An Intelligent Discussion-Bot for Answering Student Queries in Threaded Discussions

Donghui Feng, Erin Shaw, Jihie Kim, Eduard Hovy
Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA, 90292 USA
{donghui, shaw, jihie, hovy}@isi.edu

ABSTRACT
This paper describes a discussion-bot, which provides answers to students' discussion board questions in an unobtrusive and human-like way. Using information retrieval and natural language processing techniques, the discussion-bot identifies the questioner's interest, mines suitable answers from an annotated corpus of 1236 archived threaded discussions and 279 course documents, and generates a human-like reply. A novel modeling approach was designed for the analysis of archived threaded discussions to facilitate answer extraction. We compare a self-out and an all-in evaluation of the mined answers. The results show that the discussion-bot can begin to meet students' learning requests. We discuss directions that might be taken to increase the effectiveness of the question matching and answer extraction algorithms. The research takes place in the context of an undergraduate computer science course.

Categories and Subject Descriptors
H.5.2 [**Information Interfaces and Presentation**]: User Interfaces: *Theory and methods, Natural language.* H.1.2 [**Models and Principles**]: User/Machine Systems: *Human information processing*

General Terms
Algorithms, Design, Experimentation, Human Factors.

Keywords
Natural language processing, discussion-bot, threaded discussion, online learning environment.

1. INTRODUCTION
Online learning is ubiquitous, and intelligent and effective user-friendly computer mediated communication (CMC) interfaces play an increasingly important role in the acquisition of knowledge for learning. Discussion boards, also referred to as bulletin boards and conferencing systems, are integral components of most online learning environments. The Distant Education Network (DEN) at the University of Southern California Viterbi School Of Engineering offers on-campus courses that are broadcast to remote students. Online discussion boards are used in both on-campus and remote courses to facilitate instructor-student and student-student communication. Several DEN courses use an instrumented open source discussion board based on phpBB [10] known as the ISI Discussion Board (ISI DB). The ISI DB, shown in Figure 1, is a platform for evaluating new teaching and learning technologies. We use it here to evaluate new information retrieval (IR) and natural language processing (NLP) techniques.

The research takes place in the context of an undergraduate computer science course on operating systems. Students are encouraged to participate in discussions on theoretical or practical problems during their course studies and generally use the discussion board to seek answers from instructors or peers.

The course is offered every semester and is fairly consistent; the similar issues and projects are discussed each term. Students in the course make heavy use of the discussion board and responses to their questions are often time critical. If many questions are posted in a short time frame, it is unlikely they will be responded to in a timely manner. Additionally the instructor and teaching assistants answer similar questions repeatedly, either per, or across, semesters, and sometimes the answers have already been provided in the supplemental course documents.

It follows that an intelligent agent able to monitor the discussions and answer students' questions would be desirable. The agent would identify students' query interests and reply with answers from archived documents or discussions. While we wish to relieve the instructor's and TAs' burdens, we do not with to turn the discussion board into a search engine, which we believe would discourage interchanges. Thus, it is vital to provide a friendly and effective agent interface.

In this paper, we describe an intelligent discussion-bot that has been implemented within the ISI discussion board, which will automatically answer student questions and reply in an unobtrusive and human-like way. Using information retrieval and natural language processing techniques, the discussion-bot identifies users' interests, mines suitable answers from archived discussions and course documents, and generates a human-like reply. To the best of our knowledge, a novel approach to modeling threaded discussions is used to analyze past discussions to facilitate answer extraction.

IUI'06, January 29–February 1, 2006, Sydney, Australia.

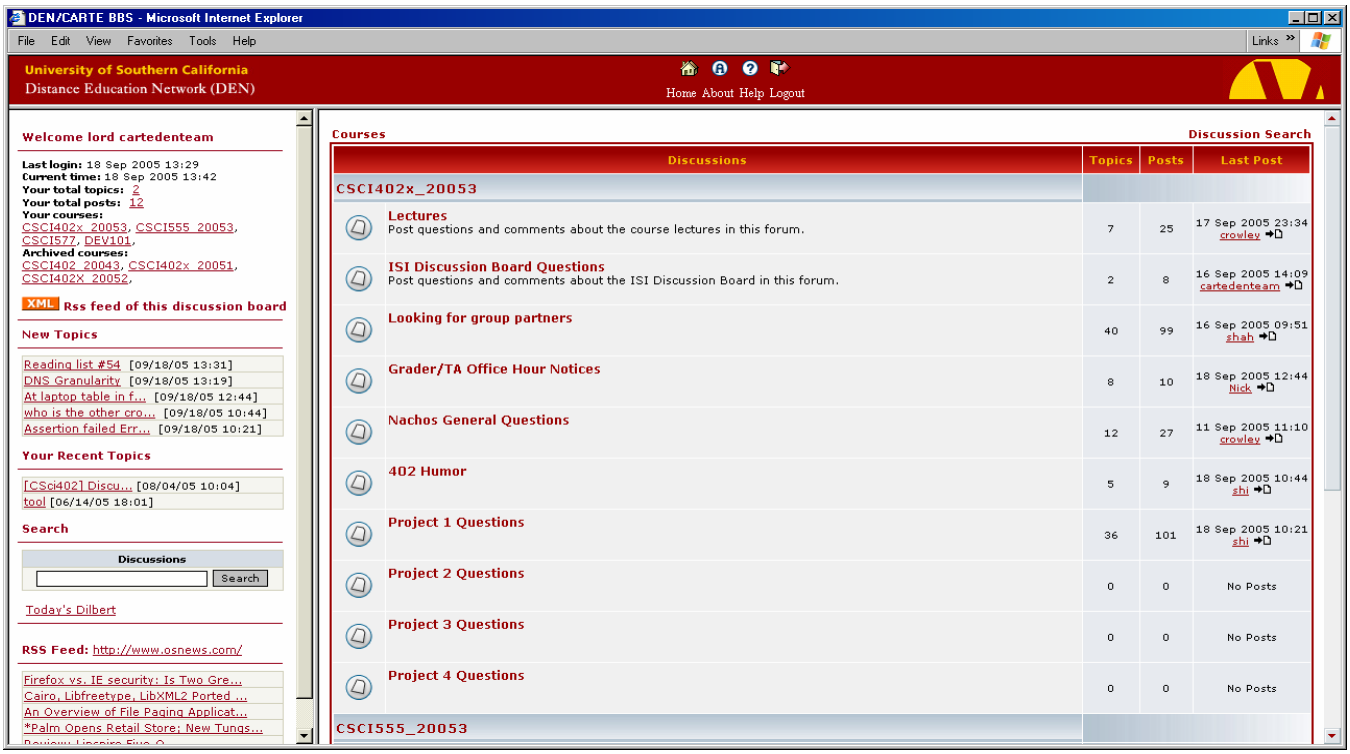

Figure 1. USC/DEN ISI DB online learning environment

The rest of this paper is organized as follows: We describe related work in Section 2 and the analysis and processing of archived data in Section 3. The details of the discussion-bot system are described in Section 4, including the system architecture and the interest-matching and an answer generation algorithm. Experimental results are presented in Section 5. We conclude in Section 6 with discussions on related issues and future work.

2. RELATED WORK

Online learning environments are ubiquitous, and used by both on campus students and remote learners in distance education. This presents challenges for intelligent tutoring system research [3, 15, 16, 17, 18]. Discussion forums are an integral and widely used component of online learning environments. Much attention has been paid to the analysis of student learning behaviors in online communications and chats [e.g. 2, 15].

A large amount of natural language interfaces have been designed for query [e.g. 11, 21]. But our work is dealing a more complex situation, in which the queries cannot be expressed with one sentence. Likewise, automatic question answering in an open-domain is the focus of much research in natural language community. Successful question answering systems tend to have similar underlying pipelines structures [e.g. 5, 6, 7, 9, 12, 20]. The general framework involves parsing questions, searching documents, and pinpointing answers. Most research focuses on factoid questions that can be answered with a short phrase. (e.g. 'Who is the Prime Minister of Australia?'). This makes it feasible to classify the answer type and target during question parsing.

However, our target situations are different: most of the students' questions are complex context questions that often include a lengthy description of the context and procedure in question. This special characteristic makes it almost impossible to represent and identify answer types in students' posts, and also more difficult to computationally discern true questions and answers in student messages.

The artificial intelligence community has developed various types of chat-bot to simulate a human interaction [1]. However, most of them focus on dialogue via single sentence or only one phrase. They generate answers based only on the immediate past input from the user and have limited ability to learn from a corpus. Therefore the dialog is likely to be shallow and discontinuous. Complex or contextual questions are difficult for a typical chat-bot to converse about. Some of the research on dialogue modeling investigates different ways to manage continuous dialogue for personal assistants [e.g. 8], but our work aims only at one-step question-reply in discussion board.

We next discuss the data involved in our task and the technical details of the discussion-bot.

3. THE DATA

For the operating systems course, we had two resources available for data mining suitable answers to student queries, the supplementary course documents and threaded discussions from past semesters. Table 1 gives the total numbers of supplementary documents and posts in the discussions.

Table 1. Numbers of archived documents and posts

Type	Documents	Posts
Numbers	279	3093

3.1 Document Processing

Course documents include reading assignments, homework and solutions, project descriptions, and instructions. Instead of matching a whole document, which is common in regular keyword searches, we aimed only to match and retrieve part of a document. We applied a natural language processing tool, TextTiling [4], to segment every whole document into semantic-related tiles and subsequently processed tile units (versus document units). Table 2 gives the numbers of documents and tiles segmented. The average number of tiles per document is 10.13.

Table 2. Numbers of documents and document tiles

Type	Documents	Document Tiles	Avg. Tiles per Doc
Numbers	279	2826	10.13

3.2 Modeling Threaded Discussion

Archived threaded discussions from previous semesters are also included in the corpus. A threaded discussion includes an initial message together with all messages posted in response to it. All responses are sequentially linked to the original message in the chorological order. Participants can read or respond to any of the messages in a thread. We used the last three semesters of discussions, resulting in 1236 threads.

Table 3. Definitions of post speech acts

Code	Name	Description
QUES	Question	Question on specific problems
COMM	Command	Command or announcement
DESC	Describe	Describe a fact or situation
CANS	Complex Answer	An answer requiring a full description of procedures, reasons, etc.
SANS	Simple Answer	An answer with short phrase or words, e.g. factoid, Yes/No
ELAB	Elaborate	Elaborate on a previous arguments or questions
CORR	Correct	Correct a wrong answer/solution with new one
OBJ	Object	Object to some argument/suggestions/ solutions
SUG	Suggest	Give advices/suggestions for some problems/solutions
SUP	Support	Support others' arguments/solutions
ACK	Acknowledge	Confirm or acknowledgement

Unlike in a flat document set, in a threaded discussion, each post plays a different role in the discussion. For example, people may make arguments, support or object to points, or give suggestions. In order to better extract useful information from the discussion, we defined a set of post speech acts based on [19]. Each message post was manually analyzed and assigned to a speech act category based on its role in the thread. Table 3 gives the specific definitions of each type of message post speech act.

4. THE DISCUSSION-BOT

Instructors, students, teaching assistants, and graders all play different roles on the discussion board. In this technical course students typically post questions about or discuss assignments, while teachers typically answer students' questions and post announcements. Students' questions are often time critical and require the response of a knowledgeable person. However, the large volume of requests makes it impossible for timely responses to all questions. As a virtual agent, the discussion-bot will mine answers automatically.

The virtual agent runs as a background server, monitoring new posts to the discussion board. We have chosen to process first-messages only, i.e., messages posted as a new topic or thread. These are typically queries, although we don't confirm this for reasons stated earlier: that because of lengthy contexts and complex descriptions, it is difficult to discern an exact question. We process only first posts because we wish the interface to be subtle and wish to avoid stifling interchanges with a 'search engine' effect. As shown in Figure 2, we then automatically process the query and generate a human-like reply.

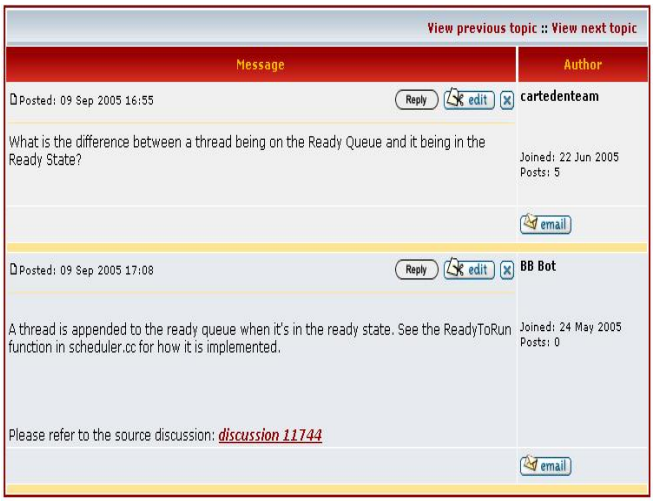

Figure 2. Example of a student query and its BB Bot response.

4.1 System Architecture

We have implemented the discussion-bot within the ISI discussion board. Figure 3 depicts the system architecture. When a query is posted to the discussion board, the discussion-bot system extracts features from the post first, e.g. the words included and their frequencies. Following that, the system tries to match student's interest in all the archived data (course documents and past discussions). A list of document tiles or posts related to the question is ranked based on predefined metrics. The answer extraction module processes the top 1 candidate in the list based on whether it is a segment of a document or a post. Different strategies are then applied to generate the answer which will be automatically presented on the discussion board as a human-like reply.

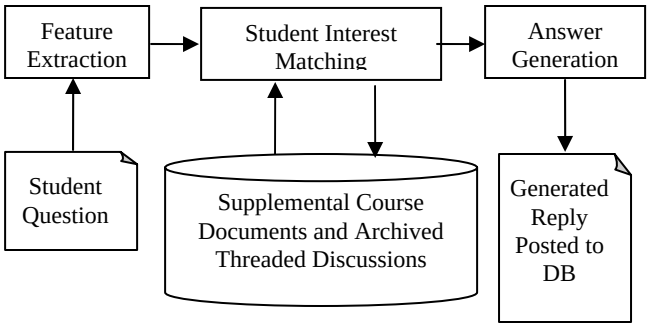

Figure 3. System architecture

4.2 Matching Students' Interests

After a question is posted, the system determines which document or message best matches the student's interest based on features extracted from the post.

Traditional question answering systems in natural language community usually apply a question-processing module to determine an answer type and extract query terms for the search engine [e.g. 5, 6, 7, 9]. The process is not feasible in our case because it is difficult to discern an exact question and thus identify an answer type and enumerated query terms. Instead, we retrieve a set of semantic-related passages that match a student's interest by directly computing a cosine similarity between question post and archived data with TF*IDF technique [14]. Here a passage refers to a document tile or a post.

Intuitively, document tiles and posts with similar words are more likely to be semantic-related. This information is represented by term frequency (TF). However, those with more general terms may be unintentionally biased when only TF is considered, so inverse document frequency (IDF) is introduced to fix the bias. This results in a more general term appearing in more data units with a smaller weight.

Mathematically, suppose we have a total of N passages in the corpus, a student query q, and corpus passages $p_1, p_2, ..., p_n$. Also suppose there are a total of m different unique words, $w_1, w_2, ..., w_m$ found in all documents and posts. Let the number of occurrences of word w_j in passage p_i be tf_{ij} and the number of passages in which word w_j is found c_j. The student's post and each passage in the corpus can be represented with a vector in the following format:

$$q = < w_{q1}, w_{q2}, ..., w_{qm} >$$

$$p_i = < w_{p_i1}, w_{p_i2}, ..., w_{p_im} >$$

where m is the total number of words in this domain, and $w = 0$ if a word is missing in that passage. Each element then, after normalization in the vectors, can be computed by

$$w_{ij} = \frac{tf_{ij} \log(N / c_j)}{\sqrt{\sum_{j=1}^{m} (tf_{ij})^2 [\log(N / c_j)]^2}}$$

When a question is posted, we can retrieve a list of semantic-related passages (post or document tile) using the cosine similarities between the query post and the passages using

$$\cos\_sim(q, p_i) = \frac{\sum_{j=1}^{m} w_{qj} * w_{p_i j}}{\sqrt{\sum_{j=1}^{m} (w_{qj})^2 * \sum_{j=1}^{m} (w_{p_i j})^2}}$$

4.3 Answer Generation

The resultant list of passages is ranked in the descendant order of the cosine similarities. Typically, we will take the top 1 result to generate the reply post. Depending on the type of passage, we generate the answers in different ways. If the passage is a document tile, we take all the text contained as the answer and attach the reference link to the original whole document.

If the passage is a post, we apply an extraction procedure to generate the answer based on the analysis of the threaded discussions using the post speech acts defined in the previous section. Typically, we take the post that appears in top position and search through the thread it comprises, applying our forward-backward answer generation algorithm that is shown in Figure 4.

1. Define boundary node set and relation rule set;
2. Start from the starting post, and go forward with DFS to construct a node list;
3. Start from the starting post, and go backward with DFS to construct a node list;
4. Combine result lists from Step 2 and Step 3;
5. Generate text answers from the list in Step 4 and attach a reference to the original thread discussion.

Figure 4. Forward-backward answer generation algorithm

The forward-backward algorithm is similar to computing state probability in a Hidden Markov Model [13], our algorithm first traverses the thread graph forward with a Depth First Search (DFS) followed by a backward traverse with DFS. If it meets the boundary test, it will stop at that level. During the DFS traverse, some extraction rules are applied, for example, a CORRECT node will void arguments in the precious node. The reply post is finally assembled with a reference to the original source, either a full document or a whole discussion.

5. EXPERIMENTS

5.1 Experimental Setup

For evaluation purposes, we used the operating systems course corpus, which includes 279 documents and 3093 posts. The archived discussions contain 1236 threads. We investigate the distribution of the length of each thread, that is, how many posts are included in each thread. Figure 5 gives the distribution of the length of each thread: 524 threads (over 40%) consist of only one post while most of the threads consist of from two to ten posts. There are very few threads that contain more than 10 posts.

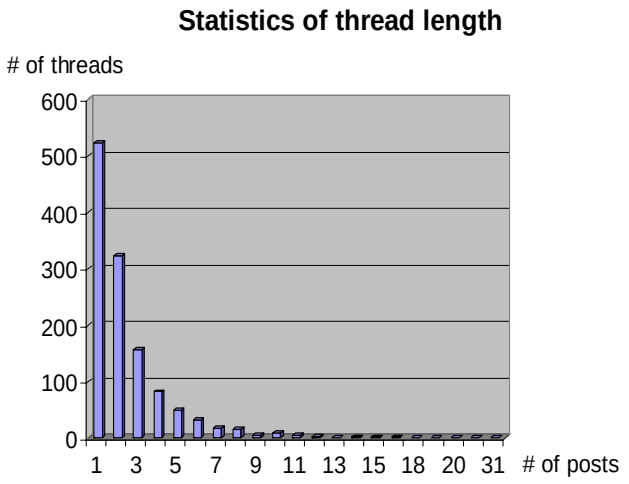

Figure 5. Statistics of thread length

With three semesters of data it was easy to carry out a realistic experiment. We chose to evaluate the most recent semester of archived discussion data against the previous two semesters of archived data. The test set contained 66 questions. There is no guarantee the questions were answered or even discussion in the corpus. The testing performance for this situation was expected to be considerably worse than for the situation where the condition has been met (i.e., an answer exists).

We write a script to "pretend" to be "a pseudo user" to post new questions on the discussion board, which could trigger the discussion-bot automatically. The human-like reply generated by the discussion-bot will be posted subsequently following the original post on the discussion board.

5.2 Results and Analysis

To better search the threaded discussions and extract the most useful information, all archived posts were manually classified as one of eleven speech acts described in Section 3. Our corpus includes a total of 2173 speech acts. Table 4 gives the percentage of speech acts found in all posts of our annotated corpus.

We find that questions comprise the biggest portion of the corpus. This is consistent with the use of the board as a technical question and answer platform. Correspondingly, answers (CANS and SANS) and suggestions comprise 39.03% of total posts. The reason we consider suggestions together with answers is that for some of the questions, it is difficult to give an exact answer and in most cases, the replies are presented as suggestions. The ratio of complex answers to simple answers is 6.3. This matches our expectation that students ask lengthy context and procedural questions instead of simple factoid or Yes/No questions.

We also investigate the relations between two consecutive posts. As each post is classified as a speech act, the relations are represented by the consecutive relations between post speech acts. Figure 6 depicts the relations for all speech acts. To make it easier to understand, we also add "START" and "END" states that refer to the start and the end of a thread discussion respectively. Each arc connecting two states is labeled with a probability from the example, there is a 78.8% probability that any given discussion will start with a question (QUES), and an 18.4% probability that it will start with a description of a situation (DESC).

previous speech act going to next SA. The information shows us how a discussion is conducted within a group of students. For

Table 4. Statistics of post speech acts in archived threaded discussions

Code	Frequency	Percentage (%)
QUES	794	36.54
COMM	11	0.51
DESC	133	6.12
CANS	372	17.12
SANS	59	2.72
ELAB	149	6.86
CORR	25	1.15
OBJ	37	1.70
SUG	417	19.19
SUP	105	4.83
ACK	71	3.27

To evaluate the quality of the automatically generated reply, human judges were asked to manually classify each of the 66 human-like replies as, *Exact Answer*, *Good Answer*, *Related Answer*, or *Unrelated Answer*. Based on these classifications, we tested both the student interest-matching algorithm and the answer generation algorithm using two different strategies. The first strategy is all-in test, which exploits the whole corpus including the thread that the original question came from. This is designed mainly for testing the answer generation module. The second strategy is self-out test, which excludes the thread that the original question came from. In this test, we tried to simulate the ideal test situation with different configurations by tuning threshold values.

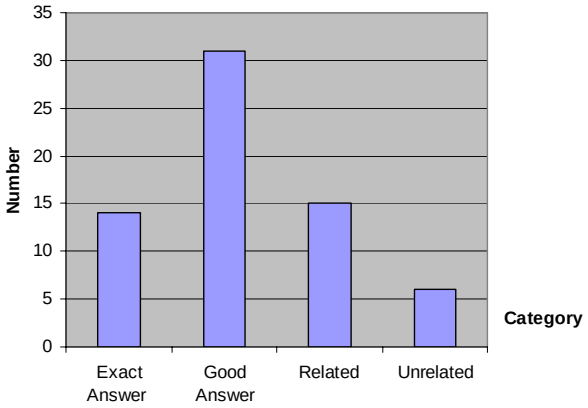

Figure 7. All-in test performance

Figure 7 gives the evaluation results for the all-in test, which tests the efficiency of the answer generation algorithm given the assumption that the student's interest is correctly matched. Even in this case, though, we are not guaranteed a response (in the case no one replied to the question), and if there was an original response, it may not have been an exact answer (it may have been classified as a suggestion, for example.) It is also possible that the

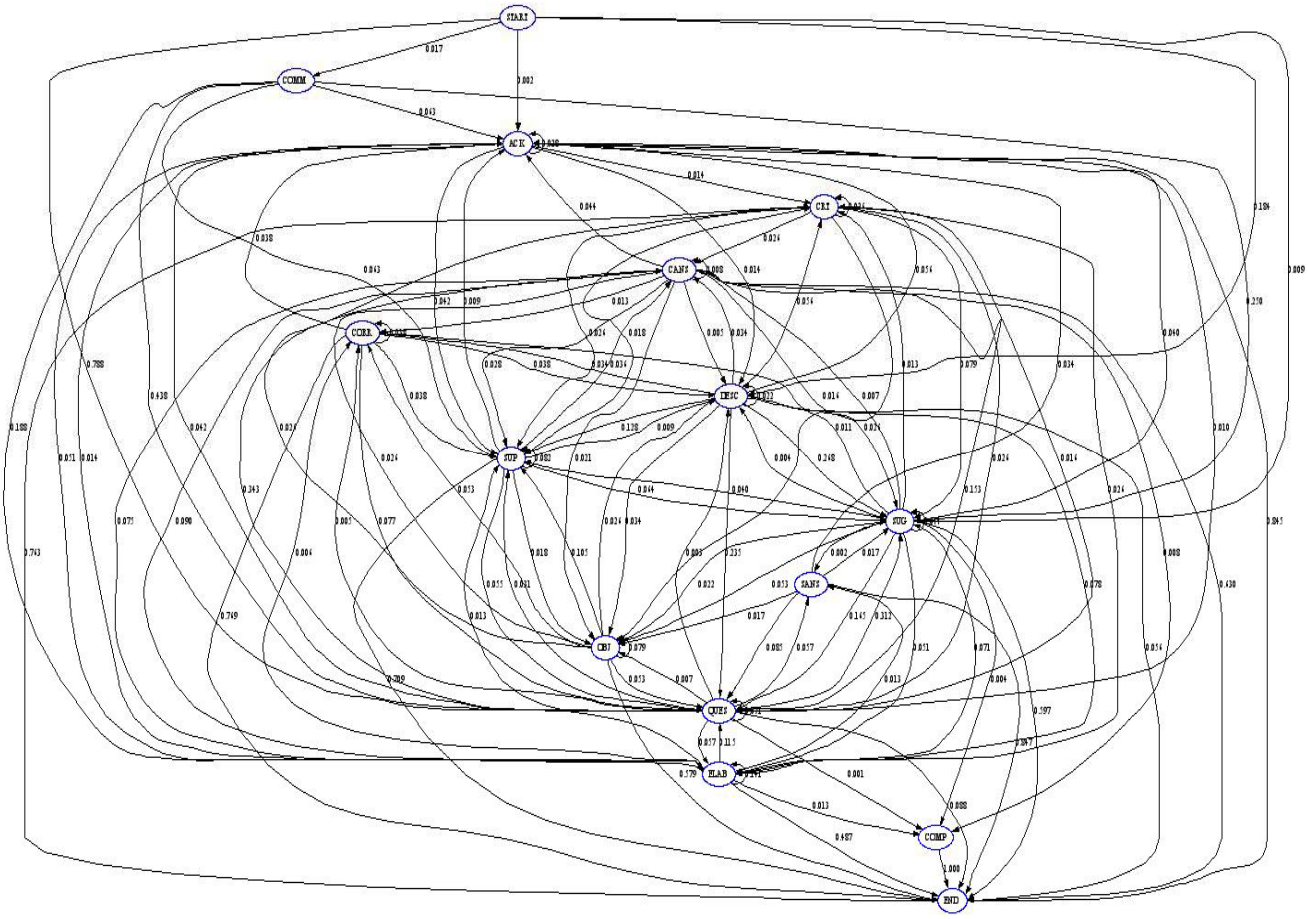

Figure 6. Threaded discussion model

target answer was removed by the extraction rules due to the complexity of post Speech Act analysis. The results are as expected.

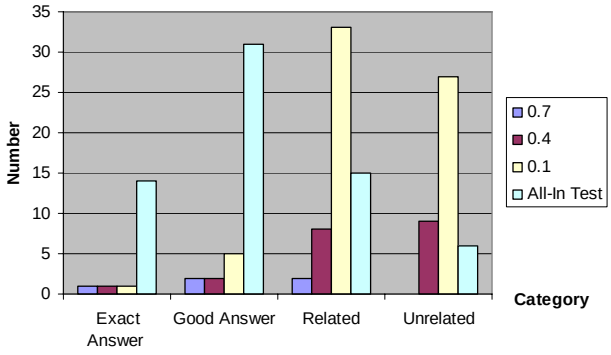

Figure 8. Self-out vs. All-in evaluation

Figure 8 gives the evaluation results for a series of self-out tests, and compares these results to the all-in results. Each self-out test corresponds to an adjustment of a threshold value on the cosine similarity score. In this case, there is no guarantee of an interest match, or that the topic was ever discussed previously (recall that over 40% of threads consist of a single post), and we see the results shift from Exact/Good Answer categories to Related/Unrelated categories. As we increase the threshold value from 0.1 to 0.7, the results falling into Related/Unrelated categories decrease greatly and the system returns less noise in the result set. These results are based on two semester's of data; we would expect to see them improve naturally over time (assuming the course is similarly taught). Although the results are not ideal, the matching, modeling, extraction, and evaluation processes show promise and merit iteration; the understanding of complex questions is very much an unsolved problem.

6. DISCUSSION AND FUTURE WORK

In this paper, we addressed one of the challenges of natural language understanding in the context of learning, to automatically reply to students' questions on a course discussion board. We implemented an intelligent discussion-bot, which can monitor students' posts, match students' interests, and generate human-like replies automatically. Evaluations on the discussion-bot gave encouraging results to meet students' learning requests.

We are currently working to acquire data from other sources: We recently began asking students to classify their own replies, and are experimenting with the Virage Video Logger, which will provide transcriptions of the class webcasts. We plan to improve the system's performances in several aspects. In the current discussion-bot system, the relations between posts are identified and analyzed manually. Automatically labeling their relations

based on language features is highly desirable. More complex reasoning-based strategies to provide more accurate and compact answers from archived threads are also being considered.

7. ACKNOWLEDGMENTS

The work was supported in part by a grant from the Lord Corporation Foundation to the Distance Education Network (DEN) at the University Of Southern California Viterbi School Of Engineering.

We thank USC Professor Michael Crowley for providing the discussion archives, Lei Ding, Feng Pan, Patrick Pantel, Deepak Ravichandran, and Mei Si for their insightful contributions and Swapnil Patel for his effective support in integrating the discussion-bot to the ISI phpBB discussion board system.

8. REFERENCES

[1] ALICE. http://www.alicebot.org/

[2] Cakir, M., Xhafa, F., Zhou, N., and Stahl, G. Thread-based analysis of patterns of collaborative interaction in chat, *AIED* 2005.

[3] Conati, C. and Zhao, X. 2004. Building and evaluating an intelligent pedagogical agent to improve the effectiveness of an educational game. In *Proceedings of the 9th international Conference on intelligent User interface*, IUI '04. ACM Press, New York, NY, 6-13.

[4] Hearst, M. A. Multi-paragraph segmentation of expository text. *Proceedings of the 32th Annual Meeting of the Association for Computational Linguistics(ACL-94)*, 9-16, Las Cruces, New Mexico, 1994.

[5] Hermjakob, U., Hovy, E. H., and Lin, C. 2000. Knowledge-based question answering. *TREC-2000*.

[6] Hovy, E.H., Gerber, L., Hermjakob, U., Junk, M., and Lin, C. 2000. Question answering in Webclopedia. *Proceedings of TREC-2000*.

[7] Moldovan, D., Harabagiu, S., Pasca, M., Mihalcea, R., Girju, R., Goodrum, R., and Rus, V. 2000. The structure and performance of an open-domain question answering system. *Proceedings of ACL-2000*.

[8] Nguyen, A. and Wobcke, W. 2005. An Agent-Based Approach to Dialogue Management in Personal Assistants. *Proceedings of the 2005 International Conference on Intelligent User Interfaces*, 137-144.

[9] Pasca, M. and Harabagiu, S. 2001. High Performance Question/Answering, in *Proceedings of SIGIR-2001*. pages 366-374.

[10] phpBB. http://www.phpbb.com/

[11] Popescu, A., Etzioni, O., and Kautz, H. 2003. Towards a theory of natural language interfaces to databases. In *Proceedings of the 8th international Conference on intelligent User interfaces* (Miami, Florida, USA, January 12 - 15, 2003). IUI '03. ACM Press, New York, NY, 149-157.

[12] Prager, J. M., Chu-Carroll, J., and Czuba, K.W.. 2004. Question answering using constraint satisfaction. *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04)*, 2004.

[13] Rabiner, L.R. and Juang, B.H. 1986. An introduction to Hidden Markov Models. *IEEE Signal Processing Magazine* 61 (1986) 4--16

[14] Salton, G. *Automatic Text Processing, The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, Reading, MA, 1989.

[15] Shaw, E. Assessing and Scaffolding Collaborative Learning in Online Discussions. In *Proceedings of the 12th International Conference on AI in Education (AIED '05)* (Amsterdam, July 19-23, 2005).

[16] Soller, A., and Lesgold, A. 2003. A computational approach to analyzing online knowledge sharing interaction, *Proceedings of AI in Education 2003*, Sydney, Australia, 253-260

[17] Suebnukarn, S. and Haddawy, P. 2004. A collaborative intelligent tutoring system for medical problem-based learning. In *Proceedings of the 9th international Conference on intelligent User interface* January 13 - 16, 2004. IUI '04. ACM Press, New York, NY, 14-21.

[18] Terrell, S. R. and Dringus, L. 2000. An Investigation of the Effect of Learning Style on Student Success in Online Learning Environment. *Journal of Education Technology Systems*, 28 (3), 231-238, 2000.

[19] Winograd, T. 1987. A Language/Action Perspective on the Design of Cooperative Work. *Human-Computer Interactions*, 3:1, pp. 3-30.

[20] Xu, J., Licuanan, A., Weischedel, R. 2003. TREC 2003 QA at BBN: Answering Definitional Questions. *Proceedings of TREC* 2003.

[21] Yates, A., Etzioni, O., and Weld, D. 2003. A reliable natural language interface to household appliances. In *Proceedings of the 8th international Conference on intelligent User interfaces* (Miami, Florida, USA, January 12 - 15, 2003). IUI '03. ACM Press, New York, NY, 189-196.