Semi-automated Analysis of Reflections as a Continuous Course Improvement Tool

Nasrin Dehbozorgi
Department of Computer Science
University of North Carolina at Charlotte
Charlotte, NC, USA

Stephen MacNeil
Department of Cognitive Science
University of California, San Diego
San Diego, CA, USA

Abstract—**This work-in-progress paper proposes a semi-automated method to analyze students' reflections. It is challenging to include reflection activities in computing classes because of the amount of time required from students to answer the reflection questions and the amount of effort required for instructors to review the students' responses. These challenges inspired us to adopt Digital Minute Paper (DMP) as a way to give students multiple, quick opportunities to stop and reflect on their experiences in class. In this way, students are given an opportunity to develop metacognitive skills and to potentially improve their performance in the class. In addition, we used these DMPs as formative feedback for the instructors to address students' problems in the class and to continuously improve the course design. Reading reflections is tedious, time-consuming, and does not scale to large classes. To extract insights from the DMPs, we created a semi-automated process for analyzing DMPs by applying natural language processing (NLP). Our process extracts unigrams and bigrams from the reflections and then visualizes related quotes from the reflections using a treemap visualization. We found that this semi-automatic analysis of the reflections is a good, low-effort way to capture student feedback in addition to helping students be more self-regulating learners.**

Keywords—Reflection, Feedback, Digital Minute Paper (DMP), Natural Language Processing (NLP), Active Learning, CS1, Design Pattern

I. Introduction

In active learning classrooms, the instructors' role shifts from lecturing to the students in designing a dynamic learning environment. In these environments, the learning activities can change on a weekly basis. Students often work in teams to leverage the opportunity that they are all in the same place at the same time. These group dynamics and varying learning activities are also impacted by varying levels of preparedness. These many factors make it hard to create a homogeneous learning experience for all students in the classroom. For example, in groups with well-prepared students, the less prepared students may not get enough time to practice. A recent study shows that students have varying levels of preparedness that continue into the data-structures course [1]. For these reasons, it is important to capture students' learning pace to better understand and improve the design of classroom activities. There are many different ways to obtain feedback from students, which vary in terms of effort from the instructor, effort from the student, and usefulness for improving the classroom experience [2]. One way to obtain feedback from students is by polling them. However, students may not take feedback surveys seriously because students may perceive them as being for the instructors' benefit rather than for their own. Another option is to give student reflective writing assignments that they can complete during class time. However, these reflective writing assignments can take valuable time away from learning and practicing with course material. Minute Papers (MP) are short reflective writing assignments that are designed to take only one minute of students' time [8]. These can be easily integrated into existing course designs with minimal disruption. This kind of feedback may be more meaningful for students because it is designed for them as a quick reflection activity which effectively improves their critical thinking and also facilitates their communication with instructor [8]. Student reflection is not the same as instructor feedback, it is designed to help students develop their metacognitive skills and to gain a better understanding of their experiences. At the same time, these reflection activities may be beneficial for providing instructors with feedback. By extracting feedback from these writings, it may be possible to make changes to the course material and pacing to continually improve the classroom experience to ensure that students are learning the material. The MP feedback mechanism addresses previous challenges that reflection can be too tedious for students in the classroom [2].

We introduced the Digital Minute Paper (DMP) in our course via our learning management system, which is a Digital format of MP [8], to evaluate the effectiveness of reflection as a learning tool and their ability to provide useful feedback for instructors. At the end of each class period, we reserved one minute for students to reflect on their experiences. The DMPs serve as a tool for student reflection, but we also used results for attendance purposes. This encouraged students to stay until the end of class. Finally, we propose a method for automated analysis of reflective responses to DMP. This analysis can inform our pacing and the material that we present in subsequent classes. In this way, DMP serves as a tool for continuous improvement in our class.

II. Background

Reflection is an important element of the learning cycle [13]. A recent systematic literature review by Sepp et al. describes how reflection is becoming more widespread in engineering education [3]. Frameworks are also beginning to emerge that guide the successful implementation of reflective learning experiences in engineering classrooms [4]. Reflection support tools are also providing students with data-driven tools to reflect on their experiences [5]. For instance, IneqDetect records students' conversations to reflect on and improve their group dynamics [5]. This guidance and scaffolding is encouraging for promoting successful reflective learning experiences for students because it reduces the effort and maximizes the benefit of reflection. Although the purpose of integrating reflection into the classroom is to help students, they have more recently been used in new ways to improve the classroom experience for students. For example, reflective writing assignments are also being used as a tool for providing instructors with feedback [6]. In addition, features extracted from reflections have been used to improve the accuracy and time-to-detect for learning analytics systems [7]. Reflections, which were previously completed by students only for personal benefit, are now also able to improve the classroom experience for the students around them.

A. Minute Paper

The minute paper (MP) has been practiced as a tool to get close-to-real-time feedback from students [8]. It requires students to briefly answer two questions at the end of class [8], [10] and [11]. These questions are; 1) what is the most important thing they have learned in the class and 2) what was most challenging for them. The instructor compiles MP responses by reading the answers and summarizing them. The goal is to address detected issues in the next class or communicate with individual students about their issues to improve students' learning. It is reported that this approach enhances students' motivation by helping them realize their instructor values their learning needs [8]. Because students' challenges are addressed almost instantly in this method, they see it more as a tool which contributes to their learning and are more committed to thoughtfully answering MP questions [8].

The first question in the minute paper encourages students to reflect on what they have learned during the class and encourages active listening and engagement [8]. Replies to the second question reveal gaps in students' understanding and highlight the most challenging concepts. This provides a chance for the instructor to clarify challenging issues before they get lodged [9]. MP is a formative assessment technique which encourages active learning and has a positive impact on student and instructor relationship [10], [11]. This method is particularly more beneficial in large classrooms where one to one communication is a challenge [8]. Educators, who apply this method perceive it as pedagogical innovation which is an invaluable learning tool to realize the effectiveness of their teaching practice [8], [12].

In this work, we apply a digital form of MP which is called Digital Minute Paper (DMP). The main advantage of the DMP is that it can be easily integrated into the LMS system and it doesn't need to be transcribed. This allowed us to use the built-in analysis features including real-time tracking and made the text accessible to NLP and text-mining. More details of DMP practice can be found in our pedagogical design pattern model reported in previous study [14], [15] and [16]. In the following section, we present our methodology in applying DMP, the automated analysis model and results of the case study.

III. METHODOLOGY

We used the DMP to support reflection for students and as a tool for continuous improvement of the design of the classroom activities. Our implementation featured a one-minute-long reflective writing activity at the end of each class. The reflective writing prompts were the same each week so that students could anticipate the question and think about it during the learning activity and develop their metacognition. The prompts that we provided to students were, "What was the most interesting concept you learned today?" and "What was the most challenging concept in the class today?". We leveraged these DMPs to also provide instructors with feedback about their students. By analyzing and responding to each of the reflections, instructors can understand which concepts students are struggling with and when to move on to new material. Instructors can change the pacing of the material, and return to old material when students indicate that they are still struggling with those concepts. Although the MP has been practiced by researchers, its drawback is the required effort and time to analyze students' responses, especially for classes with large enrollments. This inspired us to develop an automated method to analyze the DMP reflection responses. Our model for automating the analysis of the DMPs is shown in Figure 1.

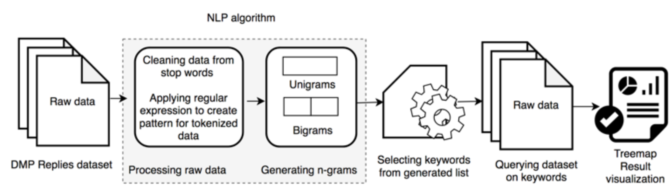

Fig.1. Our model for automating the analysis of DMP responses

In this model, first, we tokenize all of the weekly reflections and remove the stop words. We created our own set of stop words, because standard stop word dictionaries include words like "*not*" and "*or*" which have a valuable domain specific meaning in our context. Next, we extract unigrams and bigrams and sort them based on their corresponding frequencies across all the reflections for that class. Finally, we select a keyword from the list and use that as our target keyword. We query the DMP reflective response dataset based on selected keyword to create the treemap visualization of reflective statements as shown in Figure 2. The treemap partitions the space based on the number of related responses to the target keyword. The quotes presented in the treemap boxes a snippet from the longer reflective response to give a quick impression of what the student said in context. To get the full reflection text, the instructor can hover over the cell. For example, in 'Class 7' we presented the Java String API. In that week, "string" was one of the most frequently used unigrams. Figure 2 shows the visualization that resulted when we used "string" as the target keyword. The words "string" and "substring" without any other words were most common and they are presented alone in the darker colored boxes to show that. The lighter teal boxes indicate individual responses for which there was only a single quote. Irrelevant uses of the word can be removed manually at this point to fix the counts. For instance, if a student used the word "statement" we would want to verify that they meant "conditional statement" or "if statement" rather than referring to non-programming aspects. This semi-automatic approach based on regular expressions was used because NLP methods such as topic modeling are not robust to small datasets with sparse topics. We were able to quickly filter and visualize the results without necessarily having to read every reflection while ensuring that the data collected was accurately represented.

In this paper, we focused on students' responses to the challenge question: "What was the most challenging concept in the class today?" to focus on how to improve the pace of the learning material. However this methodology can be applied to the first question as well, to focus on learning outcomes. In the next section, we provide a case study where we used this approach in our CS1 classroom.

Fig 2. Our treemap visualization with "string" as the search keyword for week seven. Each box represents a partial quote from a reflection. Darker colored boxes represent multiple students who wrote the same partial quote.

A. Classroom case study

To evaluate the effectiveness of our semi-automated approach for analyzing DPM reflections for continuous course improvement, we conducted an in-class study. We collected data from an introductory computer science (CS1) class during a 5-week period. Each week we had two class sessions in which we collected data. Two of these classes were used for test-related activities and did not include reflections. This resulted in eight reflection sessions with an average of 57 responses per session. There were 63 students in the class and so our average response rate was 90.4%. Data collection is still on-going throughout the Spring 2019 semester. Using these reflections and our semi-automated approach outlined in the previous section, we were able to continually adjust the class to address misconceptions and challenges that students faced in the class. The class was taught in an active learning style with prep work at home. We introduced a short 15-minute poll everywhere quiz followed by mini-lecture in the classes that covered the challenges that we extracted from the reflections. The mini-lecture [16], [17] is one of our means for continuous improvement. The other aspect of our continuous improvement is to integrate more activities for that content in future offerings of the course.

Table 1 shows the defined goal for student learning in each class and the topics introduced in each class with the number of challenge tokens in each sessions' dataset. The data was collected from 8 sessions of CS1 class which is shown in the Table 1. In this table, we have the total number of students' replies to DMP challenge question in each class session (total number of tokens).

TABLE 1. RAW NUMBER OF TOKENS AFTER EXTRACTING KEYWORDS BASED ON N-GRAMS AND QUERYING DMP RESPONSES TO CHALLENGE QUESTIO

Topics	Topic Introduction	Number of challenge tokens in each session							
		Class 1	Class2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8
Introduction, Java, IDE	Class 1	3	0	1	0	0	0	0	0
Scanner input, output	Class 1	4	0	1	0	0	0	0	0
Math operations/concepts	Class 1	4	0	2	1	2	0	0	0
Variables	Class 1	2	0	1	0	0	0	0	0
Basic Data types	Class 1	1	0	6	0	1	0	0	0
Flowcharts (shapes)	Class 2	0	12	0	0	0	0	0	0
If statement	Class 3	0	1	4	0	6*	0	0	0
Boolean operators	Class 4	0	0	0	4	0	0	0	1
Switch statement	Class 4	0	0	0	18	17	2	0	1
Random numbers	Class 5	0	0	0	0	10	5	0	3
Math class , methods	Class 6	0	0	0	0	0	5	0	0
String class, methods and char	Class 7	0	0	0	0	0	8	25*	5
ASCII code	Class 8	0	0	0	0	0	0	0	4
Debugging and syntax, code, formatting		0	0	0	0	4	0	0	1
Total number of tokens in each session data		60	61	54	58	54	54	58	50

In Table 1, the highlighted cells indicate the classes in which the new topic was introduced. As we can see the highest number of challenge tokens in the data set are in the days that the topic was introduced. There are just two cases where the number of challenge tokens increased after the topic was introduced (marked by start*). The reason is that in 'Class 3' the basic concept of *If statement* was introduced. However, in 'Class 5' students were exposed to more complex concepts of *nested if statements*. Also, in 'Class 6' students were introduced to basics of String data type, while in 'Class 7' students had to work with string APIs which was more complicated. After extracting keywords from the whole dataset we realized that most students had debugging and syntax challenges when they worked on the conditional statement and generating random numbers. One important note is that the sum of challenge tokens in each class is not necessarily equal to the total tokens in that class since some students had no challenge in that class and so were excluded. Table 2 shows the challenge categories and their mean value in each class session

TABLE 2. PERCENTAGES OF CHALLENGE TOKENS AS A PERCENTAGE TO ACCOUNT FOR FLUCTUATIONS IN NUMBER OF TOKENS IN EACH CLASS

Topics	Topic Introduction	Number of challenge tokens in each session							
		Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8
Introduction, Java, IDE	Class 1	5.0	0	1.9	0	0	0	0	0
Scanner input, output	Class 1	6.7	0	1.9	0	0	0	0	0
Math operations/concepts	Class 1	6.7	0	3.7	1.7	3.7	0	0	0
Variables	Class 1	3.3	0	1.9	0	0.0	0	0	0
Basic Data types	Class 1	1.7	0	11.1	0	1.9	0	0	0
Flowcharts (shapes)	Class 2	0	19.7	0.0	0	0.0	0	0	0
If statement	Class 3	0	1.6	7.4	0	11.1	0	0	0
Boolean operators	Class 4	0	0	0	6.9	0	0	0	2.0
Switch statement	Class 4	0	0	0	31.0	31.5	3.7	0	2.0
Random numbers	Class 5	0	0	0	0	18.5	9.3	0	6.0
Math class , methods	Class 6	0	0	0	0	0	9.3	0	0
String class, methods and char	Class 7	0	0	0	0	0	14.8	43.1	10.0
ASCII code	Class 8	0	0	0	0	0	0	0	8.0
Debugging and syntax, code, formatting		0	0	0	0	7.4	0	0	2.0

One important note is that in Table 1 we showed the trend is such that the number of challenge tokens decreases over time (by more practice and exposure), but it did not necessarily reveal what was the most challenging topic in each session. We concluded that the top challenging topic in each session might not be the most recent introduced topic. For example, String APIs were introduced in 'Class 7' and was still the most challenging topic for students in 'Class 8', although the percentage decreased to a great extend compared to 'Class 7'.

In Table. 2 we show the mean of the number of challenge tokens (i.e. challenging topics) as a percentage to account for weekly fluctuations in the number of responses. The highlighted cells in Table 2 show the top challenging topic(s) in each class. We observed in Table 2 that the topics with the highest total mean value in all 8 classes were generally the most challenging ones for students. Thus we conclude the top 6 challenging topics for students are 1) math operations/concept, 2) flowcharts, 3) If statements , 4) switch statements, 5) random numbers and 6) strings APIs.

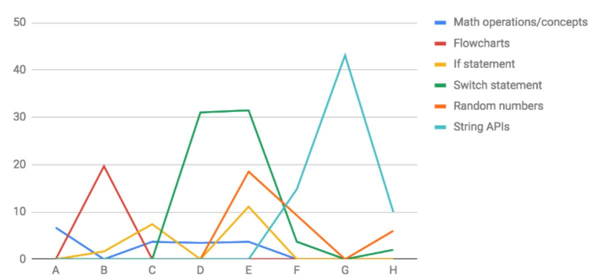

Fig. 3. Most challenging concepts over time

B. *Evaluation:*

To validate the accuracy of our semi-automated approach, we conducted a 5 point Likert Scale survey at the end of 'Class 8' to see what students think is most challenging for them. We categorized the course topics into five categories and asked students to report their mastery level on each topic (1 being very low and 5 being very high). The categories were: 1) computer software, hardware, concepts such as a compiler, bit, bytes, etc. 2) data types, variables, and identifiers 3) flowcharts and logic of programming 4) conditional statements (if/else and switch statements) and 5) strings and string APIs. Figure 4 shows students' self-reported on their mastery level.

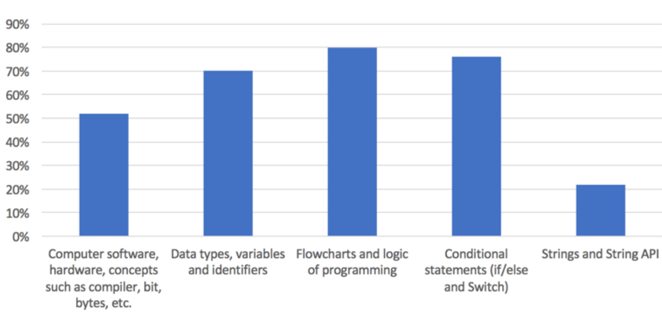

Figure 4. Students' mastery level on concepts

As shown in Figure 3, the most identified challenging topics based automated analysis of DMP result was strings APIs, switch statements (conditional statements) and flowcharts in order. Students' self-report, shown in Figure 4, supports our method's generated result, as students feel they have the lowest mastery level on string APIs, conditional statements and flowcharts in order.

IV. Discussion and Conclusion

Reflection is an important part of learning. It can also generate a lot of information about students' experiences, but analyzing these responses requires a lot of time and effort as the class size increases. In this work, we presented a semi-automated approach to make sense of students' reflections at scale by using DMP. In summary, the highlights of DMP are: 1) the duration is very short, 2) repetition of the same reflection questions each week allows students to think about them as they do their work in anticipation, 3) an ability to track students individually to identify their challenges, 4) dynamically adapt class activities based on students' problem areas. By automatically extracting keywords from students' DMPs and visualizing the corresponding quotes for a selected keyword, we could quickly get a sense for what students were struggling with during class. We used this process as a tool for continuous improvement in our CS1 class, dynamically adding mini-lectures to help students get a better handle on the most challenging aspects of the course. What we found through this process is that some topics emerge at multiple points in the course. Students encounter a concept and get some practice with it, but when we return to the concept later in the semester it is less clear for them. For example, students may understand how to use conditional logic statements, but when using them nested within iterators or conditionals, students may begin to struggle with the concept again. Our semi-automated approach helps us to continuously monitor students' understanding and easily scaled to a class with 63 students.

V. Future Work

For future work, we plan to use this approach in much larger classes. We also plan to use this approach to make sense of the curriculum across multiple different course sections. Each of our course sections follows roughly the same curriculum. It would be interesting to see whether there are trends that are unique to specific offerings, or whether there are overarching trends across multiple classes. These trends might help us to better sequence material, modify challenge level of the exams, add new activities, or adjust the pacing in subsequent semesters. We also plan to tweak our minute paper reflection questions to see what effect that has on capturing students' misconceptions and challenges. Our current approach, extracts important information from the reflections for the instructor, but with these adjustments, we may be able to scale to larger classes and capture more information about students' experiences.

VI. Acknowledgment

This work is supported by the National Science Foundation Award 1519160: IUSE/PFE: RED: The Connected Learner: Design Patterns for Transforming Computing and Informatics Education. We offer our sincere gratitude to professor Mary Lou Maher for her continuous support and feedback in accomplishing this work.

REFERENCES

[1] Celine Latulipe, Stephen MacNeil, and Brian Thompson. Evolving a data structures class toward inclusive success. In 2018 IEEE Frontiers in Education Conference (FIE 2018), San Jose, USA, October 2018.

[2] Huxham, Mark, et al. "Collecting student feedback: a comparison of questionnaire and other methods." Assessment & Evaluation in Higher Education 33.6 (2008): 675-686.

[3] Sepp, Lauren A., et al. (2015). On an upward trend: Reflection in engineering education." Proceedings of 2015 American Society of Engineering Education Annual Conference. 2015.

[4] Turns,J.A.,Sattler,B.,Yasuhara,K.,Borgford-Parnell,J.L.,andAtman, C. J., (2014). Integrating Reflection on Experience into Engineering Education. Proc. of the ASEE Annual Conf. & Exposition.

[5] Stephen MacNeil, Kyle Kiefer, Dev Takle, Brian Thompson, and Celine Latulipe. Ineqdetect: Visualizing students' conversations to increase awareness and support reflection. In Proceedings of the Global Computing Education Conference, CompEd 2019, New York, NY, USA, 2019. ACM

[6] Stone, J. (2012). Using Reflective Blogs for Pedagogical Feedback in CS1. ACM, 2 Penn Plaza, Suite 701, New York, NY 10121-0701, USA, 2012.

[7] Mohsen Dorodchi, Aileen Benedict, Devansh Desai, Mohammad Mahzoon, Stephen MacNeil, and Nasrin Dehbozorgi. A comparison of lecture-based and active learning design patterns in cs education. In 2018 IEEE Frontiers in Education Conference (FIE 2018), 2018.

[8] Stead, D. R. (2005). A review of the one-minute paper. Active learning in higher education, 6(2), 118-131.

[9] HARWOOD, W. S . (1996) 'The One-Minute Paper: A Communication Tool for Large Lecture Classes', Journal of Chemical Education 73(3): 229–30.

[10] Whittard, D. (2015). Reflections on the one-minute paper. International Review of Economics Education, 20, 1-12.

[11] Soetaert, E. (1998) Quality in the Classroom: Classroom Assessment Techniques as TQM. New Directions for Teaching and Learning. 75. Pp.47-55.

[12] CHIZMAR, J. F. & OSTROSKY, A. L . (1998) 'The One-Minute Paper: Some Empirical Findings', Journal of Economic Education 29(1): 3–10

[13] Dorodchi, M., & Dehbozorgi, N. (2017, June). Addressing the Paradox of Fun and Rigor in Learning Programming. In Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education (pp. 370-370). ACM.

[14] Dehbozorgi, N. (2017, August). Active Learning Design Patterns for CS Education. In Proceedings of the 2017 ACM Conference on International Computing Education Research (pp. 291-292). ACM.

[15] Dehbozorgi, N., MacNeil, S., Maher, M. L., & Dorodchi, M. (2018, October). A Comparison of Lecture-based and Active Learning Design Patterns in CS Education. In 2018 IEEE Frontiers in Education Conference (FIE) (pp. 1-8). IEEE.

[16] https://innovationsteched.com/design-patterns-handbook/

[17] Dehbozorgi, N. Maher, M.L., MacNeil, S. , Dorodchi, M. 2018. An Object-based Design Pattern Model for Collaborative Active Learning (Manuscript submitted to the Computer Science Education Journal)