性与独立于领域的同行相比是否有很大差异?

在这项调查中，我们使用了 Group-2 数据集，其中每个问题评论都由三位人类评分员独立注释。我们从这个 Group-2 数据集中区分出两组问题评论。

(i) Set-A:包含所有三位人类评分者都同意这些评论中表达的情感的问题评论。此集包含 1,210 条问题评论。

(ii) Set-B:由三位评分者中有两位同意这些评论中表达的观点的问题评论组成。这一套包括 357 条问题评论。

我们制定了以下零假设和备选假设，以确定最佳工具改进性能的统计显著性。

零假设-1(H_o^1):在 Set-A 的问题评论中，SentiStrength-SE 在情感检测方面的性能与其他工具相比没有显著差异。

备选假设-1(H_a^1):在 Set-A 的问题评论中，SentiStrength-SE 在情感检测方面的表现与其他工具相比存在显著差异。

零假设-2(H_o^2):在 Set-B 的问题评论中，SentiStrength-SE 在情感检测方面的表现与其他工具相比没有显著差异。

备选假设-2(H_a^2):在 Set-B 的问题评论中，SentiStrength-SE 在情感检测方面的表现与其他工具相比存在显著差异。

我们现在检查这些假设是否适用于我们比较的四种工具。以与前一节中描述的正面比较类似的方式，我们分别运行所有四个工具，包括我们在 Set-A 和 Set-B 问题评论上的 SentiStrength-SE。对于三种情感极性(即积极、消极和中立)中的每一种，我们分别在 Set-A 和 Set-B 的问题评论上计算每种工具的精度(p)、召回率(r)和 f 分数(d)。

对于 Set-A 和 Set-B 中的问题评论，与其他工具相比，最好的工具必须表现出显着改进的性能。为了测试我们的假设，在每个 Set-A 和 Set-B 数据集中，我们首先确定在所有四个工具中产生更好结果的两个工具。然后，我们检查最佳工具和第二最佳工具的性能是否存在显着差异。如果表现最好的两种工具的准确性之间存在显着差异，那么发现这种差异就足以证明表现最好的工具与其他工具之间存在显着差异。

表 9 给出了所有工具在检测每组问题评论的积极、消极和中性情绪时的指标值。如表 9 所示，对于 Set-A 中的问题评论，SentiStrength-SE 在检测积极、消极和中性情绪方面始终达到最高的精度、召回率和 f 分。

表 9

Set-A 和 Set-B 问题评论的工具准确度比较

数据	生梯- 总共	满足。	生梯- Strength-SE	生梯- 强度	NLTK	斯坦福大学 NLP
设置一个	积极的	℘	**90.15%**	70.46%	67.9%	83.39%
		𝓡	**95.33%**	91.20%	61.58%	76.66%
		𝓯	**92.67%**	79.50%	64.17%	79.89%
	负	℘	**53.66%**	28.17%	49.45%	9.66%
		𝓡	**100.00%**	**100.00%**	54.55%	86.36%
		𝓯	**69.84%**	43.96%	51.87%	17.38%
	中性	℘	**99.44%**	99.43%	77.00%	67.61%
		𝓡	**91.15%**	58.84%	54.08%	28.40%
		𝓯	**95.12%**	73.93%	63.53%	40.00%
	总体平均精度	℘	**81.08%**	66.02%	64.78%	53.55%
		𝓡	**95.49%**	83.35%	56.74%	63.81%
		𝓯	**85.88%**	65.79%	59.86%	45.76%
b	积极的	℘	**72.48%**	63.64%	67.32%	64.91%
		𝓡	96.89%	**97.92%**	70.46%	57.13%
		𝓯	**82.92%**	77.14%	68.86%	60.98%
	负	℘	**51.75%**	21.63%	50.74%	20.83%
		𝓡	**96.72%**	60.65%	55.73%	90.16%
		𝓯	**67.42%**	31.89%	53.12%	33.85%
	中性	℘	83.92%	**86.21%**	38.24%	38.88%
		𝓡	**40.87%**	21.74%	33.91%	12.17%
		𝓯	**54.97%**	34.72%	35.94%	18.54%
	总体平均	℘	**69.38%**	57.16%	52.10%	41.54%

精 度		**68.44%**	58.59%	52.72%	37.79%
		78.16%	60.10%	53.37%	53.15%

表 10

McNemar 在 Set-A 数据集中对 SentiStrength- se 和原始 SentiStrength 精度的测试的权变矩阵

被错误分类的评论#条 Both ?$_a$and ?	**84**	22	#的评论被错误分类 ?$_b$但不是由?
#条被错误分类的评论 ?$_a$但不是 by?	**511**	593	#的评论正确被?和?$_a$

这里，?$_a$= original SentiStrength 和?$_b$ = SentiStrength- se

总体平均精度表明，原始 SentiS- strength 在 Set-A 数据集中达到了第二好的精度。

现在，我们对 Set-A 数据集中的问题评论在 SentiStrength- se 和原始 SentiStrength 的精度之间进行 McNemar 测试。测试的列联表如表 10 所示。根据列联表，当 $n_{10} > n_{01}$ 时，SentiS- strength - se(?$_b$)表现更好。发现性能差异具有统计学意义，$p = 2.2 \times 10^{-16}$,$p < \alpha$。因此，McNemar 检验拒绝了我们的第一个零假设(H01)。因此，第一个备选假设(H)成立。$_a$[1]

同样，如表 9 所示，对于 Set-B 中的问题评论，Sen- tiStrength-SE 在检测所有三个极性的情绪方面获得了最高的 f 分。除了只有两种情况外，SentiStrength- SE 的准确率和召回率在所有情况下也是最高的。SentiStrength-SE 对正面情绪的召回率为 96.89%，排名第二，只比最高低 1.03%。同样，我们的 SentiStrength-SE 在检测中性情绪方面的准确率为 83.92%，也接近最佳，只比最佳低 2.29%。此外，就整体平均准确率而言，SentiS- strength 可以被认为在 Set-B 数据集上取得了第二好的性能。

与 Set-A 数据集类似，对于 Set-B，我们在 SentiStrength- se 的精度与原始 SentiStrength 之间进行 McNemar 测试，以确定这两种工具的性能之间是否存在统计学上的显著差异。测试的权变矩阵如表 11 所示。根据权变矩阵，当 n> n时，SentiStrength$_b$- se(?)优于原始的 SentiStrength(?)。使用表 $_{01}$11 的权 $_a$变矩阵 $_{10}$ 的 McNemar 测试得到 $p = 2.2 \times 10^{-16}$，因此 $p < \alpha$。因此，我们的第二个*零假设*(H02)*被拒绝*，第二个备选假设(H)成立，这表明对于 Set-B 数据集中的问题评论，SentiStrength-SE 优于原始 *SentiS*$_a^2$- strength 的性能具有统计学意义。

因此，对于 Set-A 和 Set-B 数据集，SentiStrength- se 的性能都明显优于下一个最佳性能的 SentiStrength。根据我们对 Set-A 和 Set-B 数据集的观察和统计测试结果，我们现在得出研究问题 RQ2 的答案如下:

对 RQ2 的回答:当考虑到人类评分者之间的一致和分歧时，我们的领域特定的 SentiStrength- sstill 仍然保持着显著的优势(与其领域相比)

表 11

McNemar 在 Set-B 数据集中对 SentiStrength- se 和原始 SentiStrength 精度的测试的权变矩阵

评论#分类错误的和?$_a$	97	20.	#条被?but$_b$ 错误分类的评论不是?
评论#分类错误的 By ?$_a$但不是 By？	**135**	105	正确分类的评论数 Both ?$_a$and ?

这里，?$_a$= original SentiStrength 和?$_b$ = SentiStrength- se

独立对应物)在软件工程文本中检测情感的准确性。

4.3.评估领域词典的贡献

新开发的软件工程领域词典是 SentiStrength-SE 的主要组成部分。在这里，我们进行了定量评估，以验证领域词典在准确检测软件工程文本中的情感方面的贡献。特别是，我们解决了以下研究问题:

RQ3: SentiStrength 中的领域特定词典真的有助于改进软件工程文本中的情感分析吗?

对于这个特殊的评估，我们再次使用之前介绍的 Group-2 和 Group- 3 数据集。我们调用原始的 SentiStrength 来检测这些数据集中问题评论中的情绪。然后，我们操作 SentiStrength，使其使用我们新开发的领域字典，并在同一问题评论中调用它进行情感检测。我们使用 SentiStrength*来指代原始 SentiStrength 的变体，它被强制使用我们的领域字典而不是原始的。对于这三种情感极性中的每一种，我们分别计算和比较每个数据集中工具产生的精度、召回率和 f 分。

4.3.1.原始的 SentiStrength 和 SentiStrength 的对比*

如果我们的领域词典确实有助于改进软件工程文本中的情感分析，那么 SentiStrength*的表现肯定比原始的 SentiStrength 更好。在表 12 中，我们给出了在检测每种情感极性时获得的精度(p / s)、召回率(r/t)和 F-score(ɻ)。在表中，实质性(即超过 1%)的差异以粗体标记。

如表 12 所示，在每种情况下，SentiStrength*都比原来的 SentiStrength 获得更高的 f 分。此外，除了 Group-3 数据集中的中性评论外，Sen- tiStrength*在所有情况下都显示出更高的精度。对于 Group-3 数据集中的中性评论，原始 SentiStrength 的精度仅略高 0.06%。在所有跨数据集的情况下，SentiStrength*的精度、召回率和 f 分数都高于表 12 或与**表 12** 相当

原始的 SentiStrength 和 **SentiStrength***的性能比较

数据	情绪	满足。	SentiStrength	SentiStrength*
组 2	积极的	℘	74.48%	87.56%
		ℜ	98.81%	98.28%
		ɻ	84.93%	**92.61%**
	负	℘	28.22%	**53.19%**
		ℜ	97.66%	97.65%
		ɻ	43.78%	**68.87%**
	中性	℘	96.83%	**97.94%**
		ℜ	52.42%	**81.85%**
		ɻ	68.01%	**89.18%**
第三组	积极的	℘	31.69%	**40.44%**
		ℜ	**87.79%**	82.01%
		ɻ	46.58%	**54.16%**
	负	℘	47.61%	**69.10%**
		ℜ	**78.40%**	72.65%
		ɻ	59.25%	**70.83%**
	中性	℘	91.28%	91.22%
		ℜ	56.16%	**79.54%**
		ɻ	69.54%	**84.98%**
整体平均准确率		℘	61.69%	**73.24%**
		ℜ	78.54%	**85.33%**
		ɻ	62.02%	**76.77%**

这里，SentiStrength*被迫使用我们的域字典，而不是自己的域字典。

最初的 SentiStrength。只有在 18 个案例中的两个(即在 Group-3 数据集中对积极和消极情绪的召回)中，原始 SentiStrength 的表现被认为(基本上)优于 SentiStrength*。这些观察结果也反映在表底部三行中呈现的总体平均准确性中。总体平均精度表明，SentiStrength*的性能优于原始的 SentiStrength。因此，当它被迫使用我们的新领域字典而不是原来的字典时，观察到的 SentiStrength*的准确性要高得多。为了确定我们观察结果的统计显著性，我们在 SentiStrength*的结果和原始 SentiStrength 之间执行另一个 McNemar 测试。因此，我们将零假设和备选假设表述如下:

零假设-3(H03):原始 SentiStrength 和 SentiStrength*的准确率没有显著差异。

备选假设-3(H_a^3):原 SentiStrength 和 SentiStrength*的准确率之间存在显著差异。

McNemar 检验的权变矩阵如表 13 所示。从权变矩阵中可以看出，当 $n_{10} > n_{01}$ 时，SentiStrength*($?_b$)表现出更高的精度(与原始 SentiStrength 相比)。测试得到 $p = 2.2 \times 10^{-16} < \alpha$。因此，该检验拒绝了我们的原假设(H03)。因此，*备选*假设(H)成立，$_a{}^3$ *表明*差异在统计上显著。基于这些观察和统计检验，我们得出结论，我们新创建的领域词典确实有助于情感分析的统计显着改进。因此，我们对研究问题 RQ3 的回答如下:

对 RQ3 的回答:我们新创建的领域词典对软件工程领域情感分析的改进做出了统计上显著的贡献。

4.4.我们的领域字典 vs. SentiStrength 的优化字典

原始的 SentiStrength 有一个功能，可以方便地优化特定领域的字典(SentiStrength - se, 0000)。我们想要验证我们的领域字典与 SentiStrength 针对软件工程文本优化的字典相比表现如何。具体来说，我们解决了以下研究问题:

RQ4:SentiStrength 针对软件工程文本优化的词典是否比我们创建的 SentiStrength- se 的特定领域词典表现得更好?

4.4.1.优化 SentiStrength 的字典

SentiStrength 的原始字典可以通过输入一组带注释的文本片段来针对特定领域进行优化。为了优化 SentiStrength 的软件工程领域词典，我们使用了一个由与软件工程相关的 StackOverflow 帖子/评论组成的数据集。这个 Stack Overflowposts(SOP)数据集共包含 4,423 条评论(Calefato 等人，2017a;Novielli et al.，2018a)。SOP 数据集中的每条评论都被分配了适当的情感极性(即积极、消极、中性)，这取决于它表达的是哪一种。因此，35%的帖子被贴上了积极的标签情绪和 27%的帖子被标记为负面情绪，而 38%的帖子被标记为情绪中性(Calefato 等人，2017a)。

带有情感极性标签的简单注释不足以使 SentiStrength 能够使用数据集来优化其字典。为此，SentiStrength 需要一对整数情感分数(ϱ_c，μ_c)预先分配给每个评论?其中+ 1≤ϱ_c≤+5 和− 5≤μ_c≤− 1。这些分数的解释与 2.4 节中描述的类似。$\varrho_c$and μ_c分别代表预先分配给给定文本的积极和消极情感分数?。给定的文本?标记为具有具有积极情感的，必须赋予积极情感得分 $\varrho_c > +1$。得分越高 $_c\varrho$ indicates表示积极情绪的强度/强度越高。类似地，标记为具有负面情绪的文本必须分配一个负面情绪得分 $\mu_c <- 1$。较低 $_c$的 μ 表示文本中表达的负面情绪的强度/强度较高?。在情感上被标记为中性的文本，必须分配情感分数 1，− 1。

根据上面描述的要求，我们推导出 SOP 数据集中每个评论的情感分数。为了评论?有正面情绪，我们设 $\varrho = +3$。同样，对于表达负面情绪的评论，我们设置 $\mu = -3$。我们不使用 ϱ 和 μ 域中的极值，而是选择中位数的极值。在表 14 中，我们给出了一些示例，展示了我们如何为 SOP 数据集中标记的评论分配情感分数。然后将该数据集馈送到原始的 SentiStrength，以优化其用于软件工程文本的字典。因此，我们产生了原始 SentiStrength 的另一个变体。我们将这个带有优化字典的变体称为 SentiStrength^O。

4.4.2.SentiStrength^O 和 SentiStrength 的比较*

SentiStrength ^{O} 和 SentiStrength*只是各自的字典不同。SentiStrength^O 使用优化后的字典，而 SentiStrength*使用我们为 SentiStrength- se 创建的字典。因此，比较 SentiStrength^O 和 SentiStrength*意味着比较 SentiStrength 的优化字典和我们创建的 SentiStrength- se 的软件工程领域字典。

我们调用 sentistrength^O 来检测 Group-2 和 Group-3 数据集中的问题评论中的情绪，并计算检测每种情感极性的精度(p)、召回率(r)和 F-score(ⅆ)的值。我们在表 15 中并列展示了 SentiStrength^O 和 SentiStrength*的计算度量值。

在表 15 中，我们看到 SentiStrength*总是比 SentiStrength^O 获得更高的 F-score。此外，除了 Group-3 数据集中的中性评论外，SentiStrength*在所有情况下都实现了更高的精度。对于 Group-3 数据集中的中性评论，SentiStrength*的精度仅略低于 SentiStrength^O 的 0.49%。在 18 个案例中，有 16 个案例中，SentiStrength*的召回率也大幅高于或接近于 Senti^O Strength。最后，整体平均准确率，如表格底部三行所示，表明 SentiStrength*的整体精度、召回率和 f 分都明显高于 SentiStrength^O。

为了确定我们观察结果的统计显著性，我们在表 14 的结果之间进行了另一次 McNemar 检验

表 13

McNemar 在 SentiStrength 和 SentiStrength 之间测试的权变矩阵*

被错误分类的评论#	748	334	#的评论被错误分类
和?$_a$			By ?$_b$ 但不 By ?
#条评论被错误分类为	**1527 年**	2955 年	正确评论#
?$_a$ 但不是 by?			被?和?$_a$

这里，?$_a$=原始 SentiStrength 和?$_b$ = SentiStrength*

为标记的评论分配情绪分数的示例

评论文本	情绪标签由人类评分员	(ϱ_c,μ_c)
@DrabJay:很棒的建议!	积极的	+ 3, − 1
代码改变。: -)		
那真的很臭!我怕	负	+ 1, − 3
那		
有一些，但它们似乎都是专有的	中性	+ 1, − 1

表 15

SentiStrength 和 **SentiStrength**^{O} 性能比较*

数据	情绪	满足。	**SentiStrength**^{O}	SentiStrength*
组 2	积极的	℘	74.45%	**87.56%**
		ℜ	**98.68%**	98.28%
		𝔣	84.87%	**92.61%**
	负	℘	30.12%	**53.19%**
		ℜ	97.66%	97.65%
		𝔣	46.04%	**68.87%**
	中性	℘	96.69%	**97.94%**
		ℜ	54.29%	**81.85%**
		𝔣	69.53%	**89.18%**
第三组	积极的	℘	26.00%	**40.44%**
		ℜ	**86.90%**	82.01%
		𝔣	40.02%	**54.16%**
	负	℘	47.13%	**69.10%**
		ℜ	**76.77%**	72.65%
		𝔣	58.40%	**70.83%**
	中性	℘	91.71%	91.22%
		ℜ	58.02%	**79.54%**
		𝔣	71.07%	**84.98%**
整体平均准确率		℘	61.02%	**73.24%**
		ℜ	78.72%	**85.33%**
		𝔣	68.74%	**78.82%**

这里，注意，SentiStrength 使用 ^{O} 优化后的字典
SentiStrength* 使用我们的域字典

SentiStrength^{O} 和 SentiStrength*。因此，我们将零假设和备选假设表述如下:

零假设 -4(H04):SentiStrength^{O} 与 SentiStrength* 的准确率无显著差异。

备选假设 -4(H_a^4): SentiStrength^{O} 和 SentiStrength* 的准确率之间存在显著差异。

McNemar 测试的权变矩阵如表 16 所示。从权变矩阵中可以看出，当 $n_{10} > n_{01}$ 时，SentiStrength*(?$_b$)表现出更高的准确性(与 SentiStrength^{O} 相比)。该测试得到 $p = 2.2 \times 10^{-16}$，其中 $p < \alpha$，并拒绝我们的零假设(H04)。因此，*备选假设*(H)成立，表明差异在统计上显著。SentiStrength* 显著优越的准确性意味着我们为 SentiStrength-SE 创建的领域字典优于原始 SentiStrength 的优化字典。因此，我们对研究问题 RQ4 的回答如下:

答对 RQ4:我们为 SentiStrength-SE 创建的领域字典的性能明显优于原始 SentiStrength 的优化字典。

4.5.与大型领域独立词典的比较

如前所述，领域困难是发现领域独立情感分析技术在技术文本中操作时表现不佳的主要原因之一。我们的这项工作揭示了与第 2.5.2 节中描述的相同的情况。为了克服领域困难，我们在 SentiStrength-SE 中创建了特定于领域的字典和启发式方法。然而，与现有的领域独立词典相比，我们的

表 16

McNemar 在 SentiStrength 和 Senti^{O} Strength 之间测试的权变矩阵*

被错误分类的评论#	942	140	#条评论被错误分类为
?$_a$ 但不是 by?	**1046 年**	3436 年	正确评论# 被?和?$_a$

这里，?$_a$ = SentiStrength^{O} 和?$_b$ = SentiStrength*

Domain-specificdictionary 的大小很小，有 167 个正极性条目和 310 个负极性条目。有人可能会说，一个相当大的领域独立字典可能不会遭受我们所关心的领域困难，并且可能会表现得一样好，如果不是比我们相对较小的领域特定字典更好的话。为了验证这种可能性，我们将特定领域词典的性能与大型领域独立词典进行比较。特别地，我们解决了以下研究问题：

RQ5:一个大型的领域独立词典是否能比我们为 SentiStrength-SE 创建的领域特定词典表现得更好?

4.5.1.选择一个独立于领域的字典进行比较

一般来说，有几个领域独立词典(例如，AFINN(Nielsen, 2011)、MPQA(Wilson 等人， 2009)、VADER(Hutto 和 Gilbert, 2014)、SentiWordNet(Baccianella 等人，2010)、SentiWords(L. Gatti 和 Turchi, 2016)和 Warriner 等人的词典)可用于情感分析。Islam 和 Zibran (2017a)比较了 AFINN (Nielsen, 2011)、MPQA(Wilson et al.， 2009)和 VADER(Hutto and Gilbert, 2014)词典在软件工程文本情感分析中的表现。然而，Islam 和 Zibran (2017a)的工作中所有使用的词典都可以被认为覆盖率低。另一方面，与 AFINN、MPQA 和 VADER 词典相比，SentiWordNet (Baccianella et al.， 2010)、SentiWords (L. Gatti and Turchi, 2016)和 Warriner et al.(2013)的扩展版 new (Affective Norms for English Words)词典的规模更大，覆盖率更高。

在这三个高覆盖率的大词典中，我们选择了(Warriner et al.， 2013)的扩展版新词典，其中包括 13915 个英语词，报道覆盖率为 67%(L. Gatti and Turchi, 2016)。我们选择这本字典主要有两个原因:(i)这本字典已经在软件工程研究中使用过(Mäntylä et al.， 2017;Islam and Zibran, 2018b);(ii)使用词性(POS)作为上下文来确定单词的极性，发现在软件工程文本中检测情感的准确性较低(Islam and Zibran, 2017a)。因此，我们排除了 SentiWords 和 SentiWordNet，因为这两个字典使用词性作为上下文来确定单词的极性。

4.5.2.范围转换

在 Warriner et al.(2013)的扩展新词典中，每个单词 ω 被赋予一个价值 vω，这是一个介于+ 1.0 和+ 9.0 之间的实数，表示单词 ω 的情感极性和强度/强度。单词 ω 的情感极性，表示为 sentiment(ω)，根据下面的 Eq. 7 进行解释。

$$Sentiment(\omega) = \begin{cases} Positive, & \text{if } v_\omega > +5.0 \\ Negative, & \text{if } v_\omega < +5.0 \\ Neutral, & \text{otherwise.} \end{cases} \quad (7)$$

相比之下，原始的 SentiStrength 和我们的 SentiStrength- se 都使用整数范围[-5， +5]和不同的解释来达到相同的目的。为了在 SentiStrength 中使用这个扩展的新词典，我们将扩展的新词典中每个单词的价分数从[+1.0，+9.0]范围转换为[-5，+5]范围。在这样做的过程中，vω 的分数值首先被四舍五入到最接近的整数 v n。。然后，使用表 17 中的转换比例尺，我们将[。+1，+9]范围内的每个整数价分数 v′转换为[-5，+5]整数范围 。内的?。例如，如果一个单词的原始价分数四舍五入到最接近的整数是+2，则根据表 17 所示的映射将其转换为-4。范围之间的这种转换不会改变单词的原始价强度/强度(Islam and Zibran, 2018b)。最近的一项研究(Islam and Zibran, 2018b)采用了类似的方法来进行唤醒分数的范围转换。

表 17

效价分数从[+1，+9]转换为[-5，+5]

得分在[+1，+9]	+1	+2	+3	+4	+5	+6	+7	+8	+9
得分[-5，+5]	5	4	3	2	+/-1	+2	+3	+4	+5

表 18

SentiStrength 和 **SentiStrength**^{W} 性能比较*

数据	情绪	满足。	**SentiStrength**^{W}	SentiStrength*
组 2	积极的	℘	50.17%	87.56%
		ℜ	**99.60%**	98.28%
		⊣	66.73%	**92.61%**
	负	℘	16.52%	**53.19%**
		ℜ	85.94%	**97.65%**
		⊣	27.71%	**68.87%**
	中性	℘	91.11%	**97.94%**
		ℜ	05.86%	**81.85%**
		⊣	11.01%	**89.18%**
第三组	积极的	℘	10.40%	**40.44%**
		ℜ	**96.79%**	82.01%
		⊣	18.79%	**54.16%**
	负	℘	28.33%	**69.10%**
		ℜ	70.29%	**72.65%**
		⊣	40.38%	**70.83%**
	中性	℘	75.94%	**91.22%**
		ℜ	08.23%	**79.54%**
		⊣	14.84%	**84.98%**
整体平均准确率		℘	45.41%	**73.24%**
		ℜ	61.12%	**85.33%**
		⊣	52.10%	**78.82%**

这里，Note, sentistrengths^{W} 使用 Warriner et al.(2013)的扩展版新词典。

SentiStrength*使用我们的领域字典(为 SentiStrength 创建的)

SE)

4.5.3. SentiStrengthW 和 SentiStrength 的比较*

我们将原始 SentiStrength 的原始字典替换为基于 Warriner 等人的字典创建的字典，从而创建了另一个变体，并将这个新变体称为 SentiStrength^{W}。在 Group-2 和 Group-3 数据集中，调用 senti^{W} strength 来检测 issue 评论中的情绪。然后，我们计算了 SentiStrengthin 检测 ^{W} 每个情感极性的精度(p / s)、召回率(r/ t)和 f 分(⊣)。表 18 中并列列出了 SentiStrengthW 和 SentiStrength*的计算度量值。

如表 18 所示，在 18 种情况中，有 16 种情况下，与 SentiStrengthW 相比，SentiStrength*实现了更高的精度、召回率和 F-score。SentiStrength* 仅对正面情绪的召回率略低于 SentiStrength^{W}。请注意，对于同样的情况，SentiStrength^{W} 的精度比 SentiStrength*要低得多。在每种情况下，SentiS- strength *在检测情感和中性评论时都保持了精度和召回率之间的良好平衡。这种精确度和召回率之间的平衡导致在所有情况下，SentiStrength*的 f 分都更高。表中底部三行所示的总体准确性表明，与 SentiStrengthW 相比，SentiStrength*的精度、召回率和 F-分数明显更高。为了确定观察到的准确度差异的统计显著性，我们在 Sen- tiStrengthW 和 SentiStrength* 的结果之间进行了 McNemar 测试。对于统计检验，我们将零假设和备选假设表述如下：

原假设-5(H05):SentiStrengthw 和 SentiStrength*的准确率没有显著差异。

备选假设-5(H_a^5):SentiStrengthw 与 SentiStrength*的准确率存在显著差异。

McNemar 检验的权变矩阵如表 19 所示。在权变矩阵中可以看到，SentiStrength*(?$_b$)

表 19

McNemar 在 SentiStrength 和 SentiStrength 之间测试的权变矩阵*

被错误分类的评论#	1014 年	68	#的评论被错误分类
和?a			By ?b 但不 By ?
#条评论被错误分类为	**3270 年**	1212 年	正确评论的#
?a 但不是 by?			被?和?a

这里，?a = SentiStrength 和?b = SentiStrength*

当 $n_{10} > n_{01}$ 时，表现出更高的准确性(与 SentiStrength 相比)。测试得到 $p = 2.2 \times 10^{-16}$，其中 $p < \alpha$。因此，该检验拒绝了我们的零假设(H05)。因此，备选假设(H)成立，表明 a5Sen- tiStrength* 和 SentiStrengthW 的准确性差异具有统计学意义。因此，我们对研究问题 RQ5 的回答如下:

答对 RQ5:对于软件工程文本中的情感分析，我们为 SentiStrength-se 创建的领域特定词典的性能明显优于大型领域独立词典。

4.5.4.人工调查揭示原因

我们立即进行定性调查，以揭示为什么 Warriner 等人的具有更高覆盖率的大型字典比我们较小的特定领域字典表现更差。我们从 Group-2 *数据集中*识别了一组评论 CmW，这些评论被 SentiStrength 错误分类。从 CmW 集合中，我们区分了另一个子集 *CcS*，它们被 *SentiStrength*正确分类*。然后我们从集合 cc 中随机抽取 50 条评论进行人工调查。

从人工调查中，我们发现单词含义的领域特定变化(即 2.5.2 节 1 中揭示的难度)是 SentiS- strength 准确率低的主要原因。例如，下面的中性评论被 sentis - strength 识别为既有负面情绪，也有正面情绪。

> “…**崩溃**也是出于同样的原因。在**这里**做了一些局部修复。(评论 ID: 149494)

由于“Crash”和“Fix”这两个词的存在，上述评论被错误地分类为既有积极情绪又有消极情绪，这两个词在 Warriner 等人(2013)的领域独立的新词典中分别是消极和积极极化的词。在软件工程领域，这两个词在情绪上都是中性的。由于同样的原因，在中性评论的检测上，Warriner 等人的字典的性能甚至比原始 SentiStrength 的优化和默认字典还要差。

4.6.与备选域字典的比较

回想一下，我们的 SentiStrength-SE 的域字典是只用提交消息开发的。基于不同来源的文本构建的域字典有可能提供更好的性能。因此，为了验证这种可能性，我们使用来自不同来源的文本创建了第二个域字典，并将这个新的替代字典与 sentistrength - se 的字典进行比较。特别地，我们解决了以下研究问题:

RQ6:使用来自不同来源的文本开发的域字典是否能比仅基于提交消息开发的 SentiStrength-SE 的域字典表现得更好?

4.6.1.构建一个可替代的域字典

除了用于开发 SentiStrength-SE 词典的 49 万条提交消息外，我们还获得了 1600 条 Code Review Comments (CRC)(Ahmed et al., 2017)、1795 条 JIRA Issue Comments (JIC)(Islam and Zibran, 2018b)和 4423 条 Stackoverflow 帖子

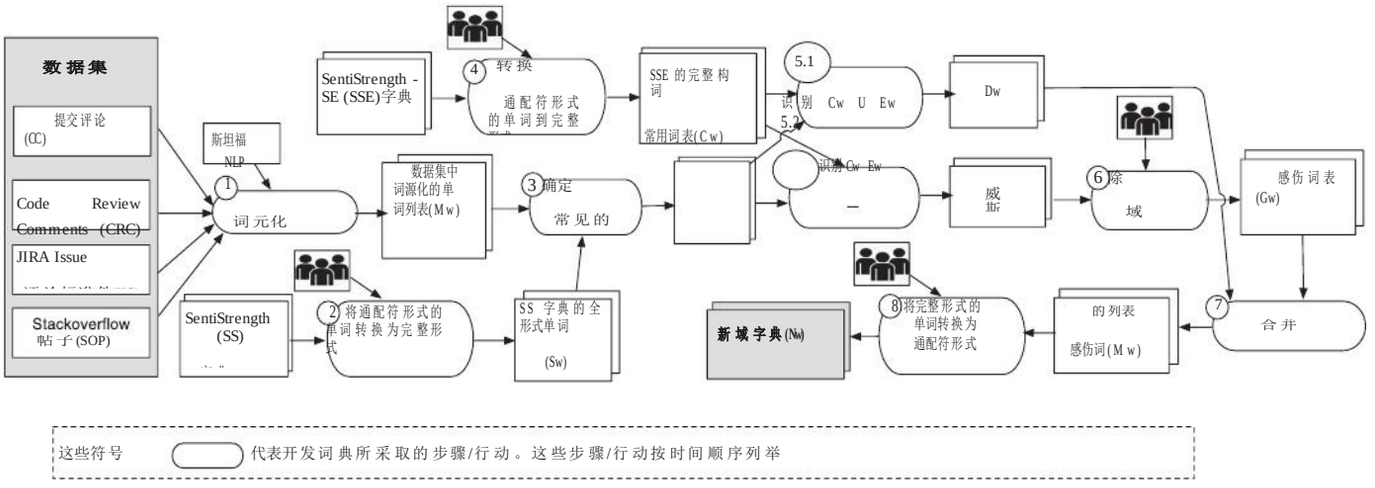

图 3所示。 开发新的替代领域词典的程序步骤

(SOP)(Calefato et al., 2017a)。我们使用来自这些不同数据集的软件工程文本来构建替代领域词典。

图 3 描述了开发这个新字典所执行的步骤/操作。在这里，在前三个步骤(即步骤 1到步骤 3)中，我们首先生成一个集合 S，其中包括来自所有四个数据集的所有不同的单词。然后，我们导出另一个集 c，其中包含 1198个单词 w，这些单词在 S和原始 SentiStrength 的领域独立字典中都是常见的。这三个步骤类似于开发我们 SentiStrength - se 的领域字典的前三个步骤(图 1)。然而，在步骤 1中，我们使用了四个数据集，而不是只提交消息。

在步骤 4中，我们将 SentiStrength-SE 字典中的通配符形成的单词转换为它们的完整形式。完整形式的词集记为 Ew。在步骤 5中，我们从 w集合 C中派生出一组词 D，使得 w D只包含 w C和 E之间共有的词。数学 w上，D= Cw∩E。www我们还推导出另一个集合 U，它包含在 C中但不在 E中的单词 w，数学上 Uwwwww= C- E。

d中的所有单词都可以被 w认为是感性的，因为这些单词也存在于 SentiStrength-SE 的字典中。我们需要在 u中识别那些在软件 w工程领域中是中立的，但在一般情况下可能是感性的词。因此，在步骤 6中，我们涉及三个人类评判员(枚举为 A, B 和 C)来独立地识别那些上下文中立的领域词。这些人类评分者与 SentiStrength-SE 领域词典开发中使用的三个评分者相同。

在表 20中，我们给出了人类评分者不同意配对的情况下的情感明智百分比。我们还根据 Fleiss-κ (Fleiss, 1971)值衡量了评分者之间的一致程度。得到的 Fleiss-κ 值为 0.691，表示独立评分者之间的一致性显著。当三个评分者中有两个认为这个词是中性的，我们就认为这个词是中性的领域词。因此，373 个单词被识别为中性领域词，我们将其从集合 u中排除，从而得到另 w一组情感词 g。然后，在步骤 7中，通过对集合 g和 d中的单词进行 w并集，我们形成了另一组单词 w M，其中 w包含了所有的情感 w词。最后，在步骤 8中，我们通过将 m中的单词恢复到它们的通配 w符形式(如果可用)，以符合 SentiStrength-SE 的字典。这个新的词集形成了我们的新领域词典(Nw)。在这个阶段，我们还确保在 SentiStrength-SE 的字典中手动添加的词(参见第 3.1.1 节)包含在新领域词典的词集中。这个备选词典包含 225 个积极和 495 个消极极化的情感条目。

表 20

评分者对情绪的解释存在分歧

情感极性	人类评分员之间的分歧		
	A、B	B、C	C一个
积极的	11.32%	12.18%	12.22%
负	12.25%	11.19%	10.21%
中性	12.15%	10.53%	13.42%

4.6.2.新字典与 SentiStrength- SE 的字典比较

我们通过用新创建的替代域字典替换 SentiStrength-SE 的域字典，创建了它的一个变体。我们称这个变体为 SentiStrength-SE^{N}。然后我们将 SentiStrength-SEN 与 SentiStrength-SE 的性能进行比较，这实际上意味着将 SentiStrength-SE 的字典与我们创建的新领域字典进行比较。

我们调用 sentistrength - se^{N} 来检测 Group-2 和 Group-3 数据集中问题评论中的情绪。然后，我们计算了 SentiStrength^{W} in 检测每个情感极性的精度(p)、召回率(r)和 f 分(ㄐ)。我们在表 21 中并列展示了由 SentiStrength-SEN 和 SentiStrength-SE 获得的计算度量值。如表 21所示，SentiStrength-SE 在检测负面情绪方面的表现略 ^{N}好于 SentiStrength-SE。另一方面，通过观察 precision 和 F- score 值，我们可以说 SentiStrength-SE 在检测积极和中性评论方面的表现并不好，尽管 SentiS- strength - se 在那些积极和中性评论中实现了略 ^{N}高的召回值。表 21 底部三行所示的总体准确率表明，SentiStrength-SEN 和 SentiStrength-SE 之间的表现没有实质性差异。为了验证观察到的差异的统计显著性，我们对 SentiStrength-SEN 和 SentiStrength-SE 的结果进行了另一次 McNemar 检验。对于统计检验，我们将零假设和备选假设表述如下:

零假设 -6(H06):SentiStrength-SEN 和 SentiStrength-SE 的准确率没有显著差异。

备选假设 -6(H_a^6):SentiStrength-SEN 和 SentiStrength-SE 的准确率存在显著差异。

McNemar 检验的权变矩阵如表 22 所示。从权变矩阵中可以看出，SentiStrength-SE^{N} ($?_a$)和 SentiStrength-SE ($?_b$)表现出几乎相等的精度。测试得到 p = 0.0040，其中 p > α 。因此，检验不能拒绝我们的原假设 *(H06)*。因此，我们得出结论，新创建的领域词典的性能与 SentiStrength-SE 的领域词典没有显著差异。我们现在将研究问题 RQ4 的答案表述如下:

表 21

SentiStrength-SE 和 **SentiStrength**^{N} -SE 的性能比较

数据	情绪	满足。	**SentiStrength-SE**^{N}	SentiStrength-SE
组 2	积极的	℘	87.82%	**88.86%**
		ℜ	98.81%	98.81%
		∮	92.99%	**93.57%**
	负	℘	**53.45%**	53.42%
		ℜ	**97.68%**	97.66%
		∮	**69.09%**	69.06%
	中性	℘	98.13%	**98.14%**
		ℜ	82.57%	**83.00%**
		∮	89.68%	**89.94%**
第三组	积极的	℘	39.16%	41.80%
		ℜ	**82.62%**	82.04%
		∮	53.13%	**55.39%**
	负	℘	**70.44%**	68.61%
		ℜ	**72.25%**	71.00%
		∮	**71.33%**	69.78%
	中性	℘	**90.71%**	90.64%
		ℜ	78.94%	**80.05%**
		∮	84.42%	**85.02%**
整体平均准确率		℘	73.28%	**73.57%**
		ℜ	**85.47%**	85.43%
		∮	78.91%	**79.06%**

这里，注意，sentistrength -se^{N} 使用新创建的域字典 SentiStrength-SE 使用自己的域字典

表 22

McNemar 在 SentiStrength-SE 和 SentiStrength-^{N} SE 之间测试的权变矩阵

被错误分类的评论#	1033 年	49	被错误分类的评论#
Both $?_a$ and ?			$?_b$ 但不是由?
#条被错误分类的评论	83	4399 年	正确注释的数量
$?_a$ 但不是 by?			被?和$?_a$

这里，$?_a$ = SentiStrength-SE^{N} and $?_b$ = SentiStrength-SE

答:to RQ6:新创建的域字典和 SentiStrength-SE 的域字典的性能没有统计学上的显著差异。

4.6.3.人工调查，确定原因

上述比较的结果似乎让我们感到惊讶，因为我们期望新创建的替代字典比 SentiStrength-SE 的表现更好。回想一下，新创建的字典比 SentiStrength- SE 的域字典更大。SentiStrength-SE 的词典有 167 个积极和 310 个消极极化的情感词，而新创建的词典则有 225 个积极和 495 个消极极化的词。虽然字典中的大量条目有助于实现高召回率，但它们也可能误导特定领域的情感分析，导致精度低。因此，我们分两个阶段手动调查这些可能性，并确定新创建的替代字典未能优于 SentiStrength-SE 的两个原因。

第一阶段调查:我们随机选择一组 5 个问题评论，其中 sentistrength -^{N} semi 对它们的情绪进行了分类，但 SentiStrength-SE 正确地进行了分类。其中一条评论如下:

“我不同意。(评论 ID: 1787887_1)

SentiStrength-SE^{N} 错误地识别了上述评论的负面情绪，因为“不同意”这个词在新创建的词典中被记录为一个负面极化的词。SentiStrength-SE 正确地将评论识别为中性，因为该词未包含在其字典中。我们为这五种评论识别出了相似的场景

随机抽取问题评论。

原因-1:新领域词典中出现的一些情感词汇，也出现在很多ground-truth数据集的中性评论中。这就是为什么sentistrength - se^{N}end将这些中立的评论错误地分类为情感评论的原因。这是高覆盖率词典的一个众所周知的问题(L. Gatti and Turchi, 2016)。

第二阶段调查:我们从新的领域词典中随机抽取 20 个固有的情感词，这些词不存在于 SentiStrength-SE 的词典中。然后，我们在 ground-truth 数据集中搜索这些词，并找到 5 个词(在所选的 20 个词中)(即“abhor”、“agony”、“骇人听闻”、“crime”和“delight”)不出现在数据集中的任何评论中。

原因 2:这意味着，尽管与 SentiStrength-SE 的词典相比，新的领域词典包含了更多的情感词，但由于这些新的情感词在使用的基础真相数据集中不存在，它们无法在情感分析中产生任何贡献。

4.7.评估启发式的贡献

除了领域字典，SentiStrength-SE 还包括一组启发式方法，以指导情感检测过程达到更高的准确性。这些专门为软件工程文本设计的启发式方法也是这项工作的主要贡献之一。在这里，我们进行了定量分析，以确定这些启发式方法在多大程度上有助于软件工程文本中的情感检测。特别地，我们解决了以下研究问题:

RQ7:集成在 sentistrength - seri中的启发式是否真的有助于改进软件工程文本中的情感分析?

我们比较了 SentiStrength- se 和 SentiStrength*的性能，以确定启发式的贡献。回想一下，SentiStrength* 指的是原始 SentiStrength 的变体，它被迫使用我们的初始域字典而不是原始的域字典。因此，SentiStrength- se 和 SentiStrength*使用相同的领域字典，它们之间唯一的区别是包含在 SentiStrength- se 中的启发式集合。因此，启发式对 SentiStrength- se 和 SentiStrength*的性能之间的任何差异负责。

我们在表 23 中给出了 SentiStrength- se和 SentiStrength*的性能。为了确定 SentiStrength-SE 中包含的启发式的效果，让我们比较表 23 中最右边的两列。我们观察到，在大多数情况下，我们的 SentiStrength- se 获得的精度、召回率和 f 分始终高于 SentiStrength*。在少数情况下，对于 Group-3 数据集，SentiStrength- se 的准确率几乎等于 SentiStrength*。整体平均准确率，如表 23 底部所示，也表明我们的 SentiS- strength - se 优于 SentiStrength*，这意味着在 SentiStrength- se 中纳入的启发式方法确实有助于提高软件工程文本中情感检测的准确性。

然而，如表 23 所示，虽然 SentiS- strength - se 的准确率比 SentiStrength*更高，但它们之间的差异并不大。因此，在这种特殊情况下，启发式的贡献似乎并不实质性，也不太可能具有统计显著性。为了验证我们的观察结果，我们在 SentiStrength*和 SentiStrength-SE 的结果之间进行了 McNemar 测试。对于测试，我们将零假设和备选假设表述如下:

原假设-5(H07):SentiStrength- se 和 SentiStrength*的准确率没有显著差异。

备选假设-5(H_a^7):SentiStrength- se 与 SentiStrength*的准确率存在显著差异。

表 23

在 SentiStrength-SE 中启发式的贡献

数据	情绪	满足。	SentiStrength-SE	SentiStrength*
组 2	积极的	℘	**88.86%**	87.56%
		ℵ	**98.81%**	98.28%
		⅁	**93.57%**	92.61%
	负	℘	**53.42%**	53.19%
		ℵ	**97.66%**	97.65%
		⅁	**69.06%**	68.87%
	中性	℘	**98.14%**	97.94%
		ℵ	**83.00%**	81.85%
		⅁	**89.94%**	89.18%
第三组	积极的	℘	**41.80%**	40.44%
		ℵ	**82.04%**	82.01%
		⅁	**55.39%**	54.16%
	负	℘	68.61%	**69.10%**
		ℵ	71.00%	**72.65%**
		⅁	69.78%	**70.83%**
	中性	℘	90.64%	**91.22%**
		ℵ	**80.05%**	79.54%
		⅁	**85.02%**	84.98%
整体平均准确率		℘	**73.58%**	73.24%
		ℵ	**85.43%**	85.33%
		⅁	**79.06%**	78.82%

*这里，SentiStrength*被迫使用我们的域字典，而不是自己的。

McNemar 测试的权变矩阵如表 24 所示。从权变矩阵中可以看出，SentiStrengthSE(?$_a$)和 SentiStrength*(?$_b$)表现出几乎相等的精度

$n_{10} \approxeq n_{01}$。检验得到 $p=0.874$，其中 p> α 。因此，检验不能拒绝我们的原假设*(H07)*。因此，我们得出结论，

在这种特殊情况下，工具中启发式的贡献并不显著。

根据我们从定量分析和统计检验中观察到的结果，我们现在将研究问题 RQ5 的答案表述如下:

对 RQ7的回答:尽管在 sentistrength - seconcs 中集成的启发式集有助于改进软件工程文本中的情感分析，但对于给定的数据集，感知到的改进在统计上并不显著。

记得,从探索性研究(第二部分)使用组- 1 部分的“黄金标准”的数据集,我们发现大部分的情感极性的误分类是由于限制(困难 D, D, D)$_1$ 使用的字典 ϖ(表 3)。因此,大多数要纠正错误分类通过使用一个域字典,留下一个相对狭窄的范围从启发式做出更大的贡献,至少在这个数据集“黄金标准”。我们对本研究中使用的数据集进行了人工调查，证实了在启发式的操作范围内存在很少的问题评论。

4.7.1.进一步的人工调查

虽然发现 SentiStrength-SE 在大多数情况下表现更好，如表 23 所示，但在 Group-3 数据集的四个情况下，SentiStrength-SE 的准确性略低于

表 24

McNemar 在 SentiStrength- se 和 SentiStrength 之间测试的权变矩阵*

被错误分类的评论#	993	78	被错误分类的评论#
Both ?$_a$ and ?			?$_b$ 但不是由?
#条被错误分类的评论	**81**	4433 年	正确评论的#
?$_a$ 但不是 by?			由?和?$_a$

这里，?$_a$ = SentiStrength- se 和?$_b$ = SentiStrength*

SentiStrength *。一个即时的定性调查揭示了这其中的两个原因，我们现在讨论一下。

首先，在某些情况下，我们用于识别情感词否定的参数设置在捕捉否定方面不足。例如，在下面的问题评论中，由于在 SentiStrength-SE 中否定配置参数设置为 5，否定词“Don’t”在“Know”之前中和了消极极化词“Hell”。“我不知道**我的 diff 程序**是怎么决定添加**看似**随机的 CR 字符的，但我现在已经把它们删除了”(评论 ID: 306519_2)

较低的否定参数可以更好地处理这个特定的问题评论，但对于其他的否定可能会表现得更差。其他可能的解决方案将在第 4.9 节中讨论。

其次，我们还发现了 Group-3 数据集中一些问题评论的负面情绪注解错误的实例，这导致 SentiStrength-SE 的准确性似乎下降了。考虑以下两个问题评论。

> “自动插入时间戳会很糟糕，因为它会限制一整条用例”(评论 ID: 1462480_2)

> “如果是这样的话，这将是糟糕的设计”(Comment ID: 748115_2)

在上述两期评论中，作者仅仅陈述了尚未发生的负面场景的可能性。这些评论并没有传达负面情绪。但是人类评判员注释了负面情绪，可能是因为考虑到在句子中使用了负面极化的单词“Bad”。

这些观察结果激励我们对 SentiStrength-SE 的成功案例，特别是失败案例进行更深入的定性调查，主要是为了探索进一步改进该工具的机会。因此，在下一节中提出了对 SentiStrength-SE 的定性评估。

4.8.SentiStrength-SE 的定性评价

虽然从对比评估中我们发现我们的 SentiStrength-SE 优于所有选定的工具，但 SentiStrength-SE 并不是一个万无一失的情感分析工具。事实上，100%的准确率不可能是一个务实的期望。尽管如此，我们对 SentiStrength-SE 进行了另一次定性评估，有两个目标:第一，确认在比较评估中发现的达到的准确性不是偶然发生的，第二，确定故障场景和进一步改进的范围。

我们首先从“黄金标准”数据集的第 2 组和第 3 组中随机抽取 150 条问题评论(50 条正面评论，50 条负面评论和 50 条情感中立评论)，其中 SentiStrength-SE 正确地检测了情感极性。从我们对这 150 条问题评论的手动验证中，我们确信 SentiStrength- SE 采用的设计决策、启发式和参数配置对情感极性的准确检测有积极的影响。

接下来，我们随机选择另外 150 个问题评论(50 个积极的，50 个消极的，50 个情感中立的)，其中 SentiStrength-SE 未能正确检测到情感极性。在对这 150 条问题评论进行人工调查后，我们发现了一些不准确的原因，其中一些在应用于 SentiStrength-SE 的设计决策范围内，其余的超出了范围，我们将在第 4.9 节中讨论。失败的原因之一是在我们新创建的领域词典中缺少情感术语。例如，SentiStrength-SE 错误地将以下评论识别为情感中性，将情感词“Stuck”误解为中性情感词，因为这个词不包括在字典中，我们将其添加到字典中

SentiStrength-SE 的释放。

“第一部分卡**在两点上**”(评论 ID: 1610758_3)

在其他一些情况下，我们发现 issue 评论中人类对情感的评分不一致，这是造成 SentiStrength-SE 不准确的原因。例如，下面的评论被人类评分者评为情感中性，尽管它包含了积极的情感术语‘谢谢’和感叹号‘!’。”**非常感谢**你，奥利弗，这么快就应用了这个**!**(评论 ID: 577184_1)

我们的调查显示，在 Group-3 中，有 200 条问题评论被人类评判员错误解读，这导致 SentiStrength-SE 在检测积极情绪方面的准确率很低，这与我们之前对这种错误解读情绪的发现是一致的。

尽管 SentiStrength- SE 的额外预处理阶段过滤掉了输入文本中不需要的内容，如源代码、URL、数值，但我们发现了几个例子，其中这些内容逃脱了过滤技术并误导了工具。

在少数情况下，我们发现我们识别专有名词的启发式方法由于没有考虑到可能的情况而不足。例如，SentiStrength-SE 在下面的问题评论中错误地计算了负面情绪。在下面的评论中可以看到，一位开发者感谢了他的同事“Harsh”。

> “谢谢 **Harsh**，这个补丁看起来不错……由于这是一个新的 API，我们不确定是否要更改它。我们暂时让它保持原样。(评论 ID: 899420)

由于没能识别出“Harsh”是一个专有名词，SentiStrength- SE 认为这个词在情感上是负面的，错误地检测到了消息中的负面情绪。我们近期的计划包括进一步扩展我们在文本中定位专有名词的启发式方法。

4.9.有效性的威胁

在本节中，我们将讨论对 SentiStrength-SE 实证评估有效性的威胁，以及我们为减轻这些威胁所做的努力。

4.9.1.建构效度和内部效度

构建效度的威胁与评价指标的适用性有关。我们使用三个指标：精度、召回率和 F-score 来评估 SentiStrength-SE 和其他工具的分类性能。这三个指标在软件工程研究中都被广泛用于类似的目的(Ahmed et al.， 2017;Blaz 和 Becker, 2016;Calefato et al.， 2017a)。仅定量分析可能无法描绘全貌，这就是我们对 SentiStrength-SE 进行定量和定性评估的原因。

度量计算的准确性取决于具有情感极性的问题评论的手动注释的正确性。因此，我们在“黄金标准”数据集中手动检查了问题评论的注释。我们确定了大约 200 条问题评论，这些评论被错误地标记为错误的情感极性。尽管如此，我们并没有排除那些错误分类的问题评论，因为它们同样影响所有工具，而不是偏袒另一个。

为了比较 SentiStrength- se 与其他工具(例如， SentiStrength, NLTK 和 stanford dnlp)的性能，我们使用了它们的默认设置。这些工具的不同设置可能会提供不同的结果，但由于它们在早期软件工程研究中的使用，我们坚持使用它们的默认设置(Guzman et al.，2014;Guzman and Bruegge, 2013;Pletea et al.， 2014;Tourani et al.，2014;Ortu 等人，2015;Calefato 和 Lanubile, 2016;乔杜里和辛德尔，2016;Garcia et al.， 2013;Jongeling et al.， 2015;Rahman et al.，2015;Rousinopoulos et al.， 2014)。

同时优化 SentiStrength 的默认字典

在软件工程领域，我们分别为正面、负面和中性的评论分配了三个常量+3、-3 和±1。有人可能会质疑选择这些特定值而不是域内其他值的原因。例如，我们可以用+2、+4 或+5 代替+3，用-2、-4 或-5 代替-3。回想一下，这些整数值不仅表示情绪的极性，还表示它们的强度/强度(sentistrength - se, 0000)。在精度、召回率和 f 分的计算中，只考虑情感极性。因此，在优化过程中，正面极化评论的+2 到+5 和负面极化评论的-2 到-5 之间的任何值都可以被使用。我们只是在中位数中选择值，而不是在域边界上选择极值。

我们将 Warriner et al.(2013)的新词典中单词的价分数范围从[+1，+9]更改为[-5，+5]，以比较其与 SentiS- strength - se 的领域词典的性能。有人可能会争辩说，范围转换可能改变了一些词的原始情感极性。我们已经考虑了这种可能性，并精心设计了转换方案，以尽量减少这种可能性。范围转换后的随机完整性检查表明不存在任何此类事件。

4.9.2.外部效度

只使用一个基准数据集(即“黄金标准”数据集)可以被认为是我们对 SentiStrength-SE 的经验评估的局限性。如果可以使用多个基准数据集，那么工作的结果可能会更具概括性。在 SentiStrength-SE 首次发布时，这个“金标准”数据集是唯一一个专门为软件工程领域精心制作的公开可用数据集(Ortu 等人，2016b;Islam and Zibran, 2017b)。有一些更新的数据集可用，但这些数据集要么不是特定于软件工程领域的，要么更具体于更狭窄的上下文(例如，代码审查，产品审查)。基准数据集中的问题评论是从开源系统收集的，因此人们可能会质疑，如果应用于来自工业/专有项目的数据集，包括我们的工具是否会有不同的表现。生成带有人工注释的大型数据集是一项乏味且耗时的任务。我们正在努力为软件工程文本中的情感分析创建第二个基准数据集。一旦完成，我们将向社区发布该数据集。

虽然在软件开发和维护的不同阶段产生的文本内容来源多种多样，但我们使用的基准数据集仅包含 JIRA 问题评论。因此，有人可能会认为，如果使用具有不同类型文本的数据集，则工具的经验比较结果可能会有很大差异。回想一下，我们的 SentiStrength-SE 工具的字典是基于提交注释创建的。因此，它在问题注释上的优越性能使我们相信该工具在其他类型的文本内容上也会表现良好。

4.9.3.可靠性

本文记录了本研究的方法，包括数据收集和分析的程序。“金标准”数据集(Ortu 等人，2016b)和所有工具(即 SentiStrength- SE (SentiStrength- SE, 0000)、SentiStrength (Thelwall 等人，2012)、NLTK (NLTK, 0000)和斯坦福 NLP(StanfordCoreNLP，最后一次访问:2018 年 6 月))都可以在网上免费获得。因此，应该有可能复制我们的工具的经验评估。

5.SentiStrength-SE 的局限性和未来的可能性

在本节中，我们讨论了 SentiStrength-SE 在设计和实现中的局限性，以及对我们的工具进行进一步改进的一些方向。在 SentiStrength-SE 的开发过程中，我们已经解决了所发现的困难

来自第 2 节中描述的探索性研究。正如我们从该工具的定性评估中发现的那样，仍有进一步改进的余地。例如，我们在第 4.8 节中观察到，缺少感伤词可能会误导 SentiStrength-SE。我们使用来自不同来源的文本为 SentiS- strength - se 创建了一个替代的新领域词典。令人惊讶的是，这个新创建的域字典并没有提供显着的性能改进。我们计划通过集成不同的词典构建方法来进一步扩展我们的领域词典(Blaz 和 Becker, 2016;Dragut et al.， 2010;Passaro et al.， 2015)。

我们采用的创建域字典的方法与其他尝试的方法相比是独特的(Blaz 和 Becker, 2016;Dragut et al.， 2010;Passaro et al.， 2015)。我们故意选择这种方法有两个原因。首先，我们想引入一种新的方法，其次，由于软件工程中情感注释文本等资源的限制，不可能采用现有的方法(Ortu 等人，2016b)。通过实证评估，我们已经证明，我们创建的领域词典对于软件工程中的情感分析是有效的。此外，在词典的开发过程中，由三位人工评价员对领域术语的识别可能会造成主观性偏差。然而，我们已经测量了评价者之间的一致性，并找到了合理的一致性，这极大地降低了威胁。

在创建我们的新领域词典时，我们使用研究生作为人类评级员，而不是来自行业的专家软件开发人员。然而，所有参与者都有至少三年的软件开发经验，这减轻了这种威胁。此外，据报道，研究生和专业软件开发人员的表现之间只存在很小的差异，特别是在涉及简单判断的小任务上(Host et al.， 2000)。

仅使用三名人类评判员可能会被认为是少数参与者。然而，在成功的软件工程研究中，两到三名评分员已经是常见的做法(Ahmed et al.， 2017;Blaz 和 Becker, 2016;Calefato 等人，2017a;Panichella et al.， 2015)。此外，通过实证评估(定量和定性)，我们已经表明，我们创建的领域词典对于软件工程文本中的情感分析是有效的。

已经提出并讨论了几种方法来确定情感分析中极化词的否定范围(Prollochs et al.， 2016;Asmi 和 Ishaya, 2012)，可以应用于提高我们的否定处理方法的性能。许多识别否定范围的复杂方法包括机器学习技术(Prollochs et al.， 2016;Morante et al.， 2008)、复杂规则(Jia et al.，2009)以及使用短语语义识别否定词(Choi and Cardie, 2008)。然而，许多现有的情感分析方法都有相对简单的方法来识别否定的范围(Panga et al.， 2002;Kennedy and Inkpen, 2006)。有趣的是，否定检测方法的性能可以可以通过域自适应来改进(Wu et al.， 2014)。在未来，我们将通过在软件工程环境中应用来评估所有提到的方法，以确定检测否定范围的最佳方法。

虽然我们过滤掉代码片段的方法可能无法正确定位所有代码部分，但过滤确实将它们最小化了。事实上，从纯文本内容中隔离内联源代码是一项具有挑战性的任务，特别是当文本可以用各种未声明的编程语言编写代码时。这样的代码分离问题可以作为一个单独的研究课题，过去曾进行过有限范围的尝试(Bacchelli et al.， 2011)。我们还计划沿着这个方向投入努力，进一步改进 SentiStrength-SE。

在这个阶段，我们没有解决困难 D_{10}、D、D_{1211}，这些都包含在我们未来的计划中。对文本中隐藏的反讽、讽刺和微妙情绪的检测确实是 NLP 中一个具有挑战性的研究课题，而且不仅仅与软件工程文本相关。即使是人类对文本中情感的解释也经常不一致，我们在“黄金标准”数据集中也发现了这一点。将基于字典的词法方法与机器学习(Reyes et al.， 2012)和其他专门技术(Balahur et al.， 2011)相结合，可以找到解决这些困难的潜在方法。我们还计划在 SentiS-strength - se 中增加正确识别疑问句的能力，以减轻难度 D_{10}。

6.相关工作

据我们所知，定性研究(第 2 节)是第一个分析公共基准数据集以揭示软件工程中情感分析面临的挑战的研究。而且，我们已经开发了第一个情感分析工具 SentiStrength-SE，专门为软件工程领域设计，我们希望它也能在其他技术领域产生卓越的性能。

除了我们的工具，只有四个突出的工具/工具包，即 SentiStrength (Thelwall 等人，2012)，Stanford NLP(StanfordCoreNLP，最后访问时间:2018 年 6 月)，NLTK(NLTK, 0000)和 Alchemy(AlchemyLanguage, 0000)，它们促进了纯文本中的自动情感分析。这些工具中的前三个已用于软件工程领域的情感分析，而 SentiStrength 在表 25 所示的研究中使用得最频繁。我们对这些研究进行分类，以便更好地理解这些工具的使用以及这些研究在软件工程领域的贡献。那些以前在软件工程领域使用但不用于情感分析的工具被排除在表格之外。值得注意的是，没有一项研究使用任何特定于领域的工具来检测情感。

Alchemy(AlchemyLanguage, 0000)是一个商业工具包，通过其发布的 api 将有限的情感分析作为服务提供。根据 Jongeling et al.(2017)的研究，Alchemy 的性能低于 SentiStrength(Thelwall et al.，2012)和 NLTK(NLTK, 0000)。NLTK 和 Stanford

表 25

情感分析工具的使用及其在软件工程中的贡献

工具	工作类型	用于软件工程研究
SentiStrength(Thelwall et al.， 2012)	的情感分析	Guzman et al. (2014);Tourani et al. (2014);Islam and Zibran (2016a,b);
	软件工程(SE)	Chowdhury and Hindle (2016);Novielli et al. (2015);古兹曼(2013);
		Ortu 等人(2016a)
	情感的应用	Guzman and Bruegge (2013);Ortu et al. (2015);Calefato and Lanubile (2016);
	SE	Garcia et al. (2013);Tourani and Adams (2016);古兹曼和马勒杰(2014);
		贾帕克迪等人(2016)
	基准测试研究	Jongeling et al.(2015、2017)
NLTK (NLTK, 0000)	SE 里的情绪分析	Pletea et al. (2014);Rousinopoulos et al. (2014)
	基准测试研究	Jongeling et al.(2015、2017)
Stanford NLP (StanfordCoreNLP，最后一次访问:6月	情感在	Rahman et al. (2015)
2018)	SE	
	基准测试研究	Jongeling et al.(2015、2017)

NLP(StanfordCoreNLP，最后一次访问:2018 年 6 月)是通用的自然语言处理(NLP)库/工具包，它期望用户具有一定的 NLP 背景，并编写脚本代码以纯文本形式进行情感分析。相比之下，SentiS-strength 是一个专用工具，它应用词法方法进行自动情感分析，并且可以在不需要编写任何脚本代码(用于自然语言处理)的情况下运行。也许，这些就是为什么在软件工程社区中，SentiStrength 比其他替代方案更受欢迎的原因之一。同样的原因也让我们选择了这个特殊的工具作为我们工作的基础。我们的 SentiStrength- se 重用了最初的 SentiStrength 的词法方法，也可以现成使用。

前面提到的所有四个工具(即 SentiStrength (Thelwall 等人，2012)、Stanford NLP(StanfordCoreNLP，最后一次访问:2018 年 6 月)、NLTK(NLTK, 0000)和 Alchemy(AlchemyLanguage, 0000))都是开发和训练用于从社交互动、网页中提取的非技术文本上操作的，当它们在软件工程等技术领域操作时，它们的表现不够好。特定领域(例如，软件工程)对固有情感词汇的技术使用严重误导了这些工具的情感分析(Pletea et al.， 2014;Tourani et al.， 2014;Jongeling et al.， 2015;Novielli et al.， 2015)，限制其在软件工程领域的适用性。我们通过开发我们的工具 SentiStrength-SE 中包含的第一个软件工程特定于主要领域的字典来解决这个问题。沿着这个方向 Mäntylä 等人(2016)开发了一个字典来捕捉软件工程文本中的情绪唤醒。

除了创建领域字典外，还探索了各种机器学习(ML)技术，如朴素贝叶斯分类器(NB)，支持向量机(SVM) (Panga 等人，2002)和逻辑回归(LR) (Choudhury 等人，2012)，以尽量减少领域难度。然而，当在特定领域的文本上操作时，这三种分类器的性能都较低(Muhammad et al.， 2013)。尽管如此，最近 Murgia 等人(2017)应用了几种 ML 技术(例如 NB、SVM)来识别情感爱、快乐和悲伤，而我们的工具 SentiStrength-SE 可以区分软件工程文本的积极性、消极性和中性性。同样，Panichella et al.(2015)使用 NB 分类器来检测软件用户评论中的情绪。然而，他们的分类器的准确性并没有被报道。这两个工具没有公开可用，无法与我们的工具 SentiS- strength - se 进行比较。此外，我们避免应用 ML 技术来实现 SentiStrength-SE，因为 ML 在情感分析方面的局限性，包括难以集成到分类器中，并且学习的模型通常在不同的文本类型或领域之间具有较差的适应性，因为它们通常依赖于在训练数据中发现的领域特定特征(Muhammad et al.， 2013)。

Blaz 和 Becker(2016)提出了三种几乎相同性能的方法，字典方法(DM)，模板方法(TM)和混合方法(HM)，用于 IT(信息技术)工作提交单中的“巴西葡萄牙语”文本的情感分析。DM 是一种纯词法方法，类似于我们的 SentiStrength-SE。尽管他们的技术可能适用于正式结构化的文本，但在处理软件工程工件(如提交注释)中经常使用的非正式文本时，这些技术可能表现不佳。相比之下，从对提交注释的经验评估中，我们的 SentiStrength-SE 在检测那些非正式软件工程文本中的情感方面具有很高的准确性。Blaz 和 Becker(2016)提出的方法是针对“巴西葡萄牙语”而不是英语编写的文本进行开发和评估的。因此，他们的方法和报告的结果不能与我们的直接比较。

与我们工作中包含的定性研究类似，Novielli 等人(2015)也对“社会程序员生态系统”中情感分析面临的挑战进行了相对简短的研究。他们还使用了 SentiStrength 来检测情绪极性，并报告了仅领域难度作为关键挑战。在他们的工作中，他们只手动研究了 100 个问题及其后续评论，以及从 Stack Exchange Data Dump(Stack Exchange Data Dump, 0000)中获得的 100 个答案及其后续讨论。相比之下，基于对公开可用的基准数据集的更深入分析，我们的研究暴露了包括领域依赖性在内的 12 个难点。此外，我们解决了这些困难的一部分，并开发了一个特定于领域的工具，用于改进软件工程文本中的情感分析。

我们的 SentiStrength-SE 是第一个软件工程领域特定的情感分析工具。在 SentiStrength-SE 发布后不久，过去几个月出现了四个特定领域的工具/工具包(即 Senti4SD (Calefato 等人，2017a)，SentiCR(Ahmed 等人，2017)，EmoTxt (Calefato 等人，2017b)和 SentiSW (Ding 等人，2018))。与我们的 SentiStrength- SE 类似，Senti4SD 也是一个软件工程领域特定的情感分析工具。Senti4SD 应用基于词典和基于关键字的特征的机器学习来检测情感。SentiSW 还应用机器学习技术在实体层面检测情感。EmoTxt (Calefato et al.，2017b)是一个开源工具包，用于从技术文本中检测六种基本情绪(即爱、喜悦、愤怒、悲伤、恐惧和惊讶)。SentiCR 的作者声明了这个工具的范围仅限于代码审查评论。同样，SentiSW 的适用性仅限于 JIRA 发行评论。EmoTxt 的报道范围是技术领域，比软件工程领域要宽。相反，SentiCR 和 SentiSW 的范围则分别局限于代码审查评论和 JIRA 发行评论的较窄领域。

进行了两项独立的研究来比较这些特定领域的情感分析工具的性能。在第一项研究中，Islam 和 Zibran (2018a)比较了 SentiS-strength - se、Senti4SD 和 EmoTxt 的性能，发现在软件工程中的情感分析工具中没有令人信服的赢家。在后来的研究中，Novielli 等人(2018b)比较了 SentiStrength-SE、Senti4SD 和 SentiCR 的性能，发现 SentiStrength-SE 的无监督方法提供了与监督技术(即 Senti4SD 和 SentiCR)相当的性能。在进行这两项比较研究时，SentiSW 还不可用。然而，SentiSW 的开发人员将其性能与 SentiStrength-SE 进行了比较，发现他们的工具在检测 JIRA 问题评论中表达的情绪方面优于 SentiStrength-SE。虽然所有这些研究都比较了领域特定工具的性能，但 Jongeling 等人(2015 年，2017 年)首次比较了领域独立工具 SentiStrength(Thelwall 等人，2012 年)、NLTK (NLTK, 0000)、Stanford NLP (StanfordCoreNLP，最后访问时间:2018 年 6 月)和 Alchemy (AlchemyLanguage,0000)的性能，以衡量它们在软件工程领域的适用性。

我们并没有声称 SentiStrength-SE 是软件工程文本中可用的情感分析工具中最好的工具。相反，通过开发和评估特定于领域的 SentiStrength-SE，我们证明，对于软件工程文本中的情感分析，特定于领域的技术比其独立于领域的同行表现得更好。如前所述，我们的 SentiStrength-SE 是软件工程中第一个用于情感分析的特定领域工具。其他研究可能从我们的工作(Islam 和 Zibran, 2017b)中获得了尝试特定领域解决方案的动机，从而产生了上面讨论的几个特定领域的工具。

7.结 论

在本文中，我们首先进行了深入的定性研究，以确定软件工程文本中自动情感分析的困难。在这些困难中，由领域依赖性引起的挑战是最主要的。主要解决以下问题:

领域难度，我们开发了一个专门为软件工程文本中的情感分析设计的领域特定词典。

我们还开发了一些启发式方法来解决其他一些已确定的困难。我们的新领域词典和启发式集成在 SentiStrength-SE 中，这是我们开发的一种工具，用于改进技术领域(特别是软件工程领域)文本内容的情感分析。我们的工具重用了 SentiStrength 的词法方法(Thelwall et al.， 2012)，这在软件工程中是最广泛采用的情感分析技术。我们的 SentiStrength-SE 是第一个专门为软件工程文本设计的特定领域的情感分析工具。

在包含 5,600 个问题评论的大型数据集(即 Group-2 和 Group-3)上，我们将我们的特定领域 SentiStrength- se 与三种最流行的领域独立工具/工具包(即 NLTK (NLTK,0000)，斯坦福 NLP (Socher 等人，2013b)和原始的 SentiStrength (Thelwall 等人，2012)进行了定量比较。实证比较表明，在检测软件工程文本内容中的情绪方面，我们的领域特定的 SentiStrength-SE 明显优于其领域独立的同类工具。

使用定量和定性评估，我们还分别验证了设计决策的有效性，包括我们在特定领域的 SentiStrength-SE 中包含的领域字典和启发式方法。从评估中，我们发现我们新创建的领域词典对软件工程文本中改进的情感分析做出了统计学上显著的贡献。然而，我们开发的启发式方法被发现对所选数据集的情感分析没有实质性影响，这些启发式方法可以最大限度地减少领域困难之外的问题。启发式的非实质性影响进一步验证了 SentiStrength-SE 准确性的提高归因于它是特定于领域的。因此，我们证明，对于软件工程文本中的情感分析，特定于领域的技术比独立于领域的技术表现得更好。

未来，我们计划通过扩展 SentiStrength-SE 并在工业/专有数据集上运行它来进一步验证这些发现。从我们的情感分析工具的探索性研究和定性评估中，我们还确定了该工具进一步改进的范围，这仍然在我们未来的研究计划中。使用 SentiStrength-SE 及其未来的版本，我们还计划使用软件工程领域的公共和专有数据集对情绪变化及其影响进行大规模研究。我们的 SentiStrength-SE 的当前版本是免费提供的(SentiStrength-SE, 0000)供公众使用。

参考文献

AlchemyLanguage:用于高级文本分析的自然语言处理。http://www。alchemyapi.com/products/alchemylanguage/sentiment-analysis。

Gold Standard Dataset Labeled with Manually Annotated Emotions。http://ansymore。uantwerpen.be /系统/文件/上传/工艺品/亚历山 德罗/MSR16 / archive3.zip。Jazzy-Java 开源拼写检查器。http://jazzy.sourceforge.net。

Stack Exchange Data Dump。https://archive.org/details/stackexchange。

Ahmed, T.， Bosu, A.， Iqbal, A.， Rahimi, S.， 2017。Senticr:用于代码审查交互的定制情感分析工具。第 32 届 IEEE/ACM 自动化软件工程国际会议论文集。106 - 111 页。

Asmi, A.， Ishaya, T.， 2012。情感分析中的否定识别与计算。第二届信息挖掘与管理进展国际会议论文集。1 - 7 页。

Bacchelli, A.， Cleve, A.， Lanza, M.， Mocci, A.， 2011。利用孤岛解析从自然语言文档中提取结构化数据。自动化软件工程国际会议论文集。476 - 479 页。

Baccianella, S.， Esuli, A.， Sebastiani, F.， 2010。Sentiwordnet 3.0:情感分析和意见挖掘的增强型词汇资源。语言资源与评价国际会议论文集。2200 - 2204 页。

Balahur, A.， Hermida, J.， Montoyo, A.， 2011。基于常识知识检测文本中隐含的情感表达。主观性和情感分析计算方法研讨会论文集。53-60 页。

Bettenburg, N.， Adams, B.， Hassan, A.， 2011。一种在非结构化数据中发现技术信息的轻量级方法。《程序理解国际会议录》。185 - 188 页。

Blaz, C.， Becker, K.， 2016。票中的情感分析为它提供支持。《挖掘软件库国际会议论文集》。235 - 246 页。

Calefato, F.， Lanubile, F.， 2016。情感信任作为分布式软件项目中成功协作的预测因子。《软件工程中的情感意识国际研讨会论文集》。3 - 5 页。

Calefato, F.， Lanubile, F.， Maiorano, F.， Novielli, N.， 2017。面向软件开发的情感极性检测。Softw Emp。Eng。352 - 1382。

Calefato, F.， Lanubile, F.， Novielli, N.， 2017。EmoTxt:从文本中进行情感识别的工具包。情感计算与智能交互论文集。

Choi, Y.， Cardie, C.， 2008。用组合语义作为结构推理的学习用于次句情感分析。自然语言处理中的经验方法会议论文集。793 - 801 页。

乔杜里，M.， Counts, S.， 2013。通过社交媒体了解工作场所的影响。计算机支持的协同工作。303 - 316 页。

Choudhury, M.， Gamon, M.， Counts, S.， 2012。开心、紧张还是惊讶?社交媒体中人类情感状态的分类。国际 AAAI 网络日志和社交媒体会议论文集。435 - 438 页。

Chowdhury, S.， Hindle, A.， 2016。表征能源意识软件项目:它们不同吗?挖掘软件存储库国际会议论文集。508 - 511 页。

Destefanis, G.， Ortu, M.， Counsell, S.， Marchesi, M.， Tonelli, R.， 2015。软件开发:礼貌重要吗?PeerJ 预印本 1-17。

Dewan, P.， 2015。走向基于情感的协同软件工程。软件工程的合作与人的方面国际研讨会论文集。109 - 112 页。

Dietterich, T.， 1998。比较监督分类学习算法的近似统计检验。j .神经。计算。10(7)，1895-1923。

丁杰，孙辉，王晓明，刘晓明，2018。议题评论的实体层面情感分析。第三届软件工程中的情感意识国际研讨会论文集。

Dragut, E.， Yu, C.， Sistla, P.， Meng, W.， 2010。情感词词典的构建。信息与知识管理国际会议论文集。1761 - 1764 页。

弗莱斯，J.， 1971。衡量许多评价者之间的名义尺度一致性。Psycholog.Bull。76(5)，378。

甘强，于勇，2015。餐厅评分:行业标准与口碑一文本挖掘与多维情感分析。《夏威夷国际系统科学会议论文集》。1332 - 1340 页。

Garcia, D.， Zanetti, M.， Schweitzer, F.， 2013。情绪在贡献者活动中的作用:gentoo 社区的案例研究。云计算与绿色计算国际会议论文集。410 - 417 页。

Godbole, N.， Srinivasaiah, M.， Skiena, S.， 2007。新闻和博客的大规模情感分析。第一届国际 AAAI 网络日志和社交媒体会议论文集。

Graziotin, D.， Wang, X.， Abrahamsson, P.， 2013。快乐的开发者生产率更高吗?软件开发人员的情感状态与他们自我评估的生产力之间的相关性。以产品为中心的软件过程改进国际会议论文集。50 - 64 页。

Guzman, E.， 2013。可视化软件开发项目中的情绪。《软件可视化会议论文集》。1 - 4 页。

Guzman, E.， Azócar, D.， Li, Y.， 2014。github 中提交评论的情感分析:一项实证研究。《挖掘软件库国际会议论文集》。352 - 355 页。

Guzman, E.， Bruegge, B.， 2013。面向软件开发团队中的情感意识。《软件工程基础国际研讨会论文集》。671 - 674 页。

Guzman, E.， Maalej, W.， 2014。用户是如何喜欢这个功能的?对应用评论进行细粒度的情感分析。国际需求工程会议论文集。153 - 162 页。

Host, M.， Regnell, B.， Wohlin, C.， 2000。以学生为研究对象:学生与专业人员在提前期影响评估中的比较研究。Softw Emp。工程 5(3)。

胡敏，刘斌，2004。挖掘和总结客户评论。知识发现与数据挖掘国际会议论文集。168 - 177 页。

Hutto, C.， Gilbert, E.， 2014。Vader:用于社交媒体文本情感分析的基于规则的简约模型。第八届国际 AAAI 博客与社交媒体会议论文集。216 - 225 页。

Islam, M.， Zibran, M.， 2016。软件工程中开发人员情感变化的探索与利用。实习生。j . Softw。创新学报，4(4)，35-55。

Islam, M.， Zibran, M.， 2016。走向理解和利用软件工程中开发人员的情感变化。《软件工程研究管理与应用国际会议论文集》。185 - 192 页。

Islam, M.， Zibran, M.， 2017。软件工程文本情感分析的词典构建方法比较。ACM/IEEE 实证软件工程与测量国际研讨会论文集。478 - 479 页。

Islam, M.， Zibran, M.， 2017。在软件工程中利用自动情感分析。挖掘软件存储库论文集。203 - 214 页。

Islam, M.， Zibran, M.， 2018。软件工程领域特定情感分析工具的比较。IEEE 软件分析、演化与再造国际会议。487 - 491 页。

Islam, M.， Zibran, M.， 2018。“deva:在软件工程的价性唤醒空间中感知情绪 tex.”第 33 届 ACM/SIGAPP 应用计算研讨会(SAC)论文集。1536 - 1543 页。

贾磊，于昌，孟伟，2009。否定对情感分析和检索有效性的影响。ACM 信息与知识管理会议论文集。1827 - 1830 页。

Jiarpakdee, J.， Ihara, A.， Matsumoto, K.， 2016。通过理解问题质量

情感方面的问答网站。软件工程中的情感意识国际研讨会论文集。12 - 17 页。

Jongeling, R., Datta, S., Serebrenik, A., 2015。选择你的武器:关于软件工程研究的情感分析工具。《软件维护与进化国际会议论文集》。531 - 535 页。

Jongeling, R., Sarkar, P., Datta, S., Serebrenik, A., 2017。关于使用情感分析工具进行软件工程研究时的负面结果。Softw Emp。Eng 1-42。

Kennedy, A., Inkpen, D., 2006。使用语境价移器的电影和产品评论的情感分类。第一版。英特尔。22(2)，110-125。

Koto, F., Adriani, M., 2015。推特情感分析的比较研究:哪些特征是好的?《自然语言在信息系统中的应用国际会议论文集》。453 - 457 页。

L. Gatti, m.g., Turchi, M., 2016。Sentiwords:衍生一个高精度、高覆盖率的情感分析词汇。IEEE 反式。影响。计算机学报，7(4)，409-421。

Lesiuk, T., 2005。音乐聆听对工作表现的影响。音乐心理学 33(2)，173-191。

Mäntylä, M., Adams, B., Destefanis, G., Graziotin, D., Ortu, M., 2016。挖掘效价、觉醒和支配——检测倦怠和生产力的可能性。《挖掘软件库国际会议论文集》。247 - 258 页。

Mäntylä, M., Novielli, N., Lanubile, F., Claes, M., Kuutila, M., 2017。Bootstrapping 一个用于软件工程中情绪唤醒的词汇。挖掘软件存储库国际会议论文集。1 - 5 页。

McDuff, D., Karlson, A., Kapoor, A., Roseway, A., Czerwinski, M., 2012。Affectaura:情感记忆的智能系统。计算系统中人因会议论文集。849 - 858 页。

Morante, R., Liekens, A., Daelemans, W., 2008。学习生物医学文本中的否定范围。自然语言处理中的经验方法会议论文集。715 - 724 页。

Muhammad, A., Wiratunga, N., Lothian, R., Glassey, R., 2013。面向情感分析的基于领域的词汇增强。SGAI 人工智能国际会议论文集。

Murgia, A., Ortu, M., Tourani, P., Adams, B., 2017。对开源系统问题报告评论中的情绪进行探索性定性和定量分析。Softw Emp。Eng。在美国。

Murgia, A., Tourani, P., Adams, B., Ortu, M., 2014。开发者会有情感吗?探索性分析软件工件中的情绪。挖掘软件存储库国际会议论文集。261 - 271 页。

Nielsen, F., 2011。A new new:微博情感分析的词表评价。ESWC 2011“微博的意义”研讨会论文集。93 - 98 页。

NLTK。情感分析的自然语言工具包。http://www.nltk.org/api/nltk。sentiment.html。

Novielli N..软件工程中用于检测情绪的工具列表。http://www。slideshare.net/nolli82/the-challenges-of-affect-detection-in-the-social-programmer-生态系统。

Novielli, N., Calefato, F., Lanubile, F., 2014。走向发现情绪在堆栈溢出中的作用。社会软件工程国际研讨会论文集。33-40 页。

Novielli, N., Calefato, F., Lanubile, F., 2015。社交程序员生态系统中情感检测的挑战。《社交软件工程国际研讨会论文集》。33-40 页。

Novielli, N., Calefato, F., Lanubile, F., 2018。堆栈溢出中情感注释的黄金标准。挖掘软件存储库国际会议论文集。第 14 - 17 页。

Novielli, N., Girardi, D., Lanubile, F., 2018。面向软件工程研究的情感分析基准研究。《挖掘软件库国际会议论文集》。799 - 808 页。

Ortu, M., Adams, B., Destefanis, G., Tourani, P., Marchesi, M., Tonelli, R., 2015。恃强凌弱的人效率更高吗?情感与问题解决时间的实证研究。《挖掘软件库国际会议论文集》。303 - 313 页。

Ortu, M., Destefanis, G., Counsell, S., Swift, S., Tonelli, R., Marchesi, M., 2016。纵火者还是消防员?敏捷软件开发中的情感。极限编程国际会议论文集。144 - 155 页。

Ortu, M., Murgia, A., Destefanis, G., Tourani, P., Tonelli, R., Marchesi, M., Adams, B., 2016。JIRA 中软件开发人员的情感面。《挖掘软件库国际会议论文集》。480 - 483 页。

Panga, B., Lee, L., Vaithyanathan, S., 2002。大拇指?使用机器学习技术的情感分类。自然语言处理中的经验方法会议论文集。79 - 86 页。

Panichella, S., Sorbo, A., Guzman, E., Visaggio, C., Canfora, G., Gall, H., 2015。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。IEEE 软件维护与演进国际会议论文集。281 - 290 页。

Passaro, L., Pollacci, L., Lenci, A., 2015。项目:一个引导意大利语情感词汇的向量空间模型。第二届意大利计算语言学会议论文集 CLiC-it。215 - 220 页。

Pletea, D., Vasilescu, B., Serebrenik, A., 2014。安全与情感:github 上安全讨论的情感分析。《挖掘软件库国际会议论文集》。348 - 351 页。

Prollochs, N., Feuerriegel, S., Neumann, D., 2016。检测金融的否定范围使用强化学习的新闻情感。《夏威夷国际系统科学会议录》。1164 - 1173 页。

邱刚，刘斌，卜杰，陈超，2009。通过双传播扩展领域情感词典。人工智能国际联合会议论文集。1199 - 1204 页。

Rahman, M., Roy, C., Keivanloo, I., 2015。使用众包知识为源代码推荐有见地的评论。源代码分析与操纵国际工作会议论文集。81 - 90 页。

Reyes, A., Rosso, P., Buscaldi, D., 2012。从幽默识别到反讽检测:社交媒体的具象语言。数据知识。工程 74,1-12。

Riloff, E., Qadir, A., Surve, P., Silva, L., Gilbert, N., Huang, R., 2013。讽刺作为积极情绪和消极情境之间的对比。《自然语言处理中的经验方法会议论文集》。704 - 714 页。

Rousinopoulos, A., Robles, G., Barahona, J., 2014。自由/开源开发者的情感分析:来自案例研究的初步发现。revsta Eletronica de Sistemas de Informacao 13(2)， 1-21。

SentiStregth-SE。情感分析工具，免费下载。http://laser.cs。uno.edu/Projects/Projects.html。

SentiStregth-SE。用于情感分析的自动领域独立工具。http:// sentistrength.wlv.ac.uk。

Sinha, V., 2016。大型软件库中 Java 源代码的情感分析。杨斯敦州立大学，美国。

Sinha, V., Lazar, A., Sahrif, B., 2016。在提交日志中分析开发者情绪。挖掘软件存储库国际会议论文集。520 - 523 页。

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C., Ng, A., Potts, C., 2013。情感树库上语义组合性的递归深度模型。自然语言处理中的经验方法会议。1631 - 1642 页。

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C., Ng, A., Potts, C., 2013。情感树库上语义组合性的递归深度模型。自然语言处理经验方法会议论文集。1631 - 1641 页。

StanfordCoreNLP，最后一次访问:2018 年 6 月。Stanford Core NLP Sentiment Annotator。http://stanfordnlp.github.io/CoreNLP/sentiment.html。

Thelwall, M., Buckley, K., Paltoglou, G., 2012。社交网络的情感强度检测。j。Soc。信息。科学。科技，63(1)，163-173。

图拉尼，P.，亚当斯，B.，2016。人类讨论对准时制质量保证的影响。《软件分析、进化和再工程国际会议论文集》。189 - 200 页。

Tourani, P., Jiang, Y., Adams, B., 2014。监控开源邮件列表中的情绪——对 apache 生态系统的探索性研究。协作型研究高级研究中心会议论文集。34-44 页。

Warriner, A., Kuperman, V., Brysbaert, M., 2013。13915 个英语引理的效价、唤醒和支配规范。Behav.Res.Meth。45(4)， 1191-1207。

Wilson, T., Wiebe, J., Hoffmann。, p., 2009。识别语境极性:短语级情感分析的特征探索。j.第一版。语言学家。35(3)，399-433。

Wrobel, M., 2013。软件开发过程中的情绪。人类系统交互国际会议论文集。518 - 523 页。

Wrobel, M., 2016。面向软件开发团队中情绪的参与性观察。计算机科学与信息系统联邦会议论文集。1545 - 1548 页。

Wu, S., Miller, T., Masanz, J., Coarr, M., Halgrim, S., Carrell, D., Clark, C., 2014。否定尚未解决:临床自然中的概括性与优化性。PLoS ONE 9 (11)， e112774。

Md Rakibul Islam 是美国路易斯安那州新奥尔良大学(UNO)计算机科学系的三年级博士生。他是 UNO 软件工程研究实验室(LaSER)的成员。他的研究重点是软件工程，特别是软件工程中的人的方面和安全。他喜欢将自己的研究领域与信息检索、数据挖掘和机器学习相结合，从数据中提取有趣的见解。他在包括 MSR, SANER, ESEM, SAC 等不同期刊和场所共同撰写了 10 多篇论文。在加入 LaSER 之前，他于 2008 年在孟加拉国库尔纳大学(Khulna University)获得理学士学位，随后在不同的软件和电信公司工作。在这些公司任职期间，他曾多次因其出色的服务而获得认可。

Minhaz F. Zibran，美国新奥尔良大学计算机科学系助理教授。他的研究兴趣包括软件工程的各个方面，特别侧重于应用源代码分析和操作来检测代码气味、程序错误和漏洞。Minhaz 的研究还包括软件工程的人类方面，包括情感分析及其对软件工程实践的影响。Minhaz 与人合著了许多学术文章，发表在 ACM 和 IEEE 赞助的国际会议和知名期刊上。Minhaz 既有教学经验，也有行业经验。他曾担任知名期刊(如 Springer EMSE, Elsevier JSS, IEEE Security & Privacy, Elsevier IST)的审稿人。他还积极参与了其研究领域的项目委员会和国际会议和研讨会的组织委员会(例如，ICDF2C ' 2018, ICPC ' 2018, SEMotion ' 2018, AffectRE ' 2018, IWSC ' 2018)。他是 IEEE 计算机学会和 ACM SIGSOFT 的专业会员。