2019

Collaboration Pattern Model for Student Participation in Problem-Solving Typed Chat

Duy Quang Bui

Collaboration Pattern Model for Student

Participation in Problem-Solving Typed Chat

Duy Quang Bui

North Carolina A&T State University

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Department: Computer Science

Major: Computer Science

Major Professor: Dr. Jung Hee Kim

Greensboro, North Carolina

2019

The Graduate College

North Carolina Agricultural and Technical State University

This is to certify that the Master's Thesis of

Duy Quang Bui

has met the thesis requirements of

North Carolina Agricultural and Technical State University

Greensboro, North Carolina

2019

Approved by:

Dr. Jung Hee Kim
Major Professor

Dr. Xiaohong Yuan
Department Chairperson

Dr. Kelvin Bryan
Committee Member

Dr. Albert Esterline
Committee Member

Dr. Clay S. Gloster, Jr.
Interim Dean, The Graduate College

Biographical Sketch

Duy Quang Bui was born on November 18, 1993 in Viet Nam. He moved to the United States in 2010 to pursue his career in science. He obtained his High School Diploma, then an Associate in Science Degree in 2014. After obtaining a Bachelor of Science degree in Computer Science from North Carolina Agriculture and Technology State University in 2017, he decided to pursuit Graduate Master Degree in Computer Science focusing on Machine Learning. He published two papers under the title “Dialogue Acts Within Typed-Chat Collaborative Problem-Solving For a University Programming Class” and “Patterns of Collaboration Dialogue Acts in Typed-Chat Group Problem-Solving” while working on this thesis.

Dedication

This work is dedicated to my family and friends who support me through the long journey. I am not a very sentimental person and like to do things in logical ways. I hope my action will be a better example of how much I care and appreciate each and every one of you.

I would like to quote my grandmother who taught me how to cook and give me the best motto to live my life “Always Remember, The Best Thing has not Come Yet.” I keep this in my heart and remind myself something better is waiting in the future.

Acknowledgments

I would like to show much appreciation to Dr. Jung Hee Kim and Dr. Michael Glass for supporting me through my academy career from Undergrad to Master. Joining the COMPS research group has changed my life in that way that I have never imagined. This two and half year journey is definitely a milestone in my life. Undoubtedly, this thesis would not be perfectly completed without you all.

Also, it is my pleasure to work along with one of the brightest and the most creative groups of colleges whom I have learned so much from. To Matthew Trotter, Tamarick Kindrick, and John Carden, if we ever have a chance to work together, it will be my honor to work with you all again.

Partial support for this work was provided by the National Science Foundation's Improving Undergraduate STEM Education (IUSE) program under Award No. 1504918. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Table of Contents

List of Figures

List of Tables

Abstract

This project measures different collaborative dialogue acts between students who are working together to solve problems in a computer programming class. In COMPS (Computer-Mediated Problem Solving) exercises students work together via online typed-chat. Transcripts of these conversations were annotated with four categories of collaborative utterance: sharing ideas, negotiating ideas, regulating problem-solving, and maintaining communication. The annotated transcripts were then applied to answer four different research questions. A) Among the several students in a conversation, there are measurable quantitative differences in dialogue behavior that correlate with the relative preparedness for solving the problem. The most prepared student not only talks more but uses a different mixture of dialogue acts than the least prepared student. B) Different Teaching Assistants have different ways of interacting with the student groups when they join the conversations. This TA's behaviors are observable from dialogue act statistics, which ultimately correlate with a more didactic and a more Socratic style. C) The patterns of short two-turn sequences of dialogues are tabulated. The higher-probability two-turn conversational exchanges form the start of a probabilistic model of dialogue behavior. D) It proved possible to train machine classifiers to recognize the different dialogue acts, with only moderate accuracy. Using the previous turns of conversational text did little to improve accuracy.

The COMPS software package was considerably enhanced during the course of this work in service of collecting more and better data. The enhanced software automatically administers

pre- and post-tests, aids in forming student groups that incorporate more and less prepared students, and organizes and links the data collected during a COMPS lab session.

This research advances toward computer assessment of student collaboration skills and aids in the understanding of collaborative learning.

CHAPTER 1

Introduction

1.1 Problem Statement

Collaborative problem-solving skills are a recent addition to the skills that schools are required to teach and students are expected to learn (Bellanca, 2010). The COMPS (Computer-Mediated Problem Solving) project at NC A&T uses a computer typed-chat modality for students working together in small groups in college classes. Typed-chat is far from atypical in education or work settings. However face-to-face mode is typical in studies of educational dialogue (Wells, 2000, p. 51). People talk differently when they are online and collaborate differently from the normal conventions (Herring, Stein, & Virtanen , 2013).This has been documented in COMPS dialogues (Glass, Bryant, Kim & Desjarlais, 2015, p. 96). The research in this thesis endeavors to explore some of the phenomena of student typed-chat dialogue and makes these findings usable to computer software that will be monitoring the computer-mediated chat. It utilizes transcripts of the collaborative labs that are conducted in COMP 167 (the 2nd semester Java programming class). As the main research tool, this study uses an inventory of dialogue acts developed for a similar purpose by the Educational Testing Service (Hao, Liu, von Davier, & Kyllonen, 2017).

The main question of this thesis is: does examining COMPS dialogues in terms of the ETS's (Educational Testing Service) dialogue acts inventory helps people understand and evaluate the dialogues?

Within this broad question, this thesis addresses several questions:

1. Within a single problem-solving conversation group, does the most prepared student behave differently, with different dialogue acts than the least prepared or the middle student? Former COMPS studies have shown that learning gains are different. This study examines whether dialogue moves are different.

2. Do the TA dialogue acts affect the dialogue? Can we identify styles of Teaching Assistant interactions simply by counting their dialogue acts?

3. Can we use machine text classifiers to accurately identify dialogue acts within the conversation? Can we identify them using only limited context, e.g. only a single turn and possibly the immediately previous turn, so that it could be used in real-time as the dialogue proceeds?

4. Using the ETS dialogue act inventory, can we identify common two-turn dialogue act pairs, short snippets of conversation such as student A suggests an idea followed by student B negotiating? These pairs would form the basis of a deeper conversation analysis of student behaviors.

1.2 Context

1.2.1 COMPS Research Context

The four parts of this research have different but overlapping applications.

1. The difference between the least and most prepared students could be helpful in designing problems or problem-prompts that engage more students better. It could also be useful in linguistic studies of educational dialogue which analyzes the different roles that students adapt within the group and within the group-cognitive exercise (Stahl, 2006).

2. From counting the dialogue acts in our data, we have identified two distinct styles of TA interaction. This can be the start of a new study of how to supervise COMPS dialogues better.

3. The machine-identification of dialogue acts has several potential uses. One is in real-time monitoring of conversations, feeding data to the professor through the COMPS dashboard. Another would be to automatically tag large quantities of dialogue data, enabling statistically significant studies.

4. Finding statistically significant patterns of two-turn conversation segments would be the beginning of building larger models, such as Markov models of how dialogues probabilistically transition from dialogue state to state.

1.2.2 Dialogue Act Research Context

Any analysis of dialogue data has to start by identifying the elementary phenomena within the data. To address the mandate to teach collaborative problem-solving skills, PISA (Program for International Student Assessment) and ETS (Educational Testing Service) has been studying how to assess these skills (Herborn, Stadler, Mustafić, & Greiff, 2018). The PISA and ETS assessments put the student to be tested into a group where the other "students" are computer agents . They communicate using natural language. This means that PISA and ETS research teams have recently been analyzing typed-chat collaborative dialogues. Even though the research results produce proprietary testing products, the ETS team has published some of their work . From studying transcripts of a great many sessions of students working together, they created and validated their own inventory of dialogue moves that students employ during their discussion (Hao et al., 2017). This inventory is new, having first appeared in print in 2017. It is unique in that it is used by a commercial assessment product. Therefore it is possibly the most tested dialogue act inventory in the literature.

1.2.3 Annotation of Dialogue Context

For most linguistic dialogue research, the researchers have to annotate the target linguistic acts within the dialogues. This usually starts with manual annotation. Researchers read transcripts and manually mark down every place they see the target phenomenon happening (Rosé et al., 2008, p. 237). In order to annotate reliably and consistently, a markup manual has to be developed: a document with guidelines and examples. Another difficulty is that it is preferable to annotate the dialogue turn-by-turn with little or no context from the previous conversation. Although we had the ETS list of dialogue categories, they did not publish their markup manual. We had to write our own, based on trial-and-error annotating and re-annotation many turns of dialogue. The good part is the manual itself becomes a base enabling new research on further dialogues.

1.2.4 COMPS Software Context

In addition to investigating the four research questions, this thesis also describes major upgrades to the COMPS software system that improved the ability to deliver exercises to classes, and at the same time improved the ability to collect data. At the beginning of this thesis work, COMPS was a stand-alone application for students to use for chat exercises, with participation from the instructors and teaching assistants. All the other activities during the lab period: checking students against a class roster, administering pre-tests, assigning students to groups, administering post-tests, after lab survey, were performed separately from the online exercises, resulting in considerable paperwork. The other main problem was linking together the data: the pre-test with the chat groups and transcripts with the post-test with the survey. This thesis describes the software which was added to the COMPS suite in order to tie all this activity together.

1.2.5 COMPS Machine Classifier Context

Previous COMPS research has studied several different dialogue phenomena. A study of domain reasoning within the dialogues --- how often the students uttered sentences which contained substance from the problem-solving --- was somewhat successful. (Willis et al., 2017, p. 11). This work was able to train machine classifiers that identified substantive utterances with reasonable accuracy. A study of conversational interactivity, attempting to identify by the computer when a student's turn addressed another student, was less successful (Glass et al., 2014, p. 31).

The author also attempts to find a pattern of collaboration based on the previous conversation of students to improve the accuracy of the machine classifier. Any finding that included in this thesis is generated from COMPS transcript and proving consistently throughout the experiment in a collaborative learning environment.

1.2.6 Limitations of this Research

There are unknown factors that may limit the generality of this research. The pattern of collaboration may be hard to apply from one system to another because of the randomness of many internal and external factors. A change in a chat interface or system of questions could produce a new pattern of interaction between students. A student's level of preparedness and knowledge of the subject can change how the group deals with the problem-solving conversation. A teaching assistant involvement can make students forfeit their group collaboration and change the conversation with the pattern to a different situation. However, there is no way to know how applicable these results are until the next researchers try to apply them to new conversations in the future.

1.3 Chapters of this Thesis

- Chapter 1 describes the research plan, its motivation, and limitations
- Chapter 2 describes COMPS and its exercises
- Chapter 3 describes related research and earlier COMPS research
- Chapter 4 describes the software updates for administering COMPS exercises and gathering coordinated data
- Chapter 5 describes the dialogue act annotation task and the markup manual
- Chapter 6 describes research results
- Chapter 7 is conclusions

CHAPTER 2

COMPS Description

2.1 COMPS (Computer-Mediated Problem Solving)

The COMPS project is a collaboration between Valparaiso University and North Carolina A&T State University, where the project is based (Desjarlais, Kim, & Glass, 2012). COMPS is built to help an instructor to gain in-depth information about what happened in his or her lab section. The instructor can choose to intervene or guide student conversation to reach the lab goal. COMPS shows real-time analysis with important data such as student participation rate and their progression within the lab. With COMPS, the instructor has the tool to make the lab exciting and engaging to every participant.

COMPS exercises are used lab sessions of the COMP 167 Java class Students work in groups of three to four (Kim, Kim, & Glass, 2016, p. 68). Lab sessions start with a pre-test, then students are assigned to a chat group. They discuss and solve the problem together using the COMPS web-delivered typed-chat interface. Students in one group are seated around the computer lab, so all communication is through the computer and recorded. Students then complete a survey and a post-test.

In a conversation, each student has to explain his or her reasoning to the group. They are instructed to obtain a consensus answer before submitting it to the lab instructor or teaching assistant. Each group dialog is being recorded, including the interactions with the lab instructors. These log files of conversations are then analyzed by our researcher, who assigns dialogue act categories to each turn of a conversation.

2.1.1 COMPS Interface

COMPS has been through several updates on its interface. The chat affordance has a very interactive feel. Students can all type at once and read and respond to each other's dialogue in real-time as they type, it isn't necessary to wait for conversational turn-taking (Glass et al., 2015). A dashboard has been tested to show measurements of the state of the conversation groups to the instructor. There are also observer windows for the lab instructor and TAs to figuratively look over the shoulders of the students, to oversee student conversations and answers.

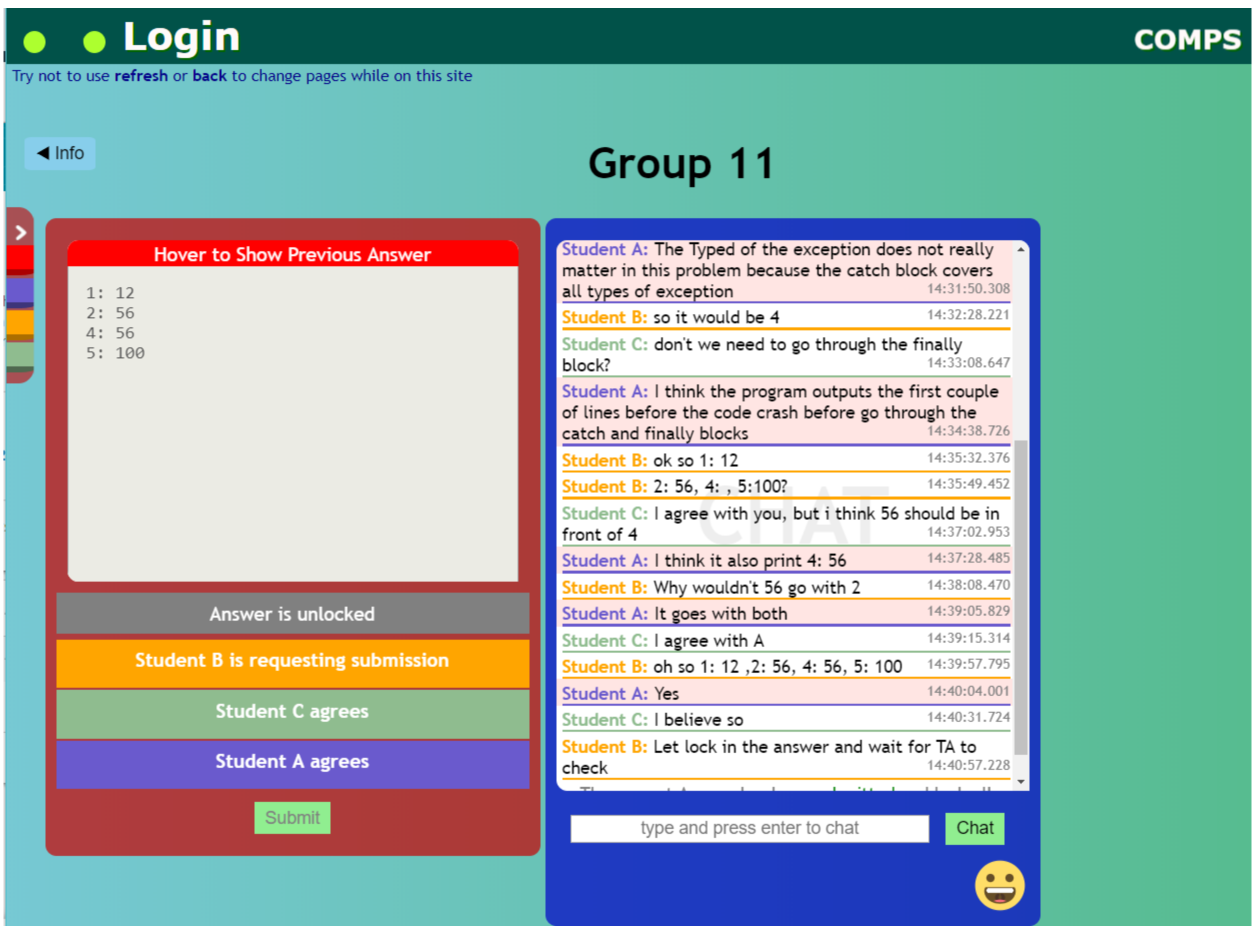

Figure 2.1. COMPS Chat Student Interface

Figure 2.1 shows our current student-facing interface. There are three main components of this interface. The Chatbox in blue on the right is where all the conversation takes place. Students have to work together to solve a set of problems that relate to their class. The answer box in red on the left allows only one student typing at the time, the group answer must be submitted by one student with an agreement of others. The answer box locks and unlocks by the lab instructor, an option to join the conversation and provide feedback in the Chatbox. This system was built to ensure all students participate in the lab. Making the students put their answer in the answer box also ensures that the agreed-on answers are clear to the students. Also the answers don't have to be teased out of the conversation transcript, by the instructor during the lab session and by the researchers later.

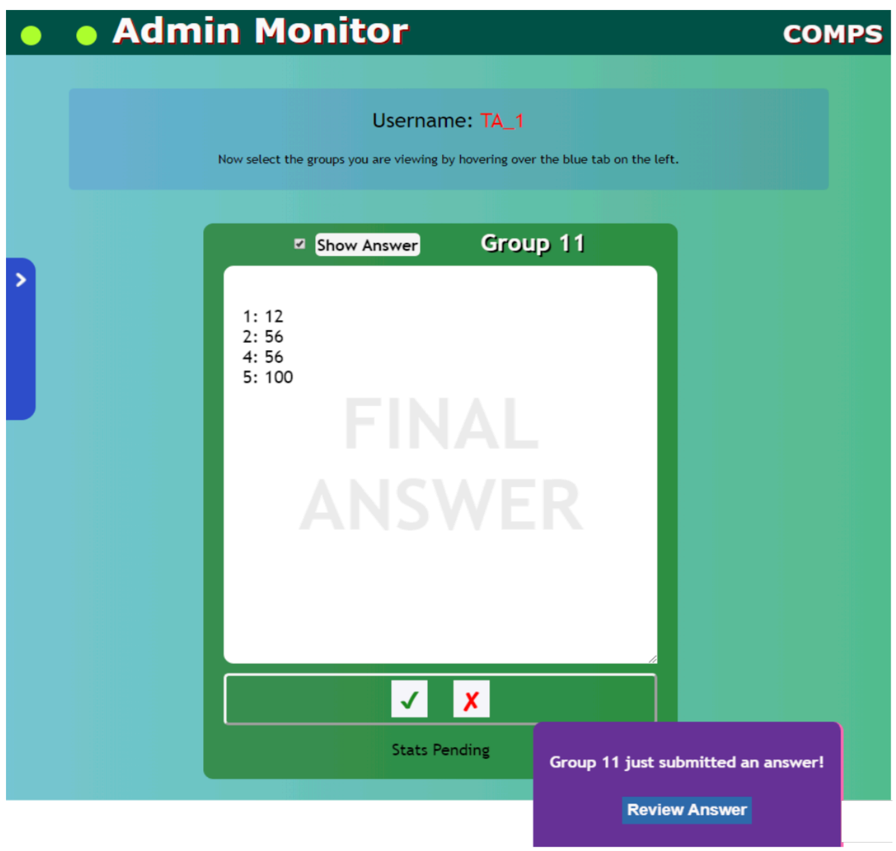

Figure 2.2. Admin Monitor Interface

Figure 2.2 is a compilation of all groups in the section with an option to the lab instructor who can join any group chat to provide instant feedback. When the group submits their final answer, a notification prompts the instructor to check for correctness. A lab observer can provide an explanation to the group.

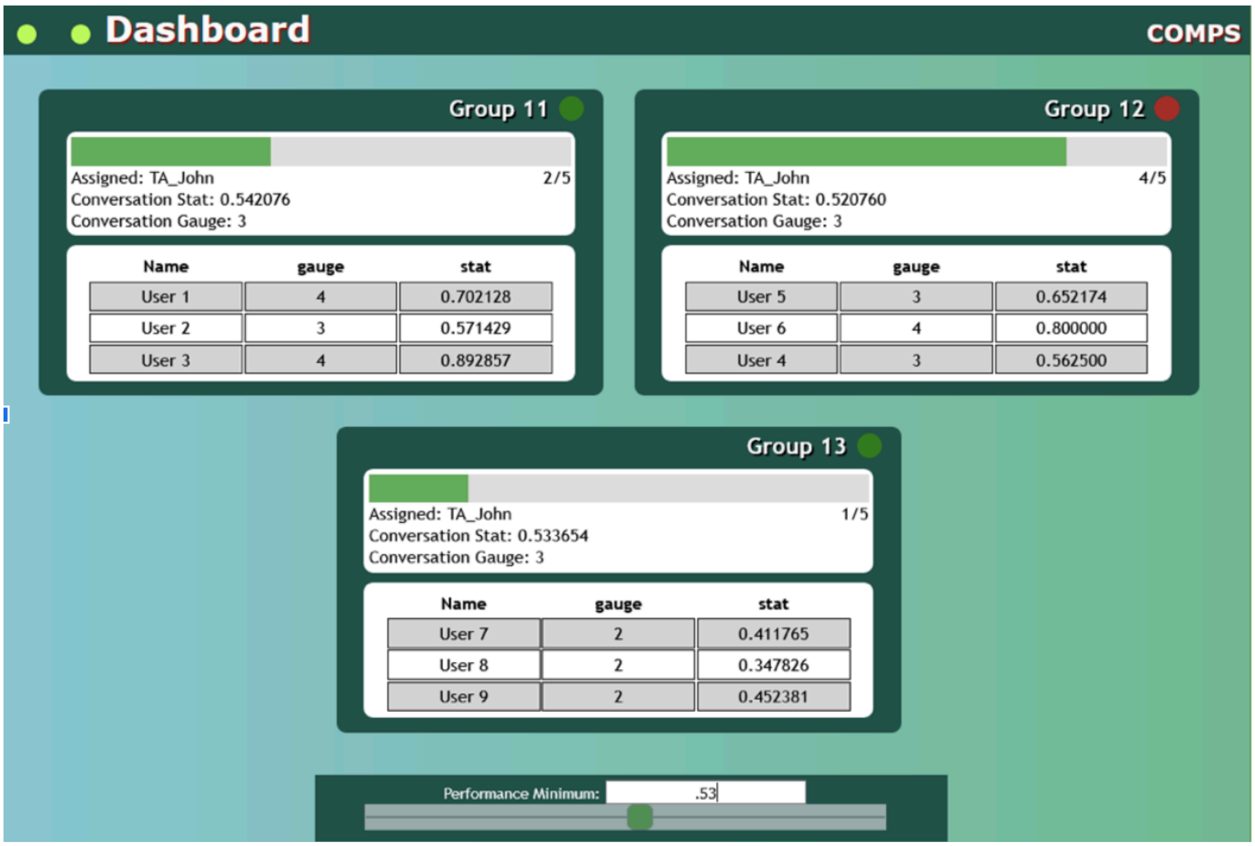

Figure 2.3. Dashboard Interface

Figure 2.3 shows the dashboard, providing the lab instructor a bird's-eye view of the lab. Each group member has their stats display such as how much they contribute to the conversation. The group which is disengaged is highlighted when its stats fall below the performance threshold. The students' stat is monitored to identify which student is lurking and not contribute to the problem-solving section. A gauge on top of each group indicates group progression which is intuitive to use. This dashboard is experimental and still in development. The overall statistic for a conversation is a combination of: a measure of whether all students are participating, and percent of dialogue turns that are judged (by a machine classifier) to contain substantive material.

2.1.2 Addition data (pretest, posttest, lab exercise, and survey)

```
public static void main( String [] args ) {
     int i = 0;
     Foo f = new Foo();
     Foo [] fooarray = new Foo[5];
      System.out.println( f.toString() );
      // A

     for (i=0; i<fooarray.length; i++)
          fooarray[i] = new Foo( i, i );
     // B
     fooarray[fooarray.length-1] = f;

     System.out.println( fooarray[fooarray.length-1].toString() );
     // C
}
```

Analyze the main method down through the line marked `// A` in the code.

P1) At the line marked `// A` in the `main` method how many objects of class `Foo` have been created?

- a) 0
- b) 1
- c) 5
- d) 6

Figure 2.4. Lab Pretest - Posttest Example

Figures 2.4 and 2.5 show part of a typical Java problem for group discussion. A copy of a pretest and posttest with answer is provided in Appendix A. This problem is multiple choice. Most problems are constructed so that student needs to utilize the Java concepts. Usually students need to analyze code, rather than writing code. In the Figure 2.4 problem, the student needs to know that creating the array which can hold 5 Foo objects does not create the 5 Foo objects which will go in the array. Furthermore the array itself is an object, but not a Foo object. So the correct answer is that 0 Foo objects have been created at point A in the code.

The student is tested for preparedness by completing a pretest before the lab takes place. The questions relate to what they recently covered in class. The pretest is straightforward with each question focusing on one topic in the lab. The posttest is usually harder in the same format. The researcher expects students to learn through working together in a collaborative lab. Learning gain is calculated as (posttest score - pretest score) / (full point score - pretest score). Earlier COMPS studies have shown that the least prepared student learns and benefits most throughout the lab (Kim et al., 2016).

Analyze the main method down through the line marked `// A` in the code and answer the A questions, then get the TA's response.

A1) At the line marked `// A` in the `main` method how many objects of class `Foo` have been created?

A2) What other object has been created and what is its type?

A3) What was printed the the line just before `// A`?

A4) Local variables are kept on the stack or the heap? Objects are kept on the stack or the heap?

A5) If `int` variables are 4 bytes and reference variables are 11 bytes, how many bytes of stack are being used?

A6) How does the `static int z` variable in the `Foo` class differ from variables `x` and `y`?

A7) At the line marked `// A` what is the value of `z`?

Figure 2.5. Lab Exercise Example

Figure 2.5 shows more questions from lab exercise . A copy of a lab exercise is provided in Appendix B. The questions are short, and they also prompt the students to explain their thought process. Each question is a stepping stone to help students understand the whole picture. The goal of the lab is to set a foundation for every student in the lecture material.

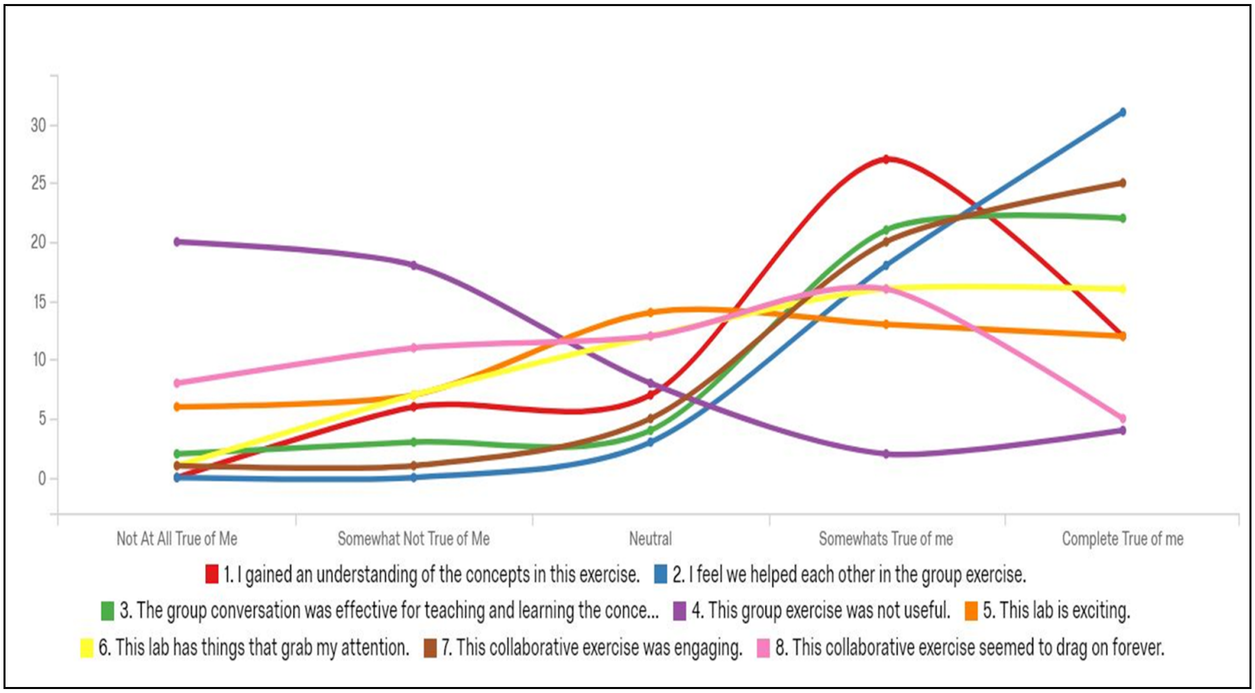

Figure 2.6. Lab survey visualization

Figure 2.6 shows the after-lab survey results. Students generally express good feelings toward the lab exercises. In addition, they can learn or establish a better understanding of class material by discussing with their classmates. This is an indication of COMPS bring change to students' normal lab routing with an exciting learning opportunity for both students and instructors alike.

CHAPTER 3

Background

3.1 COMPS Earlier Work

COMPS originated as Wooz-tutor (Kim & Glass, 2004, p. 358). Its earlier applications were to facilitate and record tutoring and collaborative problem-solving in mathematics and other STEM-related subjects.

The notion of characterizing individual turns of dialogue according to dialogue acts is due to conversation analysis, a branch of discourse linguistics (Stubbs, 1983). Conversation analysis often examines and counts extremely short sequences, such as two- and three turn sequences characterized as Initiate, Respond, and Follow up. The Initiate/Respond/Followup type of analysis informed an earlier attempt to classify and measure conversational interactivity in COMPS (Glass et al., 2014, p. 31). The machine classifier trained to identify these exchanges was not convincing enough. However, the software to process transcripts and extract features became the starting point for the machine classifier software used in this thesis.

The authors of that earlier work noted that some of the problems might have resulted from inconsistently annotated training data. The present author utilized procedures to raise the quality and consistency of manually-annotated data.

- Annotation is more reliable when there are more raters. In this research, the annotation was done by two annotators who marked up the transcripts individually. They periodically compared notes and discussed thoroughly to ensure the final consensus annotation is closest to correct. Frequently cross-checking also helped them to achieve consistent judgments.

- The annotation categories should build upon existing theory. This study adopted the dialogue act categories from the ETS research into the assessment of collaborative problem-solving (Hao et al., 2017). These categories were updated a bit when it became evident that they didn't cover all the phenomena seen in the COMPS transcripts. It was decided to use the four higher-level dialogue act categories, and not depend on the fine-grained subcategorization into over 30 sub-categories.
- Creating a markup manual, a set of instructions and examples that annotators can refer to. This manual serves as a reference manual for learning how to annotate the transcripts. It is also like case law, a record of prior decisions about example, ambiguous, and difficult cases.
- Students typing simultaneously makes the transcript annotation complicated (Glass et al., 2015). Among two conversational turns that overlap in time, it can be hard to discern from the words themselves who came first and who is responding. Therefore, when working on the transcript, the annotator has to pay close attention to the timestamp and create a working scenario of the order of conversation.

Earlier COMPS research also annotated dialogues based on other features of the dialogue, notably whether the students were reasoning about the material (Willis et al., 2017, p. 11). "Reasoning" does not refer to logical thought. It was broadly defined as the student mentioned something about the topic and also used non-topic words within the same dialogue turn. The machine classifier software for that experiment introduced topic modeling features, described below in chapter 6. This work helped produce the conversation statistics shown in the dashboard in Chapter 2 above.

3.2 Literature Survey

3.2.1 Virtual Math Team Project

VMT (Virtual Math Team) project is on a similar mission that this research is focused on (Stahl, 2006). They built an online collaborative environment where participants talk to each other through typed-chat to solve algebra and geometry problems. The VMT work describes "group cognition," the reasoning processes that emerge from the interaction of multiple people working together. The data that they collected also points out that group cognition can be difficult to understand. Among the challenges is the need to study both "Social Knowledge Building," the shared understanding that can be observed in the discourse, and the personal understanding of the participants. These processes are different from person to person and culture to culture. Without interpersonal interactivity, people are not obligated to listen very well and answer in a timely and productive manner. Group interaction changes quickly based on the different levels of knowledge sharing between group members. Members with a deeper understanding of the subject dominate the conversation.

VMT follows four pillars of computer support for collaborative learning (CSCL) which are:

- Collaborative learning with shared information and knowledge.
- Group and personal cognition.
- Computer-mediated, the people communicate via computer and the shared knowledge is visible to everybody via computer.
- Analysis of behavior and interactivity.

In this thesis, the researcher also goes through the same checklist. The researcher calculates learning gain and offers change to the lab's overall experiment. Group and personal behavior of students and teaching assistants are discussed and learned. COMPS integrates a dashboard to keep track of student's activity and bring attention to the group that is in trouble.

One lesson from the VMT conversation studies was: don't try to understand the conversations too deeply. This research deliberately focused on answering research questions using a fairly simple annotation of individual turns, without models of the psychological and social processes within and among the participants. It is interesting to note that this study also found the same problem of the more knowledgeable students unbalancing the conversation. Our dialogue act statistics indicate the group cognition may become more like a tutoring session by the most prepared student.

3.2.2 Human-Computer Interaction Institute

Several studies from The Human-Computer Interaction Institute based at Carnegie Mellon University informed how this COMPS work was done. One study attempted to measure rapport among students in a problem-solving exercise (Ogan, Finkelstein, Walker, Carlson, & Cassell, 2012, p. 11). Their results in analyzing data from the transcript have been fascinating. They report that when a group of students were playful to each other, or when students encounter a high level of face threat, the group learning gain increases. The relationship between the tutor and tutees also changes the base of how the conversation is carried out. The more comfortable they are together, the better they learn. These studies showed that it is possible to detect and measure what seems like intangible social and conversational qualities. They also trained computer classifiers to attempt to detect these qualities.

In COMPS, we start to notice patterns of social phenomena that might be detected from dialogue behaviors. In this thesis, the researcher decided to categorize two styles of Teaching Assistant as "tutor" and "mentor." A tutor-style main goal is to teach the group and lead them to answers quickly. A mentor challenges the group in critical thinking and being resourceful yet avoids giving away a direct answer to the group. As a result, group learning gain and interaction change based on the style of Teaching Assistant.

An Embodied Conversational Agent (ECA) is when the computer engages as one participant in the conversation (conversational agent). The agent is presented to the user through a named human-like image on the screen and maybe a voice (embodied). Early work on virtual humans capable of providing meaningful interaction between humans and computers in verbal or nonverbal conversation was derived from annotating transcripts in the same manner as we have done here (Cassell, Sullivan, Churchill, & Prevost, 2000). Their machine-learned behaviors used a rich set of text features drawn from several consecutive dialogue turns. This created a context in terms of conversation for the ECA to follow. Dialogue is a social behavior, ECA manages to imitate dialogue behavior because it learned from exchanges between participants. In this thesis, the researcher took the inspiration and implemented a similar idea of using the previous turn of conversation to help identify the dialogue act in the next turn.

3.2.3 TagHelper

Carolyn Rose and her team are aiming for using text classifier to classify data from CSCL system (Rosé et al., 2008, p. 237). The goal is for a computer to annotate dialogue act tags, matching dialogues that have been manually annotated. This work is analogous to the effort in this thesis. It uses a different set of dialogue act tags that works in four different dimensions. The classifier algorithms they use are decision trees, support vector machines, and Bayes.

The results show that the machine classifier only agrees 60% percent with human annotators, and the average result of 0.59 precision, 0.37 recall, and a kappa of 0.44. These results are similar to the results reported in this thesis, using linear classifiers with different text features on a similar mission.

A suggestion from Rosé is to improve the readability of the manual annotation. It is a difficult task. The requirement of clear and full instructions between our two annotators are in order to improve the classifier result. The researchers of this thesis attempt to produce a working manual and set of rules for difficult cases.

Rosé also points to the direction of using sequential or non-sequential turn as a context for the classification of dialogue acts. It is a difficult task in unknown territory that the CSCL community is heading to.

CHAPTER 4

Delivery and Data Collection of COMPS Exercises

4.1 Webpage Delivery

Until recently, the COMPS project only had an online chat application (which had been specialized for this purpose). Integrating COMPS into a classroom setting, with class rosters and lab schedules and tracking data from individual students, is a level of complexity. Integrating COMPS into a research setting, with pre- and post-tests, surveys, matching of data from different parts of the labs, is another level of complexity. The researcher wrote a web page that addresses these administrative functions (Bui , 2018). It is a one-stop web page where instructors can register classes, upload student rosters, and schedule lab sessions. This project provides the tools for professors to integrate COMPS exercises into a regular classroom curriculum. The web site delivers the lab experience. The students sign into this web page, where they are guided through pre-tests, the lab sessions, post-tests, and surveys. These disparate student activities and resulting data are hosted by different websites, but the control page provides one place for the student to log in and links together the separate records for each student. The aim is to produce a simple and friendly interface to export our COMPS curriculum to students and faculty from around the country.

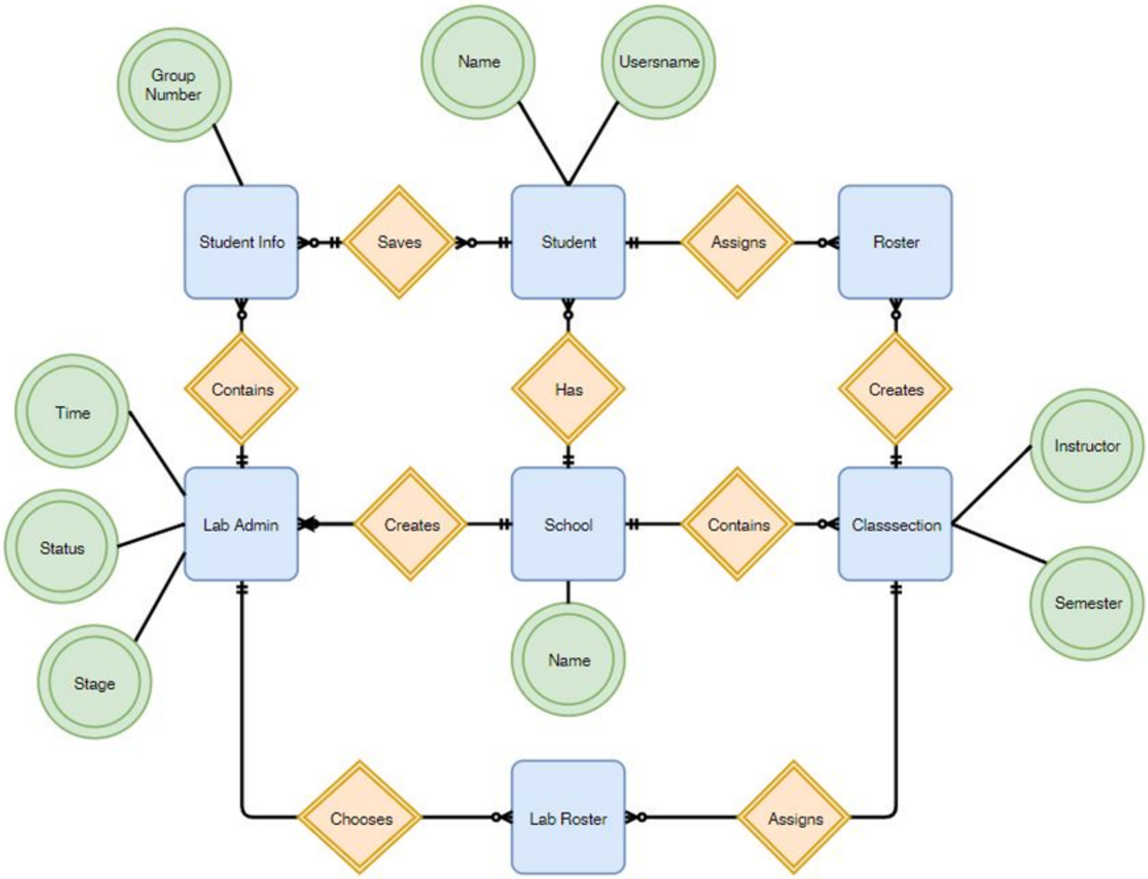

Figure 4.1. COMPS webpage design

The figure 4.1 is system design with keys function such as :

- The class roster can be extracted as CVS file from Blackboard.
- Quick and Easy setup for classes with student name and handler automatically generated and matching with other sources of data.
- Track students' activity and assign discussion group numbers.
- Professor/Teaching Assistant use web- page to sequence activities during the lab time.
- The server is available 24/7 and protected.

4.1.1 Webpage Interface and Lab Setting with Qualtrics Integrating

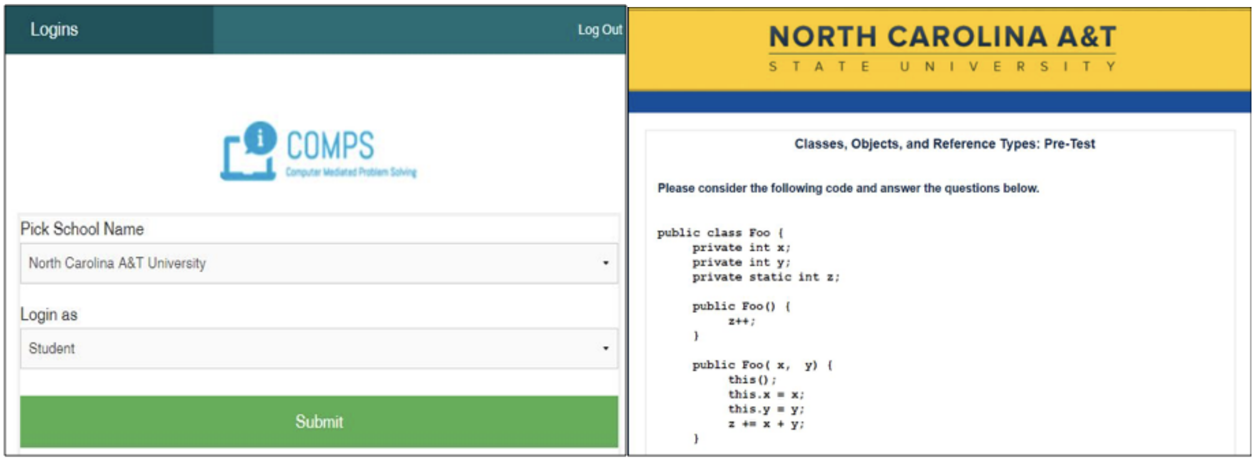

Figure 4.2. Student progressing through the webpage

The students are required to log into a webpage and select their name from the list which minimizes the inconsistently from asking students typing their names every time they perform pretest, posttest, and surveys and on Chat engine. Qualtrics is an online experiment management tool also being integrated into this webpage. Qualtrics help with auto-grading of pretest and posttest as well as hosting a survey section where data can be quickly visualizing based on research interest.

4.2 Data Collection

COMPS is collecting data from a different source and use them to learn about student interaction in a collaborative problem-solving environment. The collection of different data source including student and teaching assistant transcript, student record in requirement test, student survey are used in:

- Measure learning gain from the lab exercise.
- Calculate and determine the effectiveness of the lab itself.
- Revision of question is needed if the question is too hard.

- Strengthen students' knowledge in a particular area is needed if too many students' answer is incorrect.
- Learning of students' transcript and the data stream from the conversation.
- Understand students interest of the lab and how they feel about the lab in general.

Data collected from COMPS Chat Engine and COMPS web pages are being used to find a correlation between the conversational pattern of problem-solving to result in student learning improvement and experimenting with the lab.

4.3 Cyber Security and Privacy

With the raising of the cybersecurity issue and concern of handling students' information, COMPS Web-page follows the guidelines from the graduate research requirement and carefully protects students' data (Carden, Billie, Kim, & Glass, 2017). Transcript data that is collected in Chat Engine have been masked and send away to Valparaiso University under the anonymous name with compliment data transfer protection. Student data is saved and encrypted in one secure location including student grade in pretest, posttest, and survey and can not be accessed without the administration's permission.

4.4 Challenge and Future Work

COMPS web-page and COMPS Chat Engine still need improvement to accomplish the full delivery of COMPS exercise. The list is following from the researcher's recommendation and rank based on urgency:

- Functionally requiring that chat engine allows multiple uses of the lab at the same time.
- Admin function still in ChatEngine

- Streamlining the lab process including improvement of User Interface and adding smart feature reduces manual input from users
- Establish better communication between Qualtrics and COMPS page to enhance user experiment for both students and professors

CHAPTER 5

Markup Manual of COMPS Transcript

5.1 Markup Manual

Two annotators read and annotated each dialogue turn, where a turn most often contained one dialogue act but could contain several. They trained on several other dialogues until they achieved a satisfactory degree of inter-rater reliability. The differences in the seven multiply-annotated dialogues were then resolved by consensus.

Small amounts of dialogue before students started addressing the exercise (e.g. logging in and greeting each other) were excluded from the annotation and analysis. The segments of dialogue containing TA interaction were marked so the difference in dialogue style during TA participation could be studied.

The categories of dialogue acts were obtained from a collaborative skills assessment task under development by the Educational Testing Service (Hao et al., 2017), as follows:

- Sharing Ideas: The participant shares their idea to the group. The idea has to be task-relevant or information that contributes to the problem-solving process.
- Negotiating Ideas: The participant modifies or reacts to ideas already on the table, e.g. by agreeing/disagreeing, rephrasing, pointing out a gap, or asking for clarification.
- Regulating of Problem Solving: Examples of these dialogue moves are suggesting the next step or goal, expressing frustration, checking to understand, and evaluating whether some part of the discussion has been useful.
- Maintaining Communication: The participant engages with the group by attending to conversational or social norms or contributing to social interaction, but the turn is not part

of the cognitive work of solving the problem. Some examples are apologizing, offering help, regulating turn-taking, conversational repair, and responding with emojis or displays of affect.

Table 5.1 shows illustrative dialogue turns for each of the four categories. A few of the common sub-categories of each category are shown.

Table 5.1. Four Dialogue Act Categories with Common Sub-categories

Sub-Categories Label	**Description**	**Example Dialog**
Sharing Ideas -A1	Student gives task-relevant information (e.g ., individual respond) to the teammate	“I think for a and b they are both public and c is private”
Sharing Ideas -A2	Student point out or question a resource to retrieve task-relevant information	“ok for 2 if its called without a object then i know that it uses this.() to call it”
Sharing Ideas -A3	The student responds to the teammate's request for task-relevant information	“ encapsulation allows validation”
Negotiating Ideas -B1	The student expresses agreement with a teammates	“agree, so I think a, b, c should be doubles” “I'm wrong about 2 and ok”
Negotiating Ideas -B2	The student expresses disagreement with teammates	“i dont think calculatePayment needs institution name.”
Negotiating Ideas -B3	The student expresses the uncertainty of agreeing or disagree.	“I am not sure”
Negotiating Ideas -B4	Students ask the teammates to repeat or clarify a statement	“it would be the three formal parameters for number three right?”
Negotiating Ideas -B5	The student collaborate or change on his or her own idea	“ For that , I believe it is c”
Regulating of Problem Solving - C1	The student suggests the next step or suggests the group role before continuing	“right so a is done let's get on b” “anyone else care to be the leader?” “Are you okay with being the leader?”
Regulating of Problem Solving - C2	The student expresses confusion/ frustration or lack of understanding	“Im a bit confused, I know it all, but only how z is 26!”

Regulating of Problem Solving - C3	The student expresses progress in understanding/student reflects on what the group did	"oh I get it now so the fourth foo would add 1 to 2+2 to z is it because Z is static and x an y is non static"
Regulating of Problem Solving - C4	Student checks on understanding	"Having it be 4 when you give it to the other team is what determines the winner, based on what number they select to remove. Does that make sense?"
Regulating of Problem Solving - C5	Student shows satisfaction with the group performance	"good job everyone didn't think we would get past 1 tbh lol"
Regulating of Problem Solving - C6	The student asks or explains how to interact with the environment	"do we just click a @ symbol back and forth?"
Maintain Communication - D1	The student greets teammates or engages in small talk	"hi" "whats up" "Isn't this a blast?"
Maintain Communication - D2	Input does not make sense	"wooz? Lol"
Maintain Communication - D3	Student answers a question	"Good question"

5.2 Successful Collaborative Dialogue

The elements of successful vs. unsuccessful collaboration are based on two main factors:

- Learning (Chiong & Jovanovic , 2012, p. 81).
- Collaborative Participation (Glass et al., 2018, p. 145).

The researchers measured learning gains over the course of the lab exercise from the pretest and posttest (Kim, Kim, & Glass, 2016, p. 68). Learning gain is calculated by:

(posttest score - pretest score) / (full point score - pretest score)

However, learning gain is not by itself a marker of successful collaborative problem-solving activity. COMPS utilizes several measurements to gauge whether students are contributing. A turn participation parameter (Kim et al., 2016) measures the fraction of dialogue turns contributed by a participant on a scale of 0 (lurking) to 1 (dominating the discussion). This number is normalized so that if a student is contributing an equal fraction of the turns (1/n of the turns in an n-person group), the parameter is 0.5, it is further scaled by a logistic function so that extreme values are compressed. The COMPS project has also experimentally trained classifiers to recognize the fraction of substantive turns. A substantive turn was shallowly defined as one containing both problem-related words plus other words (Willis et al., 2017).

There was also a manual judgment of the degree of collaborative dialogue by our annotators. The analysis of the two tutoring roles depended on categorizing dialogue segments manually.

CHAPTER 6

Experimental Results

6.1 Dialogue Act Pair Analysis

6.1.1 Experiment Setup

In a verbal conversation, social convention dictates each person takes turns speaking and listening. However, this convention does not apply to COMPS. COMPS allows students to simultaneously type and see other's responses. A turn is complete when a student clicks "enter." However in COMPS sometimes students do not bother to wait for each other, they might start responding before the other student has finished typing. Or even more extreme, several students could be typing their own ideas at the same time without even pretending to be responding to each other. Consider that there are usually three and sometimes four students, plus a TA or instructor. It becomes harder to wait to take turns with more people, especially since anybody can simply start typing at any time.

Another difficulty is that when dialogue turns overlap --- the two students were typing simultaneously for part of the time --- it can be hard to know which turn to consider as the first one and which is the second one. In COMPS transcripts, the dialogue is printed in the order of the "enter" key, the timestamp of when a person finished typing. We know there are cases where this is incorrect: the person who pushed "enter" second was the person who started first. Dialogue turns then can become out of order in the transcripts. Disentangling dialogue turns can be even worse when someone doesn't bother to push "enter." Person A can pause typing, person B responds to person A, then person A responses to B, and maybe sometime later A pushes "enter." This "enter" then puts several turns of person A's dialogue into the log file, but they are all in a single turn in the text box for chatting.

So since the highly-interactive typing without waiting for COMPS interface permits students to talk with taking turns, and the turns are sometimes randomized by the enter key and the timestamps, we decided to test whether we could see in the transcripts that students were actually responding to each other (Bui, Trotter, Kim, Glass, & Kim, 2019). The hypothesis is that if students are not talking to each other, or if dialogue turns are sufficiently randomized in the transcript, then pairs of successive dialogue acts should occur randomly. In other words, in real dialogue we would expect an A turn (sharing ideas) to be followed preferentially by a B turn (negotiating) or C (regulating). So we counted the frequency of A followed by A, A-B, A-C, etc. for the sixteen possible pairs. The null hypothesis is that the successive pairs are independent, e.g. the probability P(A-A) of two dialogue acts A-A should occur with P(A) x P(A).

6.1.2 Dialogue-act Pair Experiment

We picked 1270 pairs of dialogue turns that had been annotated with collaborative dialogue acts. We considered successive pairs of dialogue acts only when both were uttered by students. The presence of Teaching Assistants or instructors will change the character of the conversation. The main concern in COMPS is student collaborative dialogue, meaning students conversing with each other. Because the presence of a TA means that almost all dialogue is back-and-forth with the TA, almost all student-student interactions happen when the TA is not present. The independent probabilities of each dialogue act were computed from the counts divided by the total. Table 6.1 shows the frequencies and counts of student dialogue acts, counted from student-student (excluding TA) pairs.

Table 6.2 shows the tabulated frequencies of the 16 possible pairs of dialogue acts of successive student-student utterances. It also shows the expected frequencies of each combination, assuming the utterances were independent of each other. The total expected pairs come to 1272 due to rounding.

We performed a chi-squared test, to see whether the distribution of 1270 events among 16 categories could have come from the expected distribution. The result $p < 10\text{-}29$ is unambiguously significant.

It is proof that we are successfully recording students interacting conversationally.

6.1.3 Discussion

The result of Tag Pair Analysis shows that some pairs that far exceed chance are:

- A-B represents sharing ideas followed by negotiating ideas.
- B-B two negotiating turns in a row.
- C-C two regulating turns in a row.
- C-D regulating followed by maintaining.
- D-D two maintaining turns in a row.
- D-C regulating following by negotiating.

This suggests that a probabilistic model of dialogue acts might be possible. The table of probabilities suggests that a cycle starts with A Sharing Idea followed by B Negotiating. It randomly switches between A and B until somebody contributes a C Regulating turn. Following C, the most common dialogue act would be another regulating turn, with possibly some D Maintaining turns interspersed. Then it probably cycles back to A.

Building a model of longer-length sequences of dialogue acts will be accomplished by tabulating longer sequences, and possibly by associating these sequences with events in the problem-solving task. Reading the dialogues suggests that probabilistic transitions are partly determined by if the problem was solved and the group is moving on the next problem. On the other hand, if the problem was not solved, the regulating can with some probability go back to negotiating.

Table 6.1. Frequency of dialogue acts in 1270 turns not involving TA

Tag	Count	Prob
A	377	0.30
B	406	0.32
C	259	0.20
D	228	0.18

Table 6.2. Tabulated frequencies of pairs of dialogue acts, compared with the assumption of independence

Pair	Count	Expected	Pair	Count	Expected
A-A	121	112	C-A	64	77
A-B	172	121	C-B	52	83
A-C	48	77	C-C	78	53
A-D	36	68	C-D	65	46
B-A	116	121	D-A	49	68
B-B	167	130	D-B	34	73
B-C	79	83	D-C	63	46
B-D	44	73	D-D	82	41

6.2 Dialogue Behavior of Teaching Assistants

6.2.1 Experiment Setup

The two dialogue-act annotators each believed they were observing two different styles of Teaching Assistant dialogue. These are labeled "tutoring" and "mentoring" behavior. Recall that the script asks students to develop and agree with an answer to one section of the problem. They signal the TA to judge the answer, possibly check their understanding, and possibly provide assistance. They proceed to the next section when the TA approves (Bui, 2019). Figure 6.1 shows a typical collaborative discussion from our dialogues, from before the students call for the TA. Figures 6.2 and 6.3 illustrate the two styles of dialogue with the TA present. The TA-participation sections, especially in the tutor behavior, sometimes show the participation rate of the TA rise above 0.6. The dialogue segment in Figure 6.2 shows 6 out of 13 turns uttered by the TA, whereas with 4 people approximately 3 turns would represent an even participation rate. In Figure 6.2 student 3 is lurking.

Participant	Text	Dialogue Acts	Sub-category
St1	public String toStrong(){ String result = null; result = lendingInstitution +' '+ PAmount +' '+ iRate +' '+ etc.	A	Sharing Idea
St2	lol yall going in i think thats right tho	D, B	Joking, Agreement on the idea
St1	we just have to explain the getters and setters now	C	Suggest next step
St3	Student 1 can u explain them	C	Check on understanding
St1	besides excapsulation, accessors make it easier to change future things mybad on the spelling	A, D	Explain Idea, small talk
St4	So everything except the setters and getters are explained right?	C	Reflects on what the group did
St1	encapsultion allows validation	A	Continue to explain Idea
St3	I dont believe we've explained the properties	C	Suggest next step

Figure 6.1. Typical problem-solving dialogue without TA involvement

When students start to address the teaching assistant, however, the dialogue often ceases to be between the students and becomes between students and TA. When the TA explains answers, we say the TA adopts a tutoring role. Otherwise, where the TA works to promote student collaborative problem-solving, we say the TA adopts a mentoring role.

Participate	Text	Dialogue Acts	Reasoning
St1	TA could you re explain the question	C	Expresses confusion
TA	for # 3?	TA-C	Check on understanding
St1	yes	D	Agree on none task-relevant
TA	Basically, you have to tell me which variables you can access from your method	TA-A	Share idea
St2	it would be the three formal parameters for number three right?	B	Ask to clarify
TA	*which class variables	TA-C	Suggest next step
St1	the principalAmount, intreset rate, and the term? because thats the values the method uses to calculate	A, B	Sharing idea, collaborate on an idea
St2	3. the principalAmount, intreset rate, and the term.....because thats the values the method uses to calculate TA?	C	Express of understanding
TA	Actually, its a bit more complicated than that.	TA-C	Suggest next step
St1	totalCurrentMortages	A	Sharing idea
TA	You have a static method for your example. Therefore you can only access a particular class var Explain why that is	TA-A	Sharing idea
St1	static methods can only access other static variables	A	Sharing idea
TA	good, you got it	TA-C	Show satisfaction of the group did

Figure 6.2. Dialogue with Teaching Assistant in Tutoring Role

Participate	Text	Label	Reasoning
St1	num 4	C	Suggest next step
St2	yes sir	D	Small talk
St3	cool	D	Small talk
St2	i suck at tracing code	D	Small talk
St3	I am working on it	C	Explain current action
St1	can we get some help	C	Express confusion
St2	TA since i cant talk, can u please help me us*	D, C	Express confusion
TA	Can you point out which part is giving you trouble in the code	TA_C	Check on understanding
St3	public string toString	C	Express confusion
TA	Do you understand the code from main()?	TA_C	Check on understanding
St2	no sir	C	Express confusion
St1	well im confused still	C	Express confusion
St2	any ideas guys? guys??TA?	C	Express confusion
TA	Ok try to discuss the 1st two lines of main and what they do	TA_C	Suggest the next step

Figure 6.3. Dialogue with Teaching Assistant in Mentoring Role

6.2.2 TA Style Experiment.

In this experiment, the annotators separated episodes of TA dialogue according to their manual judgment of the two styles. Then we counted dialogue acts in the two different batches to see if they are statistically different (Bui, Trotter, & Kim, 2019).

Each transcript was segmented into sections that correspond to one complete discussion, answer, and approval from the TA. The TA-involved segments of several dialogues were thus manually categorized as Group A (tutoring role, 178 turns) and Group B (mentor role, 120 turns).

Numerically, there are distinct differences in the dialogue acts performed by the teaching assistants between group A and group B mixtures of dialogue acts. In both cases, categories B (negotiating) and D (maintaining communication) are negligible. The other two categories are quite different. Both contain frequent regulating turns. These are often checking understanding or probing the students as illustrated in Figures 6.2 and 6.3 dialogues. However, in the tutoring role segments, the TAs contribute many more ideas into the conversation.

Therefore we decided to judge the different styles according to the frequencies of A and B dialogue acts.

Table 6.3 shows the counts and percentages of A sharing and B negotiating dialogue acts among the dialogue segments manually classified according to the two roles. The hypothesis is that the relative frequencies of these acts are the same. Chi-squared test rejects this hypothesis with $p < 0.01$.

Table 6.3. Teaching Assistant Dialogue Acts

	Group A tutoring	Group B mentoring
Sharing ideas	32%	6%
Regulating	60%	88%

6.3 Fingerprinting Participant Behavior and Preparedness

Are well-running collaborative problem-solving dialogues characterized by equal levels and styles of dialogue acts and participation? Toward fingerprinting well-running dialogues, we ranked students according to their relative level of preparedness for solving the problem. The hypothesis is that perhaps the student who enters the conversation with the most knowledge of the topic may have different contributions than the student who enters with the least knowledge. The relative preparedness was based on the pre-test scores of the three students. Rank 1 is the most prepared student in the conversation, rank 2 in the middle, and rank 3 is the least prepared (Kim, Kim, & Glass, 2016, p. 68) . Table 6.4 shows average measures of learning and collaboration according to relative student preparedness rank, taken from a set of 10 dialogues. Participation increases with increasing relative preparedness. In other words, the most knowledgeable student in the group talks more. The variety of dialogue act varies also. With more preparedness: a) sharing dialogue acts increase, b) negotiating dialogue acts decrease, c) regulating increases. One other result is that the most prepared students show zero learning gains, on average, while the others show positive learning gains from the experience (Bui, 2019).

Table 6.4. Different styles of contribution, based on relative preparedness within the group.

	Rank 1: n=10	Rank 2: n=10	Rank 3: n=10
Avg learning gain	0.0	0.1	0.5
Numb. Dialogue Acts	442	311	220
A: sharing	30%	27%	25%
B: negotiating	28%	33%	33%
C: regulating	28%	27%	21%
D: maintaining	14%	13%	22%

Rank 3 least prepared students show the largest dispersion in participation and dialogue act behavior analysis. Some rank 3 students seem to be disengaged or lurkers, resulting in a very low participation rate. Others may constantly ask for clarification, which enriches participation and negotiation dialogue acts. Others devote large numbers of turns to conversation maintenance, frequently contributing “LOL” or other comments not related to solving the problem at hand. The requirement in the script that students come to an agreement, and the TA can probe for understanding, encourages rank 3 students to be engaged.

Chi-squared tests show that the rank 3 least prepared students significantly differ in their mix of dialogue acts from rank 2 ($p< 0.05$) and from rank 1 ($p < 0.01$). A rank 3 student also participates significantly less than a rank 1, contributing fewer dialogue acts in each dialogue ($p < 0.01$). Consistent with previous COMPS results, the most prepared students showed little or no learning gain, while the least prepared showed the most (Kim, Kim, & Glass, 2016).

A conclusion is: learning gains do not seem to be caused by problem-solving participation acts. The lowest rank students had the largest learning gains, with a lower percentage of problem-solving dialogue acts in categories A) through C), and a higher proportion of category D) acts which do not contain problem-solving content. Assessing collaborative dialogues may need to take this differential into account.

6.4 Machine Classification of Dialogue Acts

The experiment was to train a linear classifier that would identify dialogue acts with a turn. The input to the classifier, each machine learning case, consists of features extracted from the turn.

Previous work by other researchers suggest that the machine classification of dialogue acts is difficult. Rose (Rosé et al., 2008, p. 237) and ECA (Cassell, Sullivan, Churchill, & Prevost, 2000) both report very low accuracy. A main difference between human and machine annotation is the human annotator understands the context of each turn of the conversation. Example: a turn with just one word “Yes” can be interpreted as B “Negotiating” or C “Regulating of Problem Solving” depending on the context of the previous turn. A more complex example of a sentence that can be understood differently is “I am behind you.” It can mean C “Showing frustration or lack of understanding” or B “Agreement with teammates.” As human annotators, the researchers can solve that by looking at the previous turns. This experiment tests if the machine can do the same. Part of this experiment is to see whether including previous dialogue context can help improve classifier accuracy. When the classifier can see features from the former turn along with the current turn, can it more reliably identify the current turn’s dialogue act?

We use a COMPS text analysis pipeline that had been built for the dashboard. This pipeline includes a classifier for student turns which express substantive content (Willis et al., 2017, p. 11). We replaced that classifier with our own for this work.

6.4.1 Topic Modeling Features

The main feature for each turn was a topic model. Each topic is a collection of words that relate to each other in one turn. Parts of the transcript are used to train a topic model with 10 topics, using 3 iterations (Willis et al., 2017, p. 11). The topic model contains a bag of words for each topic, with probabilities assigned that each word relates to that topic.

When the topic model is applied to the dialogue that will be fed to the classifier, each turn of conversation is converted into a bag of words, and then turn is modeled as a linear combination of 10 topics. The 10 values range from 0.0 to 1.0 depending on how close of each turn resemble the topic that has been trained from the earlier transcripts.

This is similar to the idea of making a "word embedding" used by neural network classifiers. The gensim Python library was used to deriving the 10 latent topic numbers for each turn.

When the context of the previous turn was included as a feature, each case had 20 numbers: 10 from the current turn and 10 from the previous turn. This was also extended to include two and three previous turns of context.

6.4.2 Other Text Features

Text features were also included in each case to improve accuracy (King, 2009, p.1755). The Linear classifier is fed with text features such as :

1. Pronunciation “you” feature with a binary value indicates one group member addresses another group member if the first 10 words in sentences. The “you” feature includes using pronouns “you,” “u,” “we,” and “us.” “You” feature possibly could be a good indication for negotiating or regulating.
2. “Question mark” feature indicates the presence of a question mark. It might be indicative of, for example, a B turn “Asking to Clarify Questions” or C “Checking on understanding.” This feature also has a binary value of 0 or 1.
3. The number of words in the turn is a feature. It might discriminate A “Sharing Idea” turns. It is very normal for a student to type a long sentence when he or she wants to explain an idea to the group. It is also very common when consecutive turns from one student are “Sharing Idea” turns. As explained in the previous chapter, the consecutive turns are grouped together as one big turn, therefore, the number of words feature is helpful in this scenario.

6.4.3 Changes to the Text, including Marking Features

The text processing pipeline also made systematic changes to the text, in order to identify more special features and to simplify the vocabulary (Glass et al., 2018, p. 145). Certain features were marked by placing special tokens in the text, synthetic “words” whose presence meant that the students had done certain things.

1. Replace emotions, hashtag, at-sign with special token for each. For example the at-sign usually means that one student is directly addressing another, which could be important for identifying the dialogue acts.
2. Spelling correction, also regularize the spelling of some words such as “soooo…”.

3. Replace caret and asterisk with tokens. For example, typing “^” or “^^^^” usually means that the student is pointing to the above turn in the dialogue, which could be important for identifying dialogue acts.
4. Reduce the vocabulary to approximately 10,000 common words. This vocabulary was enhanced from a standard English word list by adding typed-chat abbreviations (e.g. “lol” and “brb”) and other common words as found in earlier COMPS dialogues (Glass et al., 2018, p. 145). Removing uncommon words removes noise from the data. It also reduces the problem of overtraining, where the classifier may find certain unusual word classifies particular training cases, but that is not general.

Table 6.5 show an example input CSV file after feature extraction and text have been filtered before feeding to a classifier.

Table 6.5. Text After Feature Extraction and Text Processing

Raw Text	Filtered Text	Question Feature	Number of words	You Pronoun
Actually for your c) totoalCurrentMortages= private double totalCurrentMortages the answer should be private static double...	actually for your c $g01 private double $g01 the answer should be private static double	0	15	0
becuase its a class variable?	$g01 its a class variable	1	5	0
this is what i got so far..... public double calculatePayment(double principal amount, double intrest rate, string term){} for number 2	this is what i got so far public double $g01 principal amount double $g01 rate string $g01 for number 2	0	20	0
just because each Instance of the class shares the same var	just because each instance of the class shares the same var	0	11	0
ok johnathan re turn in 2	ok $g01 return in 2	0	6	0
ok...2 is public double calculatePayment(double principal amount, double intrest rate, string term) {}..... this header is public, has a return type double for the result and the parameters necessary for the calculation. TA?	$g01 is public double $g01 principal amount double $g01 rate string $g01 this header is public has a return type double for the result and the parameters necessary for the calculation ta	1	32	0

6.4.4 Linear Classifier Result and Discussion

Figures 6.4 through 6.7 show the results of dialogue act classifier experiments. Each turn of the conversation can contain multiple dialogue acts, A, B, C or D. We trained binary classifiers. The “AnotA” classifier identified (yes or no) whether dialogue act A was contained within the words of one dialogue turn. The “BnotB” classifier could be applied to the same sentences to determine whether the student had also done a B dialogue act. When it is ready for testing with the classifier, the target class is converted to a binary number of 0 and 1. With this system in place, each turn will have 4 different target classes: AnotA, BnotB, CnotC, and DnotD. Each linear classifier will focus on one target class.

For each target class, the classifier was trained using several different numbers of previous context turns.

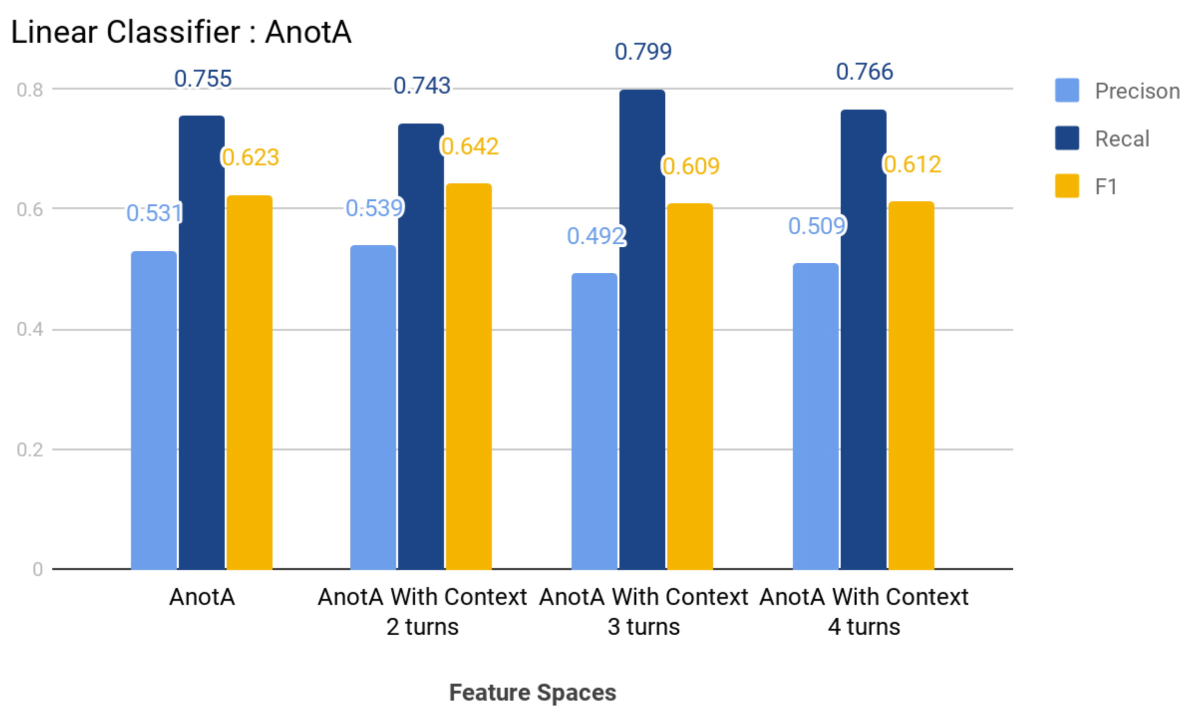

Figure 6.4. Classifier Attempt of Classifying “A” Dialogue Act vs. not-”A”

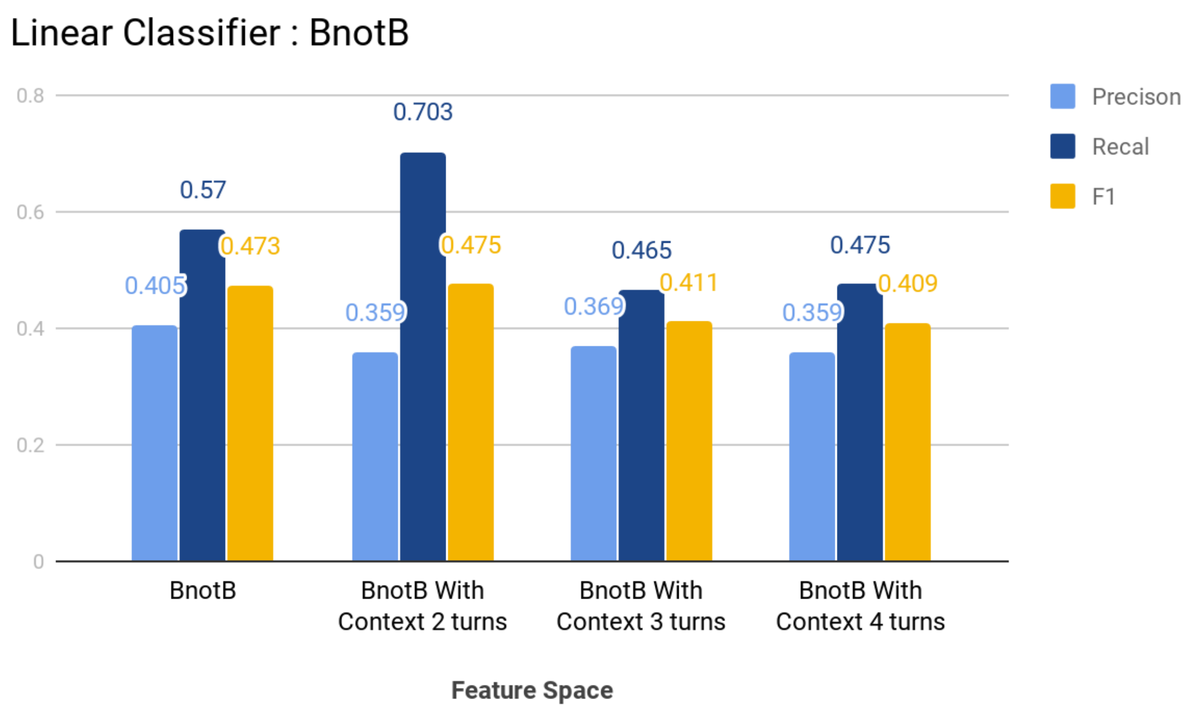

Figure 6.5. Classifier Attempt of Classifying "B" Dialogue Act vs. not-"B"

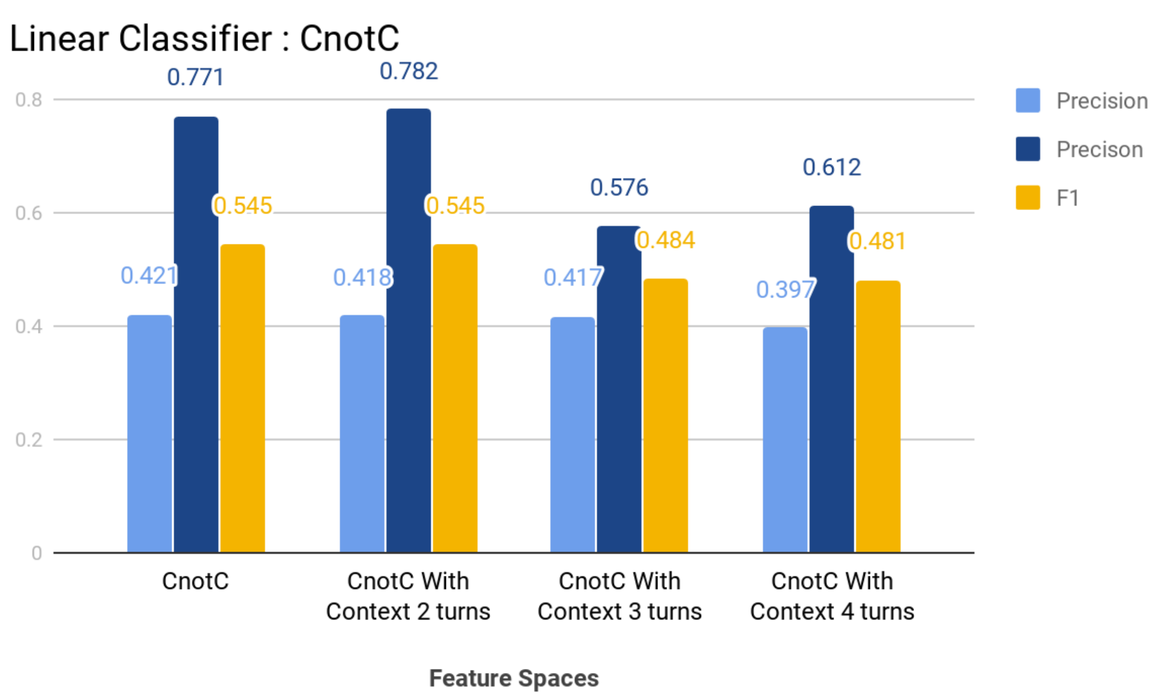

Figure 6.6. Classifier Attempt of Classifying "C" Dialogue Act vs. not-"C"

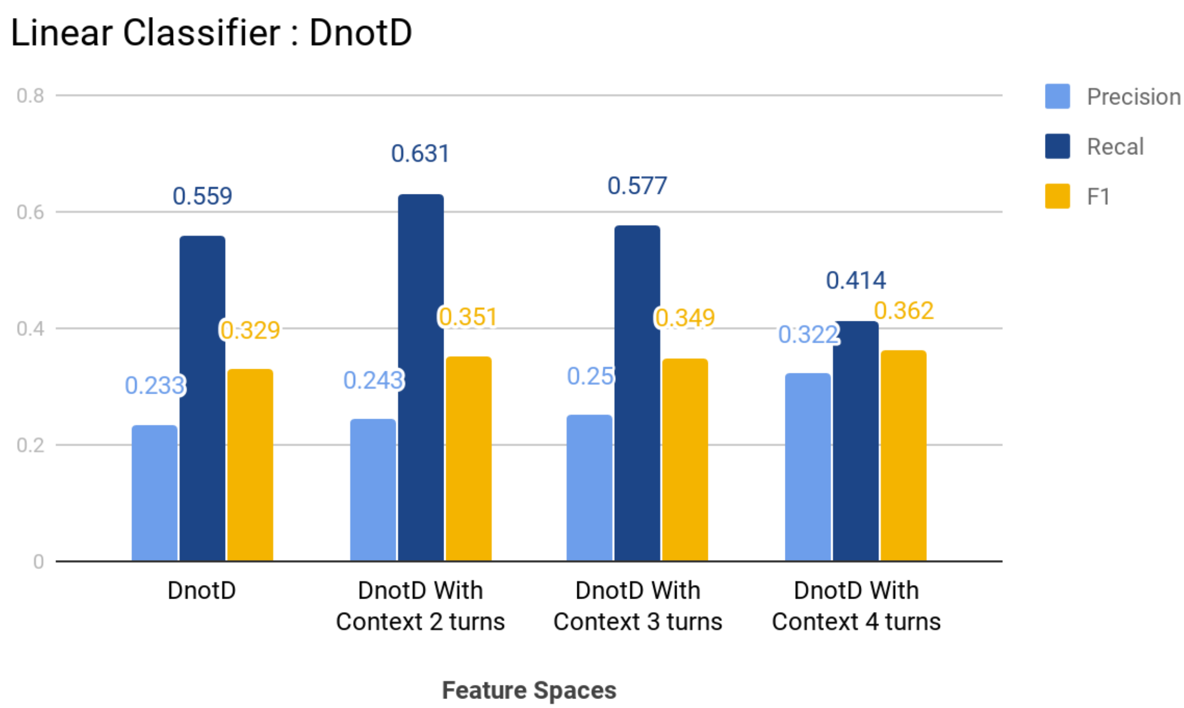

Figure 6.7. Classifier Attempt of Classifying “D” Dialogue Act vs. not-”D”

The best results were for the AnotA classifier, with F1 combined precision-recall scores in the range of 0.6 to 0.65. F1 scores for B, C, and D are in the neighborhoods of 0.47, 0.55, and 0.35.

In the four categories of collaborative dialogue acts, “A” turn has the highest precision score and “D” turn has the lowest. Adding a context turn yielded generally a slightly better results than no context, but adding more turns did not generally help.

CHAPTER 7

Conclusion and Future Work

7.1 Conclusion and Future Work

The main research goals of this study were to apply the student collaborative dialogue acts within COMPS dialogues to several research questions. The dialogue acts were simplified from a set used by Educational Testing Service in its assessment of problem-solving collaboration skills (Hao et al., 2017). The four research questions are:

1. Are student dialogue behaviors within problem-solving typed chat different for students with different levels of preparedness?
2. Can we use dialogue acts to see differences in Teaching Assistant behavior?
3. Can we train classifiers that correctly annotate dialog acts?
4. Can we start to build a model with a deeper understanding of student behavior in a collaborative environment?

The main development goal of this study was to add a new layer to the COMPS system. COMPS started as a stand-alone chat interface, which is specialized for group problem solving. The new layer turns COMPS into a deliverable curriculum and research tool, helping the instructor to manage class lists, take attendance, administer pre- and post-tests, administer surveys, assign students to groups, label and gather together the data from individual students and groups. This added software, which was developed for this project, was used for over a year in COMPS exercises in the COMP 167 class.

7.1.1 Preparedness and Learning Gain Result in Different Dialogue Acts

Within a single problem-solving conversation group, the most prepared student has contributed a higher percentage in sharing ideas tags and lower in negotiating ideas. Those students also consistently show no learning gain as is reported (Kim et al., 2016). This result shows that students who are prepared are leading the group to solve the problem. In fact it suggests that the most prepared student might be teaching the other two.

Since the most prepared two students in the group are more predictable, future research might then reduce the noise and variability by considering mainly the dialogue act contributions of the two most prepared students.

Within a new COMPS webpage update function, auto-grading and auto-ranking of student preparedness can be done in real-time. If the pre-test rankings can be used to help assign students to groups, the lab instructor can have a powerful tool to control how they want students to learn. It may also be possible to assign problems with different difficulty level appropriate to their preparedness level.

7.1.2 Tutoring vs. Mentoring Teaching Assistants

The research found that a notable cause of departure from the collaborative activity was the behavior of the teaching assistants. We also found measurable differences in the dialogue act and participation behaviors that might be useful for building text classifiers that a) recognize the teaching-assistant off-script activity, and b) possibly help detect conversation quality.

It has proven to be challenging to keep the teaching assistants on-script, promoting group collaborative discourse. Discourse obligations dictate that the TA should answer questions. For the TAs, answering students' questions is normal behavior. Answering questions is what is expected of them when interacting with students in the course. Answering questions is a large fraction of their activity during the regular non-COMPS programming lab sessions. Clearly better training of TAs should be attempted.

7.1.3 Machine Classifiers

The accuracy of the linear regression classifier is somewhat disappointing. However other researchers have also reported disappointing results. For example, Rose (Rosé et al., 2008, p. 237) shows that the machine tag classifier F1 scores of about 0.7 in tagging dialogue acts in problem-solving dialogues. This is likely not adequate for machine-annotation of individual dialogue turns.

However for assessing dialogues, it could be sufficient to estimate percentages of dialogue acts measured over multiple turns. For example, between the two TA styles, A-sharing is 32% vs 6%. For purposes of estimating a large percentage difference, a classifier that does reasonably better than chance will probably be good enough.

For future work, there are several ideas on how to improve the result.

1. In place of Topic Modeling as primary embedding word tool, we can try Word2Vec or Doc2Vec. These neural-network methods have had success in natural language applications, they could be used to generate a similar feature vector for the linear classifier (Wang, Ma, & Zhang, 2016, p. 98).
2. In terms of using context as an additional feature, there are several ideas. One is to use the other text features from the previous turns, not only the word features. For example, the presence of a question in turn 1 is often followed by an answer in turn 2. Another idea is to use the tag of the previous turn. Since the most common response to a C turn is

another C turn, for example, then the fact that turn 1 was tagged as “C” could be useful in tagging turn 2.

3. Expansion on using different classifiers such as decision trees or random forest.

The results from this study contradict the naive expectation that a healthy collaborative dialogue could be recognized by roughly equal styles of participation from all the students. However, they do suggest a possible way forward. The next step will be to endeavor to train machine classifiers to recognize and annotate the four categories of dialogue acts, in a manner similar to recognizing substantive contributions.

7.1.4 Tag Pair Analysis

Finally, we have begun analyzing sequences of dialogue acts, starting with pairs, because these often represent dialogue exchanges. We hypothesize that well-functioning collaborative problem-solving dialogue might contain characteristic patterns of dialogue exchanges.

The problem-solving cycle can mark an expected transition of one dialog act to another. The research can focus on building a longer sequence of tags into a Markov model. A foundation model has been observed within COMPS dialogues using the markup manual. The future researcher can use the same markup manual and apply it for many more problem-solving sessions.

References

Bellanca, J. A. (Ed.). (2010). 21st century skills: Rethinking how students learn. *Solution Tree Press*.

Bui, D. (May 2019).Quantitative Differences in Collaboration Dialogue Acts Between Students in Problem-Solving Discussion.8th Annual COE Graduate Poster Presentation Competition, NC A&T State University.

Bui, D (May 2018) ". Web site for Deployment of COMPS Computer-Aided Collaborative Learning Curriculum in Classroom,7th AnnualCOE Graduate Poster Presentation Competition, NC A&T State University.

Bui, D., Trotter, M., Kim, J.H. Glass M., & Kim, S. (2019). Dialogue Acts Within Typed-Chat Collaborative Problem-Solving For a University Programming Class. Fourth Annual International Technology, *Education and Development Conference INTED2019*, pp 7291-7298.

Bui, D., Trotter, M., & Kim, J. H. (2019). Patterns of Collaboration Dialogue Acts in Typed-Chat Group Problem-Solving. *In Intelligent Tutoring Systems 14th International Conference ITS 2018.* Springer, pp. 431-433.

Carden, J., Billie, K., Kim, J. H., & Glass, M (2017). Information Assurance for Computer-Mediated Chat in an Education and Research Environment, *Proceedings of 10th Annual International Conference on Education and New Learning Technologies.*

Cassell, J., Sullivan, J., Churchill, E., & Prevost, S. (Eds.). (2000). Embodied conversational agents. *MIT press.*

Chiong, R., & Jovanovic, J. (2012). Collaborative learning in online study groups: An evolutionary game theory perspective. *Journal of Information Technology Education: Research,* 11, 81-101.

Desjarlais, M. A., Kim, J. H., & Glass, M. (2012, April). COMPS Computer Mediated Problem Solving: A First Look. *In Proceedings of the Midwest AI and Cognitive Science Conference.*

Glass, M., Bryant, K., Kim, J. H., & Desjarlais, M. (2015). Come let us chat together: simultaneous typed-chat in computer-supported collaborative dialogue. *Journal of Computing Sciences in Colleges*, 31(2), 96-105.

Glass, M., Kim, J. H., Bryant, K. S., Desjarlais, M. A., Goodrum, M., & Martin, T. (2014). Toward Measurement of Conversational Interactivity in COMPS Computer Mediated Problem-Solving Dialogues. *In MAICS* (pp. 31-38).

Glass, M., Jagvaral, Y., Bouman, N., Graham, E., Kalafatis, S., Arndt, L., ... & Bryant, K. (2018). Problem-independent text analytics for real-time judgment of CSCL typed-chat dialogues. *Journal of Computing Sciences in Colleges*, 34(1), 145-154.

Hao J., Liu L., von Davier A.A., Kyllonen P.C. (2017) Initial Steps Towards a Standardized Assessment for Collaborative Problem Solving (CPS): *Practical Challenges and Strategies.* In: von Davier A., Zhu M., Kyllonen P. (eds) Innovative Assessment of Collaboration. *Methodology of Educational Measurement and Assessment.* Springer, Cham.

Herborn, K., Stadler, M., Mustafić, M., & Greiff, S. (2018). The assessment of collaborative problem solving in PISA 2015: Can computer agents replace humans?. Computers in Human Behavior.

Herring, S., Stein, D., & Virtanen, T. (Eds.). (2013). Pragmatics of computer-mediated communication (Vol. 9). Walter de Gruyter.

Kim, J. H., & Glass, M. (2004, August). Evaluating dialogue schemata with the wizard of oz computer-assisted algebra tutor. *In International Conference on Intelligent Tutoring Systems (pp. 358-367).* Springer, Berlin, Heidelberg.

Kim, J. H., Glass, M., Kim, T., Bryant, K., Willis, A., McNeil, E., & Thomas, Z. (2016, April). Student Understanding and Engagement in a Class Employing COMPS Computer Mediated Problem Solving: A First Look. *In CEUR workshop proceedings* (Vol. 1584).

Kim, J. H., Kim, T., & Glass, M. (2016). Early experience with computer-supported collaborative exercises for a 2 nd semester Java class. *Journal of Computing Sciences in Colleges, 32(2),* 68-76.

King, D. E. (2009). Dlib-ml: A machine learning toolkit. Journal of Machine Learning Research, 10(Jul), 1755-1758.

Ogan, A., Finkelstein, S., Walker, E., Carlson, R., & Cassell, J. (2012, June). Rudeness and rapport: Insults and learning gains in peer tutoring. *In International Conference on Intelligent Tutoring Systems (pp. 11-21).* Springer, Berlin, Heidelberg.

Rosé, C., Wang, Y. C., Cui, Y., Arguello, J., Stegmann, K., Weinberger, A., & Fischer, F. (2008). Analyzing collaborative learning processes automatically: Exploiting the advances of computational linguistics in computer-supported collaborative learning. *International journal of computer-supported collaborative learning,* 3(3), 237-271.

Stahl, G. (2006). Group cognition: Computer support for building collaborative knowledge. Cambridge, MA: MIT Press.

Stubbs, Michael. (1983). Discourse Analysis, U. of Chicago Press.

Wang, Z., Ma, L., & Zhang, Y. (2016, June). A hybrid document feature extraction method using latent Dirichlet allocation and word2vec. *In 2016 IEEE First International Conference on Data Science in Cyberspace (DSC) (pp. 98-103)*. IEEE.

Wells, G. (2000). Dialogic inquiry in education. Vygotskian perspectives on literacy research, 51-85.

Willis, A., Evans, A., Kim, J. H., Bryant, K., Jagvaral, Y., & Glass, M. (2017). Identifying domain reasoning to support computer monitoring in typed-chat problem solving dialogues. J*ournal of Computing Sciences in Colleges, 33(2),* 11-19.

Appendix A

An Example of Pretest and Posttest using in COMPS Exercise

1. Pre-Test

Please consider the following code and answer the questions below.

Analyze the main method down through the line marked // A in the code.

P1) At the line marked // A in the main method how many objects of class Foo have been created?

a) 0 b) 1 c) 5 d) 6

Answer: b) 1

P2) What other object has been created?

1. One integer object
2. One array of foo object
3. One integer object and one array of foo object
4. One String object

Answer: b) One array of Foo objects,

P3) What was printed the line just before // A?

1. Empty String
2. 0|0|0
3. 0|0|1
4. 1|1|1

Answer: c) 0|0|1

P4) Local variables are kept on the stack or the heap?

a)Stack

b)Heap

Answer: a) Stack

P5) Objects are kept on the stack or the heap?

a)Stack

b)Heap

Answer: b) Heap

P6) If int variables are 4 bytes and reference variables are 11 bytes, how many bytes of stack are being used by variable fooarray:

a) 0 (because arrays are on the heap)	c) 11 (reference to an array object)
b) 4 (for the length)	d) 4 + 5*11 (length plus reference to 5 Foo objects)

Answer: c) 11

P7) Assuming each Foo object requires only the space for its instance variables, and each int variable requires 4 bytes, how many bytes are needed for each Foo object?

1. 12
2. 8
3. 4
4. 0

Answer: b) 8

Now analyze the main method through // B and answer the B questions.

P8) Having run through the line marked // B, how many Foo objects have been created?

a) 1 b) 5 c) 6 d) 11

Answer: c) 6

P9) Having run through the line marked // B, what is the value of z?

a) same number as question P8 answer	c) 1
b) 1 in the first Foo object, 2 in the 2nd Foo object, etc.	d) none of the above

Answer: a)

Finally analyze to // C at the end of main and answer the C questions.

P10) At // C, How many different Foo objects are referred to from main (including variables referred to from local variables and from inside the array of Foo objects)?

a) 1 b) 5 c) 6 d) 11

Answer: b) 5

2. Post-Test

Q1) In the Baz class the line marked /* COMMENTED OUT */ was removed because of what error?

a) Correct would have been `this.s = this.tnew`	c) `s` and `tnew` are different types
b) Static method cannot access an instance variable	d) Correct would have been `s = tnew`

Answer: b) Static method cannot access an instance variable

Q2) How many how many reference variables are on the stack in the main method from threebaz?,

a) 1 int for length	c) 1 int for length, 1 ref for array obj, 3 ref for baz objects)
b) 1 int and 3 ref for baz objects)	d) 1 ref for array obj

Answer: d) 1 ref

Q3) In the Baz class the line marked /* T UPDATE */ is changing a static variable from a non-static method. Is this allowed in Java?

a) Yes

b) No

Answer: a) Yes

Q4) String objects are kept on the stack or the heap?

a) Stack

b) Heap

Answer: b) Heap

Q5) What was printed by the main method just before it executes through the line marked // A?

1. Empty String
2. Speed = 0 MPH
3. Speed = 0 M/S

4. M/S

Answer: c) Speed = 0 M/S

Q6) How many objects, and what are their types, are created in the main method just before it executes through the line // A?

a) 2 objs: 1 Baz, 1 array Baz[], 0 String	c) 6 objs: 4 Baz, 1 array Baz[], 1 String
b) 5 objs: 4 Baz, 1 array Baz[], 0 String	d) 3 objs: 1 Baz, 1 array Baz[], 1 String

Answer: A) 2. A Baz object and an array Baz[3] object;

Q7) What was printed by the main method line just before // B?

1. Empty String
2. Volume = -5 cc
3. Volume = -5 in
4. Volume = -5 kg

Answer: d) Volume = -5 kg

Q8) What was printed by the line just before // C?

1. Empty String
2. Weight = 14 kg
3. Weight = 14 gal
4. Volume = -5 cc

Answer: c) Weight = 14 gal

Q9) By the end of main, How many different Baz objects are referred to from main (including variables referred to from local variables and from inside the array of Baz objects)?

a) 1 obj: All are the same	c) 3obj: `b`; `threebaz[0]`; `threebaz[1]` and `[2]` are same
b) 4 obj: All are different	d) 2 obj: `b` and `threebaz[0]` same; `threebaz[1]` and `[2]` same

Answer: d) 2 are still visible threebaz[0] and b are the same, threbazz[1] and [2] are same.

Q10) Assuming each Baz object requires only the space for its instance variables, and each int variable requires 4 bytes and each reference to a

String object needs 11 bytes, how many bytes are needed for each Baz object?

a) 15 bytes: 4 for `x`, 11 for `s`, 0 for `t`	c) 4 bytes: 4 for `x`, 0 for `s`, 0 for `t`
b) 15 bytes: 4 for `x`, 0 for `s`, 11 for `t`	d) 26 bytes: 4 for `x`, 11 for `s`, 11 for `t`

Appendix B

An Example of Lab Exercises

Lab exercises

Answer the questions on this exercise in the answer box and have the TA check them.

```
public class Foo {
        private int x;
        private int y;
        private static int z;

        public Foo() {
                z++; }
        public Foo( x, y) {
                this();
                this.x = x;
                this.y = y;
                z += x + y; }

public String toString() {
        return x + "|" + y + "|" + z } }

public static void main( String [] args ) {
        int i = 0;
        Foo f = new Foo();
        Foo [] fooarray = new Foo[5];
        System.out.println( f.toString() );
        // A

        for (i=0; i<fooarray.length; i++)
        fooarray[i] = new Foo( i, i );
        // B

        fooarray[fooarray.length-1] = f;
        System.out.println( fooarray[fooarray.length-1].toString() ); // C }
```

Please consider the following code and answer the questions below.

Analyze the main method down through the line marked // A in the code and answer the A questions, then get the TA's response.

A1) At the line marked // A in the main method how many objects of class Foo have been created?

A2) What other object has been created and what is its type?

A3) What was printed the the line just before // A?

A4) Local variables are kept on the stack or the heap? Objects are kept on the stack or the heap?

A5) If int variables are 4 bytes and reference variables are 11 bytes, how many bytes of stack are being used?

A6) How does the static int z variable in the Foo class differ from variables x and y?

A7) At the line marked // A what is the value of z?

Now analyze the main method through // B and answer the B questions.

B1) Having run through the line marked // B, what is the value of z now?

B2) How many Foo objects have been created in total by this program?

B3) Assuming each Foo object requires only the space for its instance varables, and each int variable requires 4 bytes, how many bytes are needed for each Foo object?

B4) At the line marked // B how many bytes of heap are occupied by Foo objects

Finally analyze to // C at the end of main and answer the C questions.

C1) At // C, How many different Foo objects are referred to from main (including variables referred to from local variables and from inside the array of Foo objects)?

C2) Assuming the fooarray array requires 11 bytes for each reference to a Foo object, how many total bytes of heap are being used now? (Ignore any space needed for fooarray.length.)

C3 What was printed by the final line just before // C