软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, [doi: 10.13328/j.cnki.jos.000000]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

面向软件工程的情感分析技术研究*

陈震鹏, 姚惠涵, 曹雁彬, 刘譞哲, 梅 宏

(北京大学 信息科学技术学院计算机科学与技术系 高可信软件技术教育部重点实验室, 北京 100871)

通讯作者: 刘譞哲, E-mail: xzl@pku.edu.cn

摘 要: 情感分析在软件工程领域具有广泛的应用场景,例如,从代码提交信息中检测开发者的情绪、从程序员问答论坛中识别开发者的观点等.但是,现有的“开箱即用”的情感分析工具无法在软件工程相关的任务中取得可靠的结果.已有研究表明,导致不可靠结果的最主要原因是,这些工具无法理解一些单词和短语在软件工程领域中的特定含义.此后,研究者们开始为软件工程领域定制监督学习和远程监督学习方法.为了验证这些方法的效果,研究者们使用软件工程相关的标注数据集来对它们进行数据集内验证,即,将同一数据集划分为训练集和测试集,分别用于方法的训练和测试.但是,对软件工程领域的某些情感分析任务来说,尚无标注数据集,且人工标注数据集耗时耗力.在此情况下,一种可选的方法就是使用为了相似任务从同一目标平台上提取的数据集或者使用从其他软件工程平台上提取的数据集.为了验证这两种做法的可行性,我们需要进一步以平台内设置和跨平台设置来验证现有情感分析方法.平台内设置指的是使用提取自同一平台的不同数据集作为训练集和测试集;跨平台设置指的是使用提取自不同平台的数据集作为训练集和测试集.本文旨在数据集内设置、平台内设置、跨平台设置这三种设置下,综合验证现有的为软件工程定制的情感分析方法.最终,我们的实验结果为相关的研究者和从业者提供了具有现实指导意义的启示.

关键词: 情感分析;软件工程;数据集内设置;平台内设置;跨平台设置

中图法分类号: TP311

中文引用格式: 陈震鹏,姚惠涵,曹雁彬,刘譞哲,梅宏.面向软件工程的情感分析技术研究.软件学报. http://www.jos.org.cn/1000-9825/ 0000.htm

英文引用格式: Chen ZP, Yao HH, Cao YB, Liu XZ, Mei H. Research on sentiment analysis in software engineering. Ruan Jian Xue Bao/Journal of Software, 2021 (in Chinese). http://www.jos.org.cn/1000-9825/0000.htm

Research on Sentiment Analysis in Software Engineering

CHEN Zhenpeng, YAO Huihan, CAO Yanbin, LIU Xuanzhe, MEI Hong

(Key Laboratory of High Confidence Software Technologies, Peking University, Beijing 100871, China)

Abstract: Sentiment analysis has various application scenarios in software engineering (SE), such as detecting developers' emotions in commit messages and identifying developers' opinions on Q&A forums. However, commonly used out-of-box sentiment analysis tools cannot obtain reliable results in SE tasks and misunderstanding of technical knowledge is demonstrated to be the main reason. Then researchers start to customize SE-specific methods in supervised or distantly supervised ways. To assess the performance of these methods, researchers use SE-related annotated datasets to evaluate them in a within-dataset setting, that is, they train and test each method using data from the same dataset. However, the annotated dataset for an SE-specific sentiment analysis task is not always available. Moreover, building a manually annotated dataset is time-consuming and not always feasible. An alternative is to use datasets extracted from the same platform for similar tasks or datasets extracted from other SE platforms. To verify the feasibility of these practices, we need to evaluate existing methods in within-platform and cross-platform settings, which refer to training and testing each method using data from the same platform but not

* 基金项目: 北大百度基金资助项目(2020BD007)
收稿时间: 0000-00-00; 修改时间: 0000-00-00; 采用时间: 0000-00-00; jos 在线出版时间: 0000-00-00
CNKI 在线出版时间: 0000-00-00

the same dataset, and training and testing each classifier using data from different platforms. In this paper, we comprehensively evaluate existing SE-customized sentiment analysis methods in within-dataset, within-platform, and cross-platform settings. Finally, our experimental results provide actionable insights for both researchers and practitioners.

Key words: sentiment analysis; software engineering; within-dataset setting; within-platform setting; cross-platform setting

软件开发作为软件工程生命周期中最重要的环节之一,是一项高度协作的活动,容易受到开发人员的情感状态的影响[1][2].情感是指由一种感觉引起的态度、想法或判断[3],通常被认为具有三种极性,即,正面、负面和中立[4][5].负面的情感状态会使开发人员在软件项目中的工作效率变低[2][6]且易引入软件缺陷[7],而正面的情感状态则可以提高开发人员的生产力[8].因此,明晰开发人员的情感状态,对于软件生命周期中涉及的利益相关者来说至关重要.为了这一目的,研究者提出了问卷调查[9]、生物特征测量[10]和文本分析[11]等多种方法,用于识别开发人员的情感极性.

在这些方法中,基于开发者的文本通信记录的情感分析变得越来越流行[12],该方法具有高便利、低成本等诸多优势.许多开箱即用的基于词典的情感分析工具(例如 SentiStrength[13]等)已被广泛应用于软件工程领域的研究中.但是,最近的研究工作[14]表明,这些通用的工具无法在软件工程领域的任务中提供可靠的结果.进一步地,Islam 和 Zibran[15]发现,导致不可靠结果的最主要原因是,通用的情感分析工具无法理解一些单词和短语在软件工程领域中的特定含义.这一结果启发研究者针对软件工程领域设计特定的情感分析方法,并引发了近年来的一系列研究工作.具体而言,研究者们开始创建与软件工程相关的标注数据集,并使用机器学习或深度学习的算法,训练得到软件工程领域定制的情感分析方法[12][16][17][18].为了评价这些方法的效果,Novielli 等[19]和 Chen 等[18][20]在现有的基准数据集上对各方法进行了测试.Chen 等[18][20]的实验结果表明其提出的远程监督学习方法 SEntMoji,在各基准数据集上表现优于目前已知的方法.

现有研究[12][16][17][18][19]对于情感分析方法的测试主要采用的是数据集内(within-dataset)设置的方式,即测试数据和训练数据来自同一数据集.这种测试方式是不全面的.下面,我们结合在软件工程领域的平台上开展情感分析的两类实际应用场景具体阐述:

- **在有基准数据集的平台上开展情感分析**.如果我们需要在已存在基准数据集的平台上开展情感分析,一个直观的做法就是利用现有的该平台的基准数据集训练得到情感分类器.但是,基准数据集的数据分布、标注初衷等与实际应用场景也许存在差异,导致了应用效果的不确定性.例如,Lin 等[12]收集了 Stack Overflow 上关于 Java API 的若干文本,标注了情感极性,得到了 Stack Overflow 的一款基准数据集.但是,如果我们的应用场景并不局限于分析 Java API 相关的文本,而是分析 Stack Overflow 上各类文本的情感,那么基于 Lin 等[12]提供的基准数据集训练得到的情感分类器的表现效果则需要进一步验证.为了验证这一方法的可行性,本文拟将数据集内设置的测试方式扩展到平台内(within-platform)设置的测试方式,即测试数据和训练数据来自同一平台,但未必来自同一数据集.
- **在无基准数据集的平台上开展情感分析**.如果我们需要在尚不存在基准数据集的平台上开展情感分析,有两种可能的方法:一是从该平台上收集数据并人工标注,然后利用标注数据训练得到情感分类器;二是使用现有的软件工程领域的基准数据集训练得到情感分类器.基于目前数据集内验证的结果,我们可以推断,第一种方法可以取得较为满意的效果.但是,考虑到软件工程领域中平台众多,第一种方法耗时耗力.为了验证第二种方法的可行性,本文拟引入跨平台(cross-platform)设置的测试方式,即测试数据和训练数据来自不同平台,对软件工程领域现有的情感分析方法进行综合的测试.

针对上述应用场景,本文拟综合开展软件工程领域情感分析的数据集内验证、平台内验证以及跨平台验证,从而为研究者和从业者提供具有实际指导意义的启示.具体而言,本文采用了五种现有方法和四个基准数据集进行验证.结果表明,现有的监督学习和远程监督学习方法在平台内设置和跨平台设置下的效果明显差于数据集内设置的效果.在数据集内设置下,SEntiMoji 显著优于监督学习方法和基于词典的方法;在平台内设置和跨平台设置下,SEntiMoji 显著优于监督学习方法,但是,与基于词典的方法相比,无明显优势.研究者和从业者可以根据我们的实验结果在不同的应用场景下选择最优的方式开展情感分析任务.

本文的组织方式如下.本文第 1 节对软件工程领域的情感分析的背景和相关研究进行总结.第 2 节介绍了本文所涵盖的现有情感分析方法、基准数据集以及实验设置.第 3 节基于实验结果回答了本文提出的两个研究问题.第 4 节讨论了本文的结果对研究者和从业者的启示,以及本文的局限性.最后,第 5 节总结全文.

1 研究背景及相关工作

情感分析的主要目标是分析文本中是否蕴含了正面或负面的情感,其在软件工程领域具有较为广泛的应用.一方面,利用情感分析技术可以分析开发者对于软件制品(例如 API 等)的看法,以此辅助软件工程中的推荐系统的建设.例如,Uddin 和 Khomh[21]以及 Lin 等[12]通过分析 Stack Overflow 上开发者对于不同 API 的情感态度,以此作为 API 推荐系统的推荐依据.另一方面,利用情感分析技术可以感知开发者的情感状况,以便研究者探究开发者情感状况与软件开发过程的相互影响.例如,Guzman 等[22]探究了时间因素与开发者情感状况的关联性;Ortu 等[2]探究了问题解决时间长短与开发者情感状况的关联性.此类工作利用情感分析技术帮助研究者更好地理解开发者的情感及其行为模式,从而帮助软件开发过程中有针对性地提高开发者的开发效率.

在实际应用过程中,大多数软件工程研究者直接使用了现成的情感分析工具,例如 SentiStrength[13]、NLTK[23]和 Standford NLP[24].这些工具大多基于社交媒体数据、商品评论数据等非技术文本发展得到,但由于其开箱即用的便利性,深受软件工程研究者的青睐.其中,SentiStrength 被认为是软件工程领域的研究中使用最为广泛的情感分析工具[1][22][25][26][27][28][29].但是,许多研究者[14][30]发现,直接将这些现成的工具应用到软件工程领域的任务中无法取得可靠的结果.例如,Jongeling 等[14]发现不同的工具在软件工程领域的数据集上表现不一致,一些现有的研究工作中如果换一种情感分析工具,此前得出的结论将不再成立.为了探究这些工具无法取得可靠结果的原因,Islam 和 Zibran[15]将流行的 SentiStrength 应用于软件工程相关的数据集上,然后定性分析被错误分类的样本,归纳出了 12 种错误原因.其中,缺乏软件工程的领域知识被证明是最主要的原因,81%的错误分类样本均可归咎于它.

自此,如何引入领域知识成了软件工程领域中情感分析研究的主要方向,一系列定制化的情感分析方法被陆续提出.Islam 和 Zibran[15]基于 SentiStrength 进行改进,提出了一种新的基于词典的软件工程定制化的情感分析工具 SentiStrength-SE.除此以外,研究者们还提出了 SentiCR[16]、Senti4SD[17]等监督学习方法.这些方法基于标注语料提取特征,并使用现有的机器学习算法进行训练,得到情感分类器,效果被证明优于基于词典的方法[19].但是,标注语料的创建需要人工地对每条文本进行分析,以确定最终的情感标签.其过程耗时耗力,因此标注语料的数据量较小.Chen 等[18][20]认为标注语料的数据量限制了监督学习方法的表现,Lin 等[12]和 Chen 等[18][20]都发现这些监督学习方法在一些软件工程相关的数据集上表现不令人满意,尤其是在数据分布很不均衡的数据集上.为了解决标注语料稀缺的问题,Chen 等[18][20]使用表情符号作为情感标签的代用品,使用大量的包含表情符号的 Twitter 和 GitHub 文本来补充人工标注语料的不足,并基于此提出了一种远程监督学习方法 SEntiMoji.实验证明,SEntiMoji 在各现有的基准数据集上均胜过监督学习方法,达到了最优水平.

为了验证软件工程定制化的情感分析方法的效果,Bin 等[12]、Novielli 等[19]、Chen 等[18]均采用了数据集内设置的基准研究,即,对于每一种基准数据集,都将其划分为训练集和测试集,利用训练集来训练各方法,得到不同的情感分类器,在测试集上进行测试,然后比较各方法的表现.现有方法在数据集内设置的基准研究表现较好,这启示我们,一种可行的方法是,对于软件工程中的情感分析任务,按照任务目标去对应平台上收集文本并标注情感极性,然后基于标注数据训练得到情感分析器.但是,在实际应用中,对每个平台中每种类型的文本都建立一个标注数据集(即基准数据集)是耗时耗力的.在此情况下,一种备选方案是选用现有的软件工程领域中的基准数据集训练得到情感分析器进行应用.但是,此方案的可行性需要进一步的实验结果进行验证.为了这一目的,研究者[31]对面向软件工程中的情感分析的跨平台设置进行了一些初步探究.但是,现有研究具有以下三点局限性:首先,只覆盖了基于词典的方法以及监督学习方法,未对更先进的方法(即远程监督学习方法 SEntiMoji)进行验证;其次,只在分布较为均衡的数据集上进行了验证.考虑到软件工程平台中真实的数据分布大多极其不均衡,即中立文本远远多于正面和负面文本,只在均衡数据集上进行验证不符合真实应用场景,结果具有片面性;最后,只涉

及了跨平台设置,并以数据集内设置作为基准进行比较,并未考虑一般意义的平台内设置.为了弥补现有研究的不足,本文对软件工程中的情感分析方法进行了综合的数据集内设置、平台内设置、跨平台设置基准研究,涵盖了基于词典的方法、监督学习方法、远程监督学习方法,考虑了分布均衡和不均衡的多种数据集.

2 实验设计

本节我们将介绍软件工程领域中主流的情感分析方法(第 2.1 节)、基准数据集(第 2.2 节)、评价各情感分析方法的指标(第 2.3 节)以及本文的实验设置(第 2.4 节).

2.1 现有方法

本文采用了三类情感分析方法进行验证,具体而言,包括基于词典的方法、监督学习方法和远程监督学习方法.其中,基于词典的方法包含 SentiStrength 和 SentiStrength-SE,监督式情感分析方法包含 SentiCR 和 Senti4SD,远程监督式情感分析方法包含 SEntiMoji.所选的五种方法中,SentiStrength 并不是针对软件工程领域提出的,但是却被证明在软件工程领域现有研究中最为流行[15];其余四种方法均为近年来研究者面向软件工程提出的领域特定的方法.下面,我们将简要地介绍每种方法.

- **SentiStrength**[13]是一款开箱即用的情感分类工具,它是基于日常英语文本,而非软件工程领域中普遍存在的科技文本,而学习得到的.具体而言,SentiStrength 包含了一个内置的词典,词典中的词和短语被赋予了不同的情感强度.对于每一段输入文本,SentiStrength 可以根据该文本在其内置词典中的覆盖程度,来计算出一个正向情感分数和一个负向情感分数.基于这两者之和,报告出一个三元的分数,即,1(正面)、0(中立)、-1(负面).
- **SentiStrength-SE**[15]是一款改编自 SentiStrength 的软件工程领域特定的情感分类工具.具体而言,其作者首先将 SentiStrength 应用于 JIRA 数据集(一个软件工程领域的数据集,在第 2.2 节中会具体介绍)的 Group-1 数据上,接着分析错误样本及错误原因,并通过调整 SentiStrength 的内置字典来引入软件工程领域特定的知识,以尽可能多地解决识别出的错误,最终得到了 SentiStrength-SE.
- **SentiCR**[16]是一款监督学习式情感分析方法,起初被提出用于分析代码评审(code reviews)的情感极性.其使用词袋(bag-of-words)的词频-逆文档频率(TF-IDF)[32]作为特征,使用传统的机器学习算法用于训练情感分类器.为了复现 SentiCR,本文使用其作者推荐的梯度提升树算法[33]作为训练算法.
- **Senti4SD**[17]是一款分析软件开发者交流文本的监督学习式情感分类方法.它使用了三类特征用于情感分析,包括词典特征(基于 SentiStrength 的内置词典计算得到)、关键词特征(例如 uni-grams 和 bi-grams 特征)以及语义特征(基于大规模 Stack Overflow 文本训练的词向量计算得到).最后,它使用了 SVM 算法[34]来训练得到情感分类器.
- **SEntiMoji**[18][20]是一款针对软件工程领域提出的远程监督式学习情感分类方法.该方法包含两阶段,即表征学习阶段和训练阶段.在表征学习阶段,作者基于 Twitter 和 GitHub 上大量的包含表情符号的文本训练得到了一款表情符号预测模型 DeepMoji-SE.该模型可以将趋向于包含相同表情符号的文本表征成相似的向量,从而为文本的表征引入了情感信息.在训练阶段,作者将训练文本经由 DeepMoji-SE 表征成向量的形式,以其情感标签作为真实情况(ground-truth),训练得到最终的情感分类器.

2.2 基准数据集

本文采用了四种基准数据集进行验证,具体而言,包括 JIRA 数据集、SO1 数据集、SO2 数据集、GitHub 数据集.这四种数据集涵盖了软件工程领域中三种典型的平台,即问题追踪平台(JIRA 数据集)、问答平台(SO1 数据集和 SO2 数据集)、软件项目托管平台(GitHub 数据集).下面,我们将简要介绍每种数据集.

- **JIRA 数据集**[35]起初包含了 JIRA 问题追踪平台上提取的 5992 条问题评论,其作者将其分割为三个部分,即,Group-1 数据、Group-2 数据和 Group-3 数据.因为 SentiStrength-SE 是基于 Group-1 数据发展而来的,为了能够公平地将它和其他方法比较,我们将 Group-1 数据从 JIRA 数据集中去除.剩下的 Group-

2 和 Group-3 数据被标注了若干情绪标签,即,“love”、“joy”、“surprise”、“anger”、“sadness”、“fair”或“neutral”.与之前的工作[12][15][18][19][31]一致,我们将“love”和“joy”映射为正面情感,将“anger”、“sadness”和“fair”情绪映射为负面情感.由于“surprise”的情感二义性,我们将“surprise”标签从数据集中去除.对于 Group-2 数据,其作者给出了三个标注者对每条样本的情绪标注结果.如果一条样本被至少两个标注者标注了正面(或者负面)的情绪标签,我们将这条样本判定为正面(或者负面).在这样的标准之下,无法匹配任何一种情感标签的样本被移除.对于 Group-3 数据,其作者综合标注结果,直接给出了每条样本的最终情绪标签.在去除了 surprise 标签后,我们去除了没有情绪标签或者包含相反情感极性的情绪标签的样本.最终,JIRA 数据集包含 2573 条样本,其中 43%是正面样本,27%是中立样本,30%是负面样本.

- **SO1 数据集**[17]包含了 Stack Overflow 上提取的 4423 条文本,涵盖了该平台的四种文本类型,即,问题、回答、问题评论和回答评论.该数据集的数据源是 Stack Overflow dump 中 2008 年 7 月到 2015 年 9 月的文本全集.为了保证各类样本的均衡性,其作者基于 SentiStrength 对这些文本的情感检测结果,选取了 4800 条样本.接着,三名标注者为每条样本进行独立标注,被标注了相反情感极性的样本被去除,剩余的样本按照多数表决的策略确定情感标签.最终,SO1 数据集包含 4423 条样本,其中 35%是正面样本,38%是中立样本,27%是负面样本.
- **SO2 数据集**[12]包含了 Stack Overflow 上提取的 1500 条关于 API 的句子.每条句子被两名标注者标注了情感强度,其中-2 代表强负面,-1 代表弱负面,0 代表中立,1 代表弱正面,2 代表强正面.标注完成后,其作者解决标注冲突,并确定每条句子的情感标签.最终,SO2 数据集包含 1500 条样本,其中 9%是正面样本,79%是中立样本,12%是负面样本.区别于 SO1 数据集,SO2 数据集的作者并未刻意追求数据集中各类别样本均衡分布,使得 SO2 数据集与真实场景分布一致,即中立样本占绝大多数.
- **GitHub 数据集**[31]包含了 GitHub 上提取的 7122 条拉请求(pull requst)和提交注释(commit comment).该数据集的数据源是 Pletea 等[36]提取的 GitHub 文本,其作者按照如下步骤构造数据集:首先,遵循 SO1 数据集的标注方法,人工标注了 3931 条样本的情感极性;接着,以这些标注样本作为训练数据,使用 Senti4SD 训练得到情感分类器,应用到剩余的 GitHub 文本中,并提取 600 条被预测为正面的样本和 600 条被预测为负面的样本;最后,对提取的 1200 条样本进行人工确认,将确认预测正确的 343 条正面样本和 550 条负面样本,加入到了原先的 3931 条标注样本中,继续训练得到新的情感分类器,并重复上述的操作,以便进一步扩大样本数目.最终,GitHub 数据集包含了 7122 条样本,其中 43%是正面样本,28%是中立样本,29%是负面样本.

我们将上述四种基准数据集的统计数据汇总在表 1 中.

表 1 本文所用的基准数据集

数据集	样本数目	情感极性(类别)		
		正面	中立	负面
JIRA	2573	1104(43%)	702(27%)	767(30%)
SO1	4423	1527(35%)	1694(38%)	1202(27%)
SO2	1500	131(9%)	1191(79%)	178(12%)
GitHub	7122	3022(43%)	2013(28%)	2087(29%)

2.3 评价指标

与现有工作[18]一致,对于每个情感分类器,我们采用其在每个情感类别上所取得的精确度(precision)、召回率(recall)、F1 值(F1-score)以及在整个数据集上所取得准确率(accuracy)作为评价指标.

- 精确度代表了一种方法的精确程度.对于一个给定的情感极性 c,其精确度 precision@c 是被正确预测且情感极性为 c 的样本数目与被预测情感极性为 c 的样本总数的比值.
- 召回率代表了一种方法的敏感程度.对于一个给定的情感极性 c,其召回率 recall@c 是被正确预测且

情感极性为 c 的样本数目与情感极性为 c 的样本总数的比值.

- F1 值是精确度和召回率的结合指标.对于一个给定的情感极性 *c*,其 F1 值 F1@c 是 precision@c 和 recall@c 的调和平均值.
- 准确率衡量了一种方法做出正确预测的频率,具体而言,指的是被预测正确的样本数目与样本总数的比值.

本文为了进行综合的比较,报告了每种方法在上述所有指标上的结果.但是,我们敦促研究人员和从业人员根据实际目标确定其应重点关注的指标.

2.4 实验设置

对于基于词典的方法(即 SentiStrength 和 SentiStrength-SE),因为它们是开箱即用的工具,无需进行训练,我们直接将其应用到基准数据集上,以得到在每条样本上的预测情感标签.

对于监督学习方法(即 SentiCR 和 Senti4SD)和远程监督学习方法(即 SEntiMoji),我们采用了数据集内设置、平台内设置和跨平台设置三种方式进行验证.对于数据集内验证,我们采用了五折交叉验证的方法.具体而言,对于每一个基准数据集,我们将其随机划分成五等份,每份数据依次作为测试集,以便测试每种方法五次.每次测试,我们使用剩余的四份数据作为训练集,使用训练集训练得到每种方法对应的情感分类器,并将其应用到测试集上,得到每种方法对每条数据的预测情感标签.对于平台内验证和跨平台验证,我们选取两个不同的基准数据集,一个作为训练集,一个作为测试集,使用训练集训练得到每种方法对应得到的情感分类器,并在测试集上验证.

本文使用了诸多指标来从多角度评价每种方法的效果,因此,很难基于不同方法在某一种指标上的效果来判断其优劣.为了检验不同方法的效果是否存在显著差异,遵循现有工作[18][20]的做法,本节使用了非参数的 McNemar 检验[37].经过五折交叉验证,每种方法都对每个样本输出了一个预测标签.为了使用 McNemar 检验来比较方法 A 和方法 B,需要获取被 A 分类错但是被 B 分类正确的样本数目和被 B 分类错误但是被 A 分类正确的样本数目,并据此计算统计值.两种方法的表现存在显著差异,当且仅当统计值对应的 *p*-value 小于预设的显著性水平(通常 1%或 5%).

3 研究问题和结果

按照第 2 节中的实验设置,我们得出了现有的基于词典的工具在各基准数据集上的表现(见表 2),以及现有的监督学习和远程监督学习方法在数据集内设置、平台内设置和跨平台设置下的表现(见表 3).根据表 2 和表 3 中的结果,我们将回答本文所提出的两个研究问题.

表 2 现有的基于词典的工具的表现

测试数据	类别	SentiStrength				SentiStrength-SE			
		P	R	F1	Acc.	P	R	F1	Acc.
JIRA	Positive	.847	.889	.867	.763	.936	.922	.929	.846
	Neutral	.615	.633	.624		.708	.844	.770	
	Negative	.777	.701	.737		.874	.739	.801	
SO1	Positive	.887	.927	.907	.815	.908	.823	.863	.791
	Neutral	.923	.632	.750		.726	.784	.754	
	Negative	.671	.930	.780		.754	.759	.757	
SO2	Positive	.200	.366	.259	.693	.312	.221	.259	.778
	Neutral	.858	.767	.810		.824	.929	.873	
	Negative	.395	.433	.413		.492	.180	.263	
GitHub	Positive	.695	.764	.728	.672	.851	.746	.795	.782
	Neutral	.677	.570	.619		.734	.855	.790	
	Negative	.647	.733	.687		.810	.712	.758	

3.1 研究问题1: 对于软件工程中的情感分析任务,平台内设置的方式效果如何?

由于现有的基准数据集中只有 SO1 和 SO2 数据集来自同一平台,因此我们将这两个数据集互为训练集和测试集的情况作为平台内设置.

首先,我们基于表 3 内的结果,以数据集内设置为基准,来评价平台内设置的效果.当以 SO1 数据集作为测试集,SentiCR、Senti4SD、SEntiMoji 在平台设置中的表现差于数据集内设置的.就准确率而言,三种方法在数据集内设置下取得的准确率比平台内设置下取得的准确率分别高 0.344、0.337、0.218.换言之,SentiCR、Senti4SD、SEntiMoji 在数据集内设置下比在平台内设置下可多正确分类 34.4%、33.7%、21.8%的样本.类似地,当以 SO2 数据集作为测试集,就准确率而言,SentiCR、Senti4SD、SEntiMoji 在平台内设置下的总体效果也差于数据集内设置下的效果.除了总体效果,平台内设置在一些特定情感的召回率上也出现明显下降.例如,SentiCR 在 SO2 数据集上进行数据集内测试时,正、负面样本的召回率分别为 0.321 和 0.596,是平台内测试所取得的正、负面样本召回率的 6 倍和 15 倍.总的来说,SO1 和 SO2 数据集的平台内设置效果不佳.究其原因,可能有两个方面:

- *两者的数据分布差异较大*:SO2 数据集的数据分布极其不均衡,正面、中立、负面样本分别占 9%、79%、12%,所以以 SO2 数据集作为训练集得到的情感分类器趋向于将测试样本预测为占比最大的类别(即中立),导致正面样本和负面样本的召回率偏低,进而影响了准确率等综合指标.
- *两者的标注目标不同*:SO1 和 SO2 数据集虽然均收集自 Stack Overflow,但是 SO1 数据集是对全局文本进行采样,而 SO2 数据集却涵盖关于 Java API 的文本.一方面,算法从 SO1 数据集中学习到的情感知识不一定能在 Java API 相关的这类特定文本(即 SO2 数据集)上表现较好;另一方面,算法从 SO2 数据集中学习到的情感知识并不一定能很好地泛化到一般类型的 Stack Overflow 文本(即 SO1 数据集)上.

接着,我们分析哪种学习方法在平台内设置下表现更优.在 SentiCR、Senti4SD、SEntiMoji 三种方法中,SEntiMoji 在平台内设置的表现最优.具体来讲,SO1 和 SO2 数据集互为训练集和测试集时,共两组实验,每组计算了 10 项指标,合计 20 项指标,其中,SEntiMoji 在 17 项指标上取得最高值.当 SO1 数据集为测试集时,SEntiMoji 的优势较为明显,其所取得的准确率比 SentiCR 和 Senti4SD 分别高 0.174 和 0.152,即可以比这两种方法多正确分类 17.4%和 15.2%的样本.McNemar 检验的结果显示,当 SO1 数据集为训练集,SO2 数据集为测试集时,SEntiMoji 在 5%的显著性水平下显著优于 SentiCR(p-value = 0.040);当 SO2 数据集为训练集,SO1 数据集为测试集时,SEntiMoji 在 1%的显著性水平下显著优于 SentiCR(p-value = 0.000)和 Senti4SD(p-value = 0.000).此外,我们注意到 Senti4SD 在 20 项指标中的 16 项上超过 SentiCR,这一结果可以归因于 Senti4SD 基于大规模 Stack Overflow 文本训练的词向量计算得到语义特征.具体而言,SentiCR 仅从少量训练数据中学习到 Stack Overflow 知识,而 Senti4SD 同时从大规模 Stack Overflow 文本和少量训练数据中学习到相应的知识,因而 Senti4SD 在 Stack Overflow 相关的测试数据上分类效果更好.

最后,我们综合表 2 和表 3 内的结果,分析在平台内设置下 SEntiMoji 的效果是否优于直接使用基于词典的方法.当 SO1 数据集为测试集时,SentiStrength、SentiStrength-SE 和 SEntiMoji 所取得的准确率分别为 0.815、0.791 和 0.656.SentiStrength 可比 SEntiMoji 多正确分类 15.9%的样本,且在 10 项指标中的 6 项上取得了最高值.McNemar 检验的结果显示,SentiStrength 在 1%的显著性水平下显著优于 SEntiMoji(p-value = 0.000).当 SO2 数据集为测试集时,SentiStrength、SentiStrength-SE 和 SEntiMoji 所取得的准确率分别为 0.693、0.778 和 0.798.虽然 McNemar 检验的结果显示 SEntiMoji 与 SentiStrength 和 SentiStrength-SE 之间的表现差异在 5%的显著性水平下显著(SentiStrength: p-value = 0.000, SentiStrength-SE: p-value = 0.015),但是 SEntiMoji 的效果优势并不明显.具体而言,SentiStrength 和 SEntiMoji 分别在 5 项和 5 项指标上取得最高值.综合两种数据集作为测试集时的结果,我们发现,在平台内设置下,SEntiMoji 无法持续取得显著优于基于词典的方法的效果.

表 3 现有的监督学习和远程监督学习方法的表现

测试数据	训练数据	类别	SentiCR				Senti4SD				SEntiMoji			
			P	R	F1	Acc.	P	R	F1	Acc.	P	R	F1	Acc.
JIRA	JIRA	Positive	.950	.919	.934	.872	.881	.921	.901	.832	.947	.945	.946	.904
		Neutral	.734	.906	.811		.744	.733	.738		.822	.879	.849	
		Negative	.932	.773	.845		.837	.794	.815		.923	.866	.894	
	SO1	Positive	.872	.492	.629	.477	.877	.724	.793	.609	.934	.915	.924	.827
		Neutral	.347	.923	.504		.423	.822	.559		.671	.844	.747	
		Negative	.461	.046	.084		.644	.246	.356		.866	.685	.765	
	SO2	Positive	.704	.220	.335	.351	.623	.194	.296	.390	.904	.230	.367	.422
		Neutral	.257	.633	.366		.302	.768	.433		.307	.853	.451	
		Negative	.435	.282	.342		.568	.326	.415		.695	.304	.423	
	GitHub	Positive	.924	.897	.910	.791	.901	.824	.861	.761	.759	.952	.845	.790
		Neutral	.623	.856	.721		.577	.902	.704		.759	.600	.670	
		Negative	.829	.576	.681		.888	.539	.670		.885	.729	.799	
SO1	JIRA	Positive	.952	.509	.663	.571	.852	.908	.879	.755	.918	.861	.888	.733
		Neutral	.484	.899	.630		.794	.597	.682		.640	.753	.692	
		Negative	.484	.187	.270		.618	.783	.691		.656	.544	.595	
	SO1	Positive	.869	.921	.894	.826	.904	.916	.910	.841	.932	.941	.936	.874
		Neutral	.784	.839	.811		.832	.773	.801		.841	.843	.842	
		Negative	.834	.688	.754		.779	.843	.810		.846	.832	.839	
	SO2	Positive	.629	.332	.434	.482	.528	.151	.235	.504	.906	.445	.597	.656
		Neutral	.456	.658	.539		.457	.764	.572		.538	.848	.659	
		Negative	.433	.424	.428		.610	.585	.597		.781	.653	.711	
	GitHub	Positive	.892	.778	.831	.726	.920	.855	.886	.793	.921	.868	.894	.827
		Neutral	.666	.832	.740		.741	.790	.765		.760	.836	.796	
		Negative	.632	.512	.565		.721	.718	.719		.817	.762	.788	
SO2	JIRA	Positive	.389	.107	.168	.773	.185	.389	.251	.635	.325	.282	.302	.711
		Neutral	.819	.931	.872		.872	.683	.766		.852	.787	.818	
		Negative	.327	.202	.250		.299	.489	.371		.325	.522	.401	
	SO1	Positive	.212	.053	.085	.783	.368	.214	.271	.789	.342	.206	.257	.798
		Neutral	.798	.974	.877		.816	.958	.881		.827	.945	.882	
		Negative	.538	.039	.073		.560	.079	.138		.738	.253	.377	
	SO2	Positive	.538	.321	.402	.821	.464	.198	.278	.807	.843	.328	.473	.863
		Neutral	.883	.910	.897		.860	.925	.891		.880	.964	.920	
		Negative	.544	.596	.568		.506	.461	.482		.717	.584	.644	
	GitHub	Positive	.327	.130	.186	.760	.333	.244	.282	.779	.295	.198	.237	.783
		Neutral	.803	.934	.864		.820	.938	.875		.821	.935	.874	
		Negative	.172	.062	.091		.476	.112	.182		.632	.202	.306	
GitHub	JIRA	Positive	.901	.599	.720	.664	.695	.766	.729	.691	.852	.730	.786	.750
		Neutral	.585	.914	.714		.783	.597	.677		.725	.834	.775	
		Negative	.713	.363	.481		.605	.754	.672		.705	.648	.675	
	SO1	Positive	.856	.616	.716	.678	.816	.639	.717	.707	.862	.672	.755	.773
		Neutral	.594	.942	.728		.636	.903	.746		.707	.893	.789	
		Negative	.845	.356	.501		.814	.490	.612		.836	.696	.760	
	SO2	Positive	.493	.171	.254	.407	.410	.124	.191	.490	.775	.243	.371	.568
		Neutral	.419	.604	.495		.486	.794	.603		.505	.863	.637	
		Negative	.353	.351	.352		.537	.404	.461		.712	.454	.555	
	GitHub	Positive	.891	.820	.854	.824	.916	.858	.886	.879	.928	.883	.905	.887
		Neutral	.769	.917	.837		.856	.903	.879		.853	.924	.887	
		Negative	.867	.692	.770		.881	.865	.873		.904	.837	.869	

3.2 研究问题2: 对于软件工程中的情感分析任务,跨平台设置的效果如何?

在表 3 所展示的 16 组实验中(四个测试集,每个测试集上应用四种训练集),有 10 组实验为跨平台设置,即训练集和测试集来自不同的平台.

首先,我们基于表 3 内的结果,以数据集内设置为基准,来评价跨平台设置的效果.在四种基准数据集上,数据集内设置取得效果总体均优于跨平台设置的效果.以 JIRA 数据集为例,SentiCR、Senti4SD、SEntiMoji 在数据集内设置下所取得的准确率分别为 0.872、0.832、0.904;可是当以其他数据集作为训练集时,这三种方法所能取

得的最高准确率分别是 0.791、0.761、0.827,比数据集内设置的结果低了 0.081、0.071、0.077.换言之,跨平台设置下,SentiCR、Senti4SD、SEntiMoji 在 JIRA 数据集上至少比数据集内设置下分别少正确分类 8.1%、7.1%和 7.7%的样本.此外,我们发现,数据集的类别分布情况也会对跨平台设置的结果产生影响.具体而言,当跨平台设置中的训练集和测试集的类别分布差异较大时,所取得的效果将明显变差.如第 2.2 节所述,SO2 数据集区别于其他数据集,其作者并未刻意追求数据集中各类样本均衡分布,使得其数据分布与真实场景分布一致,即中立样本占绝大多数.因此,基于 SO2 数据集训练得到的情感分类器学习了大量中立样本的知识,倾向于将测试样本分类为中立.这导致了当以 SO2 数据集作为训练集开展跨平台验证时,在 JIRA 数据集和 GitHub 数据集中的中立样本上取得的精确度较低,进一步导致整体的准确率较低.例如,当 JIRA 数据集作为测试集时,SEntiMoji 以 SO2 数据集、SO1 数据集、GitHub 数据集作为训练集,在中立样本上所取得的精度分别为 0.307、0.671 和 0.759,取得的整体准确率分别为 0.422、0.827、0.790.可以发现,虽然均为跨平台设置,SO1 数据集和 GitHub 数据集作为训练集时在 JIRA 数据集上的表现明显优于 SO2 数据集作为训练集时的表现.

接着,我们分析哪种学习方法在跨平台设置下表现最优.在 10 组跨平台设置的实验中,McNemar 检验的结果显示,SEntiMoji 可以在 6 组实验中在 1%的显著性水平下显著优于 SentiCR 和 Senti4SD;就准确率而言,SEntiMoji 可以在 7 组实验中表现优于 SentiCR 和 Senti4SD.例如,当以 JIRA 数据集为测试集,SO1 数据集为训练集时,SEntiMoji 所取得的准确率比 SentiCR 和 Senti4SD 分别高了 0.350 和 0.218,即 SEntiMoji 可以比 SentiCR 和 Senti4SD 分别多分类正确 35.0%和 21.8%的样本.除了准确率这一综合指标,在一些特定指标上,SEntiMoji 也表现出明显优势.例如,当以 JIRA 数据集为测试集,SO1 数据集为训练集时,SEntiMoji 在负面样本上所取得的精确度比 SentiCR 和 Senti4SD 分别高 0.405 和 0.222,所取得的召回率比 SentiCR 和 Senti4SD 分别高 0.639 和 0.439.考虑开发者负面情绪检测任务,由于 SEntiMoji 在负面样本上所取得的精度较高,当 SEntiMoji 检测出开发者出现负面情绪时,其可信度比其他方法要高;由于 SEntiMoji 在负面样本上所取得的召回率较高,SEntiMoji 将比其他方法更为及时地察觉到开发者的负面情绪.综上分析,在跨平台设置上,SEntiMoji 表现整体优于其他两种监督学习方法.此外,我们注意到,Senti4SD 相较于 SentiCR,总体表现更优,这与平台内设置中的结果一致.具体而言,在 10 组跨平台设置的实验中,以准确率而言,Senti4SD 在 8 组上取得了更优的结果,这进一步说明了 Senti4SD 基于大规模 Stack Overflow 文本学习到的软件工程领域的知识对于最终的情感分类具有一定的益处.

最后,我们综合表 2 和表 3 内的结果,分析在跨平台设置下 SEntiMoji 的效果是否优于直接使用基于词典的方法.当 JIRA 和 GitHub 数据集作为测试集时,SEntiMoji 在任何一种跨平台设置下所取得的准确率均低于 SentiStrength-SE.例如,在 JIRA 数据集上,SentiStrength-SE 取得的准确率为 0.846,而 SEntiMoji 在跨平台设置下所取得最高准确率为 0.827,最低准确率为 0.422.此外,McNemar 检验的结果显示,在 JIRA 和 GitHub 数据集作为测试集的 6 组跨平台设置实验中,SentiStrength-SE 在 5 组中可以在 1%的显著性水平下显著优于 SEntiMoji.当 SO1 数据集为测试集时,SEntiMoji 以 JIRA 和 GitHub 数据集为训练集所取得的准确率分别为 0.733 和 0.827,与 SentiStrength 所取得的 0.815 的准确率相比,无明显优势;McNemar 检验的结果也显示,当 JIRA 数据集为训练集时,SentiStrength 在 1%的显著性水平下显著优于 SEntiMoji(p-value = 0.000),当 GitHub 数据集为训练集时,SentiStrength 和 SEntiMoji 的表现无显著性差异(p-value = 0.090).当 SO2 数据集为测试集时,SEntiMoji 以 JIRA 和 GitHub 数据集为训练集所取得的准确率分别为 0.711 和 0.783, 与 SentiStrength-SE 所取得的 0.778 的准确率相比,无明显优势;McNemar 检验的结果也显示,当 JIRA 数据集为训练集时,SentiStrength-SE 在 1%的显著性水平下显著优于 SEntiMoji(p-value = 0.000),当 GitHub 数据集为训练集时,SentiStrength-SE 和 SEntiMoji 的表现无显著性差异(p-value = 0.568).综上分析,我们发现,在跨平台设置下,SEntiMoji 无法持续取得显著优于基于词典的方法的效果.

4 讨论

本节,我们将讨论本文的研究结果带来的启示和本文的局限性.

4.1 启示

本文在现有的基准数据集上综合测试了软件工程领域中各情感分析方法的效果,其结果为相关的研究者和从业者提供了具有实际指导意义的启示.

对于实际应用的启示.当研究者和从业者想针对某个尚不存在基准数据集的平台开展情感分析任务时,如果对准确率等效果要求较高,应根据任务应用场景去该平台上收集数据和标注情感极性,该过程中需注意保持数据分布与实际应用场景一致.然后,以标注数据作为训练数据,使用 SEntiMoji 算法训练得到所需的情感分类器.如果该任务对准确率等效果要求不高,需要省时省力,就直接使用 SentiStrength 或 SentiStrength-SE 这些基于词典的方法,无需使用复杂的远程监督学习算法基于已有其他平台的数据集训练得到情感分类器.当研究者和从业者想针对某个已存在基准数据集的平台开展情感分析任务时,需要检查已有基准数据集的标注目的、数据分布等与实际应用场景是否一致.如果一致,应使用 SEntiMoji 作为训练算法,以基准数据集为训练数据,训练得到情感分类器.如果不一致,相应的做法参照在尚不存在基准数据集的平台开展情感分析任务时的建议.

对于学术研究的启示.过往研究[18]结果表明,SEntiMoji 在数据集内设置下可在各基准数据集上达到最优水平,效果胜过现有的基于词典的方法以及监督学习方法.在平台内设置和跨平台设置下,本文发现 SEntiMoji 仍然优于现有的监督学习方法,可是相较于基于词典的方法,无明显优势.因此,我们期待研究者们继续致力于软件工程领域的定制化情感分析,研究出在数据集内设置、平台内设置、跨平台设置下都能取得较好效果的技术.同时,本文的结果表明,数据集内设置的结果并不可以直接推广到平台内设置和跨平台设置上.因此,考虑到多样的实际应用场景,未来研究在验证新提出的定制化情感分析方法时,需要综合开展数据集内设置、平台内设置和跨平台设置下的效果验证.另外,本文发现平台内设置下现有的监督学习、远程监督学习方法效果较差,并提出了可能的原因.但是,现有的基准数据集数目较少,我们无法系统地研究每种原因的真实影响.为此,我们号召研究者在未来的研究中,多标注和开源相关的基准数据集,以供开展进一步研究.

4.2 局限性

我们将从三个方面来总结研究的局限性,包括对构成有效性(construct validity)的威胁、对内部有效性(internal validity)的威胁、对外部有效性(external validity)的威胁.

对构成有效性的威胁.JIRA 数据集原先带有不同的情绪标签,而非情感标签.为了使用该数据集进行情感分析,我们遵循先前的研究[12][18][19][31],将情绪标签映射到三元情感极性标签,并过滤掉了歧义样本.此过程可能会影响该数据集的原始分布,并降低分类任务的难度,从而影响我们对不同方法的性能的评价.但是,幸运的是,我们采用了若干基准数据集进行了综合比较,并且在其他数据集上观察到的结论与在 JIRA 数据集上观察到的无较大差异.

对内部有效性的威胁.作为一项基准研究,我们需要复现若干现有方法,对这些现有方法的配置(例如超参数的选择、机器学习算法的选择等)进行改进,可能会改善它们的表现,从而影响本文的结果.但是,为了公平比较,本文直接采用了这些方法的作者发布的脚本、推荐的超参数配置、推荐的算法来复现它们.此外,评价一个情感分析方法表现的指标众多,研究者、从业者可能会根据自己的任务特点,设计、侧重不同的评价标准.由于对于评价指标的选取将影响本文的结论,为了减少该威胁的影响,本文合计选取了领域内最常用的 10 种指标.

对外部有效性的威胁.本文涵盖了软件工程领域情感分析任务中最常用的若干基准数据集.但是,考虑到本文所选用的数据集并不能涵盖软件工程领域所有类型的文本,我们仍然不能断言我们的结论具有绝对的普适性.另外,软件工程领域的研究中涉及过非常多的情感分析方法,在本文中,我们只选择一些具有代表性的方法进行验证.

5 总结与展望

本文对现有的软件工程定制化的情感分析方法进行了综合的数据集内设置、平台内设置和跨平台设置的验证.结果表明,现有的监督学习和远程监督学习方法在平台内设置和跨平台设置下的效果明显差于数据集内

设置的效果.在三种设置下,远程监督学习方法 SEntiMoji 表现显著优于现有的监督学习方法.虽然在数据集内设置下,SEntiMoji 被过往工作证明显著优于基于词典的方法,但是,在平台内设置和跨平台设置下,SEntiMoji 与基于词典的方法相比,无明显优势.我们的结果在实际应用和学术研究层面为从业者和研究者提供了具有实际指导意义的启示.一方面,从业者和研究者可以根据我们的实验结果在不同的应用场景下选择最优的方式开展情感分析任务;另一方面,我们的结果也鼓励相关研究者继续开展软件工程定制化的情感分析方法的研究,希冀未来出现一种能在数据集内设置、平台内设置和跨平台设置下均表现最优的方法.

References:

[1] Emitza Guzman and Bernd Bruegge. Towards emotional awareness in software development teams. In: Proceedings of the Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, ESEC/FSE. ACM, 2013. 671–674.

[2] Marco Ortu, Bram Adams, Giuseppe Destefanis, Parastou Tourani, Michele Marchesi, and Roberto Tonelli. Are bullies more productive? Empirical study of affectiveness vs. issue fixing time. In: Proceedings of the 12th IEEE/ACM Working Conference on Mining Software Repositories, MSR. IEEE, 2015. 303–313.

[3] Myriam Munezero, Calkin Suero Montero, Erkki Sutinen, and John Pajunen. Are they different? Affect, feeling, emotion, sentiment, and opinion detection in text. IEEE Transactions on Affective Computing, 2014, 5(2):101–111.

[4] Bing Liu. Sentiment analysis and opinion mining. Morgan & Claypool Publishers, 2012. 1-168.

[5] Yan-Yan Zhao, Bing Qin, and Ting Liu. Sentiment analysis. Ruan Jian Xue Bao/Journal of Software, 2010, 21(8): 1834-1848 (in Chinese with English abstract).

[6] David García, Marcelo Serrano Zanetti, and Frank Schweitzer. The role of emotions in contributors activity: A case study on the GENTOO community. In: Proceedings of the 2013 International Conference on Cloud and Green Computing, CGC. IEEE 2013, 410–417.

[7] Syed Fatiul Huq, Ali Zafar Sadiq, and Kazi Sakib. Is developer sentiment related to software bugs: An exploratory study on GitHub commits. In: Proceedings of the 27th IEEE International Conference on Software Analysis, Evolution and Reengineering, SANER. IEEE 2020, 527-531.

[8] Daniel Graziotin, Fabian Fagerholm, Xiaofeng Wang, and Pekka Abrahamsson. What happens when software developers are (un)happy. Journal of Systems and Software, 2018, 140: 32–47.

[9] Miikka Kuutila, Mika V. Mäntylä, Maëlick Claes, Marko Elovainio, and Bram Adams. Using experience sampling to link software repositories with emotions and work well-being. In: Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM. ACM 2018, 29:1–29:10.

[10] Daniela Girardi, Filippo Lanubile, Nicole Novielli, Luigi Quaranta, and Alexander Serebrenik. Towards recognizing the emotions of developers using biometrics: The design of a field study. In: Proceedings of the 4th International Workshop on Emotion Awareness in Software Engineering, SEmotion@ICSE. IEEE 2019, 13–16.

[11] Fabio Calefato, Filippo Lanubile, Nicole Novielli, and Luigi Quaranta. EMTk: The emotion mining toolkit. In: Proceedings of the 4th International Workshop on Emotion Awareness in Software Engineering, SEmotion@ICSE. IEEE 2019, 34–37.

[12] Bin Lin, Fiorella Zampetti, Gabriele Bavota, Massimiliano Di Penta, Michele Lanza, and Rocco Oliveto. Sentiment analysis for software engineering: How far can we go?. In: Proceedings of the 40th International Conference on Software Engineering, ICSE. ACM 2018, 94–104.

[13] Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. Sentiment in short strength detection informal text. Journal of the American Society for Information Science and Technology, 2010, 61(12):2544–2558.

[14] Robbert Jongeling, Proshanta Sarkar, Subhajit Datta, and Alexander Serebrenik. On negative results when using sentiment analysis tools for software engineering research. Empirical Software Engineering, 2017, 22(5):2543–2584.

[15] Md Rakibul Islam and Minhaz F. Zibran. Leveraging automated sentiment analysis in software engineering. In: Proceedings of the 14th International Conference on Mining Software Repositories, MSR. IEEE 2017, 203–214.

[16] Toufique Ahmed, Amiangshu Bosu, Anindya Iqbal, and Shahram Rahimi. SentiCR: A customized sentiment analysis tool for code review interactions. In: Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering, ASE. IEEE 2017, 106–111.

[17] Fabio Calefato, Filippo Lanubile, Federico Maiorano, and Nicole Novielli. Sentiment polarity detection for software development. Empirical Software Engineering, 2018, 23(3):1352–1382.

[18] Zhenpeng Chen, Yanbin Cao, Xuan Lu, Qiaozhu Mei, and Xuanzhe Liu. SEntiMoji: An emoji-powered learning approach for sentiment analysis in software engineering. In: Proceedings of the ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE. ACM 2019, 841–852.

[19] Nicole Novielli, Daniela Girardi, and Filippo Lanubile. A benchmark study on sentiment analysis for software engineering research. In: Proceedings of the 15th International Conference on Mining Software Repositories, MSR. ACM 2018, 364–375.

[20] Zhenpeng Chen, Yanbin Cao, Huihan Yao, Xuan Lu, Xin Peng, Hong Mei, and Xuanzhe Liu. Emoji-powered sentiment and emotion detection from software developers' communication data. ACM Transactions on Software Engineering and Methodology, 2021, 30(2):18:1-18:48.

[21] Gias Uddin and Foutse Khomh. Opiner: An opinion search and summarization engine for APIs. In: Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering, ASE. IEEE 2017, 978-983.

[22] Emitza Guzman, David Azócar, and Yang Li. Sentiment analysis of commit comments in GitHub: An empirical study. In: Proceedings of the 11th Working Conference on Mining Software Repositories, MSR. ACM 2014, 352–355.

[23] Steven Bird and Edward Loper. NLTK: The natural language toolkit. In: Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions. ACL 2004, 69-72.

[24] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. ACL 2014, 55–60.

[25] Shaiful Alam Chowdhury and Abram Hindle. Characterizing energy-aware software projects: Are they different?. In: Proceedings of the 13th International Conference on Mining Software Repositories, MSR. IEEE 2016, 508–511.

[26] David García, Marcelo Serrano Zanetti, and Frank Schweitzer. The role of emotions in contributors activity: A case study on the GENTOO community. In: Proceedings of 2013 International Conference on Cloud and Green Computing, CGC. IEEE 2013, 410–417.

[27] Emitza Guzman and Walid Maalej. How do users like this feature? A fine grained sentiment analysis of app reviews. In: Proceedings of the IEEE 22nd International Requirements Engineering Conference, RE. IEEE 2014, 153–162.

[28] Parastou Tourani and Bram Adams. The impact of human discussions on just-in-time quality assurance: An empirical study on OpenStack and Eclipse. In: Proceedings of the IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering, SANER. IEEE 2016, 189–200.

[29] Vinayak Sinha, Alina Lazar, and Bonita Sharif. Analyzing developer sentiment in commit logs. In: Proceedings of the 13th International Conference on Mining Software Repositories, MSR. ACM 2016, 520–523.

[30] Robbert Jongeling, Subhajit Datta, and Alexander Serebrenik. Choosing your weapons: On sentiment analysis tools for software engineering research. In: Proceedings of the 2015 IEEE International Conference on Software Maintenance and Evolution, ICSME. IEEE 2015, 531–535.

[31] Nicole Novielli, Fabio Calefato, Davide Dongiovanni, Daniela Girardi, and Filippo Lanubile. Can we use SE-specific sentiment analysis tools in a cross-platform setting?. In: Proceedings of the 17th International Conference on Mining Software Repositories, MSR. ACM 2020, 158-168.

[32] Akiko Aizawa. An information-theoretic perspective of TF-IDF measures. Information Processing & Management, 2003, 39(1):45–65.

[33] Jerome H Friedman. Stochastic gradient boosting. Computational Statistics & Data Analysis 38, 4 (2002), 367–378.

[34] Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. Neural processing letters 9, 3 (1999), 293–300.

[35] Marco Ortu, Alessandro Murgia, Giuseppe Destefanis, Parastou Tourani, Roberto Tonelli, Michele Marchesi, and Bram Adams. The emotional side of software developers in JIRA. In: Proceedings of the 13th International Conference on Mining Software Repositories, MSR. ACM 2016, 480–483.

[36] Daniel Pletea, Bogdan Vasilescu, and Alexander Serebrenik. Security and emotion: Sentiment analysis of security discussions on GitHub. In: Proceedings of the 11th International Conference on Mining Software Repositories, MSR. ACM 2014, 348–351.

[37] Thomas G Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. Neural Computation 10, 7 (1998), 1895–1923.

附中文参考文献:

[5] 赵妍妍,秦兵,刘挺.文本情感分析.软件学报,2010,21(8):1834−1848.

陈震鹏(1994－),男,博士,主要研究领域为软件解析学.

姚惠涵(1997－),女,本科,主要研究领域为软件解析学.

曹雁彬(1996－),女,硕士,主要研究领域为软件解析学.

刘譞哲(1980－),男,博士,副教授,博士生导师,CCF 高级会员,主要研究领域为服务计算,系统软件.

梅宏(1963－),男,博士,教授,博士生导师,CCF 会士,主要研究领域为软件工程,系统软件.