IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received February 1, 2020, accepted February 17, 2020, date of publication March 6, 2020, date of current version March 19, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2979148

The Impact of Lexicon Adaptation on the Emotion Mining From Software Engineering Artifacts

MICHAL R. WROBEL
Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, 80-233 Gdańsk, Poland
e-mail: michal.wrobel@pg.edu.pl

ABSTRACT Sentiment analysis and emotion mining techniques are increasingly being used in the field of software engineering. However, the experiments conducted so far have not yielded high accuracy results. Researchers indicate a lack of adaptation of the methods of emotion mining to the specific context of the domain as the main cause of this situation. The article describes research aimed at examining whether the adaptation of the lexicon with emotional intensity of words in the context of software engineering improves the reliability of sentiment analysis. For this purpose, a new lexicon is developed in which words are evaluated as if they were used in the field of software engineering. A comparative experiment of emotion mining based on a generic and a software engineering specific lexicon does not reveal any significant differences in the results.

INDEX TERMS Affective software engineering, emotional lexicons, emotion mining, emotion recognition, sentiment analysis.

I. INTRODUCTION

Hochschild's book "The Managed Heart", published in 1983, initiated the research of emotions in the workplace. Since then many theories have been formulated to explain the impact of emotions on work [1]. The development of computer-aided emotion recognition methods in the 21st century [2] has given new impetus to undertake research on the role of emotions in various fields. According to the Web of Science database, the number of articles on emotion recognition is growing exponentially. Only 54 articles written in 1995 on this subject were indexed, 402 in 2005, and over 2,000 in 2018.

Studies on the role of emotions in the workplace are also conducted, among others, in the software engineering domain. It is generally accepted that emotions and moods affect the work of software developers, just like many other professionals [3]. Several studies revealed correlation between emotions and productivity [4]–[6]. However, the nature of this phenomenon is still not sufficiently well known. It is necessary to carry out more accurate and broader studies than those conducted so far [7]. One of the emerging directions of research is the analysis of the IT artifacts to detect the emotions of the authors. Due to the access to a large number of artifacts associated with IT projects, unlike other research methods (e.g. observations, interviews, research), such analysis allows to obtain a vast amount of information related to emotions in a relatively short time and at low cost. As a result, we could get closer to the answer to the question whether and how emotions affect the software development process.

In recent years, techniques for recognizing opinions, emotions and even moods of authors based on their writing have been widely researched [8]. For example, sentiment analysis was one of the most explored topics in computer science research, covered by tens of thousands of scientific papers [9]. A number of scientific, proprietary, as well as open tools and methods have been developed, due to their value for practical applications [10], [11].

There is no consensus regarding the terminology concerning methods and techniques aimed at recognizing human emotions based on text analysis. The most popular approach, commonly known as sentiment analysis or opinion mining, measures only the polarity of the emotions that users express in their texts. Krcadinac *et al.* defined sentiment analysis is defined as a technique that allows to recognize opinions of text authors about specific entities such as topics, events, individuals, issues, services, products, organizations, and their attributes [12]. As a result, the polarity classification is returned, that determines whether the opinion expressed in the text about a particular feature of an object is positive, negative or neutral [9]. Sentiment analysis is similarly

The associate editor coordinating the review of this manuscript and approving it for publication was Liang-Bi Chen.

determined in software engineering research. Calefato *et al.* defined sentiment analysis as the study of the polarity of text (positive and negative), which is based on lexicons [13]. Likewise, Islam and Zibran, use the term ''sentiment analysis'' as the detection of sentimental polarity (negativity, positivity, and neutrality) of text content [14].

In order to distinguish methods that are aimed at recognizing a fuller range of human emotions, than just being positive or negative, emotion mining was introduced. It focuses on discovering from text what a person feels about entities [15], [16]. In contrast to sentiment analysis, the problem of emotion mining, sometimes also called the recognition of emotions from texts, is only at an early stage of research [17]. However, due to the large interrelation of these two techniques, problems occurring in sentiment analysis in IT artifacts also occur in the case of emotion mining.

Due to the increasing use of sentiment analysis in software engineering, some researchers amphasise the need to adapt algorithms to the context of the IT domain. Jongeling *et al.* analyzed sentiment polarity using several tools in issue trackers and Stack Overflow Q&A service [18]. The results diverge significantly depending on the tool used. Moreover, none of the results were consistent with the human rater assessment. The researchers concluded that there is a need to provide sentiment analysis tools dedicated to software engineering, as the general purpose solutions were mostly trained for products reviews [18].

Similar conclusions were reached by Tourani *et al.* in their study. Its purpose was to determine if using sentiment analysis tools it is possible to detect periods of positive or negative feelings in the community of Apache projects. The researchers used SentiStrength software, which appeared to have serious problems in distinguishing between neutral IT artifacts and these with positive and negative polarity [19].

Novielli *et al.* in their paper claimed that polarity is insufficient to detect sentiments in IT artifacts in a reliable manner. They analyzed Stack Overflow posts using SentiStrenght software. Based on the study results they concluded that inducing domain-dependent lexicons may overcome the sentiment analysis limitations [20].

Members of the same cultural and social group recognize emotions more accurately than people belonging to different groups. Since many approaches to sentiment analysis and emotional mining involve the use of lexicons of words evaluated in terms of emotional intensity, it may be reasonable to develop lexicons for specific domains in order to obtain more precise results [21]. This common opinion has become the motivation to undertake study described in the paper.

The aim of the research was to examine whether domain-specific lexicons actually improve the accuracy of emotion recognition in artifacts of the software development process. For this purpose the following hypotheses were proposed:

Null Hypothesis: The difference in the results of emotion mining in software development artifacts using general and dedicated lexicons is negligible.

Alternative Hypothesis: The difference in the results of emotion mining in software development artifacts using general and dedicated lexicons is significant.

The rest of the paper is organized as follows: in Section II related work is presented. Section III describes experiment design and Section IV its execution. Finally, Section V discusses the results and Section VI presents the conclusions.

II. RELATED WORK

All research conducted so far on the relevance of adapting the assessment of words in lexicons to the field of software engineering have concerned the sentiment analysis. However, no such study has been devoted to the emotion mining.

The first lexicon of words labeled with emotional arousal dedicated to the field of software engineering was the Software Engineering Arousal lexicon (SEA). The authors claim that the analysis of the sentiments of software artifacts in which their lexicon was used is slightly more accurate when compared to the general approach [22].

Islam and Zibran, based on the state-of-the-art SentiStrength API, have developed a tool dedicated to analyzing the sentiments in the software development ecosystem. Their SentiStrength-SE contains ad hoc heuristics designed to correct the misclassifications of SentiStrength results on software engineering artifacts and thus provide better results [23].

Non-lexicon approaches to improve the sentiment analysis in software engineering were also proposed. Calefato *et al.* introduced a sentiment polarity classifier API called Senti4SD that was trained with manually annotated StackOverflow posts [13]. Ding *et al.* devloped SentiSW tool, a subject-level tool that provides sentiment classification and entity recognition [24].

However, the overview conducted by Lin *et al.* revealed that none of the above presented tools are ready for use in real environment [25]. Also another study conducted by Imtiaz *et al.* on existing sentiment analysis tools, including Senti4SD, revealed that they are unreliable, as well as inconsistencies with human raters in identifying polarity [26]

III. EXPERIMENT DESIGN

Sentiment analysis and emotion mining techniques mainly utilize vectors of the most important text features to identify the polarity or emotions [27]. To assign sentiment scores to texts, specialized lexicons with emotional evaluation of words are often employed. Such lexicons may be built either manually, which provides higher precision or automatically based on the existing corpora, resulting in wider coverage [28]. Some novel approaches introduce advanced techniques like linguistic heuristics, lexical affinity and statistical methods, such as regression analysis [27].

So far, during sentiment analysis or emotion mining studies in the software engineering domain, general purpose lexicons have been primarily used. The aim of the study described in this paper was to check whether the differences in the results of emotion mining in software development artifacts based on dedicated lexicons are significant when compared to those commonly used. For this purpose, a lexicon of words labeled with emotional intensity in the context of IT projects has been developed. Then, using both dedicated and general lexicons, an analysis of emotion mining in IT artifacts was conducted. Finally, comparison of the results allowed to verify the research hypothesis.

Three main approaches for building lexicon of word with emotional ratings can be distinguished: manual, dictionary-based and corpus-based [10]. For the purpose of this study ANEW lexicon has been selected [29]. It is one of the most popular affective lexicons which was developed using the survey method. ANEW classifies over one thousand English words in terms of emotional intensity. For each word, based on respondents' answers, the mean and standard deviation were assigned in each of the three dimensions of a VAD scale. VAD, sometimes also called PAD (Pleasure, Arousal, Dominance) is a 3D model of emotion, which assumes that any emotion might be described with three continuous dimensions of valence (pleasure), arousal and dominance [30]. For example, an emotion defined by low valance, high arousal and low dominance is in a spectrum of emotions similar to fear and terror. Whereas high valance, low arousal and low dominance determine that someone is loosened and relaxed. The emotions represented in this model may also be mapped to discrete affective states such as Ekman's six (anger, disgust, fear, happiness, sadness, surprise) [31], which is much easier to understand by humans.

A study, analogous to that performed by Bradley *et al.* during the development of ANEW lexicon [29], was conducted on a group of IT professionals.

A. WORD SELECTION

Due to the limited resource it was decided that only a selected subset of ANEW lexicon will be evaluated by the IT professionals. Experienced specialists are willing to spend only a limited amount of time on such studies. Moreover, it was not desired to include students of computer science in the study. Allowing them to participate in the experiment would certainly significantly enhance the size of the sample due to their availability at the university. However, due to their different experience it could lead to a quality reduction of the collected data. Research conducted by Salaman *et al.* revealed that during experiments in the field of software engineering students and professionals, due to their experience can not be treated equally [32]. Therefore, it was assumed that only IT professionals were included in the assessment of 50 words in the context of software development.

As a source of words selection, bug tracking system of the Eclipse project was chosen. It is based on Bugzilla, the popular Open Source software. There were selected 50 words from the ANEW lexicon, which occurred most frequently in the bug tracking system and had the highest emotional intensity. Selection of popular words, in spite of a slight modification of the lexicon, should increase the likelihood that the analysed samples will contain modified words. Emotional intensity should result in the selection of words with different emotional assessment in the context of software development than in the original lexicon. This assumption stems from the observation that technical, domain related words are mainly emotionally neutral. An example might be the word *slave*, which according the ANEW lexicon has a very low value of valence (1.84). However, in computer jargon this word is most often used as a description of the systems architecture and as such must be considered neutral. Such words, when using a general lexicons may significantly affect the final results of emotion mining. Similar conclusions were drawn by Jurado et. al in their study, where they excluded from the assessment domain-specific words such as aggressive, attach, protect, shadow or weight [33].

A bash script was developed to download bug reports and comments from the Eclipse project's Bugzilla, which contain emotional words. The script, using curl software, iteratively retrieved all bug reports and comments from the Bugzilla system, that contained selected words from ANEW lexicon. 180 words did not occur even once. In the final step 50 most frequently occurring words have been selected. In this way, the subset of ANEW lexicon, for contextual assessment was prepared.

B. QUESTIONNAIRE

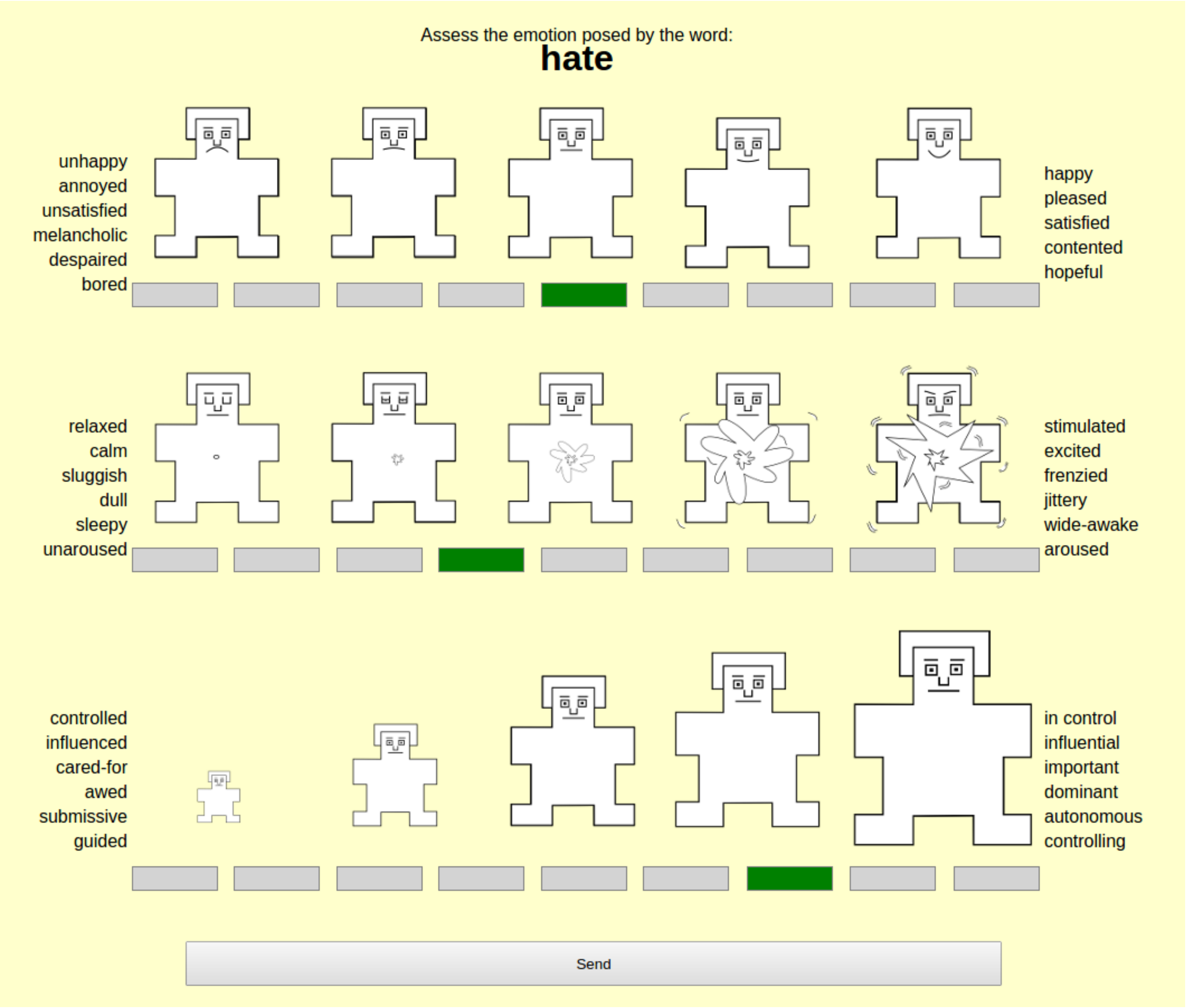
A special survey has been prepared in order to develop new emotional ratings for selected words. To provide high compatibility of the modified subset with the original lexicon, the questionnaire, as well as the guidelines were based on the original study performed by Bradly and Lang [29].

Each word has been evaluated using the Self-Assessment Manikin (SAM) form [34] that use pictographs to assess the emotional intensity in three dimensions (valence, arousal, and dominance). The graphic form of SAM eliminates inconsistencies associated with the verbal measurements while being easy to use [35].

The developed SAM form consists of three rows, each related to one dimension of emotion:

- top row – scale Happy vs. Unhappy (valence),
- middle row – scale Excited vs. Calm (arousal),
- bottom row – scale Controlled vs. In-control (dominance).

The questionnaire was developed as a web application as shown in Fig. 1. The participants were asked to assess each word using all three rows, by selecting a rectangle below the appropriate emotion intensity. The happy-unhappy scale ranges from a frown to a smile. The left extreme of this scale represents emotions such as unhappy, annoyed, unsatisfied, melancholic, despaired, or bored. The other end of the scale includes emotions such as happy, pleased, satisfied, contented, hopeful. The questionnaire also allows to describe intermediate feelings of pleasure, by selecting a rectangle below any of the other pictures. There are a total of 9 available rectangles along each rating scale that could be selected to indicate the extent of the pleasure intensity of the provided word.



Assess the emotion posed by the word:

hate

FIGURE 1. Interactive questionnaire web page.

The middle panel shows the excited or calm scale. It ranges from completely relaxed, calm, sluggish, dull, sleepy to stimulated, excited, frenzied, jittery, wide-awake, or aroused. Finally, the bottom scale distinguished between passive and active emotions. At one end of the scale (left) there are feelings characterized as completely influenced, cared-for, awed, submissive, or guided. The opposite end of this scale represents in control, influential, important, dominant, autonomous, or controlling.

The most determined respondents can evaluate all 50 words. However, it was assumed that they may interrupt the survey whenever they want. In addition, the respondents may omit these words for which they are confused about the emotional intensity or simply they do not understand their meaning. Therefore, a mechanism to ensure that all words will be evaluated by a similar number of respondents was implemented. In addition to emotional rates, the server also stores numbers representing how many times each word has been evaluated. Based on this value during the survey, each respondent identified by the session key, receives subsequently words in order from the least evaluated. When several words have the same, the smallest number of evaluations, the one which should be presented to the participant is selected randomly. With this solution it is possible to keep a similar number of evaluations of all words, even if the respondents do not complete the survey.

Respondents rate each word by clicking the gray rectangle under the corresponding SAM pictographs in each of the three rows. After clicking the ''Send'' button, which is disabled until all ratings in three rows have been selected, the data are sent to the server and stored in a database. If there are any words that have not yet been evaluated by the respondent, they are successively shown in the top of the page.

At the beginning of the survey, respondents are asked to provide general information, such as held positions and the number of years of work on the IT projects. Usually such information should be completed at the end of the survey. However, due to the nature of the on-line survey, where the participant may interrupt the study at any time, for example by closing the browser window, it was decided to gather a smaller amount of data, but at the beginning.

The software developed during this study for evaluating words using SAM forms was released under the Open Source license and can by freely used by other researchers.[1]

C. DOMAIN SPECIFIC WORDS

During the sentiment analysis or emotion mining studies conducted so far in the field of software engineering, the domain adaptations was mainly carried out by removing IT specific words from the lexicons [19], [33]. In order to assess this approach, a list of technical words that are included in the ANEW lexicon was prepared. Among all 1034 ANEW words, 22 were selected as a part a technical jargon. Words such as destroy, bullet, fat, python or bold have different emotional intensity when used in IT artifacts.

D. EMOTION MINING ALGORITHM

To conduct the evaluation of the emotional intensity of texts, naive algorithm proposed by Dodds and Danforth was applied [36]. According to this approach valence score is estimated as the average value of the valence determined on the basis of a lexicon for each word found in the text. Let v_i be a valence score of the i^{th} word, and f_i be a number of occurrence of this word in the evaluated text. The total valence score for the text is defined by the formula:

$$v_{text} = \frac{\sum_{i=1}^{n} v_i f_i}{\sum_{i=1}^{n} f_i} \qquad ([36])$$

In the same way, the total scores for the dimensions of arousal and domination are calculated:

$$a_{text} = \frac{\sum_{i=1}^{n} a_i f_i}{\sum_{i=1}^{n} f_i} \quad d_{text} = \frac{\sum_{i=1}^{n} d_i f_i}{\sum_{i=1}^{n} f_i} \qquad ([36])$$

Due to its simplicity this algorithm may not be sufficient to conduct practical emotion mining studies. More advanced approaches, such as proposed by Neviarouskaya and Aono [37] provide linguistic processing, morphological analysis, and even content awareness, and therefore they can be more reliable. However, the purpose of this research is to compare differences between evaluations of texts based on different lexicons. In this case simplicity of the algorithm should be considered rather as an advantage, not a weakness. In the approach that has been used, only the discrepancy resulting from the differences in the dictionaries will be revealed. For more complex approaches, many other factors may affect the final results.

IV. EXPERIMENT EXECUTION

The survey, which allowed to develop a subset of ANEW lexicon for software engineering domain, was carried out from early December 2015 to mid-January 2016. Software developers and other members of IT projects were invited to participate in the experiment. An invitation to participate in the survey was published on Twitter and sent directly to people from the author's mailing list. Furthermore, as a result of cooperation with the local branch of a global IT company, the information was distributed in their Intranet.

A. DEMOGRAPHY

The survey was started by 72 people, but 16 did not rate a single word, so the final number of participants was 56. On average, each of the participants rated 29.07 words. Together they made a total of 1,533 evaluations, and each word was rated 30 or 31 times. All 50 words were rated by 18 participants (32.14% of respondents), while 12 people rated less than 10 words (21.43% of respondents).

Participants differ in terms of professional experience. Only four of them had worked for less than 1 year, while 3 of them for more than 15 years. Fig. 2 shows the distribution of professional experience of participants in the following ranges: less than a year, 2 – 4 years, 5 – 8 years 9 – 14 years and above 15 years.

The survey participants were also asked to specify what their roles were in all projects in which they took part. Almost two-thirds of the participants took part in IT projects as software developers, 25 people worked as testers, 11 as analysts, 11 as architects, 10 as designers and 6 as project managers. 6 people indicated that they had other role (Fig. 2). It is worth mentioning that only 6 people have never been either a tester or a programmer.

B. SURVEY RESULTS

Based on the data collected during the survey, the mean value and standard deviation were calculated for each word in each dimension. Furthermore, separate summaries were prepared for novice and expert employees, where 5 years of professional experience was set as a boundary. Table 1 includes these words for which the absolute difference between ANEW score and the score calculated based on the survey is greater that 10% in any dimension.

Ten words with the most significant differences were highlighted in the table. These words may be assigned to one of three groups in terms of their use in IT projects. The largest group consists of words related strictly related to the computer terminology. These words are:

- bullet – used as a typographical symbol to introduce items in a list,
- fat – popular filesystem,
- python – programming language,
- detached – state of the process, object etc.
- slave – part of the client/server architecture.

[1] https://github.com/mrwrob/devanew

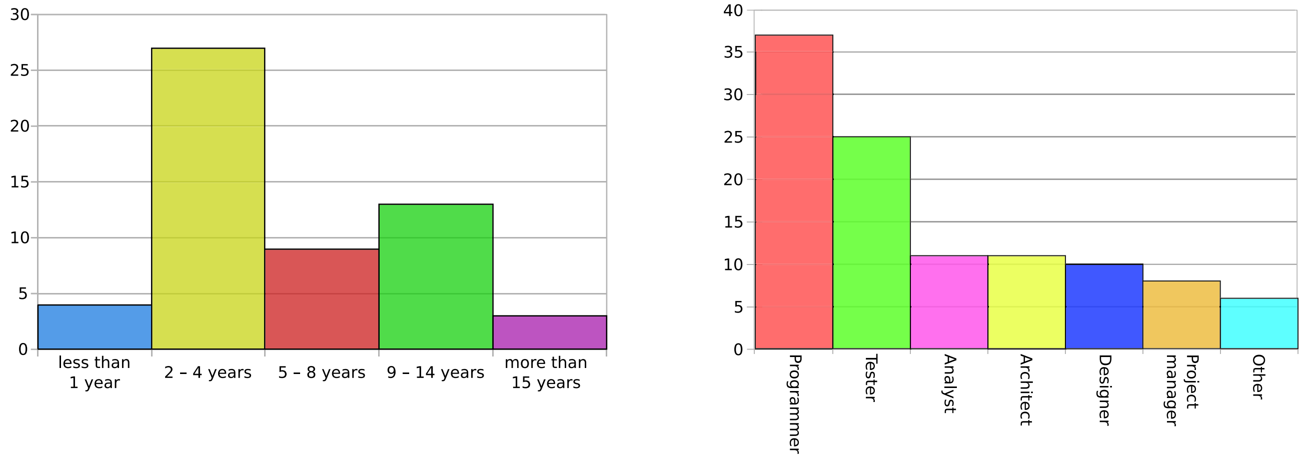


FIGURE 2. Information about the participants, on the left professional experience, o the right roles conducted in IT projects.

TABLE 1. Words that differ most significantly between lexicons.

	Dev (all)						Dev (experts)						ANEW					
	Valence		Arousal		Dominance		Valence		Arousal		Dominance		Valence		Arousal		Dominance	
	mean	s_d	mean	s_d	mean	s_d	mean	s_d	mean	s_d	mean	s_d	mean	s_d	mean	s_d	mean	s_d
accident	3,00	1,69	6,44	2,26	4,33	2,24	3,17	2,17	5,92	2,57	4,75	2,73	2,05	1,19	6,26	2,87	3,76	2,22
afraid	2,75	1,38	6,29	1,90	3,68	1,98	3,25	1,48	5,58	1,83	4,17	2,12	2,00	1,28	6,67	2,54	3,98	2,63
board	5,67	1,14	4,22	1,97	5,59	2,00	5,69	1,32	4,08	2,14	6,08	1,89	4,82	1,23	3,36	2,12	4,98	1,77
bullet	4,96	1,60	4,67	1,82	5,70	1,92	4,90	0,99	5,00	1,33	5,70	1,95	3,29	2,06	5,33	2,48	3,90	2,61
bus	5,07	1,76	4,25	2,12	4,96	2,20	5,46	1,81	4,31	2,14	5,38	2,29	4,51	1,57	3,55	1,80	4,84	1,75
contents	5,70	1,10	3,93	2,06	5,59	1,22	5,50	1,02	3,64	2,27	5,93	1,49	4,89	0,89	4,32	2,14	4,85	1,49
controlling	6,52	1,79	3,86	2,36	7,31	2,00	6,50	1,65	3,79	2,22	7,29	1,68	3,80	2,25	6,10	2,19	5,17	3,15
corrupt	2,48	1,50	6,30	1,79	4,37	2,22	2,40	1,35	6,70	1,64	4,20	2,25	3,32	2,32	4,67	2,35	4,64	2,30
damage	2,18	1,16	6,96	1,71	4,07	2,23	2,15	1,28	6,23	2,17	4,08	2,22	3,05	1,65	5,57	2,26	3,88	1,86
dead	2,07	1,51	4,89	2,85	3,68	2,64	2,15	1,41	4,85	2,67	4,31	2,81	1,94	1,76	5,73	2,73	2,84	2,32
detached	4,37	1,62	4,26	2,23	4,67	2,17	5,27	1,35	3,27	1,95	5,45	2,42	3,86	1,88	4,26	2,57	3,63	2,15
discouraged	2,64	1,13	3,57	2,38	3,54	2,49	2,82	1,33	3,73	2,15	5,00	2,93	3,00	2,16	4,53	2,11	3,61	2,01
excuse	4,24	1,72	4,66	1,99	5,00	2,24	4,50	1,95	4,21	2,29	5,21	2,52	4,05	1,41	4,48	2,29	4,07	2,10
fat	3,68	1,63	4,29	1,92	4,43	1,91	4,42	1,00	3,58	1,83	4,67	2,02	2,28	1,92	4,81	2,80	4,47	3,06
hell	2,36	1,81	6,86	1,84	4,18	2,64	2,08	1,75	7,00	1,73	3,54	2,37	2,24	1,62	5,38	2,62	3,24	2,36
hurt	2,46	1,53	6,11	1,93	4,46	2,05	2,75	1,76	6,00	1,71	4,25	1,91	1,90	1,26	5,85	2,49	3,33	2,22
inferior	2,86	1,92	4,54	2,25	3,79	2,28	3,55	2,38	4,00	2,49	4,36	2,87	3,07	1,57	3,83	2,05	2,78	2,08
lie	2,57	1,26	6,64	1,89	4,54	2,12	2,58	1,38	6,42	2,31	4,92	2,31	2,79	1,92	5,96	2,63	3,30	2,42
messy	3,04	1,48	5,93	1,38	4,44	1,69	3,09	1,70	6,27	1,10	5,27	1,74	3,15	1,73	3,34	2,37	4,75	2,15
nasty	3,44	1,67	5,56	1,65	5,33	1,80	3,18	1,66	5,82	1,66	5,09	2,17	3,58	2,38	4,89	2,50	5,00	2,17
penalty	2,52	1,28	6,78	1,50	3,07	2,04	2,80	1,40	7,00	1,63	3,00	2,49	2,83	1,56	5,10	2,31	3,95	1,97
python	6,04	1,71	4,54	1,75	5,68	2,04	5,69	1,89	4,23	2,01	5,85	2,23	4,05	2,48	6,18	2,25	4,52	2,56
rejected	2,33	1,54	5,59	2,29	3,70	2,30	3,00	1,90	5,64	2,29	3,55	2,16	1,50	1,09	6,37	2,56	2,72	2,58
reserved	5,04	1,57	3,79	1,37	5,18	2,31	4,57	1,70	4,00	1,52	5,64	2,34	4,88	1,83	3,27	2,05	4,30	1,93
shadow	4,96	1,32	3,48	1,53	4,74	1,70	5,09	1,14	3,36	1,50	5,45	1,29	4,35	1,23	4,30	2,26	4,19	1,82
slave	2,56	1,67	5,26	2,77	3,30	2,84	3,00	1,86	4,42	2,81	4,75	2,83	1,84	1,13	6,21	2,93	3,29	2,76
square	5,52	1,22	4,19	1,84	5,30	1,59	5,17	0,58	4,17	1,99	5,92	1,56	4,74	1,02	3,18	1,76	4,51	1,45
stupid	2,29	1,41	6,25	2,41	4,14	2,61	2,38	1,61	5,62	2,50	4,92	2,81	2,31	1,37	4,72	2,71	2,98	2,18
suspicious	3,74	1,46	5,33	1,98	5,04	2,26	3,82	1,72	5,45	2,02	5,18	1,99	3,76	1,42	6,25	1,59	4,47	1,99
terrible	2,11	1,31	7,04	1,63	4,19	2,22	2,18	1,60	7,36	1,03	4,18	2,82	1,93	1,44	6,27	2,44	3,58	2,34
waste	2,64	1,25	5,61	2,39	3,96	2,30	2,42	1,08	5,50	2,68	4,33	2,84	2,93	1,76	4,14	2,30	4,72	1,94

Other words were related to coding (messy, corrupt) and management (controlling, accident, rejected). Four of the distinguished words (detached, reserved, slave and accident) had a much greater difference when comparing the ratings of advanced workers and beginners. Significant differences in the evaluation of the selected word occurred mostly in more than one dimension. For four words (controlling, detached, python and slave) these differences occurred in all three dimensions, while for three words (accident, messy, rejected) only in one.

Most of the highlighted words were rated by developers as more neutral compared to the ANEW lexicon. However, in the case of the word ‘messy’, the assessment of developers compared with ANEW is lower in the dimension of Valence and much higher in the dimension of Arousal. This suggests that this word in the programming context provokes more anger than in everyday situations. For the programmer, the messy code is much more annoying than the messy room.

Among all the rated words 60% differ in at least one dimension by 10% compared with the original ANEW, including 5 in the valence dimension, 13 in the arousal and 9 in the dominance. However, taking into consideration only the assessment of experienced employees, 11 words vary significantly in the valence dimension, 15 in arousal and 17 in dominance. These differences indicate a different perception of the emotional intensity of words in professional context. This is probably due to the greater familiarity with the professional vocabulary, resulting from large work experience. For example, the difference in the evaluation of the word “detached” among novice employees compared to the original ANEW on a scale VAD is equal to (−0.11, 0.68, 0.50), while among the expert employees it is equal to (-1.41, -0.99, 1.82). As it can be noticed the biggest difference can be noticed in the Arousal dimension, where novices pointed to a greater emotional load carried by this word. On the other hand experts estimated neutral emotions as they treated the word as part of the technical jargon.

C. EMOTION MINING

In order to verify the hypothesis, emotion mining was conducted on a selected group of texts using three different lexicons:

- ANEW – original ANEW [29],
- Dev – ANEW modified based on the survey (Section III-B),
- NoDomain – ANEW without selected domain specific words (Section III-C).

As the subject of the analysis, the same texts (issues from Eclipse Project bug tracking system) were used as in the selection of words (Section III-A). For the purpose of analysis, 39 texts were selected. These are texts in which the words from the DevANEW lexicon occurred most frequently. In this way, the results of differences in emotion mining should present a worst case scenario.

The analysis was performed using the algorithm described in Section 3.4. For comparison of the emotion mining performed based on the different lexicons, the analysis was carried out on paired data. For this purpose, in each of the three dimensions (Valence, Arousal, and Dominance) absolute differences between two scores were calculated. As a result for each dimension three differences between matched pairs were given:

- $| Dev - ANEW |$ – the absolute difference between scores obtained using Dev and the original ANEW lexicons;
- $| NoDomain - ANEW |$ – the absolute difference between NoDomain and the original ANEW;
- $| NoDomain - Dev |$ – the absolute difference between NoDomain and Dev.

TABLE 2. Statistics of differences between words pairs.

		mean	25%	50%	100%
$\mid Dev - ANEW \mid$	**V**	0.17	0.03	0.07	1.59
	A	0.18	0.02	0.09	1.31
	D	0.19	0.05	0.11	0.93
$\mid NoDomain - ANEW \mid$	**V**	0.32	0.00	0.00	2.74
	A	0.12	0.00	0.00	1.12
	D	0.13	0.00	0.00	0.86
$\mid NoDomain - Dev \mid$	**V**	0.24	0.02	0.06	1.91
	A	0.16	0.04	0.13	0.92
	D	0.18	0.05	0.10	0.80

TABLE 3. Confidence interval for differences between matched pairs.

		Confidence Interval (p=0.99)
$\mid Dev - ANEW \mid$	**V**	0.17 ± 0.10
	A	0.18 ± 0.09
	D	0.19 ± 0.08
$\mid NoDomain - ANEW \mid$	**V**	0.32 ± 0.25
	A	0.12 ± 0.9
	D	0.13 ± 0.09
$\mid NoDomain - Dev \mid$	**V**	0.24 ± 0.16
	A	0.16 ± 0.07
	D	0.18 ± 0.08

Based on the collected data mean differences between matched pairs were calculated. The results are shown as boxplots in Figure 3 and in Table 2. The median of the difference between the matched pairs for all occurrence is less than 0.15, and third quartile less than 0.31. Furthermore the Confidence Interval (with 99% confidence) for the Population Mean, showed in Table 3 is very narrow.

caption needs to be more descriptive.

In order to calculate statistical significance of the difference in results of emotion mining in software development artifacts using selected lexicons, t-test for paired data was calculated. For the purpose of this test, the significance level was determined as $\alpha = 0.01$. For 38 degrees of freedom the critical value for the tests is equal: $t_{0.01} = 2.428568$

For each pair of lexicons in each dimension the value of test statistic was calculated from the following formula:

$$t^* = \frac{\bar{d}}{\frac{s_d}{\sqrt{n}}}$$

where $\bar{d}$ is the mean difference between paired values, s_d is the standard deviation and n the number of samples.

Table 4 shows the results of t-test. Each t^* value lies in a rejection region ($t^* > t_{0.01}$). This leads to the conclusion that the differences in the results of the emotion mining conducted with the examined lexicons are statistically significant.

However, to reject null hypothesis it must be defined what negligible difference in the case of the conducted research

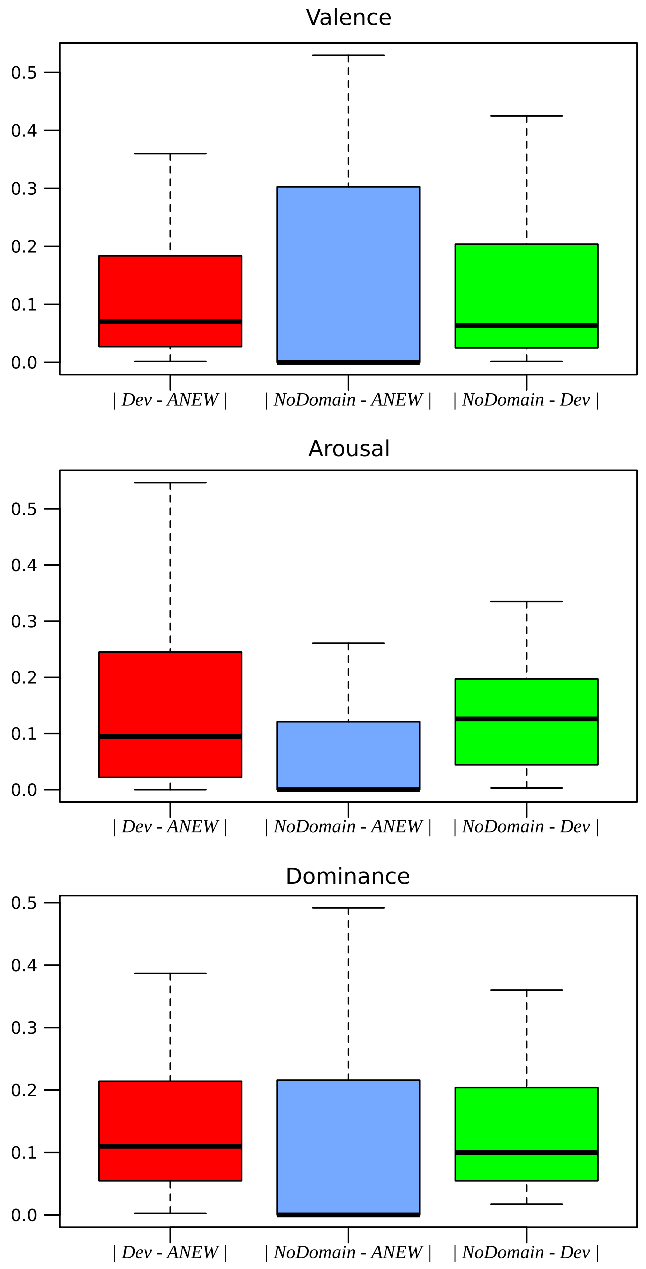


FIGURE 3. Distribution of the mean differences between matched pairs.

indicates. Emotional intensity of texts is evaluated based on the lexicons, which were developed using the survey method. Each word in the lexicon is characterized in each of the three dimensions by two values, mean value of assessment and its standard deviation. The data presented in Table 1 show that the participants' assessment, both in the ANEW and the Dev lexicons, are not completely coherent. Standard deviation for the selected words ranges from 1 to 3.5. This measure may be treated as the uncertainty metric of lexicon-based emotion mining approach.

To verify the null hypothesis $\mu_d = 1.1$ was accepted as the value of negligible difference. This is the smallest standard deviation of word assessment in any dimension of the Dev lexicon. Because this value is much grater than the mean value of differences between words in each paired lexicons, there is no need to calculate the t-test, and the null hypothesis cannot be rejected.

TABLE 4. Test statistics for differences between matched pairs.

		Test statistics t^*
$\mid Dev - ANEW \mid$	**V**	3.83
	A	4.54
	D	5.56
$\mid NoDomain - ANEW \mid$	**V**	3.04
	A	3.06
	D	3.46
$\mid NoDomain - Dev \mid$	**V**	3.43
	A	5.58
	D	5.41

V. DISCUSSION

Analysis of the test results does not give grounds to reject the null hypothesis. It showed that the differences in the results of emotion mining using general lexicon (ANEW) and a dedicated lexicon developed for the field of software engineering (Dev) are minor and can be neglected. The same, insignificant differences exist when comparing the results of emotion mining when the domain specific words were ignored during evaluation (NoDomain).

Taking into account the results provided, it may be concluded that the emotion mining from the artifacts specific to the software engineering domain can be successfully conducted with the general purpose lexicons. These conclusions also apply to the lexicon-based sentiment analysis studies. In that case, only valence dimension should be considered.

Therefore, the results of sentiment analysis and emotion mining research conducted so far in the software engineering domain (e.g., [18]–[20]) can be considered as credible. The authors' doubts about accuracy are probably related to the immaturity of emotion mining and sentiment analysis methods rather than the specific jargon of the studied field.

The research may also be a prerequisite to recognise that the use of general purpose lexicons in emotion mining, regardless of the domain, is sufficient. However, as the experiment design focused only on software engineering field, further studies in other domains should be performed.

A. THREATS TO VALIDITY

The selected algorithm of emotion mining may be considered as the main threat to the validity of the conducted study. However, due to the nature of the experiment, this naive algorithm should emphasise the differences in the results while using various lexicons. A detailed explanation of this issue is provided in Section III-D.

Another threat is at the selection of the words for the domain specific lexicon. It is possible that a different set of words would result in different results. In order to minimize this threat, artifacts with the highest number of words from the developed lexicon were selected for evaluation.

A typical threat to the validity of studies involving questionnaires is the total number of respondents. However, during this experiment, each word was evaluated by at least 30 participants, and according to the Central Limit Theorem, this allows for approximation of sample means distribution of the evaluations with a normal distribution.

The nationality of the respondents may also cause the results to be bias. The experiment was conducted in Poland and most of the participants were Poles. However IT specialist are familiar with the English computer jargon. In addition, only participants with sufficient levels of English were invited to take part in the experiment.

VI. CONCLUSION

This paper describes a study that was designed to verify the relevance of using dedicated lexicons during the sentiment analysis or emotion mining with the IT artifacts. For this purpose, a subset of the well-known ANEW lexicon was developed, in which 50 words were evaluated in the context of software development projects by IT professionals. Emotion mining experiments did not show significant differences while using general-purpose lexicons and those focused on software engineering. These results lead to the conclusion that the adaptation of the lexicons does not significantly increase the accuracy of the emotion mining studies conducted in the software engineering domain. Therefore the effort required to adapt the lexicon should rather be directed at improving the emotion recognition algorithms.

REFERENCES

[1] C. D. Fisher and N. M. Ashkanasy, "The emerging role of emotions in work life: An introduction," *J. Org. Behav.*, vol. 21, no. 2, pp. 123–129, Mar. 2000.

[2] R. A. Calvo and S. D'Mello, "Affect detection: An interdisciplinary review of models, methods, and their applications," *IEEE Trans. Affect. Comput.*, vol. 1, no. 1, pp. 18–37, Jan. 2010.

[3] M. R. Wrobel, "Emotions in the software development process," in *Proc. 6th Int. Conf. Hum. Syst. Interact. (HSI)*, Jun. 2013, pp. 518–523.

[4] D. Graziotin, X. Wang, and P. Abrahamsson, "Are happy developers more productive?" in *Proc. 14th Int. Conf. PROFES*, 2013, pp. 50–64.

[5] I. A. Khan, R. M. Hierons, and W. P. Brinkman, "Mood independent programming," in *Proc. 14th Eur. Conf. Cognit. Ergonom. Invent! Explore! (ECCE)*, 2007, pp. 269–272.

[6] M. R. Wrobel, "Towards the participant observation of emotions in software development teams," in *Proc. Federated Conf. Comput. Sci. Inf. Syst. (FedCSIS)*, Sep. 2016, pp. 1545–1548.

[7] M. R. Wrobel, "Applicability of emotion recognition and induction methods to study the behavior of programmers," *Appl. Sci.*, vol. 8, no. 3, p. 323, 2018.

[8] E. Cambria, "Affective computing and sentiment analysis," *IEEE Intell. Syst.*, vol. 31, no. 2, pp. 102–107, Mar./Apr. 2016.

[9] L. Yue, W. Chen, X. Li, W. Zuo, and M. Yin, "A survey of sentiment analysis in social media," *Knowl. Inf. Syst.*, 2018, pp. 1–47.

[10] R. Feldman, "Techniques and applications for sentiment analysis," *Commun. ACM*, vol. 56, no. 4, p. 82, Apr. 2013.

[11] B. Liu, "Sentiment analysis and subjectivity," in *Handbook of Natural Language Processing*, 2nd Ed. London, U.K.: Chapman & Hall, 2010, pp. 627–666.

[12] U. Krcadinac, P. Pasquier, J. Jovanovic, and V. Devedzic, "Synesketch: An open source library for sentence-based emotion recognition," *IEEE Trans. Affect. Comput.*, vol. 4, no. 3, pp. 312–325, Jul. 2013.

[13] F. Calefato, F. Lanubile, F. Maiorano, and N. Novielli, "Sentiment polarity detection for software development," *Empirical Softw. Eng.*, vol. 23, no. 3, pp. 1352–1382, Jun. 2018.

[14] M. R. Islam and M. F. Zibran, "Leveraging automated sentiment analysis in software engineering," in *Proc. IEEE/ACM 14th Int. Conf. Mining Softw. Repositories (MSR)*, May 2017, pp. 203–214.

[15] K. Sailunaz, M. Dhaliwal, J. Rokne, and R. Alhajj, "Emotion detection from text and speech: A survey," *Social Netw. Anal. Mining*, vol. 8, no. 1, p. 28, Dec. 2018.

[16] A. Murgia, P. Tourani, B. Adams, and M. Ortu, "Do developers feel emotions? An exploratory analysis of emotions in software artifacts," in *Proc. 11th Work. Conf. Mining Softw. Repositories (MSR)*, 2014, pp. 262–271.

[17] A. Yadollahi, A. G. Shahraki, and O. R. Zaiane, "Current state of text sentiment analysis from opinion to emotion mining," *ACM Comput. Surv.*, vol. 50, no. 2, pp. 1–33, May 2017.

[18] R. Jongeling, S. Datta, and A. Serebrenik, "Choosing your weapons: On sentiment analysis tools for software engineering research," in *Proc. IEEE Int. Conf. Softw. Maintenance Evol. (ICSME)*, Sep. 2015, pp. 531–535.

[19] P. Tourani, Y. Jiang, and B. Adams, "Monitoring sentiment in open source mailing lists: Exploratory study on the apache ecosystem," in *Proc. 24th Annu. Int. Conf. Comput. Sci. Softw. Eng.*, 2014, pp. 34–44.

[20] N. Novielli, F. Calefato, and F. Lanubile, "The challenges of sentiment detection in the social programmer ecosystem," in *Proc. 7th Int. Workshop Social Softw. Eng. (SSE)*, 2015, pp. 33–40.

[21] S. Owsley, S. Sood, and K. J. Hammond, "Domain specific affective classification of documents," in *Proc. AAAI Spring Symp., Comput. Approaches Analyzing Weblogs*, 2006, pp. 181–183.

[22] M. V. Mäntylä, N. Novielli, F. Lanubile, M. Claes, and M. Kuutila, "Bootstrapping a lexicon for emotional arousal in software engineering," in *Proc. IEEE/ACM 14th Int. Conf. Mining Softw. Repositories*. Piscataway, NJ, USA: IEEE Press, May 2017, pp. 198–202.

[23] M. R. Islam and M. F. Zibran, "SentiStrength-SE: Exploiting domain specificity for improved sentiment analysis in software engineering text," *J. Syst. Softw.*, vol. 145, pp. 125–146, Nov. 2018.

[24] J. Ding, H. Sun, X. Wang, and X. Liu, "Entity-level sentiment analysis of issue comments," in *Proc. 3rd Int. Workshop Emotion Awareness Softw. Eng. (SEmotion)*, 2018, pp. 7–13.

[25] B. Lin, F. Zampetti, G. Bavota, M. Di Penta, M. Lanza, and R. Oliveto, "Sentiment analysis for software engineering: How far can we go?" in *Proc. IEEE/ACM 40th Int. Conf. Softw. Eng.*, May/Jun. 2018, pp. 94–104.

[26] N. Imtiaz, J. Middleton, P. Girouard, and E. Murphy-Hill, "Sentiment and politeness analysis tools on developer discussions are unreliable, but so are people," in *Proc. 3rd Int. Workshop Emotion Awareness Softw. Eng. (SEmotion)*, 2018, pp. 55–61.

[27] E. Cambria, B. Schuller, Y. Xia, and C. Havasi, "New avenues in opinion mining and sentiment analysis," *IEEE Intell. Syst.*, vol. 28, no. 2, pp. 15–21, Mar. 2013.

[28] L. Gatti, M. Guerini, and M. Turchi, "SentiWords: Deriving a high precision and high coverage lexicon for sentiment analysis," *IEEE Trans. Affect. Comput.*, vol. 7, no. 4, pp. 409–421, Oct. 2016.

[29] M. M. Bradley and P. J. Lang, "Affective norms for english words (anew): Instruction manual and affective ratings," Center Research Psychophysiology, Univ. Florida, Gainesville, Fl, USA, Tech. Rep. C-1, 1999.

[30] A. Kołakowska, A. Landowska, M. Szwoch, W. Szwoch, and M. R. Wrobel, "Modeling emotions for affect-aware applications," *Inf. Syst. Develop. Appl.*, pp. 55–67, Jan 2015. [Online]. Available: http://wzr.ug.edu.pl/nauka/upload/files/Information%20systems%20development%20and%20applications.pdf#page=55

[31] A. Landowska, "Towards new mappings between emotion representation models," *Appl. Sci.*, vol. 8, no. 2, p. 274, 2018.

[32] I. Salman, A. T. Misirli, and N. Juristo, "Are students representatives of professionals in software engineering experiments?" in *Proc. 37th Int. Conf. Softw. Eng.*, vol. 1. Piscataway, NJ, USA: IEEE Press, May 2015, pp. 666–676.

[33] F. Jurado and P. Rodriguez, ''Sentiment analysis in monitoring software development processes: An exploratory case study on Github's project issues,'' *J. Syst. Softw.*, vol. 104, pp. 82–89, Jun. 2015.

[34] M. M. Bradley and P. J. Lang, ''Measuring emotion: The self-assessment manikin and the semantic differential,'' *J. Behav. Therapy Experim. Psychiatry*, vol. 25, no. 1, pp. 49–59, Mar. 1994.

[35] D. Graziotin, X. Wang, and P. Abrahamsson, ''Understanding the affect of developers: Theoretical background and guidelines for psychoempirical software engineering,'' in *Proc. 7th Int. Workshop Social Softw. Eng. (SSE)*, 2015, pp. 25–32.

[36] P. S. Dodds and C. M. Danforth, ''Measuring the happiness of large-scale written expression: Songs, blogs, and presidents,'' *J. Happiness Stud.*, vol. 11, no. 4, pp. 441–456, Aug. 2010.

[37] A. Neviarouskaya and M. Aono, ''Sentiment word relations with affect, judgment, and appreciation,'' *IEEE Trans. Affect. Comput.*, vol. 4, no. 4, pp. 425–438, Oct./Dec. 2013.

MICHAL R. WROBEL was born in Gdynia, Poland, in 1978. He received the M.S. and engineering degrees in computer science from the Gdańsk University of Technology, Poland, in 2002, and the Ph.D. degree in computer science from the Gdańsk University of Technology, in 2011.

Since 2006, he has been with the Faculty of Electronics, Telecommunications and Informatics, Department of Software Engineering, Gdańsk University of Technology. He is currently a member of the Emotions in HCI Research Group, where he conducts research on the software usability, affective computing, and software management methods. His research interest includes a modern approach to software development management, with a particular focus on the role of the human factors in software engineering.