Anger and Its Direction in Collaborative Software Development

Daviti Gachechiladze*, Filippo Lanubile†, Nicole Novielli†, Alexander Serebrenik*
* d.gachechiladze@alumnus.tue.nl, a.serebrenik@tue.nl, Eindhoven University of Technology, The Netherlands
† filippo.lanubile@uniba.it, nicole.novielli@uniba.it, University of Bari, Italy

***Abstract*—Recent research has provided evidence that software developers experience a wide range of emotions. We argue that among those emotions anger deserves special attention as it can serve as an onset for tools supporting collaborative software development. This, however, requires a fine-grained model of the anger emotion, able to distinguish between anger directed towards self, others, and objects. Detecting anger towards self could be useful to support developers experiencing difficulties; detection of anger towards others might be helpful for community management; detecting anger towards objects might be helpful to recommend and prioritize improvements. As a first step towards automatic identification of anger direction, we built a classifier for anger direction, based on a manually annotated gold standard of 723 sentences that were obtained by mining comments in Apache issue reports.**

***Keywords*-emotion mining; anger direction; issue tracking systems; collaborative software development**

I. Introduction

Software development is an inherently social activity, involving a large amount of interaction, as programmers often need to cooperate with others [25]. Recent research has provided evidence that software developers experience a wide range of emotions [18] throughout the rich ecosystem of communication channels [26]. So far, the majority of studies addressing the role of emotions in software development applied sentiment analysis [1], [7], [20], [24], that is the study of positive vs. negative orientation of a text [19]. As such, they rely on polarity as the only dimension to operationalize affect. However, polarity is only one of the possible dimensions of affect and the wide variety of affective states expressed in developers communication artifacts suggests a more fine-grained investigation of the role of emotions in collaborative software development [14], [15], [18].

Among others, negative affective states recently received particular attention [5], [13] due to their detrimental impact on developers productivity and ability to react to undesirable facts [3]. In particular, frustration may lead to poor outcomes and negative learning performance [5].

We envision emergence of tools monitoring communication between the developers, analysing the negative affect expressed in this communication and translating the analysis results into actionable insights.

To support this vision we focus on anger and all its nuances, from frustration to hostility and resentment [23], and advocate a fine-grained model of the anger emotion distinguishing between anger directed towards self, others, and objects, according to the model in Section II. Detecting anger towards *self* could be useful to design tools for supporting developers experiencing difficulties in learning a new language, solving tasks with high reasoning complexity [5], as well as in their daily programming tasks [13], thus preventing burnout and loss of productivity [12]. Conversely, timely detection of anger towards *others*, such as peers, in developers' communication messages [5], might be exploited for detection of code of conduct violations [31] or enhancing effective community management, in order to guide the contributors' behavior towards a constructive pattern of interaction and successful cooperative problem solving. Finally, detecting the expression of anger towards *objects* might be helpful to recommend and prioritize improvements based on the complaints about frameworks, programming languages or lack of documentation [5]. In particular, understanding the anger towards specific objects (e.g., APIs, app features, etc.) could be applied to user-generated content on microblogs [6] or app stores [11] to enhance software maintainance and evolution.

The closest automatic tool currently available for detecting the target of an emotion is AlchemyAPI[1] by IBM. The relation extraction feature of AlchemyAPI identifies Subject-Action-Object data from a piece of text. However, it is not able to correctly classify the emotion direction when the target is implicit, as in "Is there any progress on this issue???", which conveys anger towards collaborators, i.e., others. Futhermore, the dependence of sentiment analysis tools on the domain used for training is well known. Recent research highlights the need for emotion mining tools developed on purpose for supporting software engineering research [9], [15].

This paper represents a first step towards assessing the feasibility of automatic identification of anger direction in developers' communication. First of all, we performed an annotation study to assess if anger direction can be reliably detected by human annotators in technical texts authored by developers. As a result, we built a manually annotated gold standard of 723 sentences derived from developers' comments in the Jira-based repository of the Apache Software Foundation[2]. Both the gold standard and the annotation guidelines are publicly available[3] for research purposes and represent the first contribution of this paper (see Section III). As a second

[1] http://www.alchemyapi.com/

[2] https://issues.apache.org/jira

[3] http://goo.gl/2e6mbk

contribution, we performed a classification study to assess the feasibility of automatic classification of anger direction, which we describe in Section IV. We conclude by reviewing the related work (Section V), and providing discussion and suggesting directions for future work (see Section VI).

II. A Psychological Model of Anger Direction

Psychologists worked at decoding emotions for decades, developing theories aiming at classification of emotions and their functioning. As far as emotion mining from text is concerned [2], emotions are either considered as a continuous function of two dimensions, valence (affect polarity) and arousal (level of activation) [22], or as a finite set of individual emotions. The latter is represented by the framework of Shaver et al. [23], which includes six basic emotions, namely *love*, *joy*, *anger*, *sadness*, *fear*, and *surprise*.

When modeling anger direction, we combine the Shaver et al. definition of anger with further specification of its direction. In particular, we follow the OCC model [16] according to which emotions can be a reaction that focuses on *self*, on the *other* agent, or some properties of an *object*. For instance, a comment taken from Apache Jira "I don't have to ensure that the classloader knows groovy classes, *you* must do that"[4] expresses anger directed towards *other*.

III. A Gold Standard for Anger Direction

Starting from the comments in the Apache issue reports, we built a *gold standard* dataset by annotating angry sentences within comments with the anger direction, as shown in Figure 1. We adopt *sentences* as unit of analysis rather than each comment as a whole. Indeed, even if short, as in the case of issue tracking comments, users-contributed texts may carry different emotions [14], even with opposite polarity [29]. Furthermore, being able to analyse comments at such a fine-grained level we aim at developing a classifier which is able to clearly identify the location of the anger trigger, being it *self*, the *other* interlocutor or a specific *object*.

We started with the manually labeled emotion dataset of Ortu et al. [17], which is the best dataset available for our purpose. Since emotion annotation is a subjective task [4], before proceeding with the annotation of the anger direction, we preliminary assessed the validity of the anger label in the original dataset. We obtained 130 sentences where the authors, acting as raters, agree both on the presence of anger and on its direction. Using these sentences we developed the first prototype of the classifier for anger direction, trained in a supervised setting exploiting Support Vector Machines [8] on the features described in Section IV.

In the second step we applied this classifier to a noisier anger dataset of sentences derived from 700K comments automatically classified by the tool of Ortu et al. [17]. The classification granularity in this dataset is sentence-based. However, comments are released with just the indication of the number of sentences for which the emotion is detected. To reduce noise, we decided to use only comments composed by at most two sentences, out of which at least one was labeled as containing anger. Using this dataset and manual annotation of the anger direction we extended the annotated collection with 64 additional sentences.

Finally, in the third step we have considered sentences derived from 1.3M comments. These comments are released without any information about the emotional content [17]. We created the annotation sample using the emotion classification tool Tuktu[5] [32]. We decided to use Tuktu after comparing it against other tools for anger classification, namely Syuzhet[6] and Alchemy, on the first 130 sentences of our gold standard. In particular, we observed the highest precision for Tuktu (P=.73, R=.12, F=.21), the highest recall for Alchemy (P=.36, R=.51, F=.42), and a more balanced performance for Syuzhet, which shows the highest F-measure (P=.44, R=.5, F=.47). By optimising for precision, we reduce the number of neutral sentences misclassified as expressing anger, and then avoid annoying raters with useless annotation of neutral cases. The raters were CS graduate students trained by the first author.

The final gold standard consists of 723 anger sentences with direction labels. In particular, we have 18% sentences annotated as *self*, 9% as *other* and 73% as *object*. The interrater agreement was assessed by measuring the Fleiss' Kappa values and percentage of observed agreement among raters, that is the percentage of cases for which the raters provided the same label. Values for both metrics are reported in Figure 1. In particular, Kappa values range from moderate to substantial agreement, indicating a higher level of interrater agreement with respect to previous research on emotion annotation in Apache Jira developer comments [14].

IV. A Classifier for Anger Direction

We investigate the feasibility of building an anger direction classifier by exploiting machine learning techniques in a supervised setting, using our gold standard for training and validation. We used Weka[7], a library of machine learning algorithms. As for features, we automatically extracted uni- and bigrams using the unsupervised Weka filter *StringToWordVector*. In particular, we exploit the StringToWordVector options for calculating Term Frequency Inverse Document Frequency (TF-IDF) and performing feature selection [21].

After manually inspecting the selected features, we observed that subject/object personal and possessive pronouns as well as possessive adjectives were frequently used in their corresponding classes. Hence, we include them as additional features. To model the presence of lexicon related to *self* and *other* directed anger, we used the corresponding lexical classes of the LIWC taxonomy, a collection of about 80 word categories designed and validated by psycholinguistic research [28]. We created two new features called *LIWC-Self* and *LIWC-Other*. For each of these features we checked the sentences for respective

[4] https://issues.apache.org/jira/browse/GROOVY-1885

[5] http://www.tuktu.io/

[6] https://github.com/mjockers/syuzhet

[7] http://www.cs.waikato.ac.nz/ml/weka/

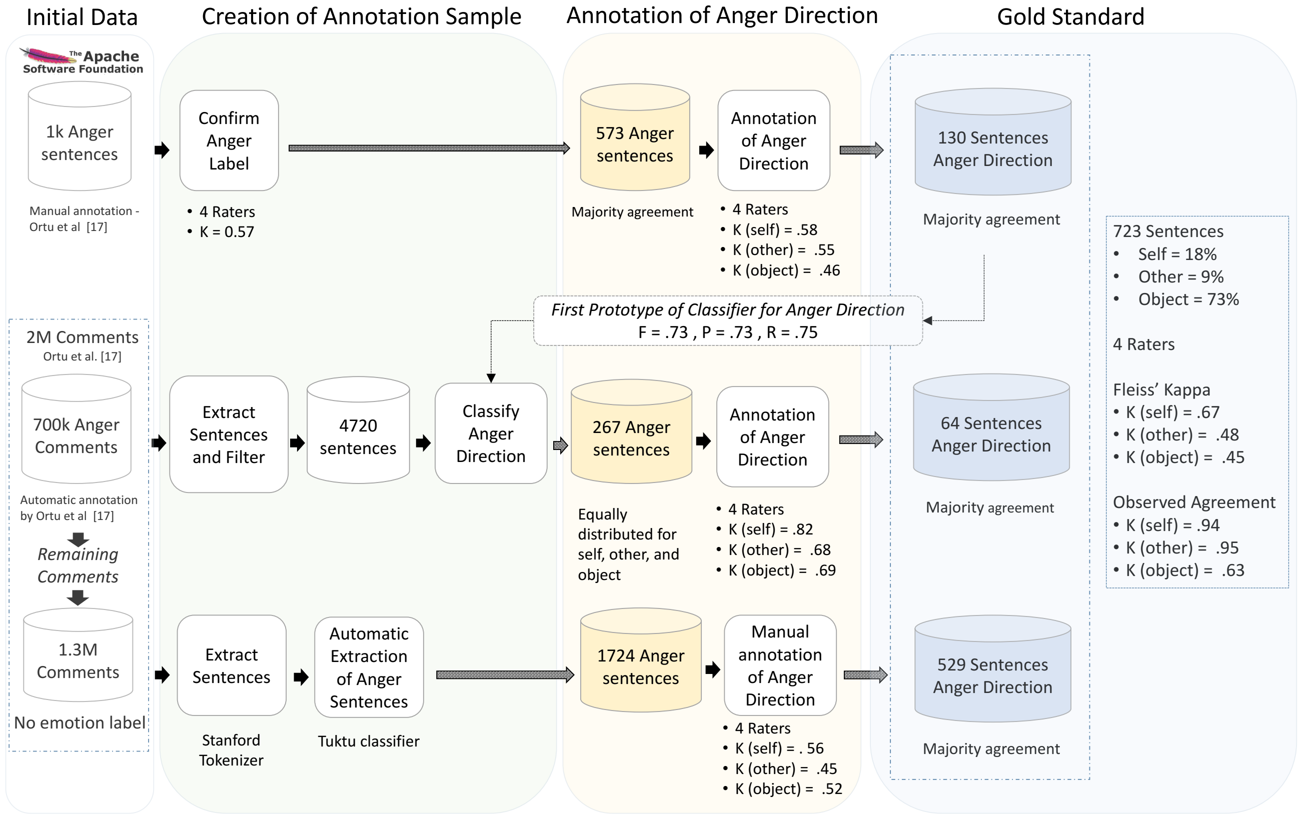

Fig. 1: Creating the Gold Standard through Manual Annotation of Anger Direction.

TABLE I
ANGER DIRECTION CLASSIFICATION RESULTS

Classifier	Class	Precision	Recall	F-Measure
SVM	Self	0.89	0.60	0.72
	Other	0.80	0.18	0.30
	Object	0.83	0.98	0.90
	Overall	0.84	0.84	0.81
J48	Self	0.69	0.57	0.62
	Other	0.38	0.24	0.29
	Object	0.83	0.91	0.87
	Overall	0.76	0.78	0.77
Naive Bayes	Self	0.53	0.82	0.64
	Other	0.30	0.57	0.39
	Object	0.91	0.68	0.78
	Overall	0.78	0.69	0.72
Baseline: Majority Class	Self	0.00	0.00	0.00
	Other	0.00	0.00	0.00
	Object	0.73	1.00	0.84
	Overall	0.53	0.73	0.61
Baseline: Random Guessing	Self	0.18	0.33	0.23
	Other	0.09	0.33	0.14
	Object	0.73	0.33	0.46
	Overall	0.33	0.33	0.33

keywords, as defined by the LIWC taxonomy, and calculated their TF-IDF scores in respect to their document class.

A classifier performance strongly depends on the setting of its input parameters, whose optimal choice heavily depends on the data being used [27]. Therefore, we used a package, called Auto-Weka[8] [30], to automatically detect the best parameters of the algorithms we used.

In Table I we report the results obtained in a 10-fold cross validation setting with the SMO Weka implementation of Support Vector Machines (SVM), J48, and Naive Bayes. We built our baseline using both the ZeroR Weka classifier, which always predicts the majority class, and random guessing. In particular, the best performance is obtained with SVM, which is kind of expected since it is regarded as the state of the art in text classification tasks [8]. However, the SVM performance reflects a bias towards the majority class *object*. By looking at the confusion matrix, we observe that low recall for *self* is due to the misclassification of 40% *self* sentences as *object*. Similarly, we observe that 79% of *other* sentences are misclassified as *object*.

V. RELATED WORK

Similarly to the analysis of emotions in software artifacts of Murgia et al. [14], we use the aforementioned framework by Shaver et al. [23]. Similarly to the tools we envision, Keertipati et al. [10] have included information about presence of negative emotion (sadness, anger, fear) to prioritize feature improvements. Differently from these works, we stress the importance of the emotion direction, specifically the anger direction. Being able to identify not only the presence of a

[8] http://www.cs.ubc.ca/labs/beta/Projects/autoweka/

negative emotion, such as anger, but also the person or object triggering the emotion, is crucial for actionable analysis and tools that support collaboration in software development.

VI. Discussion and Conclusions

In this work we envision emergence of tools monitoring communication between the developers, analysing the negative affect expressed in this communication and translating the analysis results into actionable insights. To support this vision we have conducted a preliminary study towards automatically detecting the direction of anger when developers communicate by exchanging text messages.

Our preliminary results confirm that all the three anger directions (anger towards self, others, and object) are present within comments from Apache issue reports. The preliminary classifier showed reasonable performance, suggesting that the automatic detection of the emotion direction is a realistic but challenging instrument to investigate the communication behavior among software developers.

As for annotation, the interrater agreement shows that the identification of both anger and its direction can be reliably performed by human raters, thus confirming the reliability of both our annotation schema and the resulting gold standard. However, the gold standard is highly unbalanced with *object* representing the 73% of sentences. This suggests that it is probably easier to express frustration towards something (e.g., tools or programming languages) rather than towards somebody who could be hurt and react negatively.

This unbalanced distribution of labels affects the performance of our automatic classification of anger direction. In particular, we observe that the best performing algorithm (SVM) shows high precision values for all classes, while reporting low recall for the *other* class. Thus, we highlight the need for a richer, more balanced dataset to train a robust classifier. Still, early results are encouraging and suggest that automatically detecting anger and its direction is feasible other than worth of the effort.

By sharing both our dataset and guidelines for annotation, we intend to encourage contributions from other researchers to validate and extend the gold standard dataset, and contribute to emotional awareness in software engineering.

Acknowledgment

This work is partially supported by the project "EmoQuest - Investigating the Role of Emotions in Online Question & Answer Sites", funded by the Italian Ministry for Education, University and Research (MIUR) under the program "Scientific Independence of young Researchers" (SIR).

References

[1] C. C. A. Blaz and K. Becker. Sentiment analysis in tickets for it support. In *MSR*, pages 235–246. ACM, 2016.

[2] V. Carofiglio, F. de Rosis, and N. Novielli. Cognitive emotion modeling in natural language communication. In J. Tao and T. Tan, editors, *Affective Information Processing*, pages 23–44. Springer, 2009.

[3] P. J. Denning. Moods. *Commun. ACM*, 55(12):33–35, Dec. 2012.

[4] H. A. Elfenbein and N. Ambady. On the universality and cultural specificity of emotion recognition: a meta-analysis. *Psychological bulletin*, 128(2):203–235, 2002.

[5] D. Ford and C. Parnin. Exploring causes of frustration for software developers. In *CHASE*, pages 115–116. IEEE Press, 2015.

[6] E. Guzman, R. Alkadhi, and N. Seyff. A needle in a haystack: What do twitter users say about software? In *RE*. The Association for Computer Linguistics, 2016.

[7] E. Guzman and B. Bruegge. Towards emotional awareness in software development teams. In *ESEC/FSE*, pages 671–674. ACM, 2013.

[8] T. Joachims. Text categorization with suport vector machines: Learning with many relevant features. In *ECML*, pages 137–142. Springer, 1998.

[9] R. Jongeling, P. Sarkar, S. Datta, and A. Serebrenik. On negative results when using sentiment analysis tools for software engineering research. *Empirical Software Engineering*, 2017. accepted.

[10] S. Keertipati, B. T. R. Savarimuthu, and S. A. Licorish. Approaches for prioritizing feature improvements extracted from app reviews. In *EASE*, pages 33:1–33:6, New York, NY, USA, 2016. ACM.

[11] W. Maalej, Z. Kurtanović, H. Nabil, and C. Stanik. On the automatic classification of app reviews. *RE*, 21(3):311–331, 2016.

[12] M. Mäntylä, B. Adams, G. Destefanis, D. Graziotin, and M. Ortu. Mining valence, arousal, and dominance: Possibilities for detecting burnout and productivity? In *MSR*, pages 247–258. ACM, 2016.

[13] S. C. Müller and T. Fritz. Stuck and frustrated or in flow and happy: Sensing developers' emotions and progress. In *ICSE*, pages 688–699. IEEE, 2015.

[14] A. Murgia, P. Tourani, B. Adams, and M. Ortu. Do developers feel emotions? an exploratory analysis of emotions in software artifacts. In *MSR*, pages 262–271. ACM, 2014.

[15] N. Novielli, F. Calefato, and F. Lanubile. The challenges of sentiment detection in the social programmer ecosystem. In *SSE*, pages 33–40. ACM, 2015.

[16] A. Ortony, G. L. Clore, and A. Collins. *The cognitive structure of emotions*. Cambridge University Press, 1990.

[17] M. Ortu, B. Adams, G. Destefanis, P. Tourani, M. Marchesi, and R. Tonelli. Are bullies more productive?: empirical study of affectiveness vs. issue fixing time. In *MSR*, pages 303–313. IEEE, 2015.

[18] M. Ortu, A. Murgia, G. Destefanis, P. Tourani, R. Tonelli, M. Marchesi, and B. Adams. The emotional side of software developers in Jira. In *MSR*, pages 480–483. ACM, 2016.

[19] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, Jan. 2008.

[20] D. Pletea, B. Vasilescu, and A. Serebrenik. Security and emotion: Sentiment analysis of security discussions on github. In *MSR*, pages 348–351. ACM, 2014.

[21] J. Ramos. Using tf-idf to determine word relevance in document queries. In *Instructional conference on machine learning*, 2003.

[22] J. Russell. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161–1178, 1980.

[23] P. Shaver, J. Schwartz, D. Kirson, and C. O'connor. Emotion knowledge: further exploration of a prototype approach. *Journal of personality and social psychology*, 52(6):1061–1066, 1987.

[24] V. Sinha, A. Lazar, and B. Sharif. Analyzing developer sentiment in commit logs. In *MSR*, pages 520–523. ACM, 2016.

[25] M.-A. Storey. The evolution of the social programmer. In *MSR*, pages 140–140. IEEE, 2012.

[26] M. A. Storey, A. Zagalsky, F. Filho, L. Singer, and D. German. How social and communication channels shape and challenge a participatory culture in software development. *IEEE Transactions on Software Engineering*, PP(99):1–1, 2016.

[27] C. Tantithamthavorn, S. McIntosh, A. E. Hassan, and K. Matsumoto. Automated parameter optimization of classification techniques for defect prediction models. In *ICSE*, pages 321–332. ACM, 2016.

[28] Y. R. Tausczik and J. W. Pennebaker. The psychological meaning of words: LIWC and computerized text analysis methods. *Journal of Language and Social Psychology*, 29(1):24–54, 2010.

[29] M. Thelwall, K. Buckley, and G. Paltoglou. Sentiment strength detection for the social web. *J. Am. Soc. Inf. Sci. Techn.*, 63(1):163–173, 2012.

[30] C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown. Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. In *SIGKDD*, pages 847–855. ACM, 2013.

[31] P. Tourani, B. Adams, and A. Serebrenik. Code of conduct in open source projects. In *SANER*, 2017. accepted.

[32] E. Tromp and M. Pechenizkiy. Pattern-based emotion classification on social media. In M. M. Gaber, M. Cocea, N. Wiratunga, and A. Goker, editors, *Advances in Social Media Analysis*, pages 1–20. Springer, 2015.