利用软件工程文本中改进情感分析的独特表达

孙可欣新型软件国家重点实验室
科技南京大学
南京，中国
mf20320130@smail.nju.edu.cn

慧高新型软件国家重点实验室
科技南京大学
南京，中国
ghalexcs@gmail.com

邝宏宇*新型软件国家重点实验室
南京理工大学
南京
khy@nju.edu.cn

马晓星新型软件国家重点实验室
南京理工大学
南京
xxm@nju.edu.cn

荣国平南京大学新型软件技术国家重点实验室
南京
rongp@nju.edu.cn

东邵
南京大学新型软件技术国家重点实验室
dongshao@nju.edu.cn

他张
新型软件技术国家重点实验室
南京大学
南京,中国
hezhang@nju.edu.cn

摘要-对软件工程(SE)文本的情感分析在 SE 研究中得到了广泛的应用，例如评估应用程序评论或分析 commit 消息中的开发人员情绪。为了更好地支持在 SE 任务中使用自动情感分析，研究人员构建了一个 SE 领域指定的情感词典，以进一步提高结果的准确性。不幸的是，最近的工作报告称，在分析 SE 文本中的情感时，目前主流的情感分析工具仍然无法提供可靠的结果。我们认为，造成这种情况的原因是因为 SE 文本中的情感表达方式与社交网络或电影评论中的方式有很大不同。在本文中，我们提出通过使用句子结构来改进 SE 文本中的情感分析，这是一个与构建领域词典不同的视角。具体来说，我们首先使用句子结构来识别作者是否在 SE 文本的给定子句中表达了她的情感，并进一步调整在子句中确认的情感的计算。基于四个不同数据集的经验评估表明，我们的方法可以优于两种基于字典的基线方法，并且与基于学习的基线方法相比更具泛化性。

关键词:句子结构、情感分析、软件工程、自然语言处理

我的介绍。

情感分析是对手写文本的主观性和极性的研究(通常确定为积极，中性或消极)[1]。现代软件开发过程依赖于大量的人工努力和协作，因为软件的规模明显更大，软件开发已经变得更加迭代[2]。因此，软件开发的关键绩效指标，如其质量、生产力、创造力等，由于其人性的不可分割性[3]，将不可避免地受到其参与者情绪的影响。同时，当前软件开发中激烈的人类协作在很大程度上是由不同类型的在线工具支持的，例如论坛、社区、软件库和问题跟踪工具。然后，这些工具记录了大量关于软件工程(SE)领域中开发过程的手工编写的文本。这些 SE 文本为研究者发现开发者对项目的满意或困难，即他们的积极或消极情绪，提供了一个有价值的视角。因此，为了更好地支持软件工程(如[22])和程序理解(如[25])任务，越来越多的工作[19-28]对来自不同在线工具(如应用商店[34-35]、Stack Overflow[4,32,36]、GitHub[29-31]和 JIRA[21,22])的 SE 文本应用自动情感分析。这些分析在日常 SE 实践中也是有利的，因为与传统方法[5,6,42]不同，它们不需要对开发人员进行直接观察或交互，因此不太可能阻碍他们完成分配的开发任务。

在分析 SE 文本时，大多数讨论的工作使用现成的情感分析工具，这些工具建立在与 SE 领域无关的文本上，例如电影评论[7]，或者来自典型社交网络(如 Myspace[11])的帖子。为了提高 SE 领域情感分析的性能，研究人员进一步为 SE 文本定制自动化工具，要么训练特别收集和标记的 SE 文本[13,36]，要么构建 SE 指定的字典(例如，在 SE 文本中将“失败”和“例外”标记为中性)[12]。不幸的是，当分析 Stack Overflow 讨论上的情感以帮助向开发人员推荐代码库时，Lin 等人[36]发现，目前没有情感分析工具，甚至包括两个 SE 定制的工具(即基于领域词典的工具 sentistrong -SE[12]，以及基于作者来自 Stack Overflow 的标记数据集训练的基于学习的自适应工具)，可以提供 SE 文本中开发人员情感的可靠结果。报告的负面结果不仅警告了研究人员关于当前 SE 文本情感分析的局限性，而且还要求他们进一步发现开发人员如何通过在线协作工具在 SE 文本中表达他们的情感。

对此，我们进行了密切观察，发现 SE 文本中的情感表达方式相比于普通社交媒体文本(本文称之为社交文本)中的方式更加间接和分散。具体来说，我们首先观察到，由于软件任务的整体复杂性(如修复 bug 或理解代码和功能)，SE 文本的作者往往需要在表达情感之前或之后详细描述她遇到或提出的问题。因此，与其假设整个 SE 文本(with

*邝宏宇为通讯作者

978-1-6654-1403-6/21/$31.00©2021 ieee doi
10.1109/ icpc52881.2021.00023

2021 IEEE/ACM 第 29 届国际程序理解会议(ICPC) | 978-1-6654-1403-6/20/$31.00©2021 IEEE | DOI: 10.1109/ICPC52881.2021.00023

表我。 这些样本显示了sentistrength是如何根据字典和规则工作的，并给出了一个总体结果

手机文字输入的软件	发送。分数		整体	字典或	解释
	ρ	η	结果	使用规则	
这是一个很好的功能。	2	1	1	情感词	“好”这个词的感伤得分是02;所以这句话被赋正分02。
这是一个非常好的特征。	3.	1	1	助推器词，感伤词	由于在感伤词之前的助推词“very”有+1的效果，所以这句话被赋正分03。
这不是一个好的特征。	1	2	1	伤感词消极词	由于在感伤词之前使用否定词“not”，感伤词的极性被颠倒了。
这是一个很好的特征!	3.	1	1	感伤词“!”规则	”!”会加强感伤的力量。
这是一个很好的功能。	3.	1	1	感伤词字母重复规则	在正确拼写要求的字母上方出现两次以上的重复字母，用来增强1个单位的情感强度。

一个或多个句子)作为多愁善感的，se指定的情感分析需要忽略不太可能在所有句子中表达情感的从句。然后我们观察到，由于写作更复杂，句子结构变得非常有助于更好地理解SE文本中的情感，例如，忽略虚拟语气从句或区分多义词。

在此基础上，我们提出了一种基于词典的方法，该方法使用句子结构来改进SE文本的情感分析。我们基于最先进的基于字典的工具(即SentiStrength[11])而不是再训练来构建我们的方法，因为:(1)我们可以根据我们的观察自然地将我们的启发式集成到基于字典的工具中，并明确地测试它们的效果;(2)更重要的是，基于字典的方法往往在不同类型的SE文本上具有更好的泛化性，而不需要大量的标记数据进行训练，因此我们可以使用四个不同的数据集来更好地评估我们的观察结果和提出的方法。具体来说，我们的方法包括三个主要步骤:(1)对给定的SE文本进行预处理，并将其分割成子句;(2)根据我们提出的基于SE文本句子结构的过滤规则，忽略那些不太可能表达情感的子句;(3)在识别可能的情感子句上的情感时，我们的方法进一步使用提出的调整规则来增强基于词典的情感分析结果。我们使用从三个用于软件开发的在线协作工具收集的四个数据集的先前观察来评估我们的方法:堆栈溢出、应用程序评论和JIRA。评估表明，我们的方法可以大大优于两种基于字典的基线方法[7,12]，我们的过滤器调整规则对这两种基线具有很强的互补效应。这一结果还表明，作为我们提出的过滤器调整规则基础的观察结果是有效的，因为它们可以帮助SentiStrength(最先进的基于字典的情感分析工具)在不修改其情感词字典的情况下，在SE文本上获得更好的性能。评估还表明，与仅在一个数据集上训练的基于学习的基线方法[13]相比，我们的方法在所有四个数据集上都具有更好的泛化性。

本文旨在通过基于句子结构表征SE文本中独特的表达方式来改进软件工程的情感分析。我们将我们的方法命名为SESSION(基于句子结构的软件工程情感分析)。本文做出了两个贡献:(1)我们观察并发现了SE文本中情感表达的独特性;(2)我们提高了基于词典的准确率

基于我们通过使用SE文本的句子结构从前词观察中得出的启发式方法，对SE文本进行情感分析。我们的工具是公开的[43]。

本文的其余部分结构如下。第二节介绍了基于词典的情感分析的背景，以及我们对SE文本中情感表达的观察。第三节介绍了我们的方法。第四节介绍了实验和研究问题。第五节根据实验结果回答研究问题。第六节讨论可能的威胁。第七节讨论相关工作。第八节作结论，并对今后的工作提出建议。

2se文本中情感表达的背景与观察

在本节中，我们首先介绍SESSION的基础SentiStrength[11]。然后我们讨论SE文本和社交文本在表达情感时的区别。A. SentiStrength是如何工作的

SentiStrength是一个基于字典的情感分类器，它是为普通文本开发的。它包含了一系列情感词典，包括情感词列表、助推器词列表和负面词列表。这些列表在情绪的计算中起着至关重要的作用。情感词汇表为匹配的词汇给出情感分数。助推器单词列表包含可以加强或削弱受影响的情绪分数的单词。负面词汇表中的单词用来颠倒后面一个单词的情感极性。对于输入的文本，SentiStrength会根据字典为每个单词分配情感分数，并使用次要规则来调整结果。我们使用表1中的样本来展示SentiStrength是如何根据字典和规则工作的。变量ρ和η分别指每个句子的正分数和负分数，其中+1≤ρ≤+5和-5≤η≤-1。为了更好地检测情绪，SentiStrength的默认结果同时包含了这两个分数。只有分数(1，-1)表示文本的中立性。然而，它也提供了一个“三元”选项来输出一个整体情绪，要么是积极的，要么是中立的，要么是消极的。值得一提的是，SentiStrength在不考虑输入文本中的子句数量的情况下，根据最高的积极和消极情绪分配的情感词来确定情绪得分。这种设置有助于SentiStrength专注于输入文本中最具情感色彩的部分，尤其是当文本大小较大时。我们在我们的方法中遵循相同的设置，但使用从输入文本中分割出来的子句作为我们提出的过滤器调整规则的基础。

B. SE文本与Social文本的不同表达

仔细观察 SE 文本和社会文本，我们发现两种类型的文本在表达情感方面存在明显差异。我们选择的社交文本样本是来自 SentiStrength 基准[11]的 1041 条 MySpace 评论。我们选择的 SE 文本样本是来自 Senti4SD 基准[13]的 4423 篇 Stack Overflow 帖子。接下来，我们将详细介绍我们观察到的差异。

我们首先通过比较两组样本中情感文本的百分比，发现 SE 文本倾向于表达较少的情感。在 1041 条 MySpace 评论中，有 938 条文本被手动标记为情感(积极或消极)。感伤文本的比例为 90.1%。在 4423 条 Stack Overflow 帖子中，有 2729 条文本被手动标记为感性。感性文本的比例为 61.7%。除了情感较少之外，在表达情感方面，SE 文本更加间接和分散。我们用情感密度来体现 SE 文本的这一特点。一个文本的情感密度□等于文本中情感词的数量(根据 SentiStrength 的情感词列表)□除以文本中的总单词数□。938 条 MySpace 情感文本的平均□为 0.148，而 2729 条 Stack Overflow 情感文本的平均□为 0.092。为了更直观地描述差异，我们展示了两个样本，它们的□值分别接近两组文本的平均值。代表 MySpace 的文本是“Thanks for The add Jeremy!!”得爱死那些 Macross 的玩具图了。可惜我已经没有了…，而代表堆栈溢出的是“由于在错误的环境中查找而发生错误(即，不在数据帧内)。你可以显式地指定，但那将是丑陋的，可怕的代码。按照 Iselzer 的建议使用要好得多。”可以观察到，社会文本直接表达情绪，而 SE 文本通常要先描述问题，然后再表达作者对问题的看法。另一个观察是，“错误”这个典型的社会文本贬义词，在 SE 文本讨论一个代码问题时是中性的。

然后我们观察到，由于在 SE 文本中使用长而复杂的句子来描述与发展相关的问题，SE 文本的结构更加复杂。因此，我们通过计算字符数来测量两组文本中文本的平均长度。MySpace 评论的平均长度为 102，而 Stack Overflow 帖子的平均长度为 169。为了显示这种差异，我们还选择了两个文本，它们的长度接近两个数据集的平均长度。代表 MySpace 的文本是“HAPPY BIRTHDAY BEAUTIFIL…”希望你能看到更多…更好的是，我知道你会……愿上帝保佑你，熬夜吧。”整篇文章基本上都是用祈使句来表达祝福。而代表 Stack Overflow 的文本是“我一般在导入任何东西之前做它。如果你担心你的模块名称可能与 Python 标准库名称冲突，那么改变你的模块名称!”。这篇文章的结构比较复杂，它包含了一个虚拟语气从句。

因此，我们认为，这些观察到的差异导致建立在社会文本上的现成情感分析工具提供的结果不可靠，并大大提高了难度为 SE 文本定制这些工具。情感的分散表达需要 SE 指定的工具来识别作者是否在 SE 文本的不同部分表达情感。因此，SE 文本中复杂的句子结构对我们设置过滤规则以忽略可能的中性从句，并调整规则以增强输出结果变得非常重要。我们的方法是建立在 SentiStrength 和我们提出的规则之上的。评估表明，我们的过滤器调整规则能够为 SE 文本定制 SentiStrength，即使不更新其情感字典。例如，我们的方法将忽略所讨论的 se 文本样本中包含“error”一词的句子，而不是将其修改为字典中的“neutral”。

3 建议的方法

我们提出了一个三步走的方法。首先，我们对输入的 SE 文本进行预处理，并使用 Stanford CoreNLP[37]进行分割(第 1 步)。其次，我们使用过滤规则来识别句子是否可以触发后续分析(第 2 步)。第三，我们使用调整规则来增强 SentiStrength 的原始输出(第 3 步)。值得注意的是，我们的方法没有改变 SentiStrength 的字典。在接下来的小节中，我们将对每一步进行更详细的解释。

A.第一步:预处理和分割 SE 文本

首先，我们采用定制工具 SentiStrength-SE[12]使用的预处理方法，基于正则表达式过滤掉技术词汇，过滤包含“Dear”、“Hi”、“@”等字符的名称。一个不同之处在于，我们不会过滤掉完全由大写字母组成的单词。这些词很可能表达了一种夸张的情绪，而不仅仅是技术文本的一部分。我们还保留了感叹号作为第三步输入的一部分。文本“FEAR!!!!!!!!!!”，是一个很好的例子来说明上述两种差异。除此之外，我们还会过滤掉下面括号“[]”、“{}”、“<%%>”、双引号包围的单词，因为我们认为这些单词更有可能是引语、例子，或者是技术词汇，不表达情绪。例如，在“CREATE TABLE [[With Spiteful]]]…”这句话中，“Spiteful”是一个否定词，但它是表名的一部分，并不表达情绪。类似的，“It”这句话中的否定词“tommyrot”其实是拼成“tommyrot”的。，并不表示消极情绪，因为它是作为例子引用的。此外，带有下划线符号的句子，例如“CODE_FRAGMENT”，也会被过滤，因为这个符号也是技术文本的一个特征。

其次，为了处理句子结构更复杂的 SE 文本，我们引入了斯坦福 NLP 进行分词，而不是遵循 SentiStrength 只根据标点符号对文本进行分词。我们的分词首先将整个文本(命名为段落)分成多个句子。然后，它根据标点和连词(如“because”、“but”和“so”)将每个句子分成从句。进一步，我们使用 Stanford POS tagger 对每个句子的子句中的每个单词进行标注，标注其词性(POS)标注。预处理、分段和标记的 SE 文本为我们的方法的以下步骤奠定了基础。

B.步骤 2:匹配模式以触发后续分析

为了区分作者是在表达情感还是在描述问题，我们提出了我们的过滤规则。具体地说，

任何不符合以下三种模式的句子将被过滤掉。只有符合至少一个定义模式的句子才会被认为有可能表达情绪，并将进入下一步计算其情绪得分。对模式的详细描述如下。

1)直接情绪模式。当一个给定的句子只符合以下六种情况中的一种时，它就符合直接情感模式:(1)它包含感叹号;(2)它包含SentiStrength的表情符号列表中记录的表情符号，比如“:)”;(3)根据标注的POS包含感叹词，如“哇”;(4)包含四个四字母的骂人词，分别以字母“fu”、“da”、“sh”、“he”开头;(5)其给出的从句中至少有一个以感伤词开头(“请”和“请”除外);(6)祈使句，且感伤密度大于0.3。

直观地看，前四种情况表明作者强烈地表达了自己的情感。同时，我们提出了第五和第六种情景来处理祈使句。第五种情况建议涵盖以下两个例句:“谢谢您的耐心。和“欧文，谢谢你的幻灯片。”我们在第五种情况下排除了“please”和“plz”，因为它们更有可能表达请求，而不是他们原本想表达的积极情绪。第六种情况建议涵盖以下例句:“听起来不错。”。如何计算每句话的情感密度在II.B节中讨论。

2)装饰情感模式。一个给定的句子，当它包含一个作为副词的感伤词，或者它包含一个由副词修饰的感伤词(暗示这个感伤词必须是动词或形容词)时，就符合装饰感伤模式。我们建议，在使用感伤副词，或者用副词修饰感伤词的时候，作者是下定决心要在文中表达自己的情感的，因为副词是用来表示程度或范围的。例如，在“This is very沮丧。”，副词“very”表示更深层次的挫败感(即消极情绪)。而在“the performance degradation terrible”这句话中，副词“terrible”则表示性能下降的程度太大，从而也表现了作者的消极情绪。此外，对于“always”、“even”和“still”这三个副词，我们会发现从这些词到句尾都出现了修饰过的感伤词，因为它们基于语义的覆盖范围更广。最后，我们将“how”、“sort of”和“enough”(在感伤词之后)视为副词，因为它们也极有可能表明潜在情感的程度或范围。

3)“About Me”模式:当给定的句子符合以下三种情况时，它符合“About Me”模式:(1)它的主语是“我”，并且包含一个情感词(例如“我喜欢……”);(2)包含感伤动词后接宾语“me”(例如“…使我迷惑”);(3)带有感伤形容词或名词跟在“me”后面(例如“…make me confused”);(4)包含以“my”修饰的感伤词(例如，“这是我的错”)。我们提出这四种情况，是因为我们认为作者决心用第一人称的视角来表达她的情感。相反，第三人称视角通常更倾向于描述一个事实，而不是表达情感。例如，“他讨厌 p标签，很明显”这句话就被人为地标记为中性。

4)“判断”模式:给定的句子在包含以下四种句子结构时符合“判断”模式(1)“be动词+感情形容词/名词”(例如，“it 's ugly and efficient”);(2)“代词+感伤动词”(例如，“这太糟糕了”);(3)“get +多愁善感的词”(例如，“问题变得更糟了”);(4)“多愁善感的名词+ be动词”(例如，“失败的最大原因是你的粗心”);(4)“a/an/the +形容词+名词”(“它有一个优秀的命令行界面”)。我们认为，作者通常会在对其他事物或人做出判断时表达自己的情绪，所提出的五种情况可以在很大程度上涵盖潜在的判断-表达场景。

C.第三步:调整情绪分析

我们认为句子结构也有助于更好地理解 SE 文本中表达的情感。因此，我们建议在 SentiStrength 的基础上调整规则，进一步提升结果。

1)识别虚拟语气:虚拟语气表达了作者的主观愿望、怀疑、建议或假设，但并不表达真实的情感。因此，我们忽略了虚拟语气从句中出现的多愁善感的词语。我们的方法通过将“if”和“unless”识别为给定句子的从句中的条件状语来识别虚拟语气。我们不会识别这些子句中的情感。例如，在句子“如果你真的担心这个，Java 不是适合你的语言。”的否定感伤词“担心”在虚拟语气中，因此反映不出事实，也不表达作者的情绪。

2)通过句子结构识别多义词:SentiStrength 给每个多义词打一个多义词分。然而，当感伤词根据不同的句子结构表达不同的含义时，单一的感伤评分可能会导致错误的结果。在我们的观察过程中，我们总结了几个容易导致错误的多义词。这些词被分为两组。然后我们根据 POS 标签确定第一组单词的意思，根据第二组单词与其他单词的搭配确定第二组单词的意思。

可以通过 POS 标签确认的第一组多义词如下:

Like: SentiStrength 将这个词检测为阳性。在“I like playing with you”这个句子中，“like”这个词是肯定的，它的意思是主语更喜欢做某事。然而，在“it looks like this”这句话中。，它的意思接近于“相似”，并没有表达积极的情绪。当“like”意为“相似”时，其 POS 为介词。所以当它的 POS 是介词时，我们不把这个词标记为肯定的，而是标记为中性的。

Pretty 和 Super: SentiStrength 会将这些单词检测为肯定的。在句子“She is pretty.”，“pretty”这个词是肯定的，意思是某人很有吸引力。然而，在“I'm pretty sure”这个句子中，它的意思接近于“very”，它并没有表达积极的情绪。当“pretty”的意思是“very”时，它的 POS 是副词。所以当它的 POS 是副词时，我们不把这个词标记为肯定的，而是标记为中性词。它也会起到助推器词的作用，可以加强后面的情绪词的强度，比如“very”。“Super”和“pretty”类似。当它的词性是副词，用来表示

一些高度或极端程度的东西，我们将其检测为中性，它也会起到助推器词的作用。

Block 和 Force: SentiStrength 会将这些词检测为负面的。在句子中，“缺乏训练会阻碍职业发展。，“阻碍”这个词是否定的，它指的是使行动或进步变得困难或不可能的东西，但在类似“我确定首先是代码阻碍”的句子中，它指的是被视为单个单位的东西的数量，不表达任何负面情绪。当“块”意为“一个单位”时，其 POS 为名词。所以当它的 POS 是名词时，我们不把它标记为否定的，而是标记为中性的。“Force”与“block”有相似之处。当它的 POS 为名词时，表示体力，我们将其标记为中性而不是否定。

第二组可以通过与其他词的搭配来确认的多义词如下:

说谎:SentiStrength 检测到这个词是否定的。在句子“He was lying.”，“撒谎”这个词是否定的，它的意思是偏离事实，但在类似于“网上到处都是谎言”的句子中。，它的意思接近于“在”，不表达负面情绪。当“lying”意为“be in”时，除“to”(不包括短语“lie to”)外，常与介词连用。因此，当我们认识到这种搭配时，我们不会将其标记为否定，而是将其标记为中性。

怨恨和善良:SentiStrength 检测到“怨恨”这个词是负面的，但在短语“尽管”中，整个短语代表了一种转折关系，并没有表达负面情绪。所以当在这个短语中发现时，我们不会把它标记为负面的，而是标记为中性的。“Kind”和“spite”很相似。在“kind of”这个短语中，这个短语的意思接近于“某种程度上”，这个短语没有表达任何积极的情绪。所以当我们在这个短语中发现它时，我们不会把它当作积极的，而是当作中性的。

Miss:“Miss”这个词在 SentiStrength 网站上既有正面的分数 02，也有负面的分数 02，因为当它的意思接近于“怀念”时，它经常被用来同时表达悲伤和爱。然而，当它的意思接近于“注意到不存在的东西”时，它在 SE 文本中表达的是负面情绪。根据我们的观察，当它的意思是“深情地回忆”时，后面往往会跟着人称代词。当它表示“注意到某物不在那里”时，后面跟着宾语。因此，我们会检查这个词的宾语，只有当它的宾语是人称代词时，我们才会同时计算它的积极情绪和消极情绪。

3)处理否定句。在《SentiStrength》中关于否定的原始规则将通过在否定词的正前方乘以-0.5 的因子来翻转一个情感词的极性。这个规则过度补偿和忽略了太多的否定场景，特别是对于 SE 文本。比如这段文字的情绪“不用担心，是文件的权限问题。”，会按照原来的否定规则被认定为积极，却被标注为中性。相反，在我们的方法中，否定词列表中的单词和以“t”结尾的单词(例如，“isn’t”)将中和下面三个单词中单词的情绪(“to”除外)。我们还增加了另外三个单词“nothing”、“no”和“without”(不在 SentiStrength 的原始否定列表中)，以中和它们后面第一个单词(“to”被排除在外)的情绪。新增的三个否定词否定范围有限，是因为它们的 POS 都是名词或介词，而原表中的否定词或以“t”结尾的否定词都是助动词。

表二。 感受力的分析(用三元输出)

句子	生梯。分数	
	ρ	η
这个应用真的很好[2][+1 助推器词]尽管[-4]有一些(小)缺点[-2]。	3.	4
它的字体大小会变大或变小以适应它们的空间，我不喜欢[2][*-0.5 约]。负乘数]。	2	1
如果问题解决了，我想它会更实用。	1	2
总的来说，这是一个很好的[2]应用程序。	2	1
整体结果= -1 作为 Max(ρ) < Max(abs(η))		

表3。 会话的分析(带三元输出)

句子	生梯。分数	
	ρ	η
[契合“装饰情绪模式”]这个应用真的很不错[2][+1 助推器词]尽管[多义词]有一些(小)缺点[-2]。	3.	2
[fit "'About Me 'Pattem "]它的字体大小会变大或变小以适应它们的空间，我不喜欢[被否定中和]。	1	1
如果问题解决了，我认为它会更实用。	1	—1
[fit“‘Judgement’Pattem”]总的来说，这是一款不错的[2]应用。	2	1
总体结果= 1 as Max(ρ)> Max(abs(η))		

D.通过样本 SE 文本进行总结

我们现在使用下面的示例 SE 文本来展示 SESSION 是如何工作的:“这个应用程序确实很好，尽管有一些(小)缺点。它的字体大小会变大或变小，以适应分配给它们的空间，这是我不喜欢的。如果你能解决这个问题，我相信它会更实用。总的来说，这是一款不错的应用。”这篇文章的情绪被手动标记为积极的。原始 SentiStrength 的分析和结果如表 II 所示，SESSION 的分析和结果如表 III 所示。可以观察到，基于我们提出的过滤器规则和调整规则(步骤 2 和步骤 3)，这些规则依赖于步骤 1 中预处理 SE 文本的分割和 POS 标注，SESSION 正确地识别了该文本的积极情绪，而 SentiStrength 则被文本误导，错误地将其情绪识别为消极情绪。

四、实验设置

现在我们介绍我们的实验设置来评估我们的方法。第 IV.A 节介绍了用于评估的 SE 文本的四个数据集。第 IV.B 节定义了评估建议方法性能的指标。第 IV.C 节介绍了我们的研究问题和实验设计。

A. 4个数据集的基准

我们首先引入了 Lin 等人研究的基准，并报告说目前没有情感分析工具可以提供 SE 文本[36]中表达的情感的可靠结果。它由三个数据集组成，分别建立在 1500 个 Stack Overflow 讨论、341 个应用评论和 926 个 JIRA 评论上。然后，我们引入 Calefato 等人在 4423 篇 Stack Overflow 帖子上构建的第四个数据集，以提出一个

表5 .会话和三个基线在四个数据集上的性能

数据集	工具	整体精度	积极的			中性			负		
			P	R	F	P	R	F	P	R	F
堆栈溢出 4423	SentiStrength	81.55%	88.90%	92.34%	0.906	92.76%	63.58%	0.754	66.83%	93.18%	0.778
	SESSION	86.30%	90.15%	94.70%	0.924	90.19%	75.97%	0.825	77.87%	90.18%	0.836
	SentiStrength-SE	78.86%	90.47%	82.06%	0.861	72.74%	77.80%	0.752	74.80%	76.29%	0.755
	Senti4SD	95.27%	97.25%	97.45%	0.974	95.02%	93.51%	0.943	93.15%	95.01%	0.941
堆栈溢出 1500	SentiStrength	68.00%	19.28%	36.64%	0.253	86.20%	74.98%	0.802	36.74%	44.38%	0.402
	会话	78.13%	30.89%	29.01%	0.299	85.10%	89.67%	0.873	54.10%	37.08%	0.44
	SentiStrength-SE	78.00%	31.18%	22.14%	0.259	82.72%	92.86%	0.875	50.00%	19.66%	0.282
	Senti4SD	76.93%	27.59%	30.53%	0.29	83.11%	90.51%	0.867	62.07%	20.22%	0.305
应用评价	SentiStrength	67.45%	71.81%	87.63%	0.789	4.76%	4.00% -	0.043	70.97%	50.77%	0.592
	会话	-	-	-	-	-	16.00%	-	-	-	- 0.62
		68.62%	76.17%	87.63%	0.815	9.76%		0.121	77.91%	51.54%	
	SentiStrength-SE	61.58%	74.15%	81.72%	0.777	9.59%	28.00%	0.143	80.95%	39.23%	0.528
	Senti4SD	63.93%	71.24%	86.56%	0.782	9.80%	20.00%	0.132	81.25%	40.00%	0.536
JIRA 问题	SentiStrength	81.21%	86.03%	93.45%	0.896	—	——	——	98.16%	75.63%	0.854
	SESSION	80.56%	93.13%	93.45%	0.933	—	——	——	98.55%	74.69%	0.85
	SentiStrength- se	77.21%	95.26%	90.00%	0.926	—	——	——	99.34%	71.38%	0.831
	Senti4SD	57.88%	81.55%	86.90%	0.841	—	——	——	99.65%	44.65%	0.617
						—					
						—					
						—					
						—					

基于学习的 SE 文本情感分析方法。表 IV 报告了每个数据集的文本总数，以及积极、中性和消极文本的数量。

b指标

我们首先利用三个指标来衡量三个情感极性(即积极，消极和中立)的情感分析的准确性。给定一组 S 文本，特定情感极性的精度(P)、召回率(R)和 F-测度(F)计算如下:

$$P = \frac{|S_c \cap S_c'|}{|S_c'|} \quad R = \frac{|S_c \cap S_c'|}{|S_c|} \quad F = \frac{2 \times P \times R}{P + R} \qquad (1)$$

其中 表示具有情感极性 c 的文本集， 表示通过工具分类为具有情感极性 c的文本集。F-measure是精密度和召回率的加权调和平均值。更高的 F-measure 意味着精度和召回率都高，工具性能更好。我们进一步介绍了在集合 S 上对所有三种情感极性进行情感分析的总体精度，度量总体精度计算如下:

$$\text{粽} \qquad \text{粽} = \frac{\sum_{c \in_i} | \cap |}{| |} \qquad (2)$$

其中，我们在□中对所有三个极性具有相同情感极性“c”的□中的文本数量进行累积，然后计算其在给定文本集 S 中的比例。

C.研究问题

在本文中，我们旨在研究句子结构是否可以有效地提高 SE文本的情感分析性能。因此，我们提出以下三个研究问题:

表四。 用于我们评估的数据集

数据集	句子	积极的	中性	负
栈溢出 4423	4423	1527	1694	1202
堆栈 Overflow1500	1500	131	1191	178
应用评价	341	186	25	130
JIRA 问题	926	290	0	636

RQ1:我们提出的方法在分析 SE 文本的情感方面是否优于基线?

RQ2:我们的过滤规则做出了多大的贡献?

RQ3:我们的调整规则贡献了多少?

为了研究 RQ1，我们引入了以下三个基线:(1)SentiStrength[11]，最先进的基于字典的工具，也是我们方法的基础;(2) SentiStrength-SE[12]，一个具有代表性的基于字典的工具，它构建了一个为 SE 文本指定的新字典;(3) Senti4SD[13]，一个代表性的 SE- Customized，基于学习的工具，在 Stack Overflow 4423数据集(也是我们评估的数据集的一部分)上进行训练。通过与三种基线方法的比较，我们希望发现我们的方法是否可以有更好的性能，以及我们对 SE 文本中情感表达独特性的观察是否有效。为了研究 RQ2和 RQ3，我们将分别在四个数据库上运行仅使用我们的过滤规则(SS + filter)和仅使用我们的调整规则(SS + adjust)的 SentiStrength，以进一步比较它们与 SentiStrength和 SESSION的性能。

V.结果和讨论

A. RQ1:我们提出的方法在分析 SE文本的情感方面是否优于基线?

表 5 显示了评估的四种方法的性能。首先，我们比较了 SESSION 和 SentiStrength 的性能。我们发现，SESSION 在 Stack Overflow 4423、Stack Overflow 1500和 App Reviews中的整体准确率要优于 SentiStrength。它在 Stack Overflow 1500上的整体准确率可以比 SentiStrength 高出 10%。我们之前的观察表明，社交文本比 SE 文本更多愁善感，表达方式也更直接。这种差异使得 SentiStrength 倾向于输出更多积极和消极的结果。这种倾向可以通过表 5 中 SentiStrength 实现的识别中性文本的低召回来观察。另一方面，我们提出过滤规则和调整规则来解决 SE 文本中情感表达更加间接和分散的问题。因此，我们的方法在 Stack Overflow 4423 上实现了比 SentiStrength 多 12%的召回率。提出的过滤器调整规则也是如此

表六世。 对比 session (sn)和 sentistrength (ss, m 代表手动标签)的样本

句子	米	SN	党卫军
通过添加一个条件*a != *b，很容易防止混叠。	0	0	1
如果你真的担心这个，Java 不是适合你的语言	0	0	1
为什么人们讨厌匿名块初始化器	0	0	1

有助于帮助我们的方法在评估数据集上的所有三种情绪极性的 F-Measures 中优于 SentiStrength，除了 JIRA 问题。与 Stack Overflow 的两个数据集不同，JIRA Issue 没有中性文本。因此，它给我们的过滤器调整规则留下了很少的工作空间。然而，我们的方法在 JIRA 问题上的积极情绪的 F-Measure 中仍然优于 SentiStrength，而在消极情绪的 F-Measure 中表现略差。由于 JIRA Issue 包含的负面文本比其正面文本多两倍左右，并且没有中性文本，因此 SESSION 在整体准确性上的表现略差(进一步的讨论在本节的末尾)。然后，我们使用表 VI 所示的样本文本(来自四个数据集)来演示 SESSION 如何优于 SentiStrength。在表中，第一个句子被 SentiStrength 识别为正，因为“!”，而 SESSION 可以过滤掉“!”=”作为技术文本的一部分。第二句因为“担心”被 SentiStrength 识别为否定句，而 SESSION 定位其虚拟语气，识别为中性句。因为“恨”，第三句话被 SentiStrength 识别为否定，而这句话在我们的过滤规则中不符合任何模式，SESSION将其识别为中性。

其次，我们比较了 SESSION 和 SentiStrength-SE 两者的表现。从表 V 中我们可以发现，除了 Stack Overflow 1500 和 App Reviews 上中立情绪的召回率和 F-Measure，以及 Stack Overflow 4423 上中立情绪的召回率，SESSION 在四个数据集上的总体准确性和几乎所有其他指标上都优于 SentiStrength-SE，其中 SESSION 的表现略差。这两种方法实际上都利用了 SE 文本的中性倾向。SentiStrength- se 选择建立一个 se域指定的字典，而我们的方法选择使用过滤器调整规则来增强 SentiStrength。我们发现，在 Stack Overflow 1500 上，SentiStrength-SE 和 SESSION 的总体精度差别不大。不过，为了应对中性倾向，SentiStrength- se 更新后的情感词表被缩短到只有 550 个单词，而 SentiStrength 中的原始列表有 2000 多个单词。因此，与 SESSION 和 SentiStrength 相比，SentiStrength- se 所涵盖的可能的积极和消极情绪要少得多。像“我爱。”，不会被 SentiStrength-SE 认定为多愁善感，因为它的列表中缺少多愁善感的“爱”这个词。然后，我们认为我们的观察和提出的过滤调整规则更好

表七世。对比 session (sn)和 sentistrength-se 的样本(se，m代表手动标签)

句子	米	SN	SE
乔伊，明白了!我想你是对的	1	1	0
如何正确打印一个 CString 消息框?什么都没有出现…	0	0	1
你怕商标官司吗?	0	0	1

挖掘东南语篇中独特的情感表达方式。我们使用表七所示的样本文本来演示 SESSION如何优于 SentiStrength-SE。在表中，第一句话被 SentiStrength-SE分类为中性，因为它会删除“!”，在预处理过程中。SESSION 的预处理规则将保持”!，这样文本就不会被错误分类。第二个句子被 SentiStrength-SE 分类为否定，因为单词“messagebox”与其通配符“mess*”匹配，而“mess*”在 SentiStrength-SE 的情感词列表中具有负分值 02。这个词在 SESSION 的情感词列表中不匹配，所以文本不会被错误分类。第三个句子因为“害怕”被 SentiStrength-SE 分类为否定，而这句话在我们的过滤规则中不符合任何模式，SESSION 将其识别为中性。

第三，我们比较了 SESSION 和 Senti4SD 两者的表现。从表 V 中，我们发现 Senti4SD 在其训练集 Stack Overflow 4423 上的性能明显优于 SESSION。我们认为这个结果是合理的，因为在改进的特征工程覆盖更多隐含事实 b[13]的帮助下，Senti4SD 可以更好地预测 SE 文本中的情感，特别是来自 Stack Overflow 4423 的情感，其中 Senti4SD 微调了其训练的 SVM 模型的参数进行分类。然而，当应用于其他数据集时，Senti4SD 的性能开始下降。它在 Stack Overflow 1500(与其训练集相似的数据集)上的性能低于 SESSION。同样的对比也可以在 App Reviews 上观察到。而且，它在 JIRA Issue 上的整体准确率仅为 57.88%，负面召回率仅为 44.65%。另一方面，SESSION 能够在所有情绪上实现均衡的查全率和查准率。负面文本的召回率比 Senti4SD 高出 10%-30%左右。我们认为 SESSION 实现了比 Senti4SD 更好的综合性能，特别是在泛化性方面。

我们对评估的整体观察表明，从软件工程文本开发的工具(SentiStrength-SE, Senti4SD)通常可以在情感文本中实现更高的精度，但必须付出召回率较低的代价。为社交文本开发的工具(SentiStrength)往往可以在情感文本中实现更高的召回率，但要承受精度的损失。相比之下，我们的工具，利用基于句子结构的 SE 文本中独特的情感表达，可以在准确率和召回率上取得很好的平衡表现，并且与基于学习的工具相比，具有更好的泛化能力。

B. RQ2:我们的过滤规则做出了多大的贡献?

SS + Filter 的结果如表 VIII 所示。将 SS + Filter 与 SentiStrength 的数据进行对比，我们可以发现，在两个 Stack Overflow 数据集中，SS + Filter 的整体精度优于原始工具。在 Stack Overflow 4423中，SS + Filter的整体精度提高了 2.13%，而在 Stack Overflow 1500 中，SS + Filter 的整体精度提高了 6.93%。这些规则可以有效提高感伤文本的准确率和中性文本的召回率，尤其在中性文本方面表现更好。带有过滤规则的工具的中性 f 值均高于原始工具和带有过滤规则的工具。而在另外两个数据集中，过滤规则所能带来的整体精度提升都不高。因为这两个数据集上的中性文本很少，所以在中性文本上表现更好的过滤规则比较困难

表八世。 分别分析 rule-filter 和 rule-adjust 的性能

数据集	工具	整体精度	积极的			中性			负		
			P	R	F	P	R	F	P	R	F
堆栈溢出 4423	SentiStrength	81.55%	88.90%	92.34%	0.906	92.76%	63.58%	0.754	66.83%	93.18%	0.778
	SS + Filter	83.68%	90.06%	91.94%	0.91	90.56%	70.78%	0.795	71.30%	91.35%	0.801
	SS + Adjust	84.08%	89.00%	94.89%	0.919	92.06%	68.42%	0.785	72.33%	92.43%	0.812
	SESSION	86.30%	90.15%	94.70%	0.924	90.19%	75.97%	0.825	77.87%	90.18%	0.836
堆栈溢出 1500	SentiStrength	68.00%	19.28%	36.64%	0.253	86.20%	74.98%	0.802	36.74%	44.38%	0.402
	SS + Filter	74.93%	23.31%	29.01%	0.259	85.12%	85.47%	0.853	48.23%	38.20%	0.426
	SS + Adjust	73.87%	27.43%	36.64%	0.314	86.23%	82.54%	0.843	41.62%	43.26%	0.424
	SESSION	-	-	-	-	-	-	-	-	-	- 0.44
		78.13%	30.89%	29.01%	0.299	85.10%	89.67%	0.873	54.10%	37.08%	
应用评价	SentiStrength	67.45%	71.81%	87.63%	0.789	4.76% -	4.00% -	0.043	70.97%	50.77%	0.592
	SS + Filter	-	-	-	-	10.00%	16.00%	-	-	-	-
		67.45%	75.36%	85.48%	0.801			0.123	74.44%	51.54%	0.609
	SS +调整	69.21%	73.45%	89.25%	0.806	8.33%	8.00%	0.082	74.73%	52.31%	0.615
	会话	68.62%	76.17%	87.63%	0.815	9.76%	16.00%	0.121	77.91%	51.54%	0.62
JIRA 问题	SentiStrength	81.21%	86.03%	93.45%	0.896	- - -	- - -	- - -	98.16%	75.63%	0.854
	SS + Filter	-	-	-	-	- -	- -	- -	-	-	-
		80.35%	87.91%	92.76%	0.903				97.94%	74.69%	0.847
	SS +调整	82.18%	91.28%	93.79%	0.925	- - -	- - -	- - -	98.19%	76.89%	0.862
	会话	80.56%	93.13%	93.45%	0.933	- - -	- - -	- - -	98.55%	74.69%	0.85

带来改善。综上所述，因为 SS + Filter 的 f 指标几乎都比 SentiStrength 好，所以我们可以说过滤规则实际上可以带来改进。但是，在分析情感文本过多的数据集时，它的改进会有些不稳定。

C. RQ3:我们的调整规则做出了多大的贡献?

SS + Adjust的数据见表八。我们可以发现，SS+ Adjust在4个数据集中的整体精度都优于 SentiStrength。它还可以有效提高感伤文本的准确率和中性文本的召回率。SS + Adjust的 f 指标都比 SentiStrength 好，所以我们可以说调整规则其实是可以带来提升的。相比于过滤规则，它们更擅长感伤文本。带有调整规则的工具的正 f 值和负 f 值几乎都高于原始工具和带有过滤规则的工具。对于情感文本，调整规则可以在不损失太多召回的情况下提高精度。在 Stack Overflow 4423 中，SS + Adjust的正精度(89.00%)和 SS + Filter的正精度(90.06%)是相似的。但 SS + Adjust的正召回率(94.89%)高于 SS + Filter的正召回率(91.94%)。此外，我们还发现，在分析情感文本过多的数据集时，过滤规则带来的提升会少一些。在两个具有更多中性文本的 Stack Overflow 数据集中，SS + Adjust 的整体准确率分别比原始工具高 2.53%和 5.87%。在另外两个具有更多情感文本的数据集中，SS + Adjust 的整体准确率分别比原始工具高 1.76%和 0.97%。

综上所述，我们的方法的两套规则都可以带来改进，因为它们可以有效地提高情感文本的准确性和中性文本的回忆。因为我们的规则是基于 SE 文本比社会文本更间接、更复杂的观察，所以在分析具有更多中性文本的数据集时，它们会更有帮助。更具体地说，表 VIII 显示，SESSION(同时带有过滤规则和调整规则)在 Stack Overflow 4423 和 Stack Overflow 1500 上都表现最好，而 SS + adjust在 App Reviews 和 JIRA Issue 上都表现最好。这一观察表明，虽然我们的过滤规则在处理中性文本方面做得更好，但它们也可能会进行损失

当过滤掉不能匹配任何模式的句子时，尤其是与在所有 SE 文本上执行更稳定的调整规则相比。然而，当应用于 App Reviews 和 JIRA Issue 时，我们的过滤规则仅分别使整体准确率降低了 0.59%和 1.62%。由于这两个数据集分别只包含 7%和 0%的中性文本，表明我们的过滤规则几乎没有发挥作用的空间，因此我们建议，我们的过滤规则可能造成的情绪上下文损失并不显著。然后，我们建议，由于 SE 文本中情感表达的更间接和分散的性质，我们的过滤规则和调整规则都有助于在实践中对 SE在线工具生成的 SE文本进行情感分析，其中中性文本可能占很大一部分。

此外，我们对实验结果又进行了三次观察。首先，我们对 SentiStrength 在 Lin 等人也研究过的三个数据集上的实验结果与他们的论文[36]中的结果略有不同。我们发现，这是因为 Lin 等人使用了来自 SentiStrength 的正负分数之和的符号来获得整体极性，而我们的方法使用了 SentiStrength 内置的“三元”选项来输出整体极性。通过比较，我们发现我们对 SentiStrength 的结果稍好一些，因此我们在与 SentiStrength 进行比较时没有任何偏差。其次，我们的方法的改进，虽然在所有数据集上都是平衡和稳定的，但仍然不高。我们认为这种情况是由于我们保守地选择使用基于句子结构的规则来与 SentiStrength 合作造成的。在未来的工作中，我们计划对开发人员如何在 SE 文本中表达他们的情感进行更深入的研究，并通过参考现有工作[45]来仔细建立 SE指定的词典。第三，我们发现人工标注情感的标准在不同的数据集中可能会有所不同。在我们的研究过程中，我们有一个候选的多义词“work”。当“work”是不及物动词时，它的意思是“影响某事”，可以被看作是一个积极的感伤词。然而，Stack Overflow 4423 偏向于这个候选词，而 Stack Overflow 1500 则倾向于相反，因此我们最终将这个词排除在 SESSION 的调整规则之外。我们进一步的调查表明，尽管这两个数据集都是由 Stack Overflow 创建的，但是为 Stack Overflow 1500标记情绪的参与者倾向于中立

文本而不是积极的文本或消极的文本。例如，在这个数据集中，典型的积极文本，如“我感谢你的帮助”，和典型的消极文本，如“我怀疑为什么做出这个决定”，都被标记为中性。一种可能的解释是，在参与者的意见中，这些文本的情绪，无论是确定为积极的还是消极的，都不足以令人信服地表明潜在相关 SE 任务对其他开发人员的真实状态。类似的情况也发生在 App Reviews 中，有 25 个文本被手动标记为中性，而它们包含了相当数量的情感词汇。我们调查的这些结果能够解释为什么 SESSION 和所有基线方法在 Stack Overflow 1500 的正面和负面文本以及 App Reviews 的中性文本上表现不佳。因此，我们建议，如果 SE 社区能够就在 SE 文本上手动标记情感的统一标准达成一致，以帮助研究人员(包括我们)建立更一致的数据集，旨在加强 SE 领域的情感分析研究，这将是有利的。

六、对有效性的威胁

内部威胁。对我们实验结果有效性的一个可能威胁是，我们不能保证基于斯坦福 CoreNLP 的 SE 文本分割和 POS 标注器识别 100%的准确性。然而，现有的工作已经报道，在分析具有适当句子和语法结构的上下文的文本时，现成的 NLP 工具的准确性是可以接受的，而不是分析碎片化的源代码[44]。通过额外的预处理，我们认为我们分析的 SE 文本的质量能够为我们的方法保留可用的句子结构。在我们的观察过程中，我们也没有发现斯坦福 CoreNLP 的输出有明显的错误。另一个可能的威胁是，我们的观察不够彻底和完整，无法充分利用开发人员如何在 SE 文本上表达他们的情绪，因此我们定义的规则无法涵盖我们在评估数据集上的观察发现的所有误判情绪。尽管如此，我们认为本文中定义的这些规则是一个良好的开端，因为在它们的帮助下，与基线方法相比，我们的方法能够实现更好的整体性能。我们计划在未来通过咨询现有工作(例如，心理工作量评估[34])，在心理学和社会学理论的指导下进行更深入、更全面的研究。

外部的威胁。我们的工作基于四个数据集，总共包含 7190 个 SE 文本，并带有手动标记的情感。我们的实验规模并不大，但我们仍然认为我们的发现是相关的，因为这四个数据集来自两个现有的工作[13,36]，并且是由三种不同的软件开发在线工具(Stack Overflow, App Reviews 和 JIRA)生成的。由于 Stack Overflow 的广泛流行，来自 Stack Overflow 的两个数据集能够代表开发人员通过 SE 文本的典型交互。另外两个数据集与前两个不同，包含了 SE 文本，这些文本几乎被标记为积极或消极情绪。因此，这两个数据集非常有助于验证 SESSION 是否过度强调 SE 文本中的中性情绪(两个 Stack Overflow 数据集中的大多数情绪)。我们的评估表明 SESSION 的性能在 App Reviews 和 JIRA 数据集上几乎没有下降，其中 SentiStrength-SE 和 Senti4SD(两种 se定制的基线方法)在性能上遭受明显的损失。

7相关工作

在本节中，我们将重点讨论软件工程领域中情感分析的相关研究。

A.应用于 SE的情感分析工具

开发并用于检测情绪的一套全面的开箱即用的情绪分析工具可以在其他地方找到[8,9,10]。在这些工具中，SentiStrength[11]、NLTK[39]和 stanford dnlp[37]是 SE域中常用的工具。然而，这些工具在应用于 SE 文本时表现不佳[12,36,40,41]，很大程度上是因为它们是在非技术文本上训练的。因此，一些研究通过使用 SE 文本(如 SentiCR[15]、sentistrength -SE[12]和 Senti4SD[13])来改善这种情况。SentiCR 是一种使用梯度增强树(GBT)[17]训练的监督工具，专为代码审查评论而设计。它通过计算从输入文本中提取的词袋的 TF-IDF [16] (Term Frequency - Inverse Document Frequency)来生成特征向量。SentiStrength -SE 是一个基于字典的工具，它是在 SentiStrength 的基础上通过用 SE 术语扩展固有字典而开发出来的，这是第一个针对 SE 的情感分析。Senti4SD[13]是在 Stack Overflow 的大约 4K 问题、答案和评论的黄金标准上进行训练的。在进行情感分类任务时，它利用了三种特征，包括基于字典的特征(即 SentiStrength 使用的字典)、基于关键字的特征(即从大规模 Stack Overflow帖子中提取的一元图和双元图)和语义特征(基于在 Stack Overflow帖子上训练的词嵌入)。与 Senti4SD 在语料库中利用基于关键字的特征不同，我们更注重分析 SE 文本的特征，并基于我们的密切观察创建了一组利用句子结构信息(例如，识别虚拟语气从句或区分多义词的含义)的启发式方法。此外，我们的方法是基于字典的，可以更广泛地推广到各种 SE 文本，而基于学习的方法需要大量的标记数据来训练它们的分类器[15,25,36]。

除了讨论的用于检测给定文本的情绪极性(即积极、消极和中立)的情感分析工具外，Islam 等人[14]提出了一种基于字典的工具，可以检测软件工程文本中表达的兴奋、压力、抑郁和放松。为了更好地评估情绪得分，他们的方法还集成了一套用于感知唤醒的启发式方法，但它并没有明确地利用 SE 文本中的句子结构，而在本文中，我们使用这些句子结构作为我们方法的过滤器调整规则的基础。

B.情感分析在 SE中的应用

近年来，情感分析作为 SE[18]的人为因素的一部分，越来越受到人们的关注

广泛应用于 SE 任务中[19-28]。许多研究在协作在线环境(例如，GitHub、JIRA、Stack Overflow 和 App store)中应用了情绪分析，呈现如下:Pletea 等人。[29]从 GitHub 上围绕提交和拉取请求的安全相关讨论中挖掘情绪，发现在安全相关讨论中表达的负面情绪比在其他讨论中表达的负面情绪更多。Guzman 等人[30]使用基于字典的情绪分析来检测 GitHub 中六个 OSS 项目的提交评论中表达的情绪，并显示分布式团队较多的项目在其情绪内容中往往具有更高的积极极性。Mantyla等人[21]用 VAD (Valence, Arousal, and Dominance)指标分析了包含 200 万条评论的 700,000 个 JIRA 问题。结果表明，不同类型的问题报告(例如，功能请求，改进和 Bug 报告)具有相当的 Valence 变化，而问题优先级的增加(例如，从 Minor 到 Critical)通常会增加 Arousal。Ortu 等人[22]分析了超过 560K 条 JIRA 评论中开发人员的情绪、情绪和礼貌与修复 JIRA 问题的时间之间的关系。他们发现，更快乐的开发者(在评论中表达 JOY 和 LOVE 等情绪)往往会在更短的时间内解决问题。Calefato 等人[32]从 Stack Overflow 上定量分析了一组超过 87K 个问题的情绪，发现成功的问题通常采用中性的情绪风格。Canfora et al.[34]表明，用户反馈包含使用场景、bug 报告和功能请求，可以帮助应用程序开发人员完成软件维护和进化任务。

然而，我们也需要指出，目前的情感分析工具在 SE 领域的精度和可靠性仍然低于满意度[20,36]。一个可能的原因是，许多先前的作品利用了现成的情感分析工具(如 SentiStrength[11])，这些工具建立在与 SE 域无关的文本上，而提出 SE 指定的情感分析是对[36]的挑战。因此，在本文中，我们选择首先利用 SE 文本中情感表达的独特性，然后通过将我们的过滤器调整规则集成到 SentiStrength 中来提出我们的方法。

8 结论及未来工作

越来越多的工作将情感分析应用于 SE 文本，以增强软件开发和程序理解。然而，目前的自动化情感分析，即使包括两种 SE 定制的方法，也无法在 SE 文本上提供可靠的结果。因此，我们首先观察并发现，与来自普通社交网络的文本相比，SE 文本中的情感表达更加间接和分散。然后，我们提出了一套基于 SE 文本内部句子结构的过滤和调整规则，并将这些启发式方法与主流的基于词典的方法(称为 SentiStrength)结合起来。我们基于四个不同数据集的评估表明，我们的方法总体上比三种基线方法具有更好的性能和泛化性。我们的工具现在是公开可用的[43]。

我们未来工作的可能方向如下:(1)我们计划在相关心理学和社会学理论的指导下，进一步探索开发者如何以及为什么在 SE 文本中表达他们的情绪，以便我们可以相应地微调和丰富我们的过滤-调整规则;(2)我们计划通过咨询现有的工作(例如[45])，提出一个 SE 指定的词典，进一步改进 SE 文本的情感分析;(3)我们计划进一步探索 SE 文本上表达的情感，如果被正确识别，是否从多个角度明确地与正在进行的软件开发状态相关。

鸣谢

国家重点研发计划项目(No. 2019YFE0105500)、挪威研究理事会项目(No. 309494)、国家自然科学基金项目(No. 62072227、61802173、61690204)、江苏省政府间双边创新项目(BZ2020017)、新型软件技术与产业化协同创新中心共同支持。

参考文献

[1] B. Pang 和 L. Lee，“意见挖掘和情绪分析”，Found。Trends Inf. Retr。，vol. 2, no。1-2, p. 1-135, january 2008。

[2] B. W. Boehm、A. Egyed、J. Kwan、D. Port、A. Shah 和 R. J. Madachy，“使用双赢螺旋模型:一个案例研究”，《计算机》，第 31 卷，第 31 期。7，第 33-44 页，1998 年。

[3] M. D. Choudhury 和 S. Counts，“通过社交媒体了解工作场所的影响”，载于《计算机支持的协同工作》，CSCW 2013，第 303-316 页。

M. M. Rahman, C. K. Roy 和 I. Keivanloo，“使用众包知识为源代码推荐有见地的评论”，2015 年 IEEE 第 15 届源代码分析和操作国际工作会议，SCAM 2015，第 81-90 页。

[5] M. R. Wrobel，“软件开发过程中的情感”，第六届人类系统交互国际会议，HSI 2013，第 518-523 页。

[6] D. J. McDuff, A. K. Karlson, A. Kapoor, A. Roseway 和 M. Czerwinski，“Affectaura:情感记忆的智能系统”，CHI 会议上计算系统中的人为因素，CHI 2012，第 849-858 页。

[7] R. Socher, a . Perelygin, J. Wu, J. Chuang, C. D. Manning, a . Y. Ng 和 C. Potts，“情感树库上语义组合性的递归深度模型”，载于 2013 年自然语言处理经验方法会议论文集，EMNLP 2013，第 1631-1642 页。

[8] W. Medhat, A. Hassan 和 H. Korashy，“情感分析算法和应用:一项调查”，Ain Shams 工程杂志，第 5 卷，第 5 期。4, pp. 1093-1113, 2014。

[9] A. Yadollahi、A. G. Shahraki 和 O. R. Zaiane，“从观点到情感挖掘的文本情感分析的现状”，ACM Computing Surveys (CSUR)， vol. 50, no. 11。2, pp. 1-33, 2017。

[10] A. Giachanou 和 F. Crestani，“喜欢与否:对 twitter 情感分析方法的调查”，ACM Computing Surveys (CSUR)， vol. 49, no. 5。2, pp. 1-41, 2016。

[11] M. Thelwall、K. Buckley 和 G. Paltoglou，“社交网络的情感强度检测”，《美国信息科学与技术学会学报》，第 63 卷，第 2 期。1, pp. 163-173, 2012。

[12] M. R. Islam 和 M. F. Zibran，“在软件工程中利用自动情感分析”，2017 年 IEEE/ACM 第十四届挖掘软件存储库国际会议，MSR 2017，第 203-214 页。

[13] F. Calefato、F. Lanubile、F. Maiorano 和 N. Novielli，“软件开发中的情感极性检测”，《实证软件工程》，第 23 卷，第 2 期。3, pp. 1352-1382, 2018。

[14] M. R. Islam 和 M. F. Zibran，“Deva:在软件工程文本的价唤醒空间中感知情感”，载于第 33 届 ACM 应用计算年度研讨会论文集，SAC 2018，第1536-1543 页。

[15] T. Ahmed, a。Bosu, a。Iqbal 和 S. Rahimi，“Senticr:用于代码审查交互的定制情感分析工具”，2017 年第 32 届 IEEE/ACM 自动化软件工程国际会议，ASE 2017，第106-111 页。

[16] Akiko Aizawa. 2003。TF-IDF 测度的信息论视角。信息处理与管理，39(2003)，45-65。

[17]杰罗姆·H·弗里德曼。2002。随机梯度增强。计算统计与数据分析38,4(2002)，367-378。

[10] N. Novielli, D. Girardi 和 F. Lanubile，“情感分析在软件工程研究中的基准研究”，2018 年 IEEE/ACM 第 15 届挖掘软件存储库国际会议，MSR 2018, pp. 364-375。

[19] E. Guzman 和 B. Bruegge，“迈向软件开发团队中的情感意识”，载于2013 年第 9 届软件工程基础联合会议论文集，ESEC/FSE 2013, pp. 671-674。

[20] R. Jongeling, P. Sarkar, S. Datta 和 A. Serebrenik，“在软件工程研究中使用情感分析工具时的负面结果”，Empir。Softw。Eng。，vol. 22, no。5, pp. 2543-2584，2017。

[21] M. Mantyla, B. Adams, G. Destefanis, D. Graziotin 和 M. Ortu，“挖掘效价，唤醒和支配:检测倦怠和生产力的可能性?“第 13 届挖掘软件存储库国际会议论文集”，MSR 2016，第247-258 页。

[22] M. Ortu, B. Adams, G. Destefanis, P. Tourani, M. Marchesi 和 R. Tonelli，“恃强欺弱的人更有生产力吗?”情感与问题修复时间的实证研究”，2015 年 IEEE/ACM 第 12 届挖掘软件存储库工作会议，MSR 2015，第303-313 页。

[23] M. Ortu, G. Destefanis, S. Counsell, S. Swift, R. Tonelli 和 M. Marchesi。“纵火犯还是消防员?”敏捷软件开发中的情感”，敏捷软件开发国际会议，XP 2016，第 144-155 页。

[24] M. Ortu、A. Murgia、G. Destefanis、P. Tourani、R. Tonelli、M. Marchesi 和 B. Adams，“JIRA 中软件开发人员的情感面”，2016 年IEEE/ACM 第 13 届挖掘软件存储库工作会议，MSR 2016，第 480-483 页。

[25] S. Panichella、A. Di Sorbo、E. Guzman、C. A. Visaggio、G. Canfora 和 H. C. Gall，《我如何改进我的应用程序?》分类用户评论用于软件维护和进化”，2015 年 IEEE 软件维护和进化国际会议，ICSME 2015, pp. 281-290。

[10] R. Souza 和 B. Silva，“travis CI 构建的情感分析”，2017 年 IEEE/ACM 第十四届挖掘软件存储库国际会议，MSR 2017，第 459-462 页。

[27] J. Cheruvelil 和 b.c. da Silva，“开发人员的情绪和问题重新开放”，2019 年 IEEE/ACM 第四届软件工程情感意识国际研讨会，SEmotion 2019，第29-33 页。

[28] N. Novielli, F. Calefato, D. Dongiovanni, D. Girardi 和 F. Lanubile，“我们可以在跨平台设置中使用特定于 se 的情感分析工具吗?”“第 17 届国际采矿软件存储会议论文集，MSR 2020，第 158-168 页。”

[29] D. Pletea, B. Vasilescu 和 A. Serebrenik，“安全性与情感:github 上安全性讨论的情感分析”，载于第 11 届挖掘软件存储库工作会议论文集，MSR 2014，第 348-351 页。

[30] E. Guzman、D. Azocar 和 Y. Li，“github 中提交评论的情感分析:一项实证研究”，载于第 11 届挖掘软件库工作会议论文集，MSR 2014，第352-355 页。

[31] V. Sinha, A. Lazar 和 B. Sharif，“分析提交日志中的开发人员情绪”，载于第 13 届挖掘软件库国际会议论文集，MSR 2016，第 520-523 页。

[32] F. Calefato, F. Lanubile 和 N. Novielli，“如何寻求技术帮助?”关于堆栈溢出编写问题的循证指南，《信息与软件技术》，第 94 卷，第 186-207 页，2018。

[33] W. Maalej, Z. Kurtanovic, H. Nabil 和 C. Stanik，“关于应用程序评论的自动分类”，《需求工程》，第 21 卷，第 2 期。3, pp. 311-331, 2016。

[34]王磊，顾涛，刘爱贤，姚辉，陶晓涛，陆杰，“基于内置传感器的智能手机应用的用户心理负荷评估”，IEEE 普适计算。，vol. 18, no. 6。1，第59-70 页，2019。

[35] E. Guzman 和 W. Maalej，“用户如何喜欢这个功能?应用评论的细粒度情感分析”，2014 年 IEEE 第 22 届国际需求工程会议，RE 2014, pp. 153-162。

[36] B. Lin, F. Zampetti, G. Bavota, M. Di Penta, M. Lanza 和 R. Oliveto，“软件工程的情感分析:我们能走多远?《第 40 届国际软件工程会议论文集》，ICSE 2018, pp. 94-104。

[37] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard 和 D. McClosky，“The stanford corenlp 自然语言处理工具包”，载于计算语言学协会第 52 届年会论文集:系统演示，ACL 2014，第 55-60 页。

[38]陈振中，Cao Y.，Lu X.，Q. Mei 和 X. Liu，“Sentimoji:一种用于软件工程中情感分析的表情符号驱动学习方法”，载于 2019 年第 27 届 ACM 欧洲软件工程会议联合会议和软件工程基础研讨会论文集，ESEC/FSE 2019，第 841-852 页。

[39] S. Bird，“Nltk:自然语言工具包”，载于 COL- ING/ACL 2006 年互动演示会议论文集，2006 年，第 69-72 页。

[40] S. A. Chowdhury 和 A. Hindle，“表征能源感知软件项目:它们不同吗?”，《第 13 届采矿软件存储库国际会议论文集》，MSR 2016, pp. 508-511。

[41] P. Tourani 和 B. Adams，“人类讨论对即时质量保证的影响:openstack 和 eclipse 的实证研究”，2016 年 IEEE 第 23 届软件分析、进化和再工程国际会议，SANER 2016，第 189-200 页。

b[42] W. Weimer，“当你阅读和理解代码时，你的大脑在想什么?，《第 27 届国际程序理解会议论文集》，ICPC 2019，第 1 页。

[43] 会话。基于句子结构的软件工程情感分析工具。https://github.com/huiAlex/SESSION，最后访问时间:2021 年 3 月。

[44] N. Ali, H. Cai, A. Hamou-Lhadj 和 J. Hassine，“利用词性来实现有效的自动化需求可追溯性”，Inf. software。抛光工艺。，vol. 106, pp. 126-141, 2019。

[45] D. Bollegala, D. J. Weir, and J. A. Carroll, “Using multiple sources to construct a sentiment sensitive thesaurus for cross-domain sentiment classification”, in The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, ACL 2011, pp. 132-141.