Chen, W. et al. (Eds.) (2017). Proceedings of the 25th International Conference on Computers in Education. New Zealand: Asia-Pacific Society for Computers in Education

Exploratory Analysis of Discourses between Students Engaged in a Debugging Task

Ma. Mercedes T. RODRIGO
Ateneo de Manila University, Philippines
mrodrigo@ateneo.edu

Abstract: This paper determined if and how high-performing and low-performing students differed in the language that they used as they collaborated on a debugging task. 180 students worked in pairs to debug 12 small programs with known errors. Students were segregated into high and low achievement levels based on the number of bugs they found. Chat transcripts from the pairs were analyzed using the Linguistic Inquiry and Word Count (LIWC) software. We found that high- and low-performing students only varied in terms of their use of words that implied discrepancy and sadness.

Keywords: discourse analysis, pair programming, LIWC

1. Introduction

Pair programming is a collaborative activity in which two programmers work together on a single computer in order to develop one software artifact (Williams, et al., 2000). Studies of pair programming in both industry and higher education showed that pairs are able to produce higher quality code at a faster rate than individuals. Furthermore, pair programming improves programmer confidence and job satisfaction. These benefits have been shown to transcend physical distance. Partners who were geographically separate tended to perform as well as those who were co-located (Hanks, 2005).

In recent years, researchers have tried to arrive at a more nuanced understanding of the communication between partners and how the collaborative process benefits student learning outcomes (Rodriguez, et al., 2017). Pair programming is a skill in itself, one in which partners have to manage what problem they want to solve, how the problem decomposes into elements, and the order in which these elements should be addressed (Zieris & Prechelt, 2014). These studies are important because they codify optimal discourse patterns that can then be taught to and learned by future generations of programmers.

The analysis of discourse is the quantification of utterances, i.e. spoken or written language (Chi, 1997). It involves the tabulation or counting of different kinds of utterances and the drawing of relationships between them. Within educational contexts, it is used to understand what students know or feel (De Wever, et al., 2006). While literature is rich with examples of analyses of student discourse in disciplines from psychology (e.g. Robinson et al., 2013) to math (Zemel, et al., 2007) to computer science (Romero, et al., 2013), discourse analysis of communications between partners in pair programming contexts, though, is still somewhat limited.

This paper is an initial exploratory analysis of conversations between partners engaged in a tracing and debugging task. The goal of this paper is to determine if and how high-performing and low-performing students differed in the language that they used as they collaborated. We attempt to compare the extents to which high- and low-performing students express cognitive and affective processes in their language use. We chose to focus on cognitive processes because the process of debugging involves a comprehension of programming syntax and logic and the ability to reconcile these with the program's intention. We also focused on affective processes because past research has shown relationships between emotions such as confusion (Lee, et al., 2011), curiosity and engagement (e.g. Bosch & D'Mello, 2014) and student success.

2. Methods

The data collected and analyzed for this paper were part of a nationwide study on the behaviors of pairs of students who were asked to debug 12 programs with known errors. Six universities voluntarily participated in the study: Ateneo De Davao University (ADDU), Ateneo de Naga University (ADNU), Ateneo de Manila University (ADMU), University of the Cordilleras (UC), University of San Carlos (USC), and University of South Eastern Philippines (USEP). The universities were distributed from the north to the south of the country in an attempt to get as much geographic representation as possible—UC was in Baguio, ADMU was in Manila, ADNU was in Bicol, USC was in Cebu, ADDU and USEP were in Mindanao. USEP was the only state university. All the others were private.

2.1 Participants and Test Conditions

Students aged 18-23 years old and were in their first to graduate year levels. They had taken the college-level programming courses. Ninety (90) pairs of participants composed of 112 males and 68 females. There were more males than females in the sample because there generally tended to be more males than females in information technology degree programs.

The pairs were divided into two test conditions, static (S) and dynamic (D). In the static condition, students used a custom-built slide viewer program to read the code and identify the location of the bugs. There was no need for them to correct these errors. In the dynamic condition, students located and corrected the bugs using an integrated development environment.

2.2 Data Collection and Preparation

Participants took a pre-test to establish their levels of programming proficiency. Paired students were encouraged to consult with each other using a chat program. Although they were seated together in the same room, they were spaced far enough to ensure that all communications with their partner were via chat only. Students were given one hour to debug all 12 programs. How much time they chose to spend on each program was left to them.

2.3 Student Achievement

For the pre-test, students were given one point for ever correct answer. For both the static and dynamic debugging task submissions, students were given one point for every bug that they correctly identified. The students did not incur penalties for wrong answers or blanks. To classify students as high (H) or low (L) performers, we first took the mean of the number of bugs found. Student who found less than the mean were classified as low performers.

2.4 LIWC

Conversations between pairs of students tended to be informal, hence students used a mix of English, Filipino (the national language), and local languages (Ilocano in Baguio, Bicolano in Naga, and Cebuano in Cebu and Davao). We contracted native speakers of these languages who were also proficient in English to translate the discourses to English. None of these native speakers were translators by profession.

We segregated the chats by student and then used the text processing program Linguistic Inquiry and Word Count (LIWC; Pennebaker, Booth, et al., 2015) software to analyze each student's transcript. Given a block of text, LIWC counts the number of words and then computes for the percentages of words within that text that fall under 92 categories. Our analysis is limited to the summary variables (analytic thinking, clout, authentic, emotional) and the affective (affective processes, positive emotion, negative emotion, anxiety, anger, sadness) and process categories (cognitive processes, insight, causation, discrepancy, tentative, certainty, differentiation) (Pennebaker, Boyd, et al., 2015). Analytic thinking refers to the contributor's level of formality and structure as opposed to informal, personal, narrative language. Clout reflects a level of expertise and authority as opposed to humility, tentativeness, or anxiety. Authentic implies honest, personal, and disclosing text

as opposed to guarded or distanced language. Finally, Emotional Tone refers to positivity versus anxiety, sadness, or hostility.

3. Results

We grouped the students by condition (S or D) and achievement level (H or L), resulting into four groups: static, high-performing (SH); static, low-performing (SL); dynamic, high-performing (DH); dynamic low-performing (DL). Table 1 shows the number of students per group, by gender.

Table 1: Number of students per group, by gender.

	SH	SL	DH	DL
Male	39	18	29	26
Female	23	12	10	23

On average, students typed 231 words, with a standard deviation of 233. The shortest text was 1 word, the longest was 1,505.

We used RStudio (RStudio Team, 2015) to perform two-way Analyses of Variance (ANOVAs) to compare the LIWC results among these four groups. High- and low-performing students did not vary significantly in terms of number of words used, $F(1,176)=1.161$, $p=0.28$. However, there was a main effect in terms of condition, $F(1,176)=9.55$, $p<0.01$. A post hoc Tukey HSD showed that the students in the SH group contributed more words than those in the DH ($p=0.02$) and DL groups ($p=0.04$).

3.1 Summary Variables

For Analytic Thinking, there was no main effect for condition, $F(1,176)=1.344$, $p=0.25$, or achievement level, $F(1,176)=1.001$, $p=0.32$. The interaction between the two factors was marginally significant, $F(1,176)=2.725$, $p=0.10$. However, a post hoc Tukey HSD test showed no significant difference among groups.

In terms of Clout, there was no main effect for achievement level, $F(1,176)=1.103$, $p=0.30$, or the interaction, $F(1,176)=0.53$, $p=0.47$. There was a marginally significant main effect for condition, $F(1,176)=3.134$, $p=0.08$. Once again, though, a post hoc Tukey HSD test showed no significant difference among groups.

For Authentic and Emotional Tone neither the omnibus ANOVA results nor the Tukey HSD results revealed any significant differences.

3.2 Cognitive Processes

The omnibus ANOVAs and Tukey HSDs did not yield significant results for Cognitive processes, Tentative, Certainty, and Differentiation.

The occurrence of insight words was significantly different between conditions $F(1,176)=5.82$, $p=0.02$. Insight was not significantly different between achievement levels, $F(1,176)=0.01$, $p=0.93$, nor in the interaction between the two factors $F(1,176)=0.28$, $p=0.60$. A post hoc Tukey HSD test showed no significant difference among groups.

There were some significant differences found in the occurrence of words implying causation. There was a main effect in terms of condition, $F(1,176)=11.80$, $p<0.01$, but not for achievement nor for the interaction. The post hoc Tukey HSD showed significant differences between the SH and DH groups ($p=0.03$) and the SL and DH groups ($p=0.01$). In both cases, students in the DH group used more causation words than their counterparts in other groups.

In terms of Discrepancy, we found a statistically significant difference for achievement level, $F(1,176)=9.17$, $p=<0.01$, and a marginally significant difference for condition, $F(1,176)=3.43$, $p=0.07$. The post hoc Tukey HSD only showed that the SH group tended to use more discrepancy words than those in the DL group ($p<0.01$).

3.3 Affective Processes

The omnibus ANOVAs and Tukey HSDs did not yield significant results for Affective processes, Positive emotion, Anxiety, or Anger.

Negative emotions were marginally significantly different between high- and low-performing students, $F(1,176)=3.55$, $p=0.06$. A Tukey HSD showed that high performing students tend to express marginally significantly more negative emotions than low-performing students (adjusted $p=0.06$). No other group comparisons were significant.

In terms of words expressing sadness, there was a main effect for condition $F(1,176)=8.50$, $p<0.01$, and the interaction, $F(1,176)=8.66$, $p<0.01$. A Tukey HSD showed that students in the dynamic condition expressed more sadness (adjusted $p<0.01$). It also showed that the DH group tended to express more sadness than DL group ($p<0.01$), the SH group ($p<0.01$), and SL group ($p=0.02$).

4. Discussion

The goal of this paper was to determine if and how high-performing and low-performing students differed in the language that they used as they collaborated. Based on the findings, high- and low-performing students only varied in terms of their use of words that implied causation, discrepancy, negative emotion, and sadness. High performing groups used more causation words, with sentences like ***"Because** we need to answer everything, apparently*." and "*I thought it **depends** on who was compiling*." High performing students also tended to point out discrepancies with sentences like, "*The last semicolon in line 12 is **unnecessary***." and "*Then you **should** count line 22 too."* High performing students also tended to express negative emotions such as anger, anxiety, or sadness by saying, "*What the **f***** is 4*" or "*You're **stupid"***. For the subcategory of sadness, high performers tended to apologize, "*Super **sorry** for the hassle!*"and admit confusion "*I'm now **lost** in p09*".

Although the unit of analysis was the individual rather than the pair, these findings are, to some extent, consistent with prior literature on effective programming pairs. The group of Rodriguez et al (2017) found that, generally speaking, more conversation leads to better outcomes. In specific, feedback as well as meta-comments like admitting mistakes and shortcomings, e.g. "*I think I'll need help for every number hahaha **sorry**.***"** boost learning overall.

To determine which among these five features—Word Count, Causation, Discrepancy, Negative Emotion, and Sadness—were truly predictive of student achievement, we tried to built predictive models using Weka 3.6 (Frank, Hall, & Witten, 2016). A linear regression that tried to predict the debugging scores of the students had only two features: Sadness and Discrepancy. The final model was:

$$\text{Debugging Score} = 1.7334 * \text{Sadness} + 0.6187 * \text{Discrepancy} + 12.3877$$

A ten-fold cross-validation yielded a correlation coefficient of 0.26. The model shows that Debugging Score rises with Sadness and Discrepancy words.

This model, though, has to be regarded with some caution. On average, only 0.21% of words across all students are classified as sad and the values of Sadness range from 0 to 4.17%. Similarly, on average, Discrepancy accounts for 1.98% of words. Values of Discrepancy range from 0 to 18.18%. These relatively small values imply that the large epsilon probably accounts for most of the correlation.

We therefore attempted to predict whether the students were high- or low-achievers using a J48 decision tree. The final model is as follows:

```
Discrepancy <= 1.82
|   Sadness <= 0.72: L (85.94/33.94)
|   Sadness > 0.72: H (7.08/1.08)
Discrepancy > 1.82
|   Sadness <= 0.51: H (74.82/18.82)
|   Sadness > 0.51
|   |   Sadness <= 0.87: L (8.09/1.09)
|   |   Sadness > 0.87: H (6.07/1.07)
```

Note that Sadness and Discrepancy still continued to be the most predictive features. A 10-fold cross-validation yielded a kappa of 0.24 and an accuracy of 61%. While the Kappa is considered just fair, the accuracy of the model is still better than the majority class, which is 56%.

5. Summary and Future Work

This paper was intended to be an initial analysis of discourse between pairs of programmers engaged in a debugging task. Its goal was to determine differences in cognitive and affective processes, as expressed through language, among high- and low-performing students. The study found that only the occurrence of words implying Sadness or Discrepancy differentiated high- and low-performers.

The absence of other statistically detectible differences may be attributable to a number of limitations. First, the translations captured the gist but not the full nuance of the students' discourse. Students frequently made use of slang such as such "wew" to denote "wow". When they say "joke!", they acknowledge an error, rather than attempt humor. They used "gg" to mean either "good game" or "gulong-gulo". The former is a figure of speech that denotes defeat while the latter literally translates to "extremely mixed up," an expression of confusion.

Second, LIWC is not designed to take into account the technical nature of the task. Certain words such as "problems" and "odd" were regarded as discrepancy words. When taken in context, though, students were using the former to refer to the problem number (1 through 12) that they were currently solving or that they were proceeding to solve. They used the latter because one of the problems had to do with odd and even numbers.

LIWC also did not process emoticons. Students tended to pepper their chats with smileys and similar expressions. These can all be indicators of a variety of affective states including confusion, anxiety, anger, and others that might have a relationship with overall performance.

Moving forward, we plan to continue the analysis of this dataset in a number of ways. There are many published fine-grained approaches to hand-labeling each contribution of each participant to arrive at speech acts that might imply be telling of student ability, effort, partner cohesion, or maturity of understanding. To capture these phenomena or others of interest, coding schemes will have to be developed, based on literature. For example, early work by Roschelle and Teasly (1995) described several discourse events that lead to the development of joint problem space between partners. These events include turn taking, socially distributed productions, repairs, and narrations. Farris & Sengupta (2014) attempted to trace the development of computational thinking between pairs of programmers. In more recent work of Rodriguez et al (2017), researchers categorized dialog moves between pairs of collaborating programmers as statements, opinions, explicit instructions, acknowledgements, questions, answers, feedback, or off-task statementss.

Intuition tells us that high- and low-performing students should differ in ways that manifest themselves in behavior. This analysis and others like it are attempts to discover these differences in order to arrive at learnable patterns that can then lead to a more holistic approach of educating future programmers.

Acknowledgements

I thank the Ateneo de Davao University, Ateneo de Naga University, University of San Carlos, and University of Southeastern Philippines for allowing us to conduct the eye-tracking experiment. Many thanks also to Angie Ceniza, Kristian Cordero, Joanna Feliz Cortez, Josephine dela Cruz, Jeffrel Hermias, Joshua Martinez, Thelma Palaoag, Yancy Vance Paredes, Japheth Duane Samaco, Celesamae Vicente. Lastly, thank you to the Private Education Assistance Committee of the Fund for Assistance to Private Education for the grant entitled "Analysis of Novice Programmer Tracing and Debugging Skills using Eye Tracking Data" and the Ateneo de Manila University's Loyola Schools for the grant entitled "Building Higher Education's Capacity to Conduct Eye-tracking Research using the Analysis of Novice Programmer Tracing and Debugging Skills as a Proof of Concept."

References

Bosch, N., & D'Mello, S. (2014). Co-occurring affective states in automated computer programming education. In *Proceedings of the Workshop on AI-supported Education for Computer Science (AIEDCS) at the 12th International Conference on Intelligent Tutoring Systems* (pp. 21-30).

De Wever, B., Schellens, T., Valcke, M., & Van Keer, H. (2006). Content analysis schemes to analyze transcripts of online asynchronous discussion groups: A review. *Computers & Education*, *46*(1), 6-28.

Farris, A. V., & Sengupta, P. (2014). Perspectival computational thinking for learning physics: A case study of collaborative agent-based modeling. *arXiv preprint arXiv:1403.3790*.

Frank, E., Hall, M. A., & Witten, I. H. (2016). The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016.

Hanks, B. (2005, June). Student performance in CS1 with distributed pair programming. In *ACM SIGCSE Bulletin* (Vol. 37, No. 3, pp. 316-320). ACM.

Lee, D., Rodrigo, M., Baker, R., Sugay, J., & Coronel, A. (2011). Exploring the relationship between novice programmer confusion and achievement. *Affective computing and intelligent interaction*, 175-184.

Pennebaker, J.W., Booth, R.J., Boyd, R.L., & Francis, M.E. (2015). *Linguistic Inquiry and Word Count: LIWC2015*. Austin, TX: Pennebaker Conglomerates (www.LIWC.net).

Pennebaker, J.W., Boyd, R.L., Jordan, K., & Blackburn, K. (2015). The development and psychometric properties of LIWC2015. Austin, TX: University of Texas at Austin.

Robinson, R. L., Navea, R., & Ickes, W. (2013). Predicting final course performance from students' written self-introductions: A LIWC analysis. *Journal of Language and Social Psychology*, *32*(4), 469-479.

Rodríguez, F. J., Price, K. M., & Boyer, K. E. (2017). Exploring the Pair Programming Process: Characteristics of Effective Collaboration.

Romero, C., López, M. I., Luna, J. M., & Ventura, S. (2013). Predicting students' final performance from participation in on-line discussion forums. *Computers & Education*, *68*, 458-472.

Roschelle, J., & Teasley, S. D. (1995, August). The construction of shared knowledge in collaborative problem solving. In *Computer-supported collaborative learning* (Vol. 128, pp. 69-197).

RStudio Team (2015). *RStudio: Integrated Development for R*. RStudio, Inc., Boston, MA URL http://www.rstudio.com/.

Williams, L., Kessler, R. R., Cunningham, W., & Jeffries, R. (2000). Strengthening the case for pair programming. *IEEE software*, *17*(4), 19-25.

Zemel, A., Xhafa, F., & Cakir, M. (2007). What's in the mix? Combining coding and conversation analysis to investigate chat-based problem solving. *Learning and Instruction*, *17*(4), 405-415.

Zieris, F., & Prechelt, L. (2014, September). On knowledge transfer skill in pair programming. In *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement* (p. 11). ACM.