Pedagogical Discourse: Connecting Students to Past Discussions and Peer Mentors within an Online Discussion Board

Jihie Kim and Erin Shaw

Information Sciences Institute, University of Southern California
4676 Admiralty Way, Marina del Rey CA 90292, United States
{jihie, shaw}@isi.edu

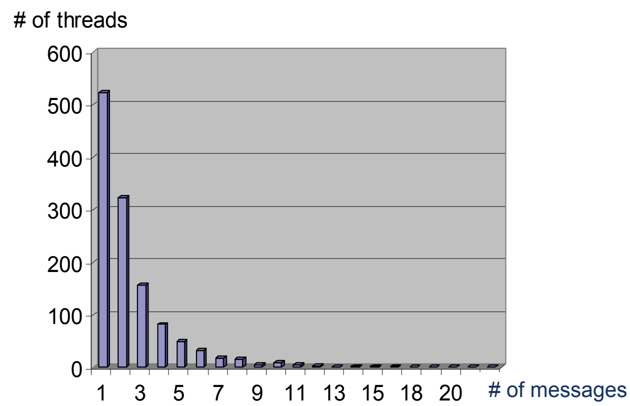
Abstract

The goal of the Pedagogical Discourse project is to develop instructional tools that will help students and instructors use discussion boards more effectively, with an emphasis on automatically assessing discussion activities and building tools for promoting student discussion participation and learning. In this paper, we present a two related participation and learning scaffolding tools that exploit natural language processing and information retrieval techniques. The PedaBot tool is designed to aid student knowledge acquisition and promote reflection about course topics by connecting related discussions from a knowledge base of past discussions to the current discussion thread. The MentorMatch tool is designed to promote student collaboration through the use of student mentors, i.e., course peers with a relatively good understanding of a particular topic. The system identifies students who often provide answers on a given topic and encourages classmates to invite mentors to participate in related discussions. Both tools have been integrated into a live discussion board that is used by an undergraduate computer science course. This paper describes our approaches to applying information retrieval and natural language processing techniques in the development of the tools and presents initial results from software instrumentation and student surveys.

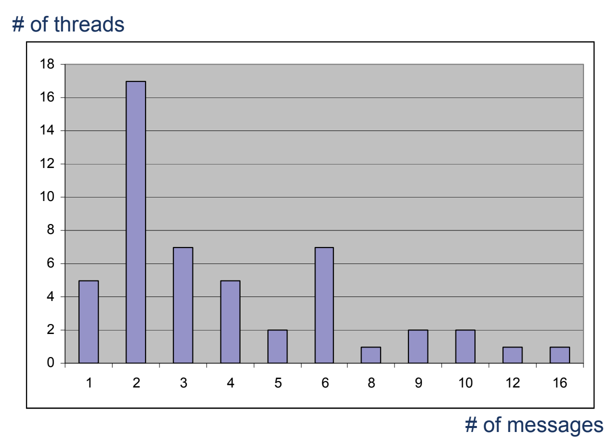
Introduction

On-line discussion boards play an important role in distance education and web enhanced courses. Studies have shown on-line discussion to be a promising strategy for promoting collaborative problem solving and discovery-oriented activities. However, existing systems for on-line discussion may not always be fully effective in promoting learning in undergraduate courses; for example, some analysis of collaborative on-line learning indicate that student participation is low or weak, even when students are encouraged to participate (Kim & Beal, 2006). Discussion threads are often very short, many consisting of only one or two messages.

This was the case for two comparable Computer Science courses that we studied. In the undergraduate courses, a distribution of discussion thread lengths, defined as number of messages per thread, is shown in Figure 1(a). In over 1000 undergraduate student discussions, most threads include only one or two messages. It is clearly the case that students do not fully exploit the collaborative problem-



(a) Messages per discussion thread in undergraduate CS course.



(b) Messages per discussion thread in graduate CS courses.

Figure 1. Discussion thread lengths in comparable undergraduate and graduate computer science courses.

solving environment and miss opportunities for deeper discussions on relevant technical issues. In contrast graduate student discussions in a comparable computer science course tend to be longer, with more messages per thread in general, as shown in Figure 1(b).

Our goal is to develop instructional tools that will help students use discussion boards more effectively. Our work takes place in the context of an undergraduate course discussion board that is an integral component of an Operating Systems course in the Computer Science department at the University of Southern California. Our students use phpBB (phpbb.com), an open source bulletin board that we have extended. The project explores opportunities for integrating AI into the environment for the purposes of learning and assessment. In this paper we describe two extensions that have been recently introduced. Pedabot connects current discussions to related past discussions and course materials, and MentorMatch connects current discussions about particular topics to student mentors based on their roles in prior discussions.

PedaBot: Scaffolding On-line Discussions with Past Discussions

PedaBot is an application for scaffolding student discussions with information from past student discussions, from the same or related courses (Kim et al., 2008). The system dynamically processes student messages or message threads, mines a corpus of relevant past discussions using information retrieval techniques, and presents the retrieved information. PedaBot was designed to aid student knowledge acquisition, promote reflection about course topics and encourage student participation in discussions. It *scaffolds* discussions in the sense that it provides different perspectives on the current discussion topic. Our hypothesis is that free-form peer-to-peer and instructor-to-peer discussions on a current topic of inquiry are inherently interesting, and that by browsing the resulting discussions and documents, students will deepen their understanding of the issues they are currently raising. In addition, we assume that responses from instructors that answered related questions in past semesters will be of particular interest to students.

Figure 2 shows a discussion thread about "terminating a function" from the Fall 2007 study. PedaBot filters and submits the current thread to the retrieval pipeline when it is posted, and responds by displaying portions of several messages that best match the student's question in the left frame. When an instructor's post or discussion in which an instructor participated is retrieved, it is highlighted. The details of the process are described below. Presenting the results in a frame on the left was thought to be unobtrusive yet still obvious and accessible. The results are updated as more messages are posted, to further refine the results.

The resulting messages are usually part of a longer thread that can be viewed by following the 'View whole discussion' link. The results may also originate from a document; in this case, a portion of the matching document is displayed. Students can rate the relevancy of the results to the current discussion and view peer ratings.

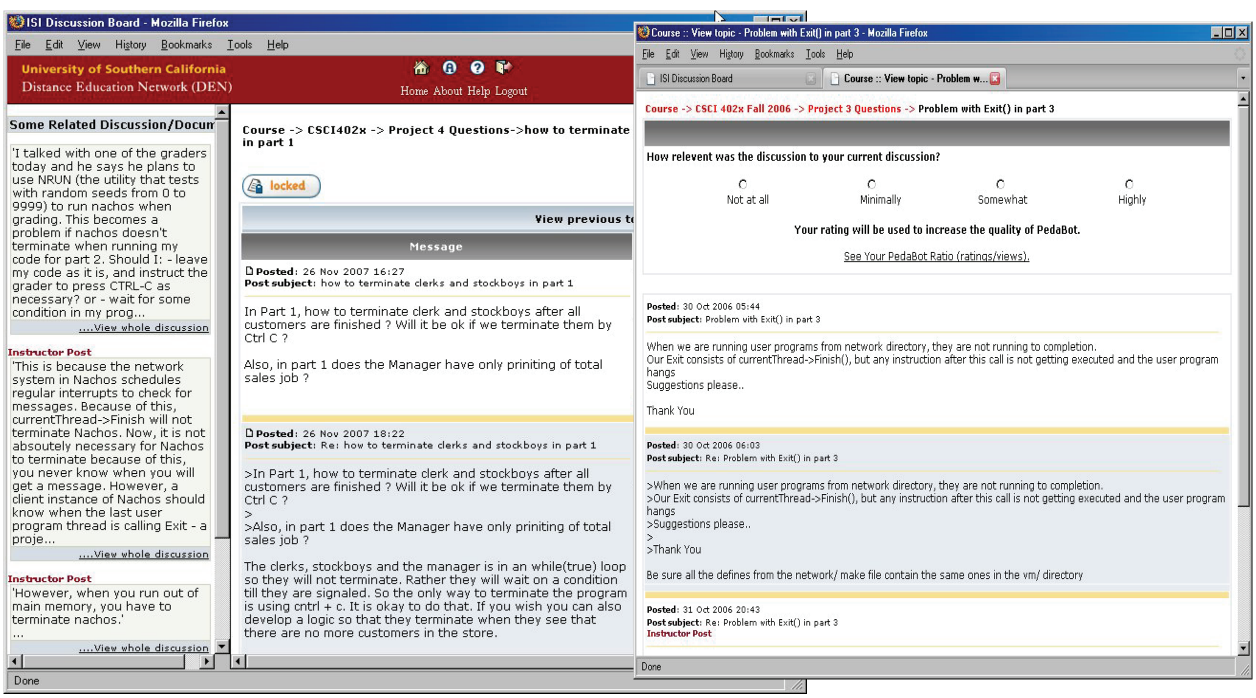


Figure 2. Relevant messages from past discussions are displayed to left. The "View whole discussion" link displays the relevant message's thread. Students can rate its relevancy. The "See your PedaBot Ratio" link displays a table of user ratings.

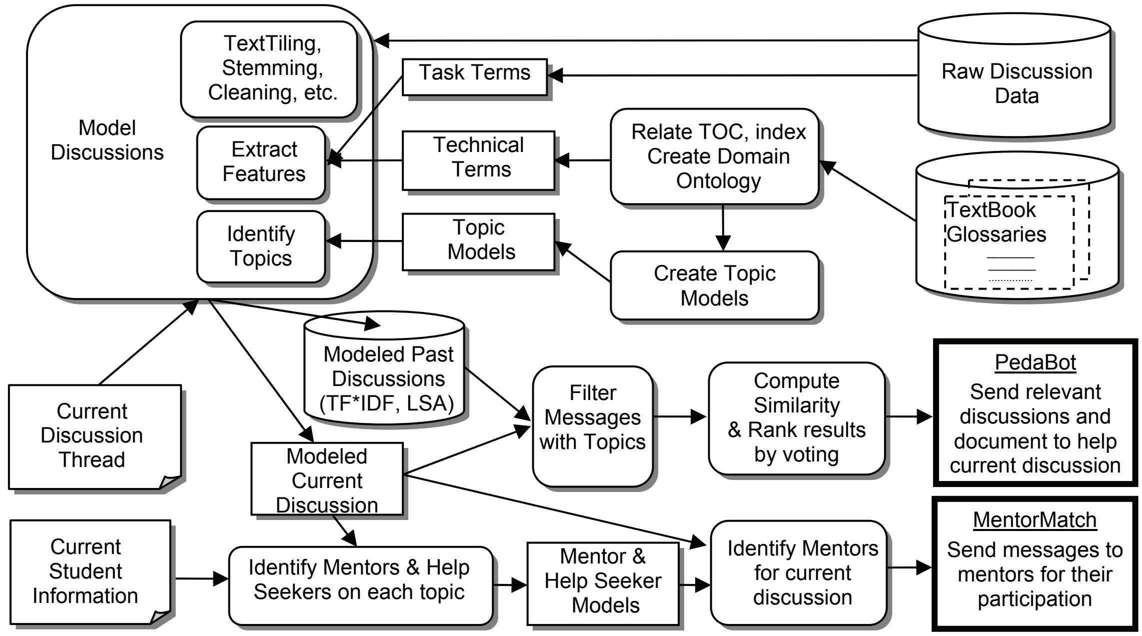


Figure 3. Data Processing Steps for PedaBot and MentorMatch.

Steps for Discussion Data Processing

Figure 3 gives an overview of the current data processing steps. The discussion corpus comprises seven semesters of discussions from the same undergraduate course, two semesters of from a related graduate course, and segments of text from related course documents. Messages from administrative forums were excluded. The total number of messages in the current corpus is 6,622.

Student messages are very informal: They are incoherent with respect to grammatical structure and noisy with respect to individual words, phrases and punctuation. There is high variance in the way students present similar information. Messages include humor and personal statements as well as technical questions and answers; and discussions about programming assignments often include snippets of code. To help model the messages, we used technical domain terms and terms related to student tasks within the course such as 'assignment' and 'project'. Typical document processing steps such as stemming and filtering are also performed at the start of the process.

Existing unigram-based models did not work well since multi-word domain terms and acronyms are dominant in student discussions (e.g. virtual memory, RPC). To build a more effective model, we improved on our original approach to semi-automatically generate a domain term dictionary with flexible multi-word term mappings and acronym mappings. The term features are extracted with the established mappings. Individual messages are modeled with a term vector of the following form:

$$M_i =< T_{i1}, T_{i2}, ..., T_{iN} > ,$$

where N is the total number of technical and task terms in the domain and $T_{ij} = 0$ if a term is missing in that message.

When a query is posted, PedaBot extracts features from the post first, e.g., the technical words and word frequencies. Following that, the system tries to match the student's interest in all archived data, both course documents and past discussions. For retrieval, we explored several combinations of topic models, TF*IDF (Term Frequency * Inverse Document Frequency) and LSA (Latent Semantic Analysis) transformations (Salton 1989, Landauer and Dumais 1997).

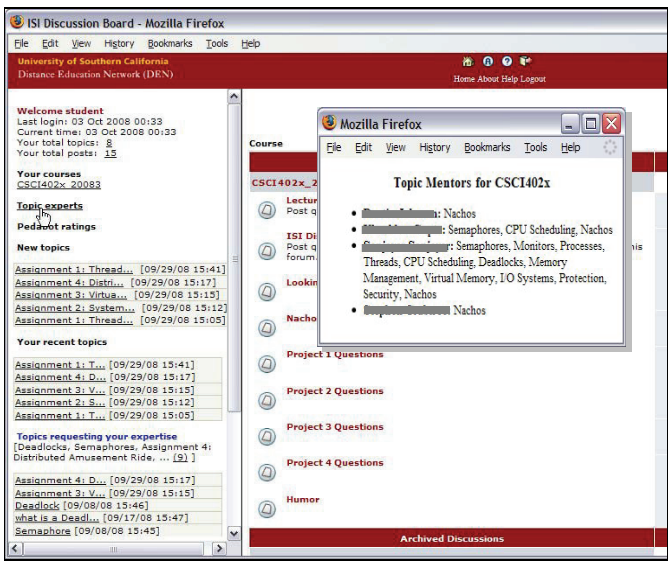
In inducing models of course topics, we rely on the technical terms described above. The system produces term weight vectors for individual topic categories. The details of the topic models are described below. When a new message is submitted to the retrieval pipeline, the system classifies the message with the topic models. The messages that do not share the topics with the new message are removed from the candidate set.

Term weights are used in calculating similarity scores between messages, such as the cosine similarity of a new message and a past message. To compute term weights we use TF*IDF, which is one of the most common ways to model term weights. LSA transforms the occurrence matrix into a relation between the terms and some *concepts*, and a relation between those concepts and the messages. We expected that LSA might reduce noise in the semantic space. We explored several different dimension settings, and applied two separate settings, (k=300 and k=75), which are commonly used in LSA applications.

Since a previous analysis with a smaller set of data showed that LSA alone did not produce better results, we explored three alternatives: TF*IDF only, TF*IDF with the topic models and a combination of TF*IDF and LSA with the topic models. Our analysis of the degree of relevance to the given student message shows that TF*IDF alone rates the best message slightly higher than other combinations (Kim et al., 2008). Unlike our expectation, the LSA options do not seem to effectively exploit concept relevancy, among the technical terms used in student discussions. Based on these results, we currently use TF*IDF.

MentorMatch: Using student mentors to scaffold participation and learning

MentorMatch scaffolds student participation and learning within discussion forums using student mentors, i.e., course peers with relatively good understanding of a particular domain topic (Shaw et al., 2009). First, we identify mentors using domain topic models, student discussion profiles, and a similarity of the topics being discussed. Second, we provide an interface that encourages classmates to invite topic mentors to participate. MentorMatch was deployed in the 2008 Fall semester.



(a). The 'Topic experts' link opens a window with mentor names and topics, inset at center. 'Topics requesting your expertise' displays links to discussions on topics of the student's expertise.

(b) Students may select a mentor to request assistance

Figure 4. MentorMatch Interface

Figure 4 shows the interface with the MentorMatch extensions. Personal information, displayed to the left, includes links to current and archived discussions, a list of new posts and a list of their own posts. To create an awareness of the new mentoring feature and encourage its acceptance as an integral part of the board, we added two new items to the panel. One is a small reference link to "Topic experts" that opens a popup window that displays the topic mentors and topics for the course (insert). The list is updated dynamically and available to all students. The other item is a list of links to current discussions on topics for which the student is a mentor, and a list of mentoring topics attributed to the student. Mentors who receive a request for help can find the referenced discussions here.

When a help-seeker initiates a new thread and posts a question, MentorMatch identifies the topics of the question based on its topic models and searches for potential mentors by matching identified topics and individual student discussion profiles. Mentors are listed as contacts at the top of the thread and the help-seeker is given the option to contact the mentors personally or automatically as shown in Figure 4(b). In either case, email is sent to the mentors inviting them to participate in the discussion.

Inducing topic models from course materials

Table 1 shows several topic categories and terms that are extracted from the Operating Systems course lecture schedule and modeled by the system.

Topic category	Terms with high weights
Semaphores	*semaphore, mutex, register, critical section,*
Monitors	*transaction, monitor, monitor lock*
Processes	*process identifier, parent, child, pcb*
Threads	*thread, kernel thread, many-to-one, signal*
Deadlocks	*deadlock, resource, avoidance, graph,*
:	…

Table 1: Example topic categories for discussion modelling

Supervised machine learning approaches to topic classification typically require a set of manually labeled data. Because student discussions are very informal, it is difficult to generate consistent labels due to high variance and noise. Messages also often contain terms relevant to multiple course topics. In place of labeled messages, we use a corpus of documents that is already indexed by related domain topics. The corpus consists of course lecture and assignment materials from both our own course and from similar courses whose materials were available online. Each topic is modeled with a representative vector using the term ontology (Feng et al., 2006b).

Student profiling with topic models

Student profiles were created to accumulate information about student activities. We track the topics in which the student participates and the types of post (Q/A) made.

Each student's message is classified using topic vectors, and the topic similarity scores are stored in the profile.

Messages are also classified based on positions in the thread, i.e. whether it is the first post or a response. More than 80% of student discussion threads start with a message containing a question (Feng et al., 2006a). This information is used to differentiate help-seekers from answer-providers. Yes/no responses and short acknowledgements will not have high similarity scores for relevant topic vectors. To incorporate these contributions into the profile, the similarity score for response message M_i and topic category T_c is computed as in the following:

$$Score\ (M_i, Tc) = w * Sim\ (M_i, Tc) + \sum_{k=1}^{i-1} w_k * Sim\ (M_k, Tc)$$

That is, the score of response message M_i includes the weighted similarity scores of previous messages, including the first message, i.e., $M_1,\ldots,M_{i-1}$. Our current implementation uses a uniform distribution over previous and current messages. The student contribution scores for all topics are accumulated over time and used to identify mentors who provide the highest number of responses on the topic. For each topic, the top three students within a given threshold are identified as mentors for the topic.

As a validation study, we analyzed whether mentors are more likely participate in a discussion. We used discussions from data from a previous semester, Spring 2007, for the analysis. For each thread, we compared the mentors identified for the first message with the actual discussion participants. We found that in 42/51 threads, at least one mentor participated.

Pilot Studies

PedaBot results

PedaBot was integrated into a live Operating Systems course discussion board in fall 2007. We show data collected from two semesters, fall 2007 and fall 2008. Among 147 discussion participants, 114 students used the feature. The usage results are shown in Table 2. Although students showed interest in PedaBot, many were unaware of the feature and its usage frequency was not high. Among students who rated the results, the average ratings are 2.46 for 2007 and 3.13 for 2008, on a scale of 4-1 (highly, somewhat, minimally, or not at all), which are consistent with our prior analysis that PedaBot retrieves moderately relevant messages. The sight increase in the average rating may be due to improvement of technical term modeling. The frequency of PedaBot use is still low and we plan to investigate strategies for promoting its use.

In Table 3, we analyze PedaBot's effect on the average number of messages in a thread in 2007. Since PedaBot was introduced mid-semester and we could compare discussions with and without it. We hoped that having PedaBot would encourage discussion and increase the length of the discussion. The number of messages per thread is a little higher with PedaBot, especially for female students (5.41 vs 2.0).

Table 2. Discussion board and PedaBot usage.

Number of students who…	2007 fall	2008 fall
Registered on discussion board	119	127
Participated in discussions	91	56
Initiated discussion threads	78	47
Viewed entire discussion context of PedaBot retrieved mgs	62	52
Rated PedaBot retrieved mgs	15	8
Average number of …		
Messages posted (#messages / #discussants)	600/91 = 6.59	543/56= 9.70
Threads participated in (#threads / #discussants)	311/91 = 3.42	231/56= 4.13
Average number of PedaBot ...		
Discussion details viewed (#viewings / #viewers)	371/62 =5.98	149/52= 2.86
Results rated (#ratings / #raters)	39/15 =2.6	8/6=1.33
Avg. rating	2.46	3.13

Table 3. Difference in thread length with and without PedaBot

Fall 2007		**with** PedaBot	**w/o** PedaBot
Average # of messages per thread	Male	426/124= 3.43	533/169 = 3.15
	Female	65/12 = 5.41	12/6 = 2.0
	All	431/127 = 3.39	543/174 = 3.12

A questionnaire was used for collecting student ratings and comments. Students found the feature relevant and somewhat useful (Kim et al., 2008).

MentorMatch results

The Mentor Match feature was deployed in October, 2008. During this time, students could elect to send a request for assistance to a mentor. Several email messages were sent. On October 29, we sent an email to all identified mentors introducing them as mentors. (In the beginning students did not understand the word 'mentor'.) The message said that due to the limited time available, unless they objected, instead of being contacted by classmates, they would be automatically sent up to five requests for assistance a week, if appropriate (i.e., if students were matched as mentors to a new discussion). Our goal was to evaluate the effect of mentor participation, especially to see if more questions would be answered by better informed students, which we hoped would happen, but also to see how a reply would affect the discussion overall.

To gauge the effect of notification, we started with the number of topics about which mentors were notified and compared the number of subsequent topic-related replies to non-topic-related replies. The comparison is shown in Figure 5. As the number of topic notifications increase, student replies tend to increase overall, however, there is more increase in non-topic related discussions.

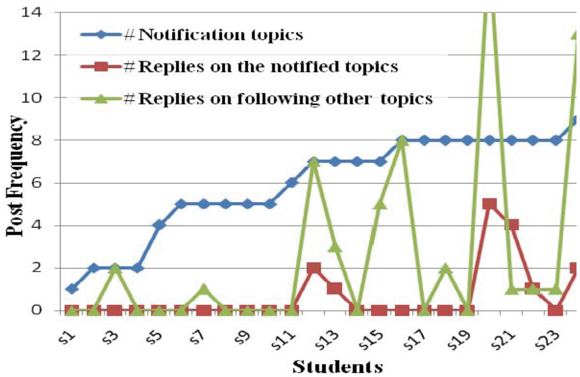


Figure 5. #Topics vs. #Replies: Topic-related replies vs. non-topic related replies after notification.

To promote use and awareness of the feature, we added two menu links: The ‘Topic mentors’ link and the ‘Topics requesting you help’ link that were described earlier. We surveyed the students to determine if they saw and/or activated the new menu links. The results are shown in Table 4. Only 52% (13/25) of the responding students opened the mentor list or noticed the menu section. Of those that noticed the links, i.e., the topic mentors, 69% (9/13) reportedly clicked through to a discussion. Students were also asked about the notification feature above the discussion thread that listed the mentors’ names and gave students the option of contacting them personally.

Table 4. Survey results. N is the eligible sample size for that question.

Number of students who…	#	N
Reported activating the ‘Topic mentors’ link	13	25
Commented that they were unaware of the link (Others said it wasn’t needed or they did not think it would help)	7	09
Noticed the ‘Topics requesting your help’ section	13	25
Reported activating a ‘Topics requesting yourexpertise’	9	13
Reported sending email to a mentor	6	25
Did not send email because didn’t notice link	6	11
Did not send because didn’t think it necessary or assumed someone else would respond	2	11
Reported receiving a request for their assistance	9	25
Reported receiving a request who responded or tried to	5	09

Students who were aware of or used the feature, were asked to rate their interest in and usefulness of the feature on a Likert scale (low=1, high=5, N=20). The average ratings are shown in Table 5. Four out of six users provided positive comments, e.g.,

- “It helped find people who could directly assist with a topic and mail them if the discussion forum didn't get a response.”
- “It was easier to communicate with the mentors and moreover easier to decide whose suggestion to take into account.”

One of the others said it could be made more useful and one said that it was introduced too late to be useful. The feature was introduced 10 weeks into a 15 week semester. We also think the timing contributed to the fact that students didn’t notice the new menu links, which blended in to the look and feel of the menu column.

Table 5. Likert scale results on interest and usefulness (N= 20).

Feature	avg	not	low	n/a	Some	high
How interesting	4.2	0	1	3	7	9
How useful	4.05	0	2	3	7	8

Related Work

Our research draws from results in the fields of topic classification, discussion mining, and collaborative learning, yet is novel in its application. There has been a lot of prior work on dialogue analysis, including tutorial dialogue (e.g. Graesser et al., 2001, Tetreault et al., 2008). Although some of the techniques are closely related, most of them focus on spoken dialogue or conversation in tutoring systems rather than threaded discussions. Other researchers worked on qualitative assessment of discussions including student reasoning (McLaren et al., 2007). Most of these systems do not provide tools for promoting student interactions. There has been increasing interest in online dialogue including email message analysis (Lampert et al. 2008). However, student discussions tend to focus on problem solving rather than task request and commitment as in email message threads in project management applications. We are investigating opportunities for complementing capabilities. Handling noisy data is a challenging task in many information retrieval applications (Knoblock et al., 2007). For high incoherence and noise in informal student discussions, we incorporate several new features including features from neighboring messages and technical term features.

System Deployment: Lessons Learned

Deployment within university

Our discussion board is accessed by students via single signon and authentication from the university’s Blackboard course management system. The students link to the board from a normal menu link that calls a web service which retrieves the username from Blackboard’s user context. Together with the course ID obtained from the menu link, the service calls a registration function on our system with information about the student requesting the new session. New students are registered on the board and valid students are confirmed. The web service sends then redirects the student to a PHPBB system gateway with authenticating information, allowing login for registered students. Once registered, students use an external password to login directly via a URL on the discussion board server.

Architecture rationale

The architecture addresses instructional challenges. First, students expect to use new communication technologies, but purchasing (university) and utilization (instructor) of always lags behind what is current. Secondly, instructors who prefer more control often have limited support. They may find it easy to maintain their own web sites but not their own discussion board. Registration, security, and maintenance are all obstacles. The architecture also addresses several technical challenges. Building our own online learning environment, i.e. duplicating the university system, is impractical. We anticipated using Blackboard's *Building Blocks* as a solution, but its API was incomplete and its integration with Blackboard was too tight: Remote development at our institute, but within a university production environment, would have been problematic.

The seamless integration maximizes ease-of-use, encourages adoption of the technology for teachers and enables development at our institute that minimizes disruptions on the production side. It allows for evolution from Blackboard, if this occurs. Finally, the open source development model maximizes code maintenance for us.

Research platform

Prior to the development of PedaBot and MentorMatch, the board was used to experiment with discussion summary, document annotation, and instant messaging tools. The board has been used by several Ph.D. students to run experiments. We coordinate these studies with the instructor before starting, and assist with the development and administration of the surveys.

Students are notified that this is a research board by the instructor when they first log in. A flash screen makes it clear that use of the new features is strictly voluntary and their use is not a course requirement.

Conclusion and Future Work

The results from our pilot study show that although many students were not fully aware of the new feature and its usage frequency was moderate, students rated the feature highly interesting. The students rated PedaBot's results as moderately relevant to their discussions. For MentorMatch, as the number of topic notifications increased student replies tended to increase overall, although within the period that we studied, there is more increase in non-topic related discussions. We plan to develop strategies for promoting the scaffolding features, including interface improvement. We will be working on the display of student and thread profiles for instructor assessment.

Optimizing models of student messages, and handling noise and variance, has been a continuous process. A deeper assessment of user activity patterns may prove useful in improving the accuracy of the models. For example, an analysis of student participation for different topics may provide hints on related topics. Combining other models may also increase the accuracy of the models. For example, speech act classification can identify whether a message contains questions or answers (Kim and Ravi 2007). This knowledge could be used to select information to send students (e.g. answers instead of additional questions), and to help profile student activities.

Acknowledgement

The work was supported by National Science Foundation, a CCLI-Phase 2 (Expansion) grant award No. 0618859.

References

Feng, D., Shaw, E., Kim, J., Hovy E. 2006a. An Intelligent DiscussionBot for Answering Student Queries. *Proceedings of IUI-06.*

Feng, D., Kim, J., Shaw, E., Hovy E. 2006b.Towards ModelingThreaded Discussions through Ontology-based Analysis. *Proceedings of AAAI-06.*

Graesser, A., VanLehn, K., Rosé, C., Jordan, P., Harter, D. 2001. Intelligent Tutoring Systems with Conversational Dialogue. *AI Magazine* 22(4):

Kim, J., Shaw, E., Ravi, S., Tavano, E., Arromratana, A. & Sarda, P., 2008. Scaffolding of On-line Discussions with Past Discussions: An Analysis and Pilot Study of PedaBot, *Proceedings of ITS-2008*.

Kim, J., Beal, C. 2006. Turning quantity into quality: Supporting automatic assessment of on-line discussion contributions. *AERA 2006*

Knoblock C., Lopresti, D., Roy, S., Subramaniam, L. 2007. Special issue on noisy text analytics. *IJDAR* 10(3-4).

Lampert, A., Dale, R., and Paris, C. 2008. The Nature of Requests and Commitments in Email Messages, *AAAI workshop on Enhanced Messaging*.

Landauer, T., Dumais, S. 1997. A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge, *Psychological Review*, 104 (2).

McLaren, B., Scheuer, O., De Laat, M., Hever, R., De Groot, R. & Rose, C. 2007. Using Machine Learning Techniques to Analyze and Support Mediation of Student E-Discussions, *Proceedings of AIED-2007*.

Ravi, S., Kim, J.: Profiling Student Interactions in Threaded Discussions with Speech Act Classifiers, *Proceedings of of AIED-2007.*

Salton, G. 1989. Automatic Text Processing, The Transformation, Analysis, and Retrieval of Information by Computer.

Saumel, K. 2000. An Investigation of Dialogue Act Tagging using Transformation-Based Learning, PhD Thesis, University of Delaware.

Shaw, E., Kim J., & Supanakoon, P. 2009. Mentor Match: Using student mentors to scaffold participation and learning within an online discussion board, *Proceedings of AIED 2009*.

Tetreault, J. Burstein, J. and De Felice, R. 2008. *Proceedings of the ACL workshop on Innovative Use of NLP for Building Educational Applications*.