PROBLEM-INDEPENDENT TEXT ANALYTICS FOR REAL-TIME JUDGMENT OF CSCL TYPED-CHAT DIALOGUES

Michael Glass, Yesukhei Jagvaral, Nathaniel Bouman, Emily Graham,
Stamatina Kalafatis, Lindsey Arndt, Melissa Desjarlais
Dept. of Computing and Information Sciences
Valparaiso University, Valparaiso, IN 46383
(219) 464-5161
michael.glass@valpo.edu

Jung Hee Kim, Kelvin Bryant
Dept. of Computer Science
North Carolina A&T State University
1605 E. Market Street, Greensboro, NC 27411
(336) 285-3695
jungkim@ncat.edu

ABSTRACT

This experiment trained text classifiers to categorize some of the conversational behaviors that might be indicative of productive online student collaborative exercises. COMPS project exercises have students working together via typed chat, solving problems in small groups. Instructors oversee these conversations. Toward the goal of aiding the instructor in locating which conversations could benefit from intervention, this experiment applies text analytics to recognize when students are using substantive vocabulary, and when they are agreeing or disagreeing with other students. The text classifiers were built from student conversations from two different schools solving problems in two different subject areas, using a vocabulary of only the more common English words.

INTRODUCTION

In COMPS (COmputer Monitored Problem-Solving) exercises students work in small groups during their class lab time, solving exercises through typed chat [1]. The exercises are oriented toward understanding and applying conceptual knowledge. To help keep the conversations productive, the instructor and possibly teaching assistants can contribute to the conversations.

As an aid toward instructor oversight, a summary dashboard is being developed which will show which of the student conversations are more likely in need of attention. Two main conversational behaviors that feed the dashboard summary of a conversation are whether students are saying something substantive and whether students are addressing each other by agreeing or disagreeing. Earlier work described a method to quantitatively estimate the first of the two behaviors, substantive utterances, by applying text classifiers to the dialogue [2]. This paper describes experiments to improve computer measurement. The accuracy of identifying substantive utterances has been increased, while at the same making that algorithm work for a broader range of student dialogues. We also report first results for recognizing agreeing and disagreeing.

The data displayed on the instructor's dashboard is the result of the following process.

- The text classifiers for identifying substantive dialogue and agreement/disagreement are trained on historical transcripts of student conversations. These transcripts have been annotated to identify instances of the target behaviors. A classifier examines the text for a dialogue turn, and decides whether the target behavior exists in that turn.
- After having been developed from historical transcripts, the classifier can be applied in real time to student conversations in COMPS lab sessions to provide a measure for the dashboard.
- In the training phase and in real-time deployment, raw dialogue text is preprocessed before being presented to the text classifier. For example spelling can be corrected and the random uppercasing that occurs in typed text can be lowercased.
- In addition to the preprocessed words themselves, other information is extracted from the dialogue for use by the computer classifier. The added features are described below in the Experiment section.

The text classifiers work best when judging dialogues that were similar to the training dialogues. Ethnically different student populations may also differ in their ways of speaking with each other. This experiment endeavors to develop classifiers that are less sensitive to the particular exercise being discussed and the particular population of students. The training data used in this experiment combined COMPS dialogue data from two different courses in different subjects with different student demographics. In addition, this experiment preprocesses the text to filter out specialized vocabulary. The dialogue texts that the classifiers are trained on, and the dialogue that is processed in real time, have been reduced to a dictionary of approximately 10,000 words.

We have used a classifier built from this training data to post dashboard values in real time during COMPS discussion labs. We have no principled way to say whether a given score represents properly functioning conversation or not, so numerical results of the classifiers are not useful for providing a grade assessment of each conversation's quality. However by accurately counting the conversational behaviors that are associated with productive conversations, the relative ordering of the scores should indicate which conversations are more or less likely to benefit from attention. Anecdotal reports from the instructors indicate that the relative ordering of the computer's judgments roughly corresponded to their own judgments. Accordingly, we evaluate the success of the experiment on how accurately the classifiers counted the dialogue phenomena of interest.

The background section describes COMPS group conversation exercises, along with the theoretical basis using for the two behaviors to monitor the health of student conversations. The next section describes the experiment in training text classifiers to recognize the behaviors. Finally we show the accuracy results of applying the classifiers to real transcripts, with a discussion for future work.

BACKGROUND

COMPS Collaborative Exercises

Figure 1 shows an extract from students discussing a COMPS problem in a Java programming class. The code and questions are visible to the students in a separate document outside the chat window. Three students A, B, and C are participating, as is a teaching assistant labeled TA. In addition to the dialogue text, Figure 1 shows the "reason" and "agree" dialogue features discussed below.

Turn	Stu	Dialogue Text	Reason	Agree
1-2	A	do this one work? public double calculatePayment(double principleAmount, double interestRate, double totalCurrentMortgages)	Y	--
3	B	should t be void?	N	Dis
4	TA	Hm make sure you guys agree first and Ill come back in a sec		
5	C	no becuase void is if you are not returning a value	Y	Dis
6-7	B	yea i know.. So to my understanding we are returning something. Ok i understand now	Y	--
8	C	I think what [student A] has works because it calls everything thta we are looking for	Y	Agr
9	A	wait we are not instanting a Morgage object correct, so doesn't that mean it's a no-arg constructor?	Y	Dis
10-12	B	See I was thinking about that. but it does have parameters lets try your original answer, agree?	Y	Agr
13	C	agree	N	Agr
14	B	public double calculatePayment(double principleAmount, double interestRate, double totalCurrentMortgages) {} @TA	N	--
15	TA	Ok you need one more modifier so it can be accessed without instanting the Mortgage class *instantiating		

Figure 1: Transcript of Discussion with Manually-Annotated Features

The dialogue example shows students writing a Java method declaration. To accomplish that as a small-group problem-solving exercise, students engage in group cognition and knowledge co-construction [3]. Group discussion forces students to verbalize the concepts and explain their reasoning [4]. The COMPS exercise protocol asks students to come to agreement on parts of the exercise, often by constructing an answer in a separate text window. Then an instructor or TA inspects the agreed-upon answer and provides feedback. During one semester of the Java class, COMPS exercises replace up to four regular programming labs as a way to reinforce student conceptual

skills that could be neglected by code-writing exercises.

The other class that provided conversation transcripts for this experiment is a mathematics class for elementary school teachers. This class routinely uses face-to-face small group problem-solving for exploratory learning of mathematics concepts. The mathematical manipulations are not difficult for college students; it is the explanation about why they work that must be learned. In the Poison game dialogues utilized in this paper, students figure out if there is a guaranteed winning strategy in a Nim-like game. The Poison exercise is a regular exercise in the curriculum. In this class, also, the instructor oversees the conversations.

Not only are the discussion topics quite different between the two classes, the populations of students are different. The programming class students attended an HBCU public institution, the mathematics class students attended a private sectarian institution.

We have shown in the programming class that in each three-person discussion group, the two less prepared students show large learning gains. Preparedness was measured before the exercise by a pretest. The most prepared student on the average doesn't show learning gains, but nevertheless the most prepared student is likely to highly rate the exercise as having worked well and also as having been helpful [1].

Transactive Behavior as a Marker of Conversation Functioning

The COMPS dashboard program detects low-functioning conversations by estimating a low prevalence of transactive dialogue turns. Transactivity is the way people interact with each other in knowledge-constructing dialogue [5]. Examples of transactive contributions are agreeing or disagreeing, providing a warrant for the other person's earlier assertion, challenging an earlier statement, or summarizing other people's statements. What makes transactivity distinct from merely advancing the argument is the social interaction. These same argumentative acts are not transactive if they were performed by one person with little participation from others.

We posit that detecting a low level of transactive behaviors will be an indication that a conversation is not likely to be productive. Among several cognitive frameworks explaining how knowledge co-construction occurs in collaborative discourse, transactive behaviors are common to all of them [6]. Therefore transactivity can be used as a marker of conversation quality even without subscribing to a particular cognitive theory. Note, however, that although knowledge co-construction requires transactive discourse, transactive discourse may occur without knowledge co-construction. We cannot posit that a high level of transactivity indicates that a conversation is necessarily productive.

Measuring Transactive Behavior

It is difficult to identify and categorize the distinct transactive acts in written dialogue. A COMPS project attempt to machine-classify which turns were initiating a conversational exchange and which were responding performed only a little better than chance [7]. Rather than identify and count individual transactive dialogue utterances, this experiment uses other markers to estimate the presence of transactivity. An experiment with spoken dialogue found that instances of people adapting their speech patterns to the other speaker correlate with people responding to each other transactively [6]. In that study the

prevalence of transactive behavior could thus be estimated even without knowing what the participants are saying.

In this work we endeavor to estimate the prevalence of transactive behavior by counting two behaviors: a) the students are talking substantively about a topic, b) the students are engaging with each other. Transactive dialogue requires both phenomena. For example, if the students are talking about the topic but not addressing each other or engaging with each others' ideas then the dialogue wouldn't be transactive.

For monitoring the state of the conversation, even somewhat unreliable detection of these phenomena might still provide enough correct data to tell the instructor which of the conversations are impoverished with respect to these behaviors, and therefore are most likely in need of attention.

EXPERIMENT IN COMPUTER RECOGNITION OF DIALOGUE PHENOMENA

The computer algorithms to recognize and count the markers of the two behaviors are described in this section. First we provide operational definitions of the phenomena. Then we describe the data that we used for training computer text classifiers to detect them. Then we describe the lexical and vocabulary transformations of the text. Finally we describe the features extracted from the text that provide additional input to the text classifier algorithm.

Operational Definitions of the Phenomena to be Recognized

a) Talking substantively about a topic. We use the shorthand name "reasoning" for the presence of a student talking about a topic. It does not mean the student is making inferences. We have operationalized the reasoning feature to be: the dialogue turn contains some material from the problem the students are working on, connected with some other words. For example, if the whole dialogue turn in a discussion of a Java program is "int x", that is not labeled reasoning. Similarly "let's try something" isn't reasoning. However, "let's try int x" is labeled as reasoning, because it contains both.

b) Engaging with each other. If a dialogue utterance contains expressions of agreement or disagreement, we surmise that it is likely that one student is addressing another. We have picked these two common dialogue acts as a proxy for measuring how frequently students engage with each other.

Dialogue Training and Testing Dataset

This experiment trained the text classifiers using dialogue transcripts from the two different classes described above in the Background section. About 4400 turns of dialogue from 32 discussion groups were were manually annotated. Statistics are shown in Table 1. Annotators were six people, with every turn annotated at least twice. Annotations were: reasoning, agreement, and disagreement. Approximately 51% of turns were labeled reasoning, 18% were labeled agreement, and 9% were labeled disagreement.

The dialogue extract illustrated in Figure 1 shows the hand-annotated features. The "agree" feature is notated "Agr" here if the student shows evidence of agreeing with a previous student's utterance, "Dis" for disagreeing, and "--" if neither is shown. Notice that the word "agree" in the dialogue is not always an indication of agreement. In turn 13

the student agrees with another. By contrast in turn 4 the TA tells the students to come to agreement, and in turn 12 student B is seeking agreement.

Table 1: Statistics of Dialogue Data

	Total	Java	Math
Number of enter-delimited chat-turns	5969	3406	2563
Number of dialogue turns (combining consecutive chat-turns from one person)	4409	2394	2015
Number of dialogues	32	19	13
Average turns per dialogue	138	126	155

Text Regularization

Chat dialogue text is regularized before text analysis. There are differences between computer chat-typing text and normative English which are accounted for during this preprocessing. In addition, the vocabulary is severely reduced, to eliminate dependence on exercise-specific words.

Both the lexical and vocabulary regularization utilize a dictionary developed for this project. In earlier work COMPS text classifiers were sensitive to the very specific vocabulary of the problems being discussed, so for this experiment we restricted the vocabulary to common words. At its base is a 10,000 word dictionary of common words available from Google [8], which was derived from the trillion-word corpus published by the n-grams project [9]. We modified this dictionary as follows:

- Typed chat employs unique words, many of them abbreviations. Examples are "idk" for "I don't know" and "brb" for "be right back." We added about 50 such words, found in our chat and in word lists harvested from the web.
- The 10,000 most common English words contained people's names, e.g. "Jane" and "Jose." We removed common names to avoid accidentally training the classifiers on the names of participants who happened to be mentioned in the training dialogue text.
- Variants of common words that were missing from the lexicon but occurred in our corpus were added, for example plurals of common nouns and alternative tenses of common verbs.

To regularize the words, we wrote algorithms to preprocess phenomena as follows. Many of these have been described by [10].

- Stretching. In typed chat letters can be repeated for emphasis. Two examples attested in our corpus were "soooooo" and "wayyy." If a letter occurs more than twice we elide the repetitions until the word matches one in the lexicon.
- @ for referring to people is stripped off and a special token added as a separate word into the text. For example "@prof" becomes "$g02 prof." Through this

special word, the text classifier can recognize that a speaker referred to somebody else, which potentially is a transactive behavior.

- A string of one or more caret characters (^), which is used in computer chat to point to earlier utterances, is replaced by a token. The caret figuratively points "up" to an earlier turn, and thus is potentially an indication of transactive behavior.
- Similarly, other special typed-chat punctuation conventions such as #hashtags and asterisks for *emphasis* are stripped off and replaced by tokens.
- Spelling errors are corrected. Correction is applied to words 4 characters or longer, except those containing digits. We compared the unknown word with words in the lexicon that differed by 1 or 2 single-character changes. Spelling correction repaired a number of typed-chat phenomena, such as restoring dropped appostrophes (e.g., "dont") and fixing impromptu abbreviations.

Text Features

Other features were derived from the regularized dialogue turns before presenting each turn to the classifier. The most salient were topic model and length of the turn. The length feature was computed as 1 + log(number_of_words), and defined as 0 when the number of words was zero. A meaningful dialogue turn could have no words, for example, if the student typed a single question mark or a smiley. The "topics" which comprise a topic model are collections of words that are likely to occur together [2, 11]. They are derived by applying the Latent Dirichlet Analysis (LDA) statistical method to the training corpus. Topics in the model sometimes contain recognizably related collections of words, but often are not comprehensible to the human reader.

RESULTS AND FUTURE WORK

Scikit-Learn was used for training and testing classifiers. As discussed in the introduction, topic models and classifiers are trained and tested on the annotated corpus, and are saved for repeated use in the dashboard. We trained linear regression classifiers for each of the "reason," "agree," and "disagree" features as shown in Figure 1. We used 60% of the data for training, randomly drawn, and 40% for testing.

Results from testing the trained classifiers is shown in Table 2. Recall is the fraction of the turns bearing the target feature which are identified; precision is the fraction of identified results which are correct. The discrimination threshhold value for the regression results (the value of the regression result that was used to discriminate between declaring the target as present or absent) was picked to maximize the F1 value, a statistic which balances precision and recall. AUC is the area under the precision vs. recall curve.

Compared to previous work, we have improved the linear classifier F1 score for identifying reasoning from 0.59 to 0.83 [2]. This improvement occurred at the same time as we restricted the vocabulary, broadened the variety of dialogue transcripts, and tested on a larger set, all of which work to make the classification task more difficult.

Table 2. Accuracy of Regression Classifiers

Target Attribute	Precision	Recall	F1	AUC
Reasoning	0.748	0.938	0.832	0.879
Agreement	0.481	0.661	0.557	0.776
Disagreement	0.222	0.333	0.267	0.818

The agreement and disagreement classifiers detect a subset of the ways that students interact with each other. We will experiment with other text features that might help the classifiers detect interaction. Other features we can extract from the text which might be indicative of interaction include you-pronouns (e.g. “you,” “us,” “we”) that indicate one person is addressing another, and smileys and emojis, which express affect to other people.

Manual inspection of the topic models shows that we have not entirely achieved problem independence. A success is that we observe topics containing non-specific reasoning words and agreement words, such as “correct,” “wrong,”, “yeah,” and “agree.” In an earlier experiment topics often related to specific Java problems. They contained Java language tokens and variable names from the code in the Java exercises [2]. Since the specialized words are gone, these topics no longer appear. However the topic models clearly still contain problem-specific topics, albeit made of common words. In one topic the most prominent few words include “team,” “we,” and “they.” This topic comes from the mathematics dialogues where the students would divide into teams and experimentally play the game they are analyzing. Other topics contain Java tokens that are also common words, such as “object,” “method,” and “static.” We will test how well the topic model works with dialogues from other problems.

CONCLUSIONS

The COMPS project is developing software to monitor students engaging in group problem-solving chat exercises. The theoretical basis is to measure whether the conversation contains transactive dialogue turns, which is estimated by measuring how often the dialogue turns are substantive and how often the students are addressing each other.

The accuracy of the substantive-contribution classifier has been improved. At the same time its functioning has been partially distanced from dependence on the vocabularies of particular problems and particular student populations. This was achieved by limiting its input to common English words, and by training it on several divergent conversation topics and student populations. We have used the substantive-contribution classifier, combined with a measure of whether all students are participating, to post dashboard values in real time during Java COMPS labs. Anecdotal reports from the instructors indicate that the relative ordering of the computer’s judgments roughly corresponded to their own judgments.

ACKNOWLEDGMENT

Partial support for this work was provided by the National Science Foundation's Improving Undergraduate STEM Education (IUSE) program Award No. 1504918 and

MSEED program Award No. 1068346. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Glass, M., Kim, J. H., Kim, T. Early experience with computer supported collaborative exercises for a 2nd semester Java class, *Journal of Computing Sciences in Colleges*, 32 (2), 96-105, 2016.

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

[3] Stahl, G., ed. *Studying Virtual Math Teams*, Springer, 2009.

[4] Koschmann, T. Understanding understanding in action, J*ournal of Pragmatics*, 43 (2), 435-437, 2011.

[5] Weinberger, A., Fischer, F. A framework to analyze argumentative knowledge construction in computer-supported collaborative learning. *Computers & Education* 46 (1), 71-95, 2006.

[6] Gweon, G., Jain, M., Mcdonough, J., Raj, B., Rosé, C. P. Measuring prevalence of other-oriented transactive contributions using an automated measure of speech style accommodation, *International Journal of Computer-Supported Collaborative Learning*, 8 (2), 245-265, 2013.

[7] Glass, M., Kim, J. H., Bryant, K., Desjarlais, M. Indicators of conversational interactivity in COMPS problem-solving dialogues, Third Workshop on Intelligent Support for Learning in Groups (ISLG), 2014.

[8] Google, google-10000-english (online corpus file), 2017, `https://github.com/first20hours/google-10000-english`

[9] Franz, A., Brants, T. All our n-grams belong to you. *Google Research Blog*. 2006, h`ttps://research.googleblog.com/2006/08/all-our-n-gram-are-belong-to-you.html`.

[10] Bieswanger, M. Micro-linguistic structural features of computer-mediated communication, in Herring, S., Stein, D., Virtanen, T., eds., *Pragmatics of Computer-Mediated Communication*, De Gruyter Mouton, 2013.

[11] Řehůřek, R., Sojka, P. Software framework for topic modeling with large corpora, *Proc. LREC Workshop on New Challenges for NLP Frameworks*, 45-50, 2010.