Comparing the Communication Tone and Responses of Users and Developers in Two R Mailing Lists: Measuring Positive and Negative Emails

Marc J. Lanovaz[1] and Bram Adams[2]
[1] *École de psychoéducation, Université de Montréal, Canada*
[2] *MCIS, Polytechnique Montréal, Canada*

***Abstract*—Despite a growing number of automated sentiment detection tools, the impact of the tone of communication in software engineering communities (e.g., end users vs. developers), and the way in which it conveys emotions, has only seen limited research. To address this issue, we examined the prevalence and characteristics of different tones in the R-help user and R-devel developer mailing lists over a ten-year period as well as their relation to replies. Our analyses show that more than 90% of posts contained generally neutral tones. That said, posts with negative and positive tones were typically longer than neutral ones. When emotional tones were displayed in replies, they were most likely to match the tone of the initial post. Overall, our results suggest that different tones may affect responding in asynchronous communications.**

***Keywords*-Sentiment Mining, Mailing Lists, Mining Software Repositories**

I. Introduction

Programmers and developers often use asynchronous communication mechanisms (e.g., message boards, mailing lists) to ask questions and share information with their peers [1], [2]. These asynchronous exchanges may range from cordial and upbeat to condescending or even hateful [3]. Examining how communication tone affects the behavior of others appears important to foster positive working communities. For example, knowing that adopting a negative tone decreases the probability of receiving a reply may influence posting etiquette on software mailing lists or online discussion forums. Contrarily, guidelines may be tailored to favor responses with a positive tone if those tend to increase participation.

In recent years, researchers have developed tools to identify sentiment and emotions in text-based communication related to software projects, such as SentiStrength-SE, Senti4SD, and EmoTxt [4]. The main limitation of these tools is that they typically perform best on the data sets on which they were trained [4], [5], [6]. Nevertheless, automated sentiment detection tools open many opportunities for researchers to efficiently measure communication tone in large data sets [7].

Apart from an early exploratory study [8], no researcher has validated the use of any of the previously mentioned tools with mailing lists. We selected SentiStrength-SE because it allowed us to categorize messages as having a positive or negative tone. In contrast, researchers designed EmoTxt and Senti4SD to return a specific emotion (e.g., anger, surprise, joy) [9], [10], which was not the intent of the current study. We examined emotions in two R mailing lists as the R programming language has one of the fastest growing user and developer communities [11]. The first list, R-help, targets mostly users of R whereas the second list, R-devel, is geared towards developers. Our research questions were:

- What is the prevalence of posts with negative and positive tones in the mailing lists?
- Do negative and positive posts differ in length and thread depth from neutral posts?
- Do replies differ based on the tone expressed in the initial post?
- Do results differ across users and developers?

II. Our Approach

A. The Data

We downloaded all emails published from 2008 to 2017 on the R-help and R-devel mailing lists [12], then used R's tm.plugin.mail package to parse the data and create a matrix containing each email's UNIX timestamp, number of characters, sentiment score (see below), thread number and thread depth. When extracting the email content and counting the number of characters, we removed any line starting with ">", "$" and , "[", as these symbols typically preceded text from the previous message or code output. After removing emails with no content, the R-help data set contained 235,309 email messages divided into 78,970 threads, while the R-devel data set had 25,771 emails divided into 7,354 threads. Our data and scripts are freely available on the Open Science Framework [13].

B. Detecting Communication Tone

We used SentiStrength-SE to extract the tone expressed by the content of each of the messages. SentiStrength-SE is a lexical sentiment mining approach for software engineering-related documents. It provides two values that range from -4 (most negative) to +4 (most positive). The first value represents the most negative tone (or sentiment) expressed in the post and the second value the most positive tone in the same post. To obtain one sentiment score per email, we use SentiStrength-SE's scale output, which simply adds the two values together. To facilitate the analyses and to remain conservative in our classification, we considered a post as neutral if the scale

output ranged from -1 to +1, as positive if the score was 2 or more, and as negative if the score was -2 or less.

III. Calibrating SentiStrength-SE

To calibrate the SentiStrength-SE tool for our data sets, we first conducted a qualitative analysis of 100 randomly-selected posts (50 negative and 50 positive) from the R-help mailing list to identify potential sources of misclassifications. The three most common causes of misclassifications for negative posts were related to the family name of a very active user ("Graves"; 4 cases), to the word "Poisson" being confused with "poison" (4 cases), and to the word "loss" (e.g., "packet loss"; 2 cases). For positive words, the expression "goodness of fit" led to three misclassifications and the word "fine" to two misclassifications. Moreover, we found three misclassifications due to positive or negative words in quotes following signatures.

To address the previous issues and improve the validity of our analyses, we made the two following changes prior to conducting our quantitative analysis. First, we changed the weights of the following words to zero in SentiStrength-SE: "grave", "poison", "loss", "goodness", and "fine". Second, we deleted the signatures and associated quotes in the email content by removing all text that followed the symbols " -- ".

During our calibration, we also made some interesting observations that should be considered when interpreting our results. The first observation was that most of the negative posts had nothing to do with flamewars or with insults directed towards others, but rather with the way (tone) in which someone communicates a message in an online venue. In 21 of 50 cases, the words that made a post negative involved apologies for asking a simple question or for not understanding something. The second most common use of negative words was to qualify a problem or solution, or to make a self-deprecating statement (e.g., "stupid solutions", "looks ugly"), which we observed in 8 cases. In contrast, 27 of 50 positive posts expressed gratitude (e.g., "thanks in advance", "appreciate your help"). Praise directed towards the solution to a problem or towards another person were also common (18 cases).

IV. Our Findings

A. What is the Communication Tone in R-help and R-devel?

We began our analyses by examining the proportion of positive, negative, and neutral posts across each data set. In R-help, 95% of posts presented neutral tones. Only 2.3% and 3.1% of posts expressed negative and positive tones, respectively. In R-devel, neutral posts represented 92% of messages, with negative and positive posts each representing 4% of messages.

Interestingly, we found less emotional tones than other researchers in JIRA issue comments, StackOverflow posts, and code review comments [3], [4], [10]. One hypothesis for this discrepancy is that our criterion for categorizing a message as positive or negative may have been more stringent than prior studies. Another hypothesis is that issue reports, comments, and online posts may be more conducive to producing emotional tones than the more traditional question/answer format of a mailing list, even though the "customer" in the R community has a reasonable technical background (since R is a programming language).

B. Do Negative and Positive Posts Differ?

Next, we examined whether the length and thread depth of posts differed based on tone. The median lengths of messages for R-help and R-devel, respectively, were 339 and 471 characters for neutral posts, 574 and 723 for negatives posts, and 633 and 709 for positive posts. Our Kruskal-Wallis tests indicate that the difference across tones was statistically significant for both lists ($p < 0.0001$). In other words, negative and positive posts tended to be longer than neutral posts.

For all subsequent analyses, we only kept the initial emails in a thread ("depth 1") as well as all direct replies to an initial email ("depth 2"), which reduced the size of R-help to 227,759 messages and R-devel to 23,629 messages. Table I displays the distribution of posts across tones and both thread depths. The chi-square test conducted to examine the difference in tone across the two thread depths was statistically significant for R-help ($p < 0.0001$) and R-devel ($p = 0.008$). We were more likely to observe negative tones in replies than in initial posts when compared to positive tones. This pattern was less pronounced in the R-devel mailing list.

C. Do Replies Differ Across Tones?

Table II shows the distribution of the number of replies following an initial post, grouped by the tones in the initial post. A chi-square test shows a statistically significant difference in their distribution ($p < 0.0001$) for R-help. The same test was nonsignificant for the R-devel data. Closer examination of the R-help distribution shows that the largest difference is due to the negative posts being the least likely to receive replies when compared to those that had neutral or positive tones.

For our final analysis, we examined the tone expressed in the first reply ("depth 2") to an initial email ("depth 1"). Because we used timestamps to identify the first reply, all messages in the .mbox files that did not contain a UNIX timestamp were removed from the data set at this point. Furthermore, we also deleted single-post threads as they did not contain a reply to analyze, which left 54,004 threads for R-help and 4,989 threads for R-devel. Table III shows the frequency of tones in the first reply, grouped by tones in the initial post. The chi-square tests were statistically significant for both lists ($p < 0.0001$). Regardless of the tone expressed in the initial post, the replies were overwhelmingly neutral. When emotional tones were displayed in replies, they were most likely to match the tone from the initial post.

V. Discussion

The results remained generally consistent across users and developers. The main differences across the two mailing lists were (a) developers showed marginally more positive and negative tones, (b) the messages were longer in the R-devel

Table I: Frequency distribution of thread depth by tones.

	R-help			R-devel		
	Tone			Tone		
Thread Depth ↓	Negative	Neutral	Positive	Negative	Neutral	Positive
1	1,183 (23%)	74,853 (35%)	2,933 (42%)	263 (29%)	6,772 (31%)	317 (35%)
2	3,945 (77%)	140,782 (65%)	4,063 (58%)	638 (71%)	15,093 (69%)	576 (65%)

Table II: Frequency distribution of the number of replies, grouped by tones in the initial post.

	R-help			R-devel		
	Tone			Tone		
Number of Replies ↓	Negative	Neutral	Positive	Negative	Neutral	Positive
0	441 (37%)	22,834 (30%)	933 (32%)	84 (32%)	2,112 (31%)	119 (37%)
1	265 (23%)	20,254 (27%)	724 (25%)	77 (29%)	1,778 (26%)	75 (24%)
2	178 (15%)	12,363 (17%)	454 (15%)	36 (14%)	1,000 (15%)	42 (13%)
3 or more	299 (25%)	19,402 (26%)	822 (28%)	66 (25%)	1,882 (28%)	81 (26%)

Table III: Frequency distribution of tones in the first reply, grouped by tones in the initial post.

	R-help			R-devel		
	Tone of the Initial Post			Tone of the Initial Post		
Tone of the First Reply ↓	Negative	Neutral	Positive	Negative	Neutral	Positive
Negative	56 (8%)	965 (2%)	45 (2%)	16 (9%)	144 (3%)	4 (2%)
Neutral	657 (90%)	49,384 (96%)	1,801 (91%)	153 (87%)	4,372 (95%)	160 (82%)
Positive	17 (2%)	950 (2%)	129 (7%)	7 (4%)	101 (2%)	32 (16%)

list (regardless of emotion) and (c) the frequency distribution of replies differed. In the R-help list, negative posts were most likely to receive no reply. In contrast, the chi-square test was not significant for the R-devel mailing list. This discrepancy suggests that developers may be less influenced by tone when choosing to reply to messages.

Our study has some limitations that should be noted. First, the design of our study was not experimental, which prevents us from determining the exact mechanisms responsible for the observed differences. At this point, we can only hypothesize as to why we observed differences across tones. Moreover, our parsing procedures remained imperfect despite our effort at removing signatures and previously quoted text. For example, signatures not preceded by " -- " were not removed.

Another issue is related to the algorithms designed to detect the emotional tones. For example, a post may include five sentences each containing moderately positive words with a score of +1, but a single sentence with a fairly strong negative word of -3, which would lead to a post being classified as being negative (-2). A solution may be to conduct a sentence-by-sentence analysis of sentiment or, alternatively, the algorithms may include weights to correct for message length. As observed during our calibration, removing specialized statistical and software engineering terms (e.g., "goodness of fit" and "packet loss") seems essential for any detection tool designed to analyze tone amongst developers and programmers. Our results should thus inform designers of sentiment detection tool to improve their algorithms and to train them to consider context [10]. That said, our analyses found differential patterns across posts categorized as positive and negative, suggesting that the algorithms do capture signal amongst the noise.

A final issue that merits further consideration is that we conducted our study on two subject systems, the R-help and R-devel mailing lists, which is a threat to external validity. Therefore, the extent to which our conclusions are applicable to other communities remains an open question, although these lists consider both end users and developers. To extend the results, researchers could apply our methodology to mailing lists of other projects, or consider other forms of asynchronous communications such as StackOverflow [14].

VI. Implications for Practice

This paper examined the use of the SentiStrength-SE sentiment mining tool on mailing list communication, showing that a calibration is important to obtain meaningful results. Therefore, developers and programmers should consider calibrating sentiment detection tools prior to using them in practice. These results may also help developers and programmers improve sentiment detection tools, or at least to specialize them to different communication media.

As expected, most of the exchanges on the mailing lists were neutral, which is good news for developers and programmers working in open source communities who rely substantially on mailing list communication for their daily operations. While a negative tone decreased the likelihood of receiving of reply for end users only, the tone of an email typically mimicked that of the initial message the email was replying to for both users and developers. As such, it makes sense for developers and programmers in open source communities to encourage neutral/positive tones in emails, either through guidelines or active moderation. This recommendation echoes learning theories [15] that consider positive tones as enablers of increasing participation.

In sum, the comparison between R-help and R-devel showed how the presence of negative/positive tones can correlate with the type of audience (i.e., developer vs. end user) and exchanged messages of a mailing list. Developers and programmers may use the results to improve the quality and flow

of their asynchronous communications, which may potentially lead to increased participation in online communities. That said, the replication of our study with asynchronous communications from a wider range of open source projects is key to examine the generality of our findings.

REFERENCES

[1] P. C. Rigby and A. E. Hassan, "What can oss mailing lists tell us? a preliminary psychometric text analysis of the apache developer mailing list," in *Proceedings of the fourth international workshop on mining software repositories*. IEEE Computer Society, 2007, pp. 23–30.

[2] B. Vasilescu, A. Serebrenik, P. Devanbu, and V. Filkov, "How social q&a sites are changing knowledge sharing in open source software communities," in *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*. ACM, 2014, pp. 342–354.

[3] A. Murgia, P. Tourani, B. Adams, and M. Ortu, "Do developers feel emotions? an exploratory analysis of emotions in software artifacts," in *Proceedings of the 11th working conference on mining software repositories*. ACM, 2014, pp. 262–271.

[4] M. R. Islam and M. F. Zibran, "A comparison of software engineering domain specific sentiment analysis tools," in *2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 2018, pp. 487–491.

[5] 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 *2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE)*. IEEE, 2018, pp. 94–104.

[6] M. R. Islam and M. F. Zibran, "Sentistrength-se: Exploiting domain specificity for improved sentiment analysis in software engineering text," *Journal of Systems and Software*, vol. 145, pp. 125–146, 2018.

[7] ——, "Leveraging automated sentiment analysis in software engineering," in *Mining Software Repositories (MSR), 2017 IEEE/ACM 14th International Conference on*. IEEE, 2017, pp. 203–214.

[8] P. Tourani, Y. Jiang, and B. Adams, "Monitoring sentiment in open source mailing lists -– exploratory study on the apache ecosystem," in *Proceedings of the 2014 Conference of the Center for Advanced Studies on Collaborative Research (CASCON)*, Toronto, ON, Canada, November 2014, pp. 34–44.

[9] F. Calefato, F. Lanubile, and N. Novielli, "Emotxt: a toolkit for emotion recognition from text," in *2017 Seventh International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*. IEEE, 2017, pp. 79–80.

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

[11] D. Robinson, "The impressive growth of r," https://stackoverflow.blog/2017/10/10/impressive-growth-r/, 2017.

[12] "R: Mailing lists," https://www.r-project.org/mail.html, 2019.

[13] M. J. Lanovaz, "Data and r code," https://osf.io/ts5nq/?view_only=75387361aa184c18a794df5838346363, 2019.

[14] B. Lin, F. Zampetti, R. Oliveto, M. Di Penta, M. Lanza, and G. Bavota, "Two datasets for sentiment analysis in software engineering," in *2018 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 2018, pp. 712–712.

[15] A. Catania, *Learning*, 5th ed. Sloan Publishing, 2013.

VII. AUTHOR BIOGRAPHIES

Marc J. Lanovaz is associate professor in the École de psychoéducation at the Université de Montréal. His research interests include the development and validation of software technology to facilitate the delivery of services to individuals with developmental disabilities and their families. Lanovaz received his PhD in educational psychology from McGill University (Canada). Contact him at marc.lanovaz@umontreal.ca.

Bram Adams is associate professor at Polytechnique Montréal. His research interests include software release engineering, mining software repositories and the impact of human affect on software development. Adams received his PhD in computer science engineering from Ghent University (Belgium). Contact him at bram.adams@polymtl.ca.