# Profiling Student Learning from Q&A Interactions in Online Discussion Forums

**De Lin ONG, Kyong Jin SHIM* & Swapna GOTTIPATI**
*School of Computing and Information Systems, Singapore Management University, Singapore*
*kjshim@smu.edu.sg

**Abstract:** The last two decades have witnessed an explosive growth in technology adoption in education. Proliferation of digital learning resources through Massive Open Online Courses (MOOCs) and social media platforms coupled with significantly lowered cost of learning has brought and is continuing to take education to every doorstep globally. In recent years, the use of asynchronous online discussion forums has become pervasive in tertiary education institutions. Online discussion forums are widely used for facilitating interactions both during the lesson time and beyond. Numerous prior studies have reported benefits of using online discussion forums including enhanced quality of learning, improved level of thinking beyond the classroom, collaborative knowledge building, and enhanced participation by shy or intimidated students. By monitoring and analyzing students' activities in online discussion forums, instructors can intervene and manage students' learning. For the instructor to employ appropriate intervention measures, both quantitative and qualitative analyses of students' participation are important. To mitigate the challenge of the sheer volume of conversation threads in online discussion forums, we present a text mining approach to profiling student learning based on Q&A interactions. Firstly, we perform text classification to categorize conversations into two categories: non-programming-related and programming-related. Secondly, from the programming-related conversation threads, our method categorizes students into four participation proficiency types based on their Q&A activities. Next, our method determines whether a student adopts more explicit or implicit expression behavior in Q&A activities. We evaluate our approach on the second-year computing course, Web Application Development II. Finally, we share the lessons learned in this teaching process.

**Keywords:** Profiling student learning, online discussion forum, Q&A interactions, slack

## 1. Introduction

Today, the Internet is ever present in the lives of many people globally. Proliferation of digital learning resources through Massive Open Online Courses (MOOCs) and social media platforms coupled with significantly lowered cost of learning is continuing to bring education to every doorstep. Leaders of educational institutions found online learning to be a critical component in their long-term strategic planning (Allen & Seaman, 2015). Online learning allows a wider range of individuals – such as full-time workers, caretakers, and mothers – to flexibly start and continue their learning journey (Ragusa & Crampton, 2017).

In recent years, the use of asynchronous online discussion forums has become pervasive in tertiary education institutions. Online discussion forums are widely used for facilitating interactions both during the lesson time and beyond. The traditional online discussion forum is typically situated within a Learning Management System (LMS). Today more than ever, as we live through the COVID-19 pandemic where virtual distance learning is commonplace, the benefits of online discussion forums are being revisited.

In online discussion forums, conversations can be neatly organized by course topics by using threads or channels. Digitally saved and viewed anytime anywhere, online discussion forums allow for learning much more flexibly compared to traditional modes of learning restricted to physical classrooms. Picciano's study (Picciano, 2002) found that the learner's participation in online discussion forums led to enhanced learning in terms of quantity and quality. Meyer's study reports that participation in online discussions is associated with improved level of thinking beyond the classroom

(Meyer, 2003). Another prior study reported a positive correlation between the learner's final grade and his engagement in online discussion forums (Bliuc et al., 2010).

Online discussion forums enhance the participation of shy or intimidated learners who would otherwise keep quiet in face-to-face lessons (Groeling, 1999; Al-Salman, 2009; Gerbic, 2010). Further, online discussion forums promote networking with other learners such that they can build knowledge collaboratively (Gilbert & Dabbagh, 2005). By monitoring and analyzing learners' activities in online discussion forums, instructors can intervene and manage students' learning (Stephens-Martinez, 2014; Jiang et al., 2015). Understanding the extent to which learners participate in the online discussion forum is important. Simply quantifying learners' participation such as the number of posts in online discussion forums does not adequately portray their contributions (Mazzolini & Maddison, 2007). For the instructor to employ appropriate intervention measures, qualitative analyses involving verification of each student's contribution are still deemed important. However, it can be quite challenging for the instructor to manually inspect the sheer volume of messages in an online discussion forum. One way to mitigate this problem is to employ an automated approach to the identification of relevant discussion forum threads. Further, giving instructors the ability to automatically profile learners based on learners' participation will allow instructors to quickly identify and narrow down to those learners that need most help at the moment.

Therefore, the goal of our paper is to present a text mining approach to profiling student learning based on Q&A interactions in Slack, a popular messaging application designed for teamwork. Originally built for businesses, by March 2020, over 3,000 colleges and universities had adopted Slack in their classrooms (Slack, 2020). Slack, Discord, Microsoft Teams, and Google Hangouts are similar platforms, and we choose Slack due to some free features compared to other tools (Tuhkala & Kärkkäinen, 2018). Firstly, we perform text classification to categorize conversations into two categories: non-programming-related and programming-related. Secondly, from the programming-related conversation threads, our method categorizes learners into four participation proficiency types based on their Q&A activities. Next, our method determines whether a learner adopts more explicit or implicit expression behavior in Q&A activities. We evaluate our approach on the second-year computing course, Web Application Development II. Finally, we share the lessons learned in this teaching process.

## 2. Background

### *2.1 Web Application Development Curriculum*

In our university's Information Systems undergraduate program, all students must complete core courses including Web Application Development I (WAD I) and Web Application Development II (WAD II). WAD I focuses on server-side web development using HTML and PHP. WAD II is a follow-up course whose focus is on front-end web development, and it builds on top of WAD I by introducing students to HTML, CSS, Bootstrap and responsive UI, JavaScript, APIs, and Vue.js as part of the curriculum. In WAD II, 4–5-person group projects allow students to explore beyond-the-curriculum frameworks and tools such as ReactJS, AngularJS, Tailwind CSS, D3.js, Git, and other front-end frameworks/libraries as well as visualization and animation APIs. WAD II is a young course which was offered as a core course for the first time in the Fall 2020 semester.

### *2.2 Using Slack to facilitate Q&A in Web Application Development Courses*

In early March 2020, all lessons at our university moved to fully online due to the COVID-19 pandemic. As a result, WAD I lessons and assessments abruptly moved to online via Zoom and our university's LMS with little preparation. At the time, only a few WAD I sections and faculty members had been using Slack to facilitate Q&A. While our lessons were still conducted fully on-campus, most students sought help from instructors and teaching assistants (TAs) via physical means by arranging face-to-face consultations on campus. Thus, while Slack served a good purpose for information dissemination, it was still not fully utilized as more students resorted to face-to-face Q&A and were not familiar with or comfortable with participating in online Q&A discussions.

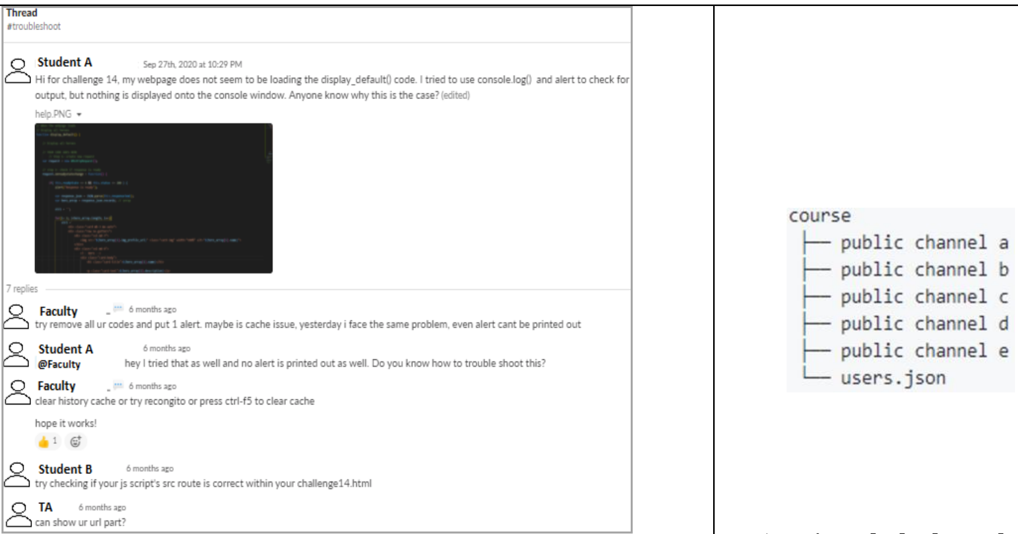
In the subsequent Fall semester, our university allowed a small percentage of courses to adopt hybrid classroom mode while a greater majority had to conduct all lessons fully online. Given the very hands-on nature of web development, WAD II instructors opted for the hybrid classroom option. In the hybrid classroom mode, students would take turns on a bi-weekly basis to alternate between online learning (work-from-home) and on-campus learning (in a designated classroom with appropriate social distancing measures in place) such that only about half of the class (~ 24 students per section) were allowed in a classroom each week. One week prior to the first lesson, only one-third of the students indicated in a survey that they would attend physical lessons. With a larger percentage of students learning-from-home while attending synchronous in-class lessons via Zoom, we determined that there would be a large gap in students' learning in terms of seeking help which would have been done in a much more convenient and efficient way by simply waiving a hand to ask questions or turning to peers sitting nearby in the classroom.

To overcome this challenge, some of the WAD II instructors adopted Slack to facilitate Q&A both inside-of-the-lesson and outside-of-the-lesson. Unlike the existing online discussion forum built into our university's LMS, Slack offers many useful and fun features suitable for programming-related courses and for the younger generation of students. For example, Slack offers a wide range of "stickers" (or emojis) that users can use to give "thumbs up" to show gratitude and "clapping" to congratulate and encourage others. As human body language and verbal tone may not fully translate in text messages, stickers or emojis serve as viable alternate means for communicating nuanced meaning (Tan, 2017). Further, Slack's "channels" provide an efficient way of organizing and searching through topic-based conversations. Lastly, Slack's API and data export feature (freely available for public channels) would allow for deep analyses of conversation threads in terms of identifying trending topics, profiling student learning based on Q&A activities and their interactions with other users, and development of automated analyses via programmatic means.

## 3. Methodology

### *3.1 Slack Terminology*

Instructors can create Slack workspaces for free. The free version comes with maximum 10,000 messages limit, but there is no limit to the number of users that can join a workspace.




**(Slack conversation thread in #troubleshoot channel)** **(Unzipped Slack workspace)**

*Figure 1.* Slack Conversation Thread & Unzipped Slack Workspace from Data Export.

Inside a workspace, one or more 'channels' can be created. A 'channel' is equivalent to a discussion 'forum', and each channel is a separate space where multiple conversation 'threads' can occur. Figure 1 (left) shows an actual conversation thread from WAD II course where usernames are anonymized. The thread reads from top to bottom, and it was started by 'Student A' posting a question (message) about a JavaScript issue. Subsequently, three other users (Faculty, Student B, and TA) replied to Student A's message. Similar to popular social media platforms such as Facebook, Instagram and Twitter, Slack allows users to 'mention' other users using '@' (at) symbol. Mentioned users are then notified.

### 3.2 Dataset

At the start of the semester, instructors and TAs regularly checked all threads to determine which threads are of importance to the teaching team for intervention purposes. In particular, the WAD II course teaching team was interested in Q&A discussions on course-related topics. To automate the identification of programming-related threads, we downloaded the WAD II Slack workspace via Slack's export feature so that computer programs could perform analyses. The data was extracted as zip files via Slack's management console. Figure 1 (right) depicts an example of an unzipped workspace. The exported file contains all public channels as well as user data (in users.json file), and it does not contain Direct Messages (DMs) as DMs are private message exchanges between pairs of individuals. A total of 1,772 messages from threads were used for our analyses.

### 3.3 Identification of Programming-Related Q&A Messages

In this section, we describe a text classification method for identifying programming-related Q&A messages. The objective was to classify Q&A messages into either programming-related or non-programming-related. As shown in Figure 2, we first performed text preprocessing which included lowercasing all words, removing all the Slack's mentions, normalizing certain words, and expanding contractions and slangs. The resulting texts were then transformed into Word2Vec (Mikolov et al., 2013) vectors to train a classifier. Next, we labeled our dataset so that our model could learn to classify. Various models were experimented with the Word2Vec vectors and labels using Positive-Unlabeled Learning (Sansone et al., 2019). An evaluation of the various models was done against F1, Precision, and Recall. We chose the best model based on the highest F1 score, and the best model would be used to predict whether a message was programing related or not at a later stage. As there were no labels for identifying which messages were programming-related, we needed rules or heuristics for labeling them. As WAD II course focuses on front-end web development, we used certain rules to automatically label the dataset to determine if the message was programming-related or not.

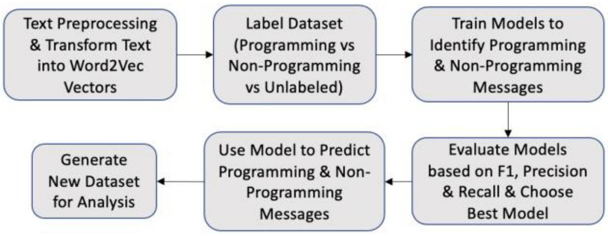



*Figure 2.* Summarized Workflow for Text Classification.

At the initial stage of labeling, we used three labels. The three labels were: 1) programming related, 2) non-programming-related, 3) un-labeled. With Positive-Unlabeled learning, we converted unlabeled text into either programming-related or non-programming-related. The rationale behind having a label called 'unlabeled' was because we used certain rules to label our dataset. However, our rules were not comprehensive, and it might not have picked up certain characteristics of web development-related terms such as HTML, CSS, and JavaScript keywords. Hence, instead of simply passing these texts as non-programming-related, we used Positive-Unlabeled Learning so that the classifier could learn the hidden characteristics of the unlabeled dataset by itself.

```
def label_message(text):
    If text contains <= 1 word:
       Return non-programming-related label
    Else:
       If the results for HTML tags are valid:
          Return programming-related label
       Else If there are results for JavaScript:
```

```
          Return programming-related label
       Else If there are results for CSS with ";":
          Return programming-related label
       Else If there are results for CSS with {}:
          Return programming-related label
       Else If the results for HTML tags are invalid:
          Return non-programming-related label
       Else:
          Return unlabeled
```

*Figure 3*. Pseudocode for Labeling Messages.

```
def find_html(text):
   Find HTML tags using <> within text
   If there are any HTML tags:
       Check if links and symbols are valid html tags or are programming-related
   Do another check to remove other stop words
   Return result
def find_javascript(text):
    Find Javascript using ";" and return result in a list
def find_css_colons(text):
    Find CSS using ":" and return result in a list
def find_css_brackets(text):
    Find CSS using "{" and "}" and return result in a list
```

*Figure 4*. Pseudocode for checking if a Message Contains HTML, CSS, JavaScript.

We first checked if there were more than one word found in the message text (Figure 3). If the text was only one-word long, the text was non-programming-related as HTML, CSS & JavaScript could not be done with a single word. Next, we performed a series of checks to see if HTML, CSS, and JavaScript code was found in the text (Figure 4). We checked if there were any HTML tags in the message using regular expressions. As URLs were also embedded in tags, we had to include certain URLs as stop words. If there were any HTML tags, we labeled them as programming-related. If the text contained ";", it was likely JavaScript. Hence, we labeled it as programming-related. We performed similar checks subsequently for CSS by checking the existence of ";", "{", and "}".

After performing all checks for HTML, CSS, and JavaScript, if the text was not yet labeled, we double checked to see if the text had previously been identified as containing HTML syntax but filtered away as invalid. In that case, we assigned non-programming-related label. Having non-programming-related labels was important as these labels would serve as reliable negatives for our classifiers to be trained using Positive-Unlabeled Learning.

To determine the classifier to be used for our research, we experimented with k-nearest neighbors (k-NN), Support Vector Machine (SVM), and XGBoost algorithms. We only used Word2Vec representation as our features to train the classifiers. Classifiers were trained using Positive-Unlabeled Learning which assumed two-class classification, but there were no labeled negative examples for training. The training data was only a small set of labeled positive examples and a large set of unlabeled examples.

There are mainly four methods in Positive-Unlabeled Learning: Rocchio, Naive Bayesian Classifier, Spy technique, and 1-DNF method. The Spy technique inserts positive labels (spies) into the unlabeled dataset. A prior study (Liu et al., 2003) reported the results of using two-step strategy for learning a classifier from positive and unlabeled data with a detailed evaluation of all the combinations of methods. Their results indicated that the models trained using the Spy technique achieved fairly high F scores (above ~0.8 macro-averaged F-score). Therefore, in our study, we decided to use the Spy technique. Classification performance results are shown in Table 1. SVM achieved the best result in terms of F1 and recall while XGBoost achieved the highest precision. Thus, for the final predictive model, we chose the SVM classifier with the highest F1 score.

Table 1. *Classifier Performance*

| Classifier | F1 | Recall | Precision |
|---|---|---|---|
| K-Neighbors | 0.859 | 0.810 | 0.914 |
| SVM | 0.897 | 0.886 | 0.909 |
| XGBoost | 0.872 | 0.823 | 0.929 |

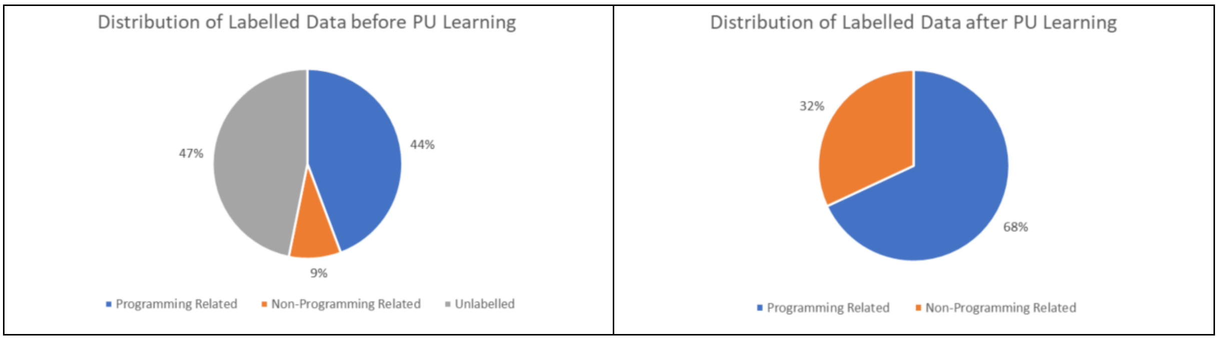



*Figure 5*. Distribution of Labeled Data before & after PU Learning.

As shown in Figure 5 (left), using our rule-based (heuristic) approach of labeling, 44% of the messages were labeled as programming-related, and 47% were labeled as non-programming-related. Further, after performing Positive-Unlabeled Learning, we could capture even more programming-related messages. As shown in Figure 5 (right), 68% of the messages were now labeled as programming-related and 32% labeled as non-programming related. Next, we performed an error analysis by calculating model confidence.

Table 2. *Top 5 Messages Incorrectly Predicted to be Non-Programming-Related by PU Learning*

| Message | Model Confidence |
|---|---|
| when you use float: right, the space beside the container with float:right becomes "empty" and the html code that comes after will be "moved up" to fill that "empty" space. when you add float: right to #side-bar, the code that comes after in the html code which is the main-footer is "moved up" and takes the "empty" space to the left of your side-bar which is empty when you did float:right. you should create "empty" space to the right of the main-content so that the side-bar container can "move up" | 92.9% |
| might want to try refer to this <https://www.w3schools.com/css/css_link.asp.You> could have possibly just clicked the link the style applied may not have been reflected | 89.2% |
| The clear property specifies how elements should behave when they bump into each other on the page. 1.left: the left side of the element will not touch any other element within the same containing element. 2.right: the right side of the element will not touch any other element within the same containing element. 3.both:neither side of the element will touch any other element within the same containing element. 4.none: the element can touch either side | 87.1% |
| can try the following to push the list style inwards to your content list-style-position: inside; | 86.1% |
| Try the code in the following sequence In your chrome browser: 1. Directly visit/load post.php 2. Go to form.html, fill out the form and press SUBMIT (which will lead you to post.php) 3. Go to form2.html, fill out the form and press SUBMIT (which will lead you to post.php) 4. Go to form_empty.html, fill out the form and press SUBMIT (which will lead you to post.php) | 81.9% |

Table 3. *Top 5 Messages Correctly Predicted to be Programming-Related by PU Learning*

| Message | Model Confidence |
|---|---|
| Hi guys, Please remember to add the <h1> tag with you own name. There are some websites which missed out this portion. | 51.0% |
| hi anyone know why such behaviour is displaying? I am currently using font-size:150% for the button which by right should display 1.5x bigger than the rest of the font and it looks fine when rendered as a website but when rendered in mobile version, the font shrinks and looks smaller in comparison to the rest of the text (Web Vs Mobile-Iphone 5) | 42.7% |
| prof, for $status in add, if we didnt prompt the user for values of status, can we | 41.8% |

| | |
|---|---|
| just use variable of $status = 'A' from the start and not use :status? | |
| ```echo "Hello, my name is $person";``` | 41.2% |
| navigate to the 2 folders, inside look for api/config/database.php inside line 9 ```private $password = "";``` change password to empty if your on wamp. or if you manually changed your own mysql password | 38.0% |

Table 2 shows the top five messages that were previously labeled as programming-related (by rule-based approach), and our Positive-Unlabeled Learning model incorrectly predicted them to be non-programming-related. We observe that the model confidence is in the higher range. Table 3 shows the top five messages that were previously labeled as programming-related (by rule-based approach), and our Positive-Unlabeled Learning model correctly predicted them to be programming-related. The model exhibits a lower overall confidence in this case.

This indicates that our rule-based approach might not be perfectly suitable for labeling messages despite high F1, precision, and recall scores. The rules we used might be more suited to identify code blocks, but the messages found in Slack are in the form of Q&A "discussions" where users not only post code examples but also write their thoughts and explanations in English. This can result in texts where plain English is mixed with code snippets which may not necessarily be syntactically correct (e.g. missing closing semicolon). Therefore, the heuristics in our rule-based approach must be flexible enough to account for such cases. One way to mitigate this problem might be to leverage human judges to assess and label the initial pool of Q&A messages, and the heuristics can be further refined based on verified human-labelled messages.

### *3.4 Q&A Participation Proficiency Types*

From the programming-related messages, we quantified the number of questions and the number of answered made by each student. Note that we only included 'explicit' expressions for the purposes of determining proficiency types *(and this is further elaborated in Section 3.5).*

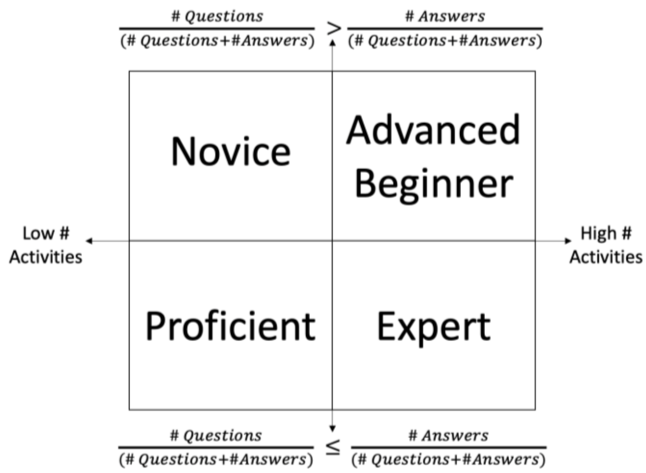



*Figure 6*. Q&A Participation Proficiency Types.

We categorized students into four Q&A participation proficiency types based on the number of questions and the number of answers they contributed in the Slack channels of the WAD II course. Figure 6 shows a quadrant of the four proficiency types. The *novice* are users who are not active and tend to ask more questions than to provide answers. The *advanced beginner* are users who are active and tend to ask more questions than to provide answers. The *proficient* are users who are not active and tend to provide more answers than to ask questions. The *expert* are users who are active and tend to provide more answers than to ask questions.

### *3.5 Q&A Participation Expression Types*

As described in Section 2.2, Slack has great support for a wide range of stickers to convey meaning. We define 'explicit' behavior to be direct textual verbalization of questions or answers. For example, if a student posts a piece of code and writes *"I don't understand why this code"* in the message field, this is considered an 'explicit' expression. On the other hand, if the student presses a "confused" sticker to *indicate* that he is confused (instead of writing out the text), we consider this an implicit expression.


*Figure 7*. Implicit vs. Explicit Expression Types.

As shown in Figure 7, if the number of stickers is more than or equal to the number of questions and answers, we categorize the student as exhibiting more of an 'implicit' expression behavior. On the other hand, if the number of stickers is less than the number of questions and answers, the student is considered to be exhibiting more of an 'explicit' expression behavior where he tends to verbalize questions and answers without using stickers.

## 4. Findings & Discussions

### *4.1 Q&A Participation Proficiency Types & Expression Types*

From the programming-related messages, using the Q&A Participation Proficiency quadrant (Figure 6) and Expression Types scale (Figure 7), we were able to profile student learning as shown in Figure 8.

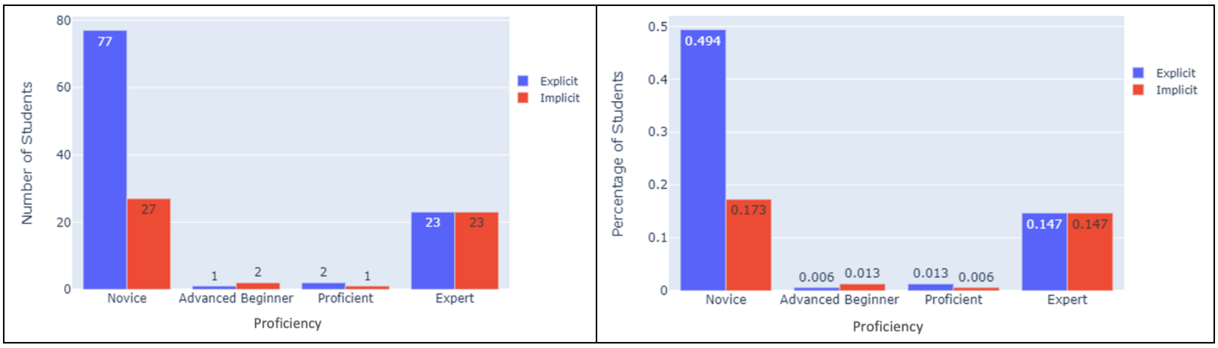



*Figure 8*. Student Learning Profiles (Proficiency Type & Interaction Type).

As shown in Figure 8 (right), over 66% of the students are categorized as "Novice" and over 29% of the students are categorized as "Expert" in programming-related Q&A discussions in Slack. Only a small number of students belong to "Advanced Beginner" and "Proficient" categories. Overall, the distribution of Q&A participation proficiency types appears to be un-even across the four proficiency types. There are two important findings that are valuable to WAD II course instructors.

**Finding (A) *"Novice" Students*:** They are not active and tend to ask more questions than to provide answers. There are two possible scenarios behind this: 1) "Novice" students can seek help independently (via Internet search or other means). Thus, they do not have any or too many questions to post or 2) they need help but are shy or intimidated to post questions publicly. Figure 8 shows the results from our end-term analysis where we profiled all students at the end of the semester over all Slack Q&A threads created up until that point. To employ appropriate and timely intervention measures, it is recommended that instructors perform regular profiling of student learning. By profiling student learning on a more frequent basis throughout the semester, instructors can verify whether students fall into the first scenario or the second one by giving frequent pop-quizzes or frequent knowledge checks.

1) In the first scenario, pop-quizzes will help students verify their own knowledge of course topics. It will also indicate to instructors which of the "Novice" students need help. Instructors can encourage the "Novice" students who perform well on pop-quizzes to answer other students' questions (in which case, they could bump up to "Proficient" or even "Expert" levels).
2) In the second scenario, instructors and TAs must intervene and offer help with course content. If students are too shy to post questions on their own, instructors can encourage

them to check Slack channels regularly and give stickers ('implicit' expression type) to questions posted by other students that they resonate with. Although it is an 'implicit' form of expression, from the sticker types, instructors can reasonably infer the students' intention and be able to tell whether certain students need help. In the exported Slack dataset, all stickers are shown as emoji texts (e.g. :confused:, :help:, :im_lost) that can be easily parsed and analyzed to detect sentiment.

**Finding (B) *"Expert" Students*:** the presence of over 29% of "Expert" students is very promising as they actively partake in answering other students' questions and helping maintain WAD II Slack workspace a vibrant discussion space. In terms of expression type, they are equally split into "Explicit" and "Implicit". Manual inspection of Slack threads reveals that the "Expert" students actively give stickers to other students in a form of encouragement (e.g. claps, cheers, thumbs up). Additionally, Q&A threads where "Expert" students who exhibit "Explicit" expression type point us to the next direction in our research. As mentioned in Section 2.1, WAD II had its first run as a core course in Fall 2020, and it repeats every Fall semester. From the Fall 2020 Slack Q&A threads, we can mine Frequently-Asked-Questions and corresponding answers provided by the "Expert" and "Explicit" students to build a knowledge repository which can be re-used in the future. Such a knowledge repository will help instructors better understand which topics are confusing to students and thus need more revision and hands-on exercises. It will also greatly help new students quickly find answers to programming-related questions.

### *4.2 Q&A Participation and Performance*

We used Pearson correlation coefficient scores between Q&A activities in Slack and student course grade to analyze the impact of online discussion forum participation on grades (Table 4). There is no significant evidence suggesting any correlation. In our analysis, we observe that while Q&A activities are highly skewed, student grades are normally distributed. This observation explains the lack of correlations between Q&A activities and student grades.

Table 4. *Correlations between Q&A Activities and Student Performance*

| | # Answers | # Questions | # Reactions | # Messages | Total # Activities |
|---|---|---|---|---|---|
| Pearson Coefficient r | 0.207 | 0.191 | 0.152 | 0.221 | 0.242 |
| P-Value | 0.0124 | 0.0215 | 0.0684 | 0.0075 | 0.0034 |

## 5. Limitations & Future Work

This research work is based on the case study of one faculty member to profile student learning. While we do not aim to generalize our findings, the technical steps of text classification involving heuristics and machine learning as well as categorization of students into Q&A participation proficiency types and expression types suggest a viable "automated" way of analyzing large volumes of conversation threads. As mentioned in Section 3.3, the existing rule-based approach can be further improved to identify programming-related messages. In our analysis, the percentage of "Novice" students was quite high at 66%. In our case study, we were not able to conclusively tell whether these students belong to the first or second scenario as described in Section 4.1. During the case study period, due to the COVID-19 pandemic, it was logistically challenging to plan pop-quizzes and fully automate student profiling. We plan to conduct another larger scale case study now with a fully automated student profiling system.

## 6. Conclusion

In this paper, we present a text mining approach to profiling student learning based on Q&A interactions in online discussion forums. Firstly, we perform text classification to categorize conversations into two categories: non-programming-related and programming-related. Secondly, from the programming-related conversation threads, our method categorizes students into four participation proficiency types (Novice, Advanced Beginner, Proficient, Expert) based on their Q&A activities. Proficiency types are determined based on the number and ratio of questions and answers posted by students. Next, our method determines whether a student adopts more explicit or implicit expression behavior in Q&A activities. Students resorting to stickers (emojis) more than words are considered to be exhibiting "implicit" expression type. Students that tend to use more English words to verbalize their thoughts and emotions than stickers are considered to be exhibiting "explicit" expression type. We evaluated our approach on a second-year undergraduate course, Web Application Development II.

## Acknowledgements

We would like to thank all teaching team members and students who participated in this study.

## References

Allen, I. E., & Seaman, J. (2015). Grade Level. Tracking Online Education in the United States. Babson Survey Research Group and Quahog Research Group, LLC.

Al-Salman, S. M. (2009). The role of the asynchronous discussion forum in online communication. *Journal of Instruction Delivery Systems*, 23(2), 8–13.

Bliuc, A., Goodyear, P., Ellis, R. (2010). Blended learning in higher education: How students perceive integration of face-to-face and online learning experiences in a foreign policy course. *Research and Development in Higher Education: Reshaping Higher Education*, Melbourne, July 6–9, vol. 33, pp. 73–81.

Gilbert, P. K., Dabbagh, N. (2005). How to structure online discussions for meaningful discourse: a case study. *Br J Educ Technol.*, 36(1):5–18.

Gerbic, P. (2010). Getting the blend right in new learning environments: A complementary approach to online discussions. *Education and Information Technologies*,15, 125–137.

Groeling, T. (1999). Virtual Discussion: Web-based Discussion Forums in Political Science. *Paper presented at the 1999 National convention of the American Political Science Association,* Atlanta, Georgia.

Hrastinski, S. (2008). What is online learner participation? A literature review. *Computers & Education*, 51(4), 1755–1765.

Jiang, Z., Zhang, Y., Liu, C., Li, X. (2015). Influence analysis by heterogeneous network in MOOC forums: what can we discover? *Int Educ Data Min Soc*, 242–249.

Liu, B., Dai, Y., Li, X., Lee, W. S., & Yu, P. S. (2003). Building text classifiers using positive and unlabeled examples. *Third IEEE International Conference on Data Mining*, 179-186.

Mazzolini, M., & Maddison, S. (2007). When to jump in: The role of the instructor in online discussion forums. *Computers & Education.*, 49. 193-213. 10.1016/j.compedu.2005.06.011.

Meyer, K. (2003). Face-to-face versus threaded discussions: The role of time and higher-order thinking. *Journal of Asynchronous Learning Networks*, 7(3), 55–65.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *ICLR*.

Picciano, A. G. (2002). Beyond student perceptions: Issues of interaction, presence, and performance in an online course. *Journal of Asynchronous Learning Networks*, 6(1).

Ragusa, A. T., & Crampton, A. (2017). Online learning: Cheap degrees or educational pluralization? *British Journal of Educational Technology*, 48(6), 1208–1216.

Sansone, E., Natale, F. D., & Zhou, Z. (2019). Efficient Training for Positive Unlabeled Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41, 2584-2598.

Slack. (2020, March 16). Distance learning thrives in Slack. https://slack.com/intl/en-sg/blog/collaboration/distance-learning-in-slack

Stephens-Martinez, K., Hearst, M. A., Fox, A. (2014). Monitoring moocs: which information sources do instructors value? *Proceedings of the first ACM conference on Learning@ scale conference*, p. 79–88.

Tan, R. L. (2017, July 23). How emojis have changed the way people communicate. https://www.straitstimes.com/lifestyle/how-emojis-have-changed-the-way-people-communicate

Tuhkala, A., & Kärkkäinen, T. (2018). Using Slack for computer-mediated communication to support higher education students' peer interactions during Master's thesis seminar. *Education and Information Technologies.*, 23. 10.1007/s10639-018-9722-6.

Wise, A. F., Marbouti, F., Hsiao, Y., Hausknecht, S. (2012). A survey of factors contributing to learners 'listening' behaviors in asynchronous online discussions. *J Educ Comput Res*, 47(4):461–80.