Automated Classification of Collaboration Skills in Typed-Chat Collaborative Problem-Solving *

Jung Hee Kim[1], Joelle Banks[1], Duy Bui[2], Michael Glass[3]
[1]Computer Science Department
North Carolina A&T State University
1601 E. Market St.
Greensboro, NC 27411
jungkim@ncat.edu, jlbanks@aggies.ncat.edu
[2]L3Harris
2235 Monroe Street, Herndon VA 20171
duyquangbui111893@gmail.com
[3]Computing and Information Sciences
Valparaiso University
Valparaiso, IN 46493
michael.glass@valpo.edu

Abstract

This experiment trained classifiers to monitor the dialogue of students working together in a Java programming class. The classifiers recognized four activities within the problem-solving conversation: sharing ideas, negotiating, regulating, and maintaining conversation. These dialogue acts are characteristic of problem-solving conversations. This experiment trained classifiers that utilize specific words in the dialogue. It also trained classifiers that use a statistical topic model built from the dialogue transcripts. If dialogue acts can be recognized, then the counts of student interactions could be used for computer monitoring of online student collaborative group exercises.

1 Introduction

This study works toward computer identification of the collaboration dialogue acts of students working together solving problems. The students are in a Java programming class. In this class, small-group problem-solving exercises are administered as a way for students to learn and apply conceptual knowledge. As they work, student conversations are partly monitored by teaching assistants and the instructor [6].

A goal of the COMPS project is to provide computational assistance in overseeing the student conversations. The computer could judge in real time whether the conversation groups are productive or could benefit from intervention. The computational model should work for a broad variety of college classes [11]. It should also detect a variety of conversational phenomena, e.g. students becoming frustrated or not working together.

Dialogue acts are the different actions that a person can take while interacting with other people in a conversation. For example, in a problem-solving conversation some dialogue acts might be advancing a new idea or referring to some information from the problem statement. Other dialogue acts are disagreeing with another person, summarizing earlier ideas, and suggesting the next problem-solving move. Categorizing the parts of the conversation in terms of dialogue acts provides a basis for analyzing or assessing the conversation. Relative frequencies of dialogue acts are potentially diagnostic. For example if students never negotiate then it is not likely that all students are engaged in the problem-solving task. Counting pairs of successive dialogue acts can reveal interactive behaviors, e.g. [1]. The theoretical basis of this project is by first analyzing a dialogue as a sequence of small dialogue acts, it could be possible to approximately assess a variety conversational phenomena. This method does not depend on knowing the substance of the students' conversation. Thus it could be generally applicable to student small-group problem-solving exercises.

Prior work toward machine-assessment of COMPS project Java dialogues targeted linguistic phenomena that were less specific to problem-solving conversations: initiate-respond pairs [4] and whether students were contributing substantive turns and agreeing or disagreeing [3, 11]. Compared with dialogue acts, the phenomena studied earlier were less directly related to collaborative problem-solving skills. For example, when students are chatting about their summer vacations their conversation will contain initiate-respond pairs and could contain agreement/disagreement.

In this experiment, 1395 turns of dialogue were annotated manually according to a simplified set of four problem-solving dialogue acts. The annotated dialogue turns were used for training and testing four different classifiers, one for recognizing each type of dialogue act. The features from the text utilized by the classifiers were derived from the dialogue text in three ways:

- the presence or absence of individual words, using the most frequent English words attested in the transcripts. These did not include words which were specific to the topic of the conversation, which was analyzing Java code.
- individual words including Java-related words such as "method" and "double".
- topic modeling [9], a statistical technique based on word-co-occurrences which models each turn as a combination of a small number of latent feature values.

A motivation for training with different feature sets is to train classifiers that are independent of the particular problems the students were discussing. A danger is that machine learning can utilize the variable names from the Java code under discussion. This classifier won't work as well when presented with students discussing a different problem. By being careful in selecting individual words as features, we were able to experiment with a classifier that is not cognizant of any words from the computer programming domain. We also experimented with a classifier that was cognizant of Java programming terms generically, but excluded words from the specific problem.

This paper describes the dialogue act categories and the transcripts of student dialogues. It then reports on experiments in training classifiers for the acts.

2 Background

2.1 Dialogue Act Categories

For this study we adopted categories of dialogue act as defined by Hao et al. [5]. Each different dialogue act corresponds to a skill that students can evince while engaged in collaborative problem-solving (CPS). The list of dialogue acts was developed for the purpose of assessing student CPS ability. The published scheme has 33 skills among four different categories. For purposes of this experiment we utilize only the categories, as shown in Table 1.

2.2 Theoretical Background

The theoretical underpinning of this experiment is: it is possible to detect fingerprints of various collaborative problem solving dialogue phenomena from the dialogue acts. For example [2]:

- Counts of successive dialogue-actions in CPS discussions show sequences of sharing followed by negotiating occur much more frequently than chance.

Table 1: Dialogue Act Categories, after Hao et al. [5]

	Dialogue Act Category	Definition
A	Sharing ideas	Student advances an idea or points to problem-relevant information.
B	Negotiating ideas	Student agrees/disagrees, rephrases or completes or elaborates, identifies a conflict or a gap or modifies the ideas from teammates, modifies or updates own previous ideas.
C	Regulating problem-solving	Metacognitive processes for the team: identifying the goals, expressing lack of understanding, suggesting next step, reflect on problem-solving process, etc.
D	Maintaining communication	Student engages in social interactions, apologizes, corrects spelling, offers help, prompts other students, etc.

- The counts of dialogue interactions are distinctly different when the students are working among themselves, versus when the teaching assistant is involved in the conversation.
- The different roles of students within a discussion can be discerned by counting dialogue acts. The student who was most prepared at the start of the conversation (measured by pre-test) produces more sharing dialogue acts. A less-prepared student negotiates relatively more often. The least prepared student engages in relatively more frequent conversation maintenance.

The skills categories of Hao et al. [5] correlate with categories of transactive activities cataloged by Weinberger and Fischer [10] that are often used in dialogue analysis. Transactivity is the social mode of knowledge co-construction, the ways that students can interact while engaging in a group cognitive exercise. For example the "sharing" skill corresponds to an "externalization" transactive act. Skills such as rephrasing or identifying a conflicting idea, which are labeled "negotiating" by Hao et al., are "integration-oriented" and "conflicted-oriented" consensus building transactive acts. The correspondence between student CPS skills and transactive acts indicates that these categories of dialogue act are likely to be fruitful for dialogue analysis.

Student	Text	Categories
S1	its calling the method foo and then it is reading throught he array	A
S1	the*	D
S2	yea but what its it printing out	B, C
S3	"foo" is an array, not method	B
S3	0|0|0	A
S1	your right my mistake. so they are both printing 0|0|0	B
S3	this is what I think the second one is printing out 0|0|0 1|1|2 2|2|4 3|3|6 0|0|0	B
S1	yes i agree your using the array values in the tostring method so its fromatted the same but has different values	B
S3	yeh and foo[4] wouldve been 4|4|8 but then you have that line of code foo[foo.length-1] = f; which sets foo[4] to f	B
S1	ok so we have an answer	C
S3	so im bout to type the answer so he can check it	C

Figure 1: Transcript of Discussion with Manually-Annotated Dialogue Acts.

2.3 COMPS Project Collaborative Exercise Dialogues

Figure 1 shows an extract from students discussing a problem in a second semester Java programming class. The code and questions are visible to the students in a separate document outside the chat window. Three students S1, S2, and S3 are participating. This transcript shows the categories of dialogue acts for each turn.

The dialogue example shows students analyzing Java code. The students do not execute the code, COMPS project exercises emphasize learning and operationalizing Java concepts. Figure 2 shows part of the Java code under discussion. The students in the Figure 1 transcript are debating the result of the print statements, the proposed answers are in the form of numbers and strings the code would print out.

Part of the theory of group problem-solving exercises is that group discussion forces students to verbalize concepts and explain their reasoning [7]. To encourage this process, the exercise protocol asks students to come to agreement on parts of the exercise. Students positively affirm their agreement by clicking a button. Then an instructor or TA inspects the agreed-upon answer and provides feedback. The Java concepts involved in solving the exercise in

```
public class Foo {
    private int x, 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() );

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

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

Figure 2: Java Code Discussed by Student Problem-Solving Group.

this experiment included: classes, instance variables, static variables, constructors with different signatures, arrays, and instantiating objects within an array. In addition to the printed output discussed Figure 1, other questions concern how many objects were created at various points in the code (including the array itself), and how many objects are still accessible.

The Java terms associated with these concepts occur frequently in the dialogues. This raises possibility that the machine classifier of dialogue acts might be trained to work well with Java collaborative problem-solving exercises if it were cognizant of Java terms.

3 Experiment

3.1 The Dialogue Data

The dialogue data for this experiment consisted of 1385 turns of typed dialogue, from transcripts of 8 collaborative discussions involving 24 students. Each discussion was approximately 1 hour. The discussion exercises were administered during the programming lab time of a 2nd semester university Java class. The groups were also joined at intervals by teaching assistants. Different dialogue act patterns occurred when a TA was participating in the discussion. As the goal of this experiment is to study student dialogue for purposes of assessing whether the students are working together. The segments of discussion with the TA present were thus not included in the data for this experiment.

An artifact of typed-chat conversation is that the speaker can end a typed chat turn by pressing "enter" and then continue typing, maintaining the conversational focus without anybody intervening. In spoken dialogue this might be a momentary pause in a dialogue turn. When one person had control of the conversation for several successive typed-chat turns, we combined them into a single dialogue turn for classification. Merging successive turns from the same person single turns resulted in a data set with 870 turns.

Every turn was annotated by two coders, who then resolved disagreements. One turn could contain several dialogue acts. Figure 1 illustrates a turn where the student first agrees with the preceding student (a negotiating act), then focuses the discussion on deciding the program's printed result (a regulating problem-solving act). The result was 1270 dialogue acts. Table 2 shows the frequencies of the four categories of acts.

Table 2: Distribution of Dialogue Act Categories.

Category	Count	Pct
A. Sharing	377	30%
B. Negotiating	406	32%
C. Regulating	259	20%
D. Maintaining	228	18%

3.2 Feature Sets for Classification

Machine classifiers require extracting a set of features from each dialogue turn. The classifier algorithm then learns to predict the dialogue act categories within the turn based on the values of the features derived from the turn.

To be generally useful, it is important to train classifiers that don't require the presence of words that are specific to the problems under discussion. The word "foo" in these dialogues is the name of a variable. It tends to occur in dialogue acts that are sharing or negotiating but not in dialogue that is simply maintaining conversation. The presence or absence of "foo" could be excluded from the feature set, so that the classifier won't utilize its predictive association.

The first feature set is a vector indicating the presence (1) or absence (0) of 41 different words. For each turn, a vector is produced from the words in that turn. The set was produced from the 50 most common words in the dialogue transcripts. Words that were specific to the problem under discussion, mostly names of variables in the Java code, were excluded [3].

Another feature set started with the 41 common word features but further excluded common English words which were being used in their Java context. These included, e.g., "static" and "class." This feature set of 33 common words was thus even further removed from the specific problems the students were discussing.

A final feature set is a Latent Dirichlet Analysis topic model [9] derived from the transcripts. The topic model contained a 10 number vector for each turn. Conceptually, a topic model is a form of dimensionality reduction. The original text of a dialogue turn could be modeled as a high-dimensional vector with one dimension for the presence/absence of each vocabulary word. In the topic model, each dimension represents bundles of words which tend to co-occur in the same contexts. For the topic model, words which were not in a lexicon of the 10,000 most common English words were excluded.

For the 41-word and 33-word feature models, we checked whether the vocabulary words were likely to contribute information to a machine classifier algorithm. For each word and each dialogue act, a Fischer exact test showed the likelihood that the word contributed information toward diagnosing the presence of the dialogue act. The confusion matrix of presence/absence of word and presence/absence of dialogue act was counted. The Fischer test computed the likelihood that the co-occurrences of that word and dialogue act could be due to chance. Most of the words were significantly related to at least one dialogue act at $p \leq 0.05$.

4 Results and Conclusions

4.1 Classifier Experiment

Scikit-Learn was used for training linear regression classifiers. We used 60% of the data for training, randomly drawn, and 40% for testing. Each classifier was trained and tested with the three feature sets: the topic model features, the 33 common English words, and the set of 41 words including Java concept terms. Table 3 shows the F1 combined precision and recall scores.

4.2 Conclusion and Future Work

The three different feature sets performed similarly. The topic model features, which utilized the largest set of words, was more accurate at recognizing sharing. Sharing is the dialogue act which most directly expressed the ideas of the

Table 3: Classification accuracy of dialogue acts using different feature sets.

Feature Set	Sharing	Negotiating	Regulating	Maintaining
Topic model	0.623	0.473	0.545	0.329
Generic words only	0.546	0.534	0.556	0.388
Plus Java terms	0.568	0.510	0.563	0.437

problem. The small sets of generic words more accurately recognized negotiating acts, such as agreement/disagreement. Achieving problem-independent recognition of dialogue acts might be harder for some types of acts.

Although these classifiers are not very accurate, it might be possible to detect anomalies in the relative percentages of dialogue acts. Table 2 shows that each category of dialogue act can be expected to occur frequently. If no regulating turns or no negotiating turns were occurring, for example, even a poor classifier might reveal their relative absence in a typical 150-turn conversation. An earlier COMPS dashboard assessed conversations based simply on relative occurrence of substantive turns and agreement within the dialogues [3].

A deeper model of assessing dialogue would count successive pairs of dialogue actions. The interaction between two students can be recognized by how the second responded to the first, e.g. sharing followed by negotiating. Counting these interactions entails correctly tagging both acts in the pair. Higher classification accuracy will be needed for successfully recognizing these interactions in student dialogues.

Future work includes finding better-performing feature sets. The models used in this experiment used bags-of-words, where the order of the words in sentences did not contribute. We will try word embedding feature vectors that are more sensitive to sequences of words, for example the doc2vec model [8]. We will also develop and test a revised, simpler, annotation rubric, that we hypothesize may result in more consistent categorization.

Potential future work includes applying the same experiment to the threads in class-related discussion board postings, where the students interact with each other but do not post their dialogue turns in real-time interactions.

References

[1] Duy Bui, Jung Hee Kim, and Michael Glass. Dialogue act pairs for automated analysis of typed-chat group problem-solving. In *Intelligent Tutoring Systems, 16th International Conference*, pages 378–381. Springer, 2020.

[2] Duy Quang Bui. Collaboration pattern model for student participation in problem-solving typed-chat. Master's thesis, Computer Science Dept., North Carolina A&T State University, 2019.

[3] Michael Glass, Yesukhei Jagvaral, Nathaniel Bouman, Emily Graham, Stamatina Kalafatis, Lindsey Arndt, Melissa Desjarlais, Jung Hee Kim, and Kelvin Bryant. Problem-independent text analytics for real-time judgment of CSCL typed-chat dialogues. *J. Comput. Sci. Coll.*, 34(1):145–154, October 2018.

[4] Michael Glass, Jung Hee Kim, Kelvin Bryant, and Melissa Desjarlais. Indicators of conversational interactivity in COMPS problem-solving dialogues. In *Third Workshop on Intelligent Support for Learning in Groups (ISLG)*, 2014.

[5] Jiangang Hao, Lei Liu, Alina A von Davier, and Patrick C Kyllonen. Initial steps towards a standardized assessment for collaborative problem solving (cps): practical challenges and strategies. In *Innovative Assessment of Collaboration*, pages 135–156. Springer, 2017.

[6] Jung Hee Kim, Taehee Kim, and Michael Glass. Early experience with computer-supported collaborative exercises for a 2nd semester java class. *J. Comput. Sci. Coll.*, 32(2):68–76, December 2016.

[7] Timothy Koschmann. Understanding understanding in action. *Journal of Pragmatics*, 2(43):435–437, 2011.

[8] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196, 2014.

[9] Radim Rehurek and Petr Sojka. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Citeseer, 2010.

[10] Armin Weinberger and Frank Fischer. A framework to analyze argumentative knowledge construction in computer-supported collaborative learning. *Computers & Education*, 46(1):71–95, 2006.

[11] Angelica Willis, Ashana Evans, Jung Hee Kim, Kelvin Bryant, Yesukhei Jagvaral, and Michael Glass. Identifying domain reasoning to support computer monitoring in typed-chat problem solving dialogues. *J. Comput. Sci. Coll.*, 33(2):11–19, December 2017.