A Machine Learning Approach to Assessing Knowledge Sharing During Collaborative Learning Activities

Amy Soller, Janyce Wiebe, Alan Lesgold

University of Pittsburgh
Pittsburgh, PA 15260
soller@pitt.edu, wiebe@cs.pitt.edu, al@pitt.edu

ABSTRACT

Students bring to a collaborative learning situation a great deal of specialized knowledge and experiences that undoubtedly shape the collaboration and learning processes. How effectively this unique knowledge is shared and assimilated by the group affects both the process and the product of the collaboration. In this paper, we describe a machine learning approach, Hidden Markov Modeling, to analyzing and assessing on-line knowledge sharing conversations. We show that this approach can determine the effectiveness of knowledge sharing episodes with 93% accuracy, performing 43% over the baseline. Understanding how members of collaborative learning groups share, assimilate, and build knowledge together may help us identify situations in which facilitation may increase the effectiveness of the group interaction.

Keywords

Assessing collaborative learning, interaction analysis, knowledge sharing, dialog coding, machine learning

INTRODUCTION

A group of students gather around a table to solve a problem, and begin to exchange the knowledge each brings to bear on the problem. Each group member brings to the table a unique pool of knowledge, grounded in his or her individual experiences. The combination of these experiences, and the group members' personalities and behaviors will determine how the collaboration proceeds, and whether or not the group members will effectively learn from and with each other (Brown and Palincsar, 1989; Dillenbourg, 1999; Webb & Palincsar, 1996).

If we take a closer look at the interaction in this group, we might see that the way in which a student shares new knowledge with the group, and the way in which the group responds, determines to a large extent how well this new knowledge is assimilated into the group, and whether or not the group members learn the new concept. It is reasonable to assume that, in effective knowledge sharing conversation, the presentation (sharing) of new concepts and ideas would initiate questioning, explaining, and critical discussion. Studying the interaction that provokes and follows knowledge sharing events may help us assess the ability of the group to assimilate new information that group members naturally bring to bear on the problem.

In this paper, we describe a machine learning approach, Hidden Markov Modeling, to identifying, analyzing, and assessing on-line knowledge sharing conversations. We begin by discussing work related to analyzing knowledge sharing conversations, and then describe how Hidden Markov Modeling was used to assess these conversations. The fourth section reports on the results of an experiment in which this technique was successfully used to classify instances of effective and ineffective knowledge sharing interaction. We conclude by discussing the implications of this research, and pointing to a few open-ended questions.

KNOWLEDGE SHARING

We define a *knowledge sharing episode* as a series of conversational contributions (utterances) and actions (e.g. on a shared workspace) that begins when one group member introduces new knowledge into the group conversation, and ends when discussion of the new knowledge ceases. New knowledge is defined as knowledge that is unknown to at least one group member other than the knowledge sharer. In general, analyzing knowledge sharing episodes involves the following three steps:

1. Determining which student played the role of knowledge sharer, and which the role(s) of receiver
2. Analyzing how well the knowledge sharer explained the new knowledge
3. Observing and evaluating how the knowledge receivers assimilated the new knowledge

The use of Hidden Markov Models to accomplish step (1) above is described in (Soller and Lesgold, in press). In this paper, we describe their application to steps (2) and (3). Studying the effectiveness of knowledge sharing involved collecting sequences of interaction in which students shared new knowledge with their peers, and relating these sequences to the group members' performance on pre and post tests. The tests targeted the specific knowledge elements we expected the students to share and learn during the experiment. To ensure that high-quality knowledge sharing opportunities exist, each group member was provided with a unique piece of knowledge that the team needed to solve the problem. This *knowledge element* was designed to mirror the sort of unique knowledge that students might naturally bring to the problem from their own experiences. By artificially constructing situations in which students are expected to share knowledge, we single out interesting episodes to study, and more concretely define situations that can be compared and assessed.

In order for a knowledge element to be shared "effectively", three requirements must be satisfied (F. Linton, personal communication, May 8, 2001):

the individual sharing the new knowledge (the "sharer") must show that she understands it by correctly answering the corresponding pre and post test questions

the concept must come up during the conversation, and

at least one group member who did not know the concept before the collaborative session started (as shown by his pre-test) must show that he learned it during the session by correctly answering the corresponding post-test question.

In this paper, we focus on situations in which criteria (1) and (2) are satisfied, since these criteria are necessary for studying how new knowledge is assimilated by collaborative learning groups. Other research has addressed how students acquire new knowledge (criteria 1, Gott & Lesgold, 2000), and how to motivate students to share their ideas (criteria 2, Webb & Palincsar, 1996).

Experiments designed to study how new knowledge is assimilated by group members are not new to social psychologists. *Hidden Profile* studies (Lavery, Franz, Winquist, and Larson, 1999; Mennecke, 1997), designed to evaluate the effect of knowledge sharing on group performance, require that the knowledge needed to perform the task be divided among group members such that each member's knowledge is incomplete before the group session begins. The group task is designed such that it cannot be successfully completed until all members share their unique knowledge. Group performance is typically measured by counting the number of individual knowledge elements that surface during group discussion, and evaluating the group's solution, which is dependent on these elements.

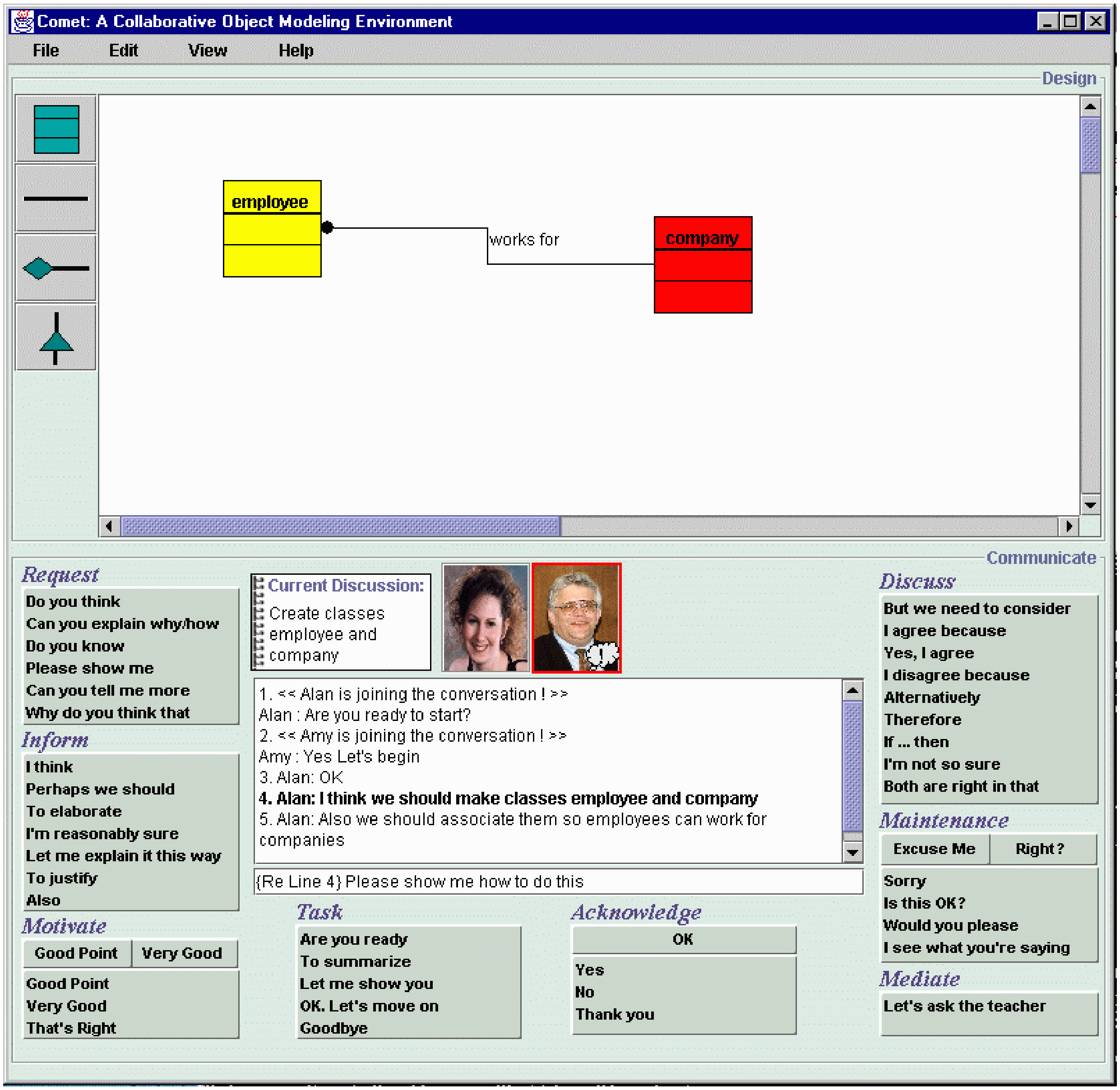
Surprisingly, studying the process of knowledge sharing has been much more difficult than one might imagine. Stasser (1999) and Lavery et al. (1999) have consistently shown that group members are not likely to discover their teammates' hidden profiles. They explain that group members tend to focus on discussing information that they share in common, and tend not to share and discuss information they uniquely possess. Moreover, it has been shown that when group members do share information, the quality of the group decision does not improve (Lavery et al., 1999; Mennecke, 1997). There are several explanations for this. First, group members tend to rely on common knowledge for their final decisions, even though other knowledge may have surfaced during the conversation. Second, "if subjects do not cognitively process the information they surface, even groups that have superior information sharing performance will not make superior decisions (Mennecke, 1997)." Team members must be motivated to understand and apply the new knowledge.

At least one study (Winquist and Larson, 1998) confirms that the amount of unique information shared by group members is a significant predictor of the quality of the group decision. More research is necessary to determine exactly what factors influence effective group knowledge sharing. One important factor may be the complexity of the task. Mennecke (1997) and Lavery et al.'s (1999) tasks were straightforward, short-term tasks that subjects may have perceived as artificial. Tasks that require subjects to cognitively process the knowledge that their teammates bring to bear may reveal the importance of effective knowledge sharing in group activities. In the next section, we describe one such task.

EXPERIMENTAL METHOD

In our experiment, five groups of three were each asked to solve one Object-Oriented Analysis and Design problem using a specialized shared workspace, while communicating through a structured, sentence opener interface. The communication interface, shown on the bottom half of Figure 1, contains sets of sentence openers (e.g. "I think", "I agree because") organized in intuitive categories (such as Inform or Discuss). To contribute to the group conversation, a student first selects a sentence opener. The selected phrase appears in the text box below the group dialog window, where the student may type in the rest of the sentence. Each sentence opener is associated with a particular conversational intention, given by a subskill and attribute. For example, the opener, "I think" corresponds to the subskill (or category) "Inform", and the more specific attribute, "Suggest".

Sentence openers provide a natural way for users to identify the intention of their conversational contribution without fully understanding the significance of the underlying communicative acts (Baker & Lund, 1997, McManus & Aiken, 1995). The categories and corresponding phrases on the interface represent the conversation acts most often exhibited during collaborative learning and problem solving in a previous study (Soller et al, 2001). Further details about the functionality of the communication interface can be found at http://lesgold42.lrdc.pitt.edu/EPSILON/Epsilon_software.html.



Current Discussion:
Create classes employee and company

1. << Alan is joining the conversation ! >>
Alan : Are you ready to start?
2. << Amy is joining the conversation ! >>
Amy : Yes Let's begin
3. Alan: OK
4. Alan: I think we should make classes employee and company
5. Alan: Also we should associate them so employees can work for companies

{Re Line 4} Please show me how to do this

Figure 1. The shared OMT workspace (top), and sentence opener interface (bottom)

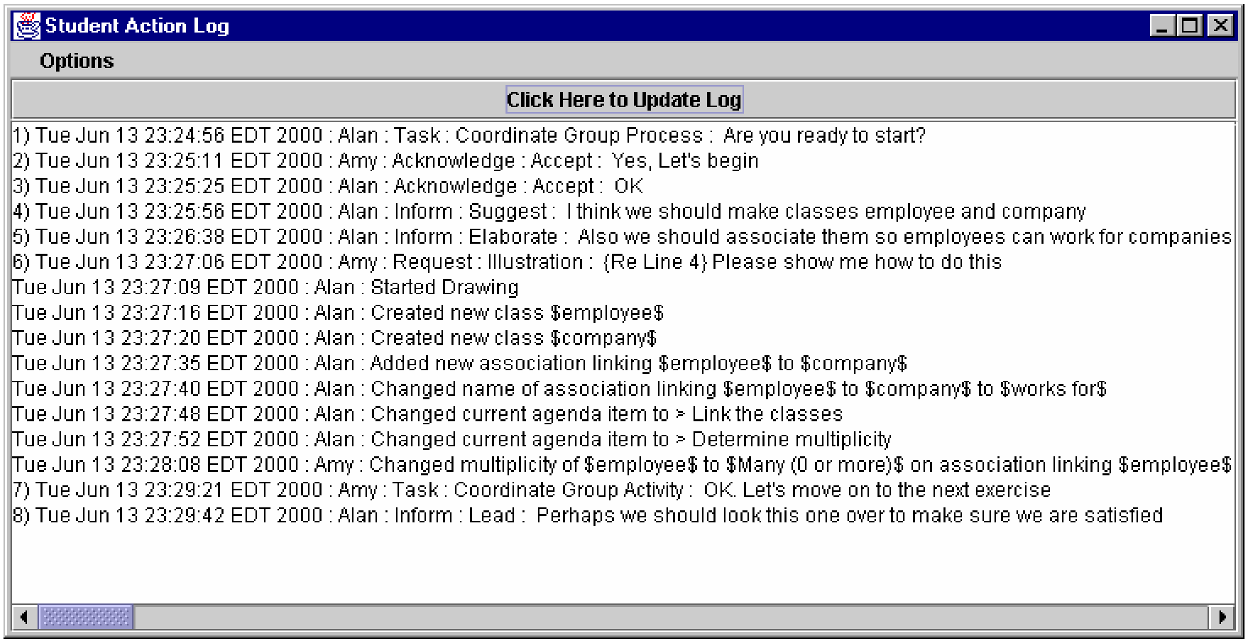
The specialized shared workspace is shown on the top half of Figure 1. The workspace allows students to collaboratively solve object-oriented design problems using Object Modeling Technique (OMT) (Rumbaugh, Blaha, Premerlani, Eddy, and Lorensen, 1991), an object-oriented analysis and design methodology. Software engineers use methodologies such as OMT to construct graphical models for optimizing their designs before implementation, and to communicate design decisions. These models are also useful for preparing documentation, or designing databases. Object-oriented analysis and design was chosen because it is an open-ended domain usually done in industry by teams of engineers with various expertise, so it is also an inherently collaborative domain. An example of an OMT design problem is shown below.

Exercise: Prepare a class diagram using the Object Modeling Technique (OMT) showing relationships among the following object classes: school, playground, classroom, book, cafeteria, desk, chair, ruler, student, teacher, door, swing. Show multiplicity balls in your diagram.

The shared OMT workspace provides a palette of buttons down the left-hand side of the window that students use to construct objects, and link objects in different ways depending on how they are related. Objects on the shared workspace can be selected, dragged, and modified, and changes are reflected on the workspaces of all group members.

Subjects. Five groups of three students each participated in the study. The subjects were undergraduates or first-year graduate students majoring in the physical sciences or engineering, none of which had prior knowledge of Object Modeling Technique. The subjects received pizza halfway through the four hour study, and were paid at the completion of the study.

Procedure. The five groups were run separately. The subjects in each group were asked to introduce themselves to their teammates by answering a few personal questions. Each experiment began with a half hour interactive lecture on OMT basic concepts and notation, during which the subjects practiced solving a realistic problem. The subjects then participated in a half hour hands-on software tutorial. During the tutorial, the subjects were introduced to all 36 sentence openers on the interface. The subjects were then assigned to separate rooms, received their individual knowledge elements, and took a pre-test. Individual knowledge elements addressed key OMT concepts, for example, "Attach attributes common to a group of subclasses to a superclass." Each knowledge element was explained on a separate sheet of paper with a worked-out example. The pre-test included one problem for each of the three knowledge elements. It was expected that the student given knowledge element #1 would get only pre-test question #1 right, the student given knowledge element #2 would get only pre-test question #2 right, and likewise for the third student. To ensure that each student understood his or her unique knowledge element, an experimenter reviewed the pre-test problem pertaining to the student's knowledge element before the group began the main exercise. Students who missed the pre-test problem on their knowledge element were asked to reread their knowledge element sheet and rework the missed pre-test problem, while explaining their work out loud (Chi et al., 1989).



```
1) Tue Jun 13 23:24:56 EDT 2000 : Alan : Task : Coordinate Group Process :  Are you ready to start?
2) Tue Jun 13 23:25:11 EDT 2000 : Amy : Acknowledge : Accept :  Yes, Let's begin
3) Tue Jun 13 23:25:25 EDT 2000 : Alan : Acknowledge : Accept :  OK
4) Tue Jun 13 23:25:56 EDT 2000 : Alan : Inform : Suggest :  I think we should make classes employee and company
5) Tue Jun 13 23:26:38 EDT 2000 : Alan : Inform : Elaborate :  Also we should associate them so employees can work for companies
6) Tue Jun 13 23:27:06 EDT 2000 : Amy : Request : Illustration :  {Re Line 4} Please show me how to do this
Tue Jun 13 23:27:09 EDT 2000 : Alan : Started Drawing
Tue Jun 13 23:27:16 EDT 2000 : Alan : Created new class $employee$
Tue Jun 13 23:27:20 EDT 2000 : Alan : Created new class $company$
Tue Jun 13 23:27:35 EDT 2000 : Alan : Added new association linking $employee$ to $company$
Tue Jun 13 23:27:40 EDT 2000 : Alan : Changed name of association linking $employee$ to $company$ to $works for$
Tue Jun 13 23:27:48 EDT 2000 : Alan : Changed current agenda item to > Link the classes
Tue Jun 13 23:27:52 EDT 2000 : Alan : Changed current agenda item to > Determine multiplicity
Tue Jun 13 23:28:08 EDT 2000 : Amy : Changed multiplicity of $employee$ to $Many (0 or more)$ on association linking $employee$
7) Tue Jun 13 23:29:21 EDT 2000 : Amy : Task : Coordinate Group Activity :  OK. Let's move on to the next exercise
8) Tue Jun 13 23:29:42 EDT 2000 : Alan : Inform : Lead :  Perhaps we should look this one over to make sure we are satisfied
```

Figure 2. The student action log dynamically records all student actions and conversation

The subjects were not specifically told that they hold different knowledge elements, however they were reminded that their teammates may have different backgrounds and knowledge, and that sharing and explaining ideas, and listening to others' ideas is important in group learning. All groups completed the OMT exercise on-line within about an hour and fifteen minutes. During the on-line session, the software automatically logged the students' conversation and actions (see Figure 2). After the problem solving session, the subjects completed a post-test, and filled out a questionnaire. The post-test, like the pre-test, addressed the three knowledge elements. It was expected that the members of effective knowledge sharing groups would perform well on all post-test questions.

The next section describes the findings from this study, gives a brief introduction to the analysis method, Hidden Markov Models, and discusses how we used them to train a computer to recognize instances of effective and ineffective knowledge sharing.

RESULTS

Four of the five groups showed both instances of effective knowledge sharing and instances of ineffective knowledge sharing. Recall from the section on knowledge sharing that in order for a knowledge element to be effectively shared, three requirements must be satisfied: (1) the individual sharing the new knowledge (the “sharer”) must show that she understands it by correctly answering the corresponding pre and post test questions, (2) the concept must come up during the conversation, and (3) at least one group member who did not know the concept before the collaborative session started (as shown by his pre-test) must show that he learned it during the session by correctly answering the corresponding post-test question (F. Linton, personal communication, May 8, 2001).

Since there were 15 subjects, there were a maximum of 30 possible opportunities for effective knowledge sharing: 2 opportunities for each student to learn the other 2 students’ elements. Ten of these were effective (i.e. they met all 3 criteria), and two students did not meet criteria (1), eliminating 4 opportunities. We are now in the process of determining why the students did not take advantage of the other 16 opportunities.

The student action logs (e.g. Figure 2) from the five experiments were parsed by hand to extract the dialog segments in which the students shared their unique knowledge elements. Fourteen of these *knowledge sharing episodes* were identified, and tagged as either effective or ineffective (this process is described later in this section). These sequences do not directly correspond to the 30 opportunities in the previous paragraph, since one episode may result in 2 students learning, or one student may learn across several episodes. The knowledge sharing episodes were used to train a system to analyze and classify new instances of knowledge sharing. We now describe the training algorithm, and how it was applied.

A Brief Introduction to Hidden Markov Models

Hidden Markov Models (HMMs) were used to model the sequences of interaction present in the knowledge sharing episodes from the experiment. HMMs were chosen because of their flexibility in evaluating sequences of indefinite length, their ability to deal with a limited amount of training data, and their recent success in speech recognition tasks. We begin our introduction to HMMs with an introduction to Markov chains.

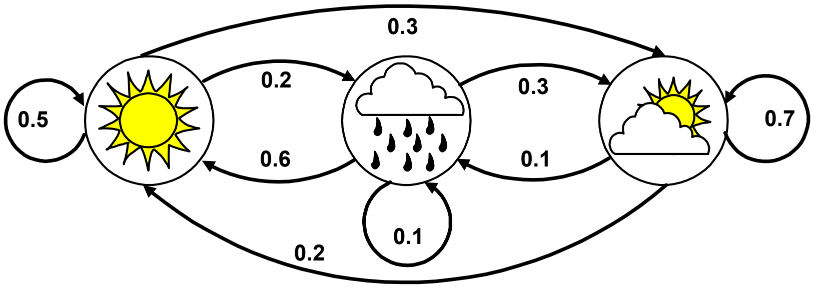


Figure 3. A Markov chain describing the probability of various weather patterns

Markov chains are essentially probabilistic finite state machines, used to model processes that move stochastically through a series of predefined states. For example, a model of the weather might include the states sunny, rainy, and overcast (see Figure 3). The probability of entering a rainy state after visiting a sunny state might be 0.2, the probability of entering an overcast state 0.3, and the probability of another sunny state 0.5. In other words, if today is sunny, there is a 20% chance that tomorrow will be rainy, a 30% chance that tomorrow will be overcast, and a 50% chance that it will be sunny again. In Markov chains, the arcs describe the probability of moving between states. The probability of a sequence of states is the product of the probabilities along the arcs. So, if today is sunny, then the probability that tomorrow will be rainy, and the next day overcast (0.2)(0.3) = 0.06.

Hidden Markov Models generalize Markov Chains in that they allow several different paths through the model to produce the same output. Consequently, it is not possible to determine the state the model is in simply by observing the output (it is “hidden”). Markov models observe the Markov assumption, which states that the probability of the next state is dependent *only* upon the previous state. This assumption seems limiting, however efficient algorithms have been developed that perform remarkably well on problems similar to that described here. Hidden Markov Models allow us to ask questions such as, “How well does a new (test) sequence match a given model?”, or, “How can we optimize a model’s parameters to best describe a given observation (training) sequence?” (Rabiner, 1989). Answering the first question involves computing the most likely path through the model for a given output sequence; this can be efficiently computed by the Viterbi (1967) algorithm. Answering the second question requires training an HMM given sets of example data. This involves estimating the (initially guessed) parameters of an arbitrary model repetitively, until the most likely parameters for the training examples are discovered. The explanation provided here should suffice for understanding the analysis in the next section. For further details on HMMs, see Rabiner (1989) or Charniak (1993).

Coding the Interaction

The fourteen knowledge sharing episodes varied in length from 5 to 62 contributions, and contained both conversational elements and action events. The top part of Figure 4 shows an example of one such sequence. The sentence openers, which indicate the system-coded subskills and attributes, are italicized. The bottom part of Figure 4 shows the actual sequence that is used to train the HMM to recognize similar knowledge sharing sequences.

Student	Subskill	Attribute	Actual Contribution (Not seen by HMM)
A	Request	Opinion	*Do you think* we need a discriminator for the car ownership
C	Discuss	Doubt	*I'm not so sure*
B	Request	Elaboration	*Can you tell me more* about what a discriminator is
C	Discuss	Agree	*Yes, I agree* because I myself am not so sure as to what its function is
A	Inform	Explain/Clarify	*Let me explain it this way* - A car can be owned by a person , a company or a bank. I think ownership type is the discrinator.
A	Maintenance	Apologize	*Sorry* I mean discriminator.

Actual HMM Training Sequence
A-Request-Opinion
C-Discuss-Doubt
B-Request-Elaboration
C-Discuss-Agree
A-Inform-Explain
A-Maintenance-Apologize
Sequence-Termination

Figure 4. An actual logged knowledge sharing episode (above), showing system coded subskills and attributes, and its corresponding HMM training sequence (below)

Some of the extracted sequences included actions that students took on the workspace. These actions were matched to a list of predetermined "productive" actions – those that were expected to lead students to a model solution. Productive actions were labeled as such, and included in the sequence with the name of the student who took the action (e.g. A-Productive-Action).

The system codes were obtained directly from the sentence openers that students choose to begin their contributions, and may not accurately reflect the intention of the contribution. For example, a student might choose the opener, "I think", and then add, "I disagree with you". Each sentence opener is associated with one subskill and attribute pair that most closely matches the expected use of the phrase; however even having gone through sentence opener training (described in the previous section), students may not always use the openers as expected. In order to determine to what degree the students used the openers as they were intended, 2 researchers recoded 3 of the 5 dialogs (selected at random). Tables 1 and 2 show the agreement between the 2 coders (A and B) and between each of the coders and the system, averaged over all 3 dialogs. As shown by the tables, agreement between the raters and the system was high for the subskill case, and reasonable for the attribute case (Carletta et al., 1997).

Table 1. Agreement statistics for subskill codes
codes

Coder 1	Coder 2	% Agreement	κ
A	B	87.0	.85
A	System	90.1	.88
B	System	86.4	.84
Average of A & B	System	88.25	.86

Table 2. Agreement statistics for attribute

Coder 1	Coder 2	% Agreement	κ
A	B	71.2	.71
A	System	85.5	.73
B	System	71.5	.60
Average of A & B	System	78.49	.66

The next section describes the results of training Hidden Markov Models to assess the effectiveness of the 14 knowledge sharing episodes. This analysis was done using the system codes (those based on the sentence openers that the students selected), however similar results were obtained when the recoded dialogs were substituted as test sequences.

Assessing the Effectiveness of Knowledge Sharing Episodes

Two 6 state Hidden Markov Models were trained[1]. The first was trained using only sequences of effective knowledge sharing interaction (we call this the effective HMM), and the second using only sequences of ineffective knowledge sharing (the ineffective HMM). Testing the models involved running a new knowledge sharing sequence – one that is not used for training – through both models. The output from the effective HMM described the probability that the new test sequence is effective, and the output from the ineffective HMM described the probability that the new test sequence is ineffective. The test sequence was then classified as effective if has a higher path probability through the effective HMM, or ineffective if its path probability through the ineffective HMM was higher. Since the probabilities in these models can be quite small, we usually take the log of the path probability, which results in a negative number. The largest path probability is then given by the smallest absolute value.

Since HMMs "learn" by generalizing sets of examples, training the HMMs to model effective and ineffective knowledge sharing meant collecting sequences of interaction indicative of effective and ineffective interaction. The transcripts from the experiment described earlier were parsed, and 14 situations were identified in which the students discussed the unique knowledge elements each learned before the problem solving session began. These 14 sequences were tagged as being either effective or ineffective. A sequence is considered effective if at least one of the students receiving the new knowledge did not know it before the session (as shown by his pre-test) and demonstrated that he learned it during the session (as shown by his post-test). Recall that the pre and post tests directly target the three knowledge elements that the students are expected to share during the group problem solving session (see section entitled, "Experimental Method"). A sequence is considered ineffective if a knowledge element was discussed during the episode, but none of the receiving students demonstrated mastery of the concept on the post test.

Of the 14 knowledge sharing sequences identified, 7 were found to be effective and 7 were found to be ineffective. Because of the small dataset, we used a 14-fold cross validation approach, in which we tested each of the 14 examples against the other 13 examples (as training sets), and averaged the results. Figure 5 shows the path probabilities of each test sequence through both the effective and ineffective HMMs. The y-value shows the log of the Viterbi path probability (Rabiner, 1989). This value is highly dependent on the length of the test sequence (longer sequences will produce smaller probabilities), and so will vary for each sequence. Notice that the path probabilities of the 7 effective test sequences (labeled E1 through E7) were higher through the effective HMM, and the path probabilities for 6 of the 7 ineffective test sequences (labeled I8 through I14) were higher through the ineffective HMM, resulting in an overall 92.9% accuracy. The baseline comparison is chance, or 50%, since there is a 1/2 chance of arbitrarily classifying a given test sequence as effective or ineffective. The HMM approach successfully performed at almost 43% above the baseline.

[1] Before choosing the 6 node HMM, we experimented with 3, 4, and 5 node HMMs, obtaining similar (but not optimal) results. Performance seemed to decline with 7 or more states.

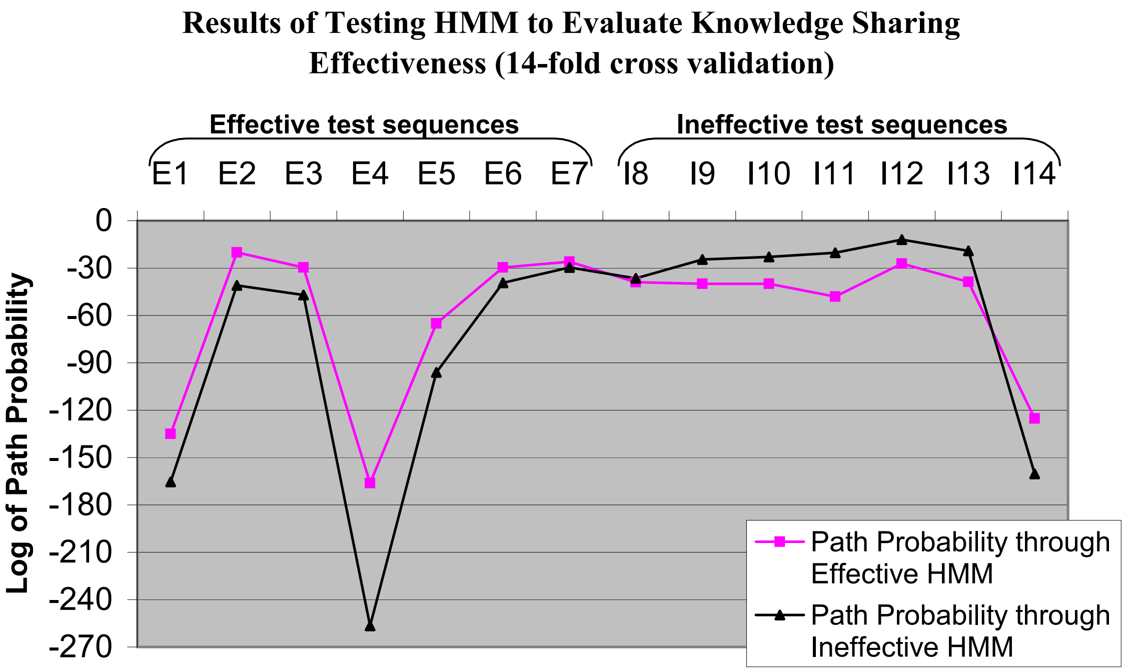


Figure 5. Viterbi path probabilities of each test sequence through both the effective and ineffective HMMs

The analysis in this section shows that artificial intelligence models of collaborative interaction may be useful for identifying when students are effectively sharing the new knowledge they bring to bear on the problem. Once we have discovered a situation in which students are not effectively interacting, we can formulate hypotheses about the various facilitation methods that might help the students collaborate more effectively.

DISCUSSION AND FUTURE WORK

Determining from a sequence of coded interaction, such as that shown in Figure 4, how well new knowledge is assimilated by the group is a very difficult task. Other researchers have explored a number of different methods, including finite state machines (McManus & Aiken, 1995), fuzzy inferencing (Barros & Verdejo, 1999), decision trees (Constantino-Gonzalez & Suthers, 2000; Goodman, Hitzeman, Linton, & Ross, 2001), rule learning (Katz, Aronis, & Creitz, 1999), and plan recognition (Muehlenbrock & Hoppe, 1999), for analyzing collaborative learning interaction (see Jermann, Soller, and Muehlenbrock, 2001, for a review of different approaches). Why does the HMM approach work so well? The models are trained to represent the possible ways that a student might share new knowledge with his teammates, and the possible ways that his teammates might react. The HMM, in this case, is therefore a sort of compiled conversational model. This means that, for example, the effective model includes a compilation of the conversational patterns students use when knowledge is effectively built by the group members. Our next step is to take a closer look at the differences between the effective and ineffective sequences in order understand the qualitative differences. For example, we might expect to see more questioning and critical discussion in effective knowledge sharing episodes, and more acknowledgement in less effective episodes (Soller, 2001).

The long-term goal of this project is to support learning groups on-line by mediating situations in which new knowledge is not effectively assimilated by the group. Understanding *why* a knowledge sharing episode is ineffective is critical to selecting a proper mediation strategy. A knowledge sharer may need help in formulating sufficiently elaborated explanations using, for example, analogies or multiple representations. Or, a knowledge receiver may need encouragement to speak up and articulate why he does not understand a new knowledge element. Research is now underway to develop a generalized model of ineffective knowledge sharing that includes models in which new knowledge is not effectively conveyed by the sharer, and models in which new knowledge is not effectively assimilated by the receivers. A system that can differentiate between these cases may be able to better recommend strategies for supporting the process of knowledge sharing during collaborative learning activities.

CONCLUSION

Students bring to a collaborative learning situation a great deal of specialized knowledge and experiences that will undoubtedly shape the collaboration and learning processes. How effectively this unique knowledge is shared and assimilated by the group affects both the process and the product of the collaboration.

In this paper, we describe a novel approach to assessing the effectiveness of knowledge sharing conversation during collaborative learning activities. Our approach involves applying a machine learning technique, Hidden Markov Modeling, to differentiate instances of effective from ineffective knowledge sharing interaction.

The experiment we described here was designed specifically to collect instances of knowledge sharing during collaborative learning. These instances were coded to reflect both task and conversational events, and used to train two 6 state Hidden Markov Models. The models, when tasked to determine the effectiveness of new sequences of knowledge sharing interaction, correctly classified 92% of these sequences, a 42% improvement over the baseline. The preliminary results of this study are promising. We are now collecting more data so that we may confirm and elaborate on these findings

Our research goal is to analyze the knowledge sharing process, and identify situations in which facilitation might help to increase the effectiveness of the group interaction. Studying the interaction that provokes and follows knowledge sharing events may help us assess the ability of the group to assimilate new information that group members naturally bring to bear on the problem.

Understanding and supporting students' knowledge sharing behavior is a complex endeavor, involving analysis of student learning, understanding, conversation, and physical actions. But the results of this effort can be applied to analyzing and supporting other complex aspects of collaborative learning, such as the joint construction of shared knowledge, and cognitive conflict. Furthermore, this research may help to define guidelines about the limits on the kinds of support a collaborative learning system, in general, might offer.

ACKNOWLEDGEMENTS

The Kappa statistics presented here would not have been possible without the partnership and dedication of Tricia Chirumbole. Thanks also to Kwang-Su Cho and Patrick Jermann for helping to run the experiment, and for their suggestions and insights. This research was supported by the U.S Department of Education, grant R303A980192, and an Andrew Mellon Predoctoral Fellowship.

REFERENCES

Baker, M., and Lund, K. (1997). Promoting reflective interactions in a computer-supported collaborative learning environment. *Journal of Computer Assisted Learning*, 13, 175-193.

Barros, B., and Verdejo, M.F. (1999). An approach to analyse collaboration when shared structured workspaces are used for carrying out group learning processes. *Proceedings of the Ninth International Conference on Artificial Intelligence in Education*, 449-456.

Brown, A. and Palincsar, A. (1989). Guided, cooperative learning and individual knowledge acquisition. In L. Resnick (Ed.), *Knowledge, Learning and Instruction* (pp. 307-336), Lawrence Erlbaum Associates.

Carletta, J., Isard, A., Isard, S., Kowtko, J., Doherty-Sneddon, G., and Anderson, A. (1997). The reliability of a dialogue structure coding scheme. *Computational Linguistics,* 23, 13-32.

Charniak, E. (1993). *Statistical Language Learning*. Cambridge, MA: MIT Press.

Chi, M. T. H., Bassok, M., Lewis, M. W., Reinmann, P., and Glaser, R. (1989). Self-Explanations: How students study and use examples in learning to solve problems. *Cognitive Science*, 13, 145-182.

Constantino-Gonzalez, M., and Suthers, D. (2000). A coached collaborative learning environment for Entity-Relationship modeling. *Proceedings of the 5th International Conference on Intelligent Tutoring Systems*, Montreal, Canada, 324-333.

Dillenbourg, P. (1999). What do you mean by "Collaborative Learning". In P. Dillenbourg (Ed.) *Collaborative Learning: Cognitive and Computational Approaches* (pp.1-19). Amsterdam: Elsevier Science.

Goodman, B., Hitzeman, J., Linton, F., and Ross, H. (2001). Intelligent agents for collaborative learning: Predicting the roles of dialogue participants. Manuscript in preparation.

Gott, S., and Lesgold, A. (2000). Competence in the workplace: How cognitive performance models and situated instruction can accelerate skill acquisition. In R. Glaser (Ed.) *Advances in Instructional Psychology: Vol. 5. Educational Design and Cognitive Science* (pp. 239-327). Mahwah, NJ: Lawrence Erlbaum Associates.

Jermann, P., Soller, A., & Muehlenbrock, M. (2001). From mirroring to guiding: A review of state of the art technology for supporting collaborative learning. *Proceedings of the First European Conference on Computer-Supported Collaborative Learning*, Maastricht, The Netherlands, 324-331.

Katz, S., Aronis, J., and Creitz, C. (1999). Modelling pedagogical interactions with machine learning. *Proceedings of the Ninth International Conference on Artificial Intelligence in Education*, LeMans, France, 543-550.

Lavery, T., Franz, T., Winquist, J., Larson, J. (1999). The role of information exchange in predicting group accuracy on a multiple judgment task. *Basic and Applied Social Psychology*, 2(4), 281-289.

McManus, M. and Aiken, R. (1995). Monitoring computer-based problem solving. *Journal of Artificial Intelligence in Education*, 6(4), 307-336.

Mennecke, B. (1997). Using group support systems to discover hidden profiles: An examination of the influence of group size and meeting structures on information sharing and decision quality. *International Journal of Human-Computer Studies*, 47, 387-405.

Muehlenbrock, M., and Hoppe, U. (1999). Computer-supported interaction analysis of group problem solving. *Proceedings of the Third International Conference on Computer Support for Collaborative Learning (CSCL '99)*, Stanford, California, 398-405.

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorensen, W. (1991). *Object-Oriented modeling and design*. Englewood Cliffs, NJ: Prentice Hall.

Rabiner, L. (1989). A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), 257-286.

Soller, A. (2001) Supporting social interaction in an intelligent collaborative learning system. *International Journal of Artificial Intelligence in Education*, 12, in press.

Soller, A., and Lesgold, A. (in press). Modeling the process of knowledge sharing. In U. Hoppe & M. Ikeda (Eds.) *New Technologies for Collaborative Learning*. Kluwer Academic Publishers.

Stasser, G. (1999). The uncertain role of unshared information in collective choice. In L. Thompson, J. Levine, and D. Messick (Eds.) *Shared Knowledge in Organizations* (pp 49-69). Hillsdale, NJ: Erlbaum.

Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, 13, 260-269.

Webb, N., & Palincsar, A. S. (1996). Group processes in the classroom. In D. Berlmer & R. Calfee (Eds.), *Handbook of Educational Psychology* (pp. 841-873). New York: Simon & Schuster Macmillan.