

Towards automatic scaffolding of on-line discussions in engineering courses

1. Introduction

On-line discussion boards play an important role in distance education and web enhanced courses. Recent studies have pointed to on-line discussion board as a promising strategy for promoting collaborative problem solving courses and discovery-oriented activities [Scardamalia & Bereiter, 1996]. However, other research indicates that existing systems for on-line discussion may not always be fully effective in promoting learning in undergraduate courses. For example, some analyses of collaborative on-line learning indicate that student participation is low or weak, even when students are encouraged to participate [Palloff & Pratt 1999; Kim & Beal, 2006]. As course enrollments increase, with some introductory courses enrolling several hundred students, the heavier on-line interaction can place a considerable burden on instructors and teaching assistants. Discussion threads are often very short, many consisting of only one or two messages, and students do not fully exploit the collaborative problem solving environment where they could discuss relevant technical issues with one another. We are developing instructional software tools that can automatically assess student participation and promote interactions by sending responses to student messages.

In this paper, we present a novel software tool that applies data mining and information retrieval techniques for guiding student discussions. Given a discussion thread, the tool retrieves useful information from past student discussions in related courses and automatically presents the retrieved information on the discussion board. Our hypothesis is that we can scaffold discussions by sending messages from past students who had similar assignments or problems regarding course topics. We expect when a discussion thread is short or stagnant, such intervention may promote further participation from students.

We first semi-automatically extract domain terms from textbooks specified for the courses and task related terms from discussion corpus, and use them in modelling individual messages with term vectors. We apply LSA (Latent Semantic Analysis) [Landauer and Dumais, 1997] and TF*IDF (term frequency and inverse document frequency) [Salton, 1989] in finding useful information from past student discussions. The tool exploits the discussions from the same undergraduate course in past eight semesters and related graduate courses in two semesters. We performed an initial analysis of retrieved messages. The analysis focused on the degree of relevancy to the current student question. The preliminary results indicate that the tool can retrieve moderately relevant information from past discussions.

2. Approach to retrieving relevant messages in past discussions

The course we are currently focusing on is an undergraduate Operating Systems course at the University of Southern California. The course is held every semester, and the instructor has been using our discussion board software for the last seven semesters. The new scaffolding tool is called PedaBot (Pedagogical discussion assistant). PedaBot will be used in the Operating Systems course from the Fall 2007 semester.

Figure 1 shows a screenshot from the current system. The right hand side window shows a question posted by a student and the left panel shows three relevant messages retrieved by the system. The student can see the whole discussion thread where individual messages are retrieved from by clicking "View whole discussion".

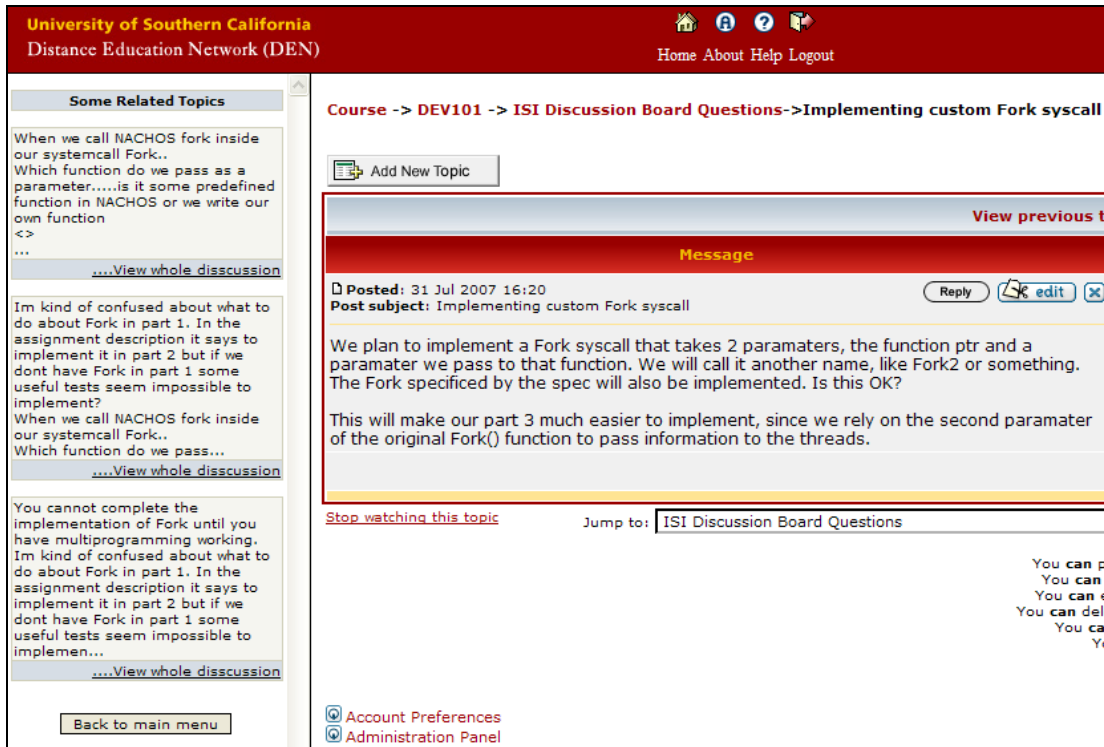


Figure 1. PedaBot: Retrieving relevant messages from past discussions

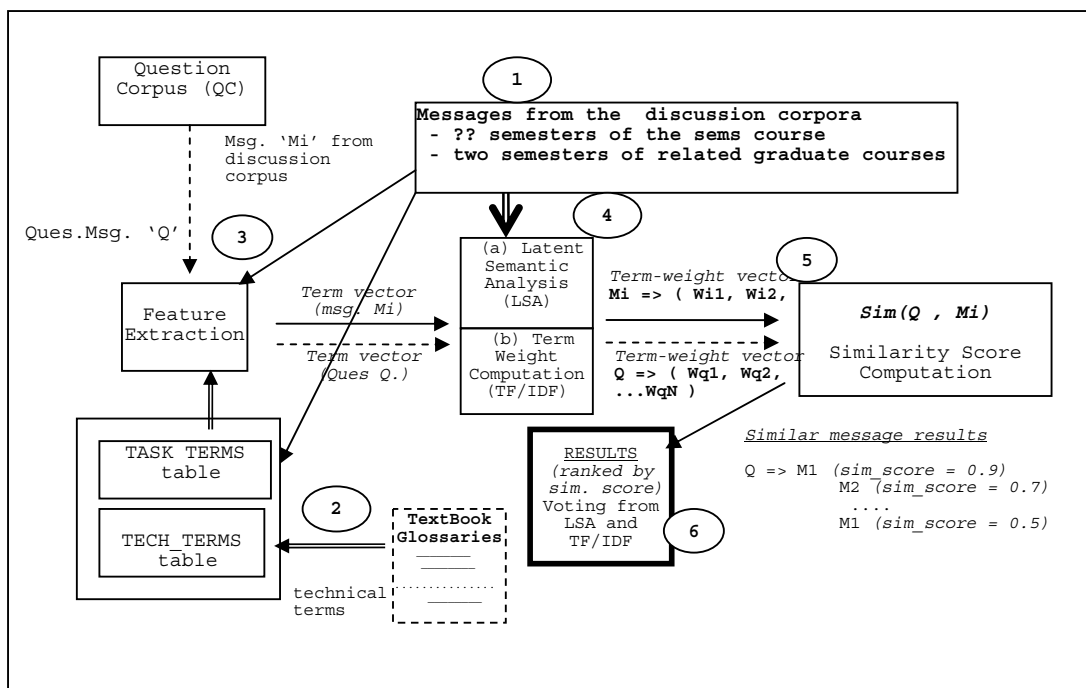


Figure 2. Steps involved in retrieving relevant messages

We use discussion corpora from two Operating Systems courses: an undergraduate level course and a graduate-level course. The undergraduate course is held every semester, and

students and instructors have access to discussion boards to exchange information regarding various topics covered in the course.

2.1 Representing Messages with Domain Terms

We first removed messages from administrative forums that contain non-technical discussions. The undergraduate discussion corpus was created with data from eight semesters, and the graduate discussion corpus was obtained from two semesters. The total number of messages in the corpus was 14434. Step 1 in Figure 2 represents this process.

We found that discussion data from students, especially undergraduate students, are highly incoherent and noisy. The raw data includes humorous messages and personal announcement as well as technical questions and answers. Student messages are very informal and there are high variances in the way they present similar information. A lot of messages on programming assignments also include programming code. Due to the noise and the highly informal nature of messages posted by students in the discussion forums, in modeling the messages for scaffolding, we use technical terms and the terms related to student tasks during the course. The system also applies typical document processing steps including stemming and cleaning [Ravi and Kim, 2007].

The technical terms that we use were extracted from the glossaries of the undergraduate and graduate text books. The glossaries were automatically scanned and processed. We created a “TECH_TERMS” table that contains all the technical terms (2233 terms) from both courses. We also manually selected additional 1587 terms that are related with student tasks by examining frequent terms in discussions (TASK_TERMS). This corresponds to step 2 in Figure 2.

Individual messages are then modeled with a term vector of the following form (Step 3 in Figure 2):

$$M_i = \langle T_{i1}, T_{i2}, \dots, T_{iN} \rangle$$

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

2.2 Modeling Messages for Retrieval

This section describes step 4 in Figure 2 where messages in past discussions are modeled with TF*IDF and LSA.

TF*IDF

After term vectors are created for all messages in the corpus, the weights for the terms are computed. The weights are used in calculating similarity scores between messages, such as similarity of a new message and a message in a corpus.

In computing term weights, we use TF*IDF (term frequency * inverse document frequency) [Salton, 1989]. TF*IDF is one of the most common ways to model term weights. Messages with same technical terms are more likely semantically related. This information is captured in TF (term frequency). TF tends to weight the commonly occurring terms more and give low weights to rare technical terms. IDF (inverse document frequency) fixes it by introducing general importance of the term. Equation 1 describes the method to compute individual TF*IDF weight values for each term.

Equation 1. TF-IDF feature weight computation

$$W_{ik} = TF_{ik} * \log(N / nk)$$

T_k = term k in document (message) M_i
 TF_{ik} = frequency of term T_k in document (message) M_i
 IDF_k = inverse document frequency of term T_k in C
 N = total number of documents (messages) in the discussion corpus C
 nk = the number of messages in C that contain T_k
 $IDF_k = \log(N / nk)$
 W_{ik} = TF-IDF weight for term k in document (message) M_i

The term vector for each message is converted into a corresponding term-weight vector, where each term-weight represents the TF*IDF measure for a particular technical term existing in the message.

LSA

LSA (Latent Semantic Analysis) transforms the occurrence matrix into a relation between the terms and some *concepts*, and a relation between those concepts and the messages. Thus the terms and documents are now indirectly related through the concepts [Landauer and Dumais, 1997]. LSA has been used in various educational applications including dialogue support for intelligent tutoring systems and essay grading. We used two different dimension settings, 300 and 75, that are often used in various LSA applications.

2.3 Retrieving Relevant Messages by Combining the Results with TF*IDF and LSA

Equation 2. Cosine similarity computation (between message/document D_i and question Q)

$$sim(Q, D_i) = \frac{\sum_{j=1}^t w_{q_j} w_{d_{ij}}}{\sqrt{\sum_{j=1}^t (w_{q_j})^2 \sum_{j=1}^t (w_{d_{ij}})^2}}$$

$D_i = (W_{di1}, W_{di2}, \dots, W_{dit})$
 Feature-weight vector representing document/message D_i in the discussion corpus.

 $Q = (W_{q1}, W_{q2}, \dots, W_{qt})$
 Feature-weight vector representing question message Q .

 W_{dit} = TF-IDF or LSA weight for feature term t in message D_i .
 W_{qt} = TF-IDF or LSA weight for feature term t in question message Q .

Given a new message posted on the discussion board, a new term vector is created with term weights with TF/IDF. We use cosine similarity to determine relevance between the new message and a message in the corpus (Equation 2). This measures the cosine of the angle between the two term-weight vectors representing the two messages. Messages are ranked by their scores in order to find the most relevant messages. Similarly, for LSA, we calculate the cosine similarity of the vector with each document's "concept" vector that was generated by LSA.

Once all the similarity scores are computed, we select the messages with the similarity score higher than 0.5. We then combine the results from TF*IDF and LSA options based on average rankings of the retrieved messages. The top three messages are sent as related messages (Step 6 in Figure 2).

3. Preliminary analysis of PedaBot message relevancy

We have performed an initial analysis of the messages retrieved by PedaBot. Our analysis has focused on the degree of relatedness to the given student question. We created a new message corpus with a separate discussion data from a recent semester, Fall 2006. We extracted the first messages in all the threads in the new corpus. We randomly picked 20 messages and used them in triggering the above procedure and retrieving relevant messages from the discussion corpus. Two human evaluators rated the system responses in terms of degree of relevancy with 1 through 5 Likert scale. The kappa value for the best messages is 0.61. Table 1 shows the average relevancy of top three responses retrieved by PedaBot. The preliminary results indicate that the tool can retrieve moderately relevant information from past discussions. Table 2 shows MRR (mean reciprocal rank) for the best message selected by the evaluators. The current results show that TF*IDF rates the best message higher than the LSA options. It seems that the current LSA options do not fully exploit concept relevancy among the technical terms. We are investigating different settings for LSA.

Table 1: Average relevancy ratings for retrieved messages

M1	M2	M3
3.2	3.875	3.175

Table 2: MRRs and average rankings for LSA and TF*IDF

	MRR for the best message
LSA 300	0.58
LSA 75	0.46
TF*IDF	0.70

4. Related Work

In the area of online learning, much attention has been paid to the analysis of student learning behaviors in online communications. Various frameworks have been proposed for analyzing and guiding students in computer mediated communication. They include dialogue based tutoring software [Graesser et al., 2001; Rose et al., 2001], collaborative discussions [Shaw, 2005], email and chat exchanges [Cakir et al., 2005], knowledge sharing [Soller & Lesgold, 2003] and general argumentation [Feng et al., 2006, Painter et al., 2003], but none have been sufficiently relevant or fine-grained to facilitate data mining and scaffolding for student discussions.

5. Discussion and future work

Depending on the situation at hand, different tutorial scaffolding strategies can be used such as generating questions and comments, inviting participation and clarification, and referring students to past discussions of a related topic. In order to improve the usefulness of the scaffolding capability, we are investigating various ways to characterize student messages and discussion threads. For example our speech act classifiers can identify whether a message contains questions or answers [Ravi and Kim, 2007]. We are also developing a monitoring capability so that PedaBot can send messages based on the focus of the discussion and only when a discussion thread is short or stagnant.

References

- Cakir, M., Xhafa, F., Zhou, N., and Stahl, G. (2005). Thread-based analysis of patterns of collaborative interaction in chat, *Proc. of AI in Education*.
- Feng, D., Kim, J., Shaw, E., and Hovy E. (2006b), Learning to Detect Conversation Focus of Threaded Discussions. *Proc. of the Joint HLT Conf./Annual Mtg of the NA Chap of the Assoc. for Computation Linguistics*.
- Graesser, A.C., Person, N., Harter, D., & TRG (2001). Teaching tactics and dialog in AutoTutor. *International Journal of AI in Education*, 12.
- Kim, J. and Beal, C (2006), Turning quantity into quality: Supporting automatic assessment of on-line discussion contributions, *AERA 2006*.
- Landauer, T. and 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).
- Painter, C., Coffin, C., and Hewings, A. 2003. Impacts of Directed Tutorial Activities in Computer Conferencing: A Case Study. *Distance Education*, 24(2)
- Rose C., Jordan, P., Ringenberg, M., Siler, S., VanLehn, K. Weinstein, A. (2001), Interactive Conceptual Tutoring in Atlas-Andes.
- Ravi, S., Kim, J. (2007), Profiling Student Interactions in Threaded Discussions with Speech Act Classifiers, *Proceedings of AI in Education*.
- Ravi, S., Kim, J. , and Shaw E. (2007), Mining On-line Discussions: Assessing Technical Quality for Student Scaffolding and Classifying Messages for Participation Profiling, *Educational Data Mining workshop in AIED2007*.
- Scardamalia, M., & Bereiter, C. 1996. Computer support for knowledge building communities. In T. Koschmann (Ed.), *CSCL: Theory and practice of an emerging paradigm*. Mahwah NJ: Erlbaum.
- Shaw, E. 2005. Assessing and Scaffolding Collaborative Learning in Online Discussions. *Proceedings of AI in Education*.
- Soller, A., and Lesgold, A. (2003), Computational Approach to Analyzing Online Knowledge Sharing Interaction, *Proceedings of AI in Education*.