Towards Identifying Unresolved Discussions in Student Online Forums

Jihie Kim, Jia Li, and Taehwan Kim
University of Southern California
Information Sciences Institute
4676 Admiralty Was, Marina del Rey, CA, U.S.A
{jihie, jiali, taehwan}@isi.edu

Abstract

Automatic tools for analyzing student online discussions are highly desirable for providing better assistance and promoting discussion participation. This paper presents an approach for identifying student discussions with unresolved issues or unanswered questions. In order to handle highly incoherent data, we perform several data processing steps. We then apply a two-phase classification algorithm. First, we classify "speech acts" of individual messages to identify the roles that the messages play, such as question, issue raising, and answers. We then use the resulting speech acts as features for classifying discussion threads with unanswered questions or unresolved issues. We performed a preliminary analysis of the classifiers and the system shows an average F score of 0.76 in discussion thread classification.

1 Introduction*

Online discussion boards have become a popular and important medium for distance education. Students use discussion forums to collaborate, to exchange information, and to seek answers to problems from their instructors and classmates. Making use of the dialog to assess student understanding is an open research problem. As the class size increases and online interaction becomes heavier, automatic tools for analyzing student discussions are highly desirable for providing better assistance and promoting discussion participation. In this paper, we present an approach for automatically identifying discussions that have unresolved issues or unanswered questions. The resulting discussions can be reported to instructors for further assistance.

We present a two-phase machine learning approach where the first phase identifies high level dialogue features (speech acts such as question, issue raising, answer, and acknowledgement) that are appropriate for assessing student interactions. The second phase uses speech acts as features in creating thread classifiers that identify discussions with unanswered questions or unresolved issues. We also describe an approach where thread classifiers are created directly from the features in discussion messages. The preliminary results indicate that although the direct learning approach can identify threads with unanswered questions well, SA based learning provide a little better results in identifying threads with issues and threads with unresolved issues.

2 Modeling Student Discussions

Our study 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. We obtain our data from an existing online discussion board that hosts student technical discussions. Total 291 discussion threads (219 for training and 72 for test) with 1135 messages (848 for training and 287 for test) from two semesters' discussions were used for this study. 168 students participated in the discussions.

2.1 Discussion Threads

Unlike prototypical collaborative argumentation where a limited number of members take part in the conversation with a strong focus on solving specific problems, student online discussions have much looser conversational structure, possibly involving multiple anonymous discussants. Student

*

discussions are very informal and noisy with respect to grammar, syntax and punctuation. There is a lot of variance in the way that students present similar information. Messages about programming assignments include various forms of references to programming code. Figure 1 shows an example discussion thread that is relatively technical and formal. The raw data include humorous messages and personal announcements as well as technical questions and answers.

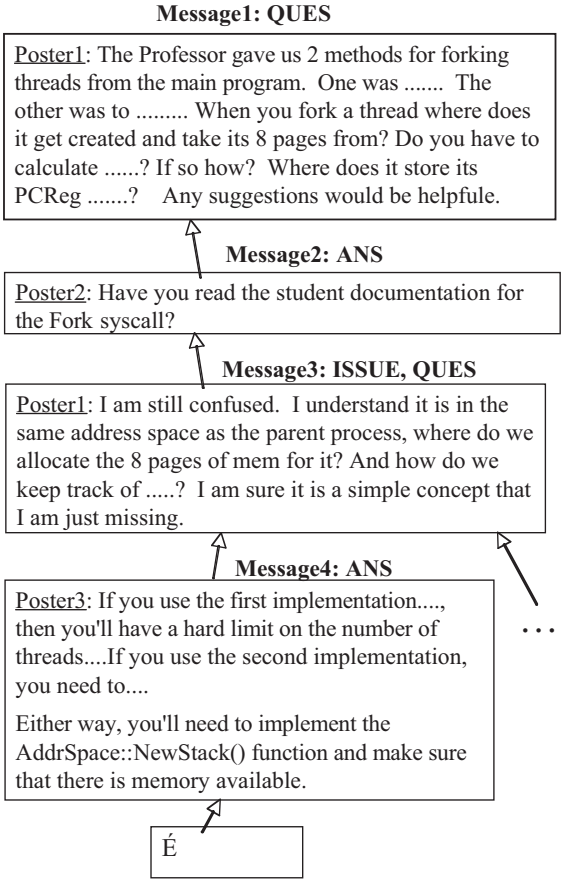


Figure 1. An example discussion thread

The average number of messages per discussion thread in our undergraduate course is 3.9, and many discussion threads contain only two or three messages. Discussions often start with a question from a student on a project or an assignment. In some cases, the discussion ends with an answer that follows the question. In some other cases, the original poster may raise additional issues or ask questions about the answer. The discussion can continue with the following answer from another student as in Figure 1. However, sometimes the discussion ends with hanging issues or questions without an answer.

2.2 Speech Acts in messages: Identifying roles that a message plays

For conversation analysis, we adopted the theory of Speech Acts (SAs) to capture relations between messages (Austin, 1962; Searle, 1969). Each message within a discussion thread may play a different role. A message could include a question for a particular problem, or it could contain an answer or suggestion with respect to a previous question in the thread. Messages can include question, answer, acknowledgement, and objection. Since SAs are useful in understanding contributions made by students in discussions, and are natural indicators for unanswered questions or unresolved issues, we use SAs as features for classifying discussion threads in a two phase learning as described below.

Table 1. Speech Act Categories and Kappa values

SA Category	Description	kappa
QUES	A question about a problem, including question about a previous message	0.94
ANS	A simple or complex answer to a previous question. Suggestion or advice	0.72
ISSUE	Report misunderstanding, unclear concepts or issues in solving problems	0.88
Pos-Ack	An acknowledgement, compliment or support in response to a prev. message	0.87
Neg-Ack	A correction or objection (or complaint) to/on a previous message	0.85

We divide message roles into several SA categories, extending the approaches presented in (Kim et al., 2006; Kim and Ravi 2007). We focus on the categories that are relevant to the problem of identifying discussion threads with unanswered question or unresolved issues.

The message might contain a question about a particular problem (QUES) or report a misunderstanding, unclear concepts or issues in solving a problem (ISSUE). It might propose an answer or suggestion with respect to a previous question in the thread (ANS). Finally, a message might acknowledge the previous message with support

(Pos-Ack) or show disagreement or objection (Neg-Ack). SAs relate a pair of messages that has a 'reply-to' relation. A pair of messages can be labeled with multiple SAs, and a message can have multiple SAs with more than one messages. This allows us to capture various relations among messages. Table 1 describes the categories we are focusing on and the kappa values from two annotators. Figure 1 shows SA relations between message pairs.

During annotation of the corpus, the annotators marked the cues that are relevant to a particular SA category as well as the SA categories themselves. Such information provides hints on the kinds of features that are useful. We also interviewed the annotators to capture additional cues or indicators that they used during the annotation. We iterated with several different annotation approaches until we reach enough agreement among the annotators on a new dataset that was not seen by the annotators before.

Table 2 shows the distribution statistics of each SA category among the whole training and test corpus. Since a message may have more than one SA, the percentage sum of all SAs doesn't equal to 1. As we can see, Pos-Ack and Neg-Ack are experiencing lacking data problem which is one of the challenges we are facing for SA classification.

Table 2. Statistics for each Speech Act Category

SA Category	Training set		Test set	
	# of msgs	Percentage	# of msgs	Percentage
QUES	469	55.31%	146	50.87%
ANS	508	59.91%	176	61.32%
ISSUE	136	16.03%	46	16.03%
Pos-Ack	78	9.20%	30	10.45%
Neg-Ack	23	2.71%	8	2.79%

3 Message Speech Act Classifiers

In this section, we first describe how raw discussion data is processed and show the features generated from the data, and we then present the current SA classifiers.

3.1 Discussion Data Pre-processing

Besides typical data preprocessing steps, such as stemming and filtering, which are taken by most NLP systems, our system performs additional steps to reduce noise and variance (Ravi and Kim 2007).

We first remove the text from previous messages that is automatically inserted by the discussion board system starting with righ angle braket (>) when the user clicks on a *"Reply to"* button. We also apply a simple stemming algorithm that removes *"s"* and *"es"* for plurals. Apostrophes are also converted to their original forms. E.g., "I'm" is converted to "I am". For discussions on programming assignment, the discussion included programming code fragments. Each section of programming code or code fragment is replaced with a single term called code. Similar substitution patterns were used for a number of categories like filetype extensions (*".html", ".c", ".c++", ".doc"*), URL links and others. Students also tend to use informal words (e.g. *"ya", "yeah", "yup"*). We substitute some of such words with one form ("yes"). For words like *"which", "where", "when", "who" and "how"*, we used the term categ_wh. We do not replace pronouns ("I", "we", "they",) since they may be useful for identifying some SAs. For example, "You can" may be a cue for ANS but "I can" may not.

We also apply a simple sentence divider with simple cues (punctuation and white spaces such as newline) in order to captures the locations of the features in the message, such as cue words in the first sentence vs. cues in the last sentence.

3.2 Features for Speech Act Classification

We have used six different types of features based on input from the annotators.

F1: cue phases and their positions: In addition to SAs (e.g. QUES), the human annotators marked the parts within the message (cue phrases or sentences), which helped them identify the SAs in the message. In order to overcome data sparseness, we generate features from the marked phrases. That is, from each phrase, we extract all the unigrams, bigrams, trigrams (sequence of 1/2/3 words) and add them to the feature set. We also added two separate unigrams, three separate unigrams and a unigram and a bigram combinations since the annotations in the training data indicated that they could be a useful pattern. All the cues including separate cues such as two unigrams are captured and used for a single sentence. Positions of the cues are included since in longer messages the cues in the beginning

sentences and the ones in the end sentences can indicate different SAs. For example, THANK in the beginning indicates a positive answer but THANK in latter part of the message usually means politeness (thank in advance).

F2: Message Position: Position of current message within the discussion thread (e.g. the first message, the last message, or middle in the thread).

F3: Previous Message Information: SAs in the previous message that the current message is replying to.

F4: Poster Class: Student or Instructor.

F5: Poster Change: Was the current message posted by the same person who posted the message that the current message is replying to?

F6: Message Length: Values include Short(1-5words), Medium(6-30 words), and Long(>30 words).

F1 is a required feature since the annotators indicated cues as useful feature in most cases. All the others are optional.

3.3 Speech Act Classifiers

We applied SVM in creating binary classifiers for each SA category using Chang and Lin (2001). Also, Transformation-based Learning (TBL) was applied as it has been successfully used in spoken dialogue act classification (Samuel 2000; Brill 1995). It starts with the unlabeled corpus and learns the best sequence of admissible "transformation rules" that must be applied to the training corpus to minimize the error for the task. The generated rules are easy to understand and useful for debugging the features used. TBL results are also used in generating *dependencies* among SA categories for **F3**, i.e. which SAs tend to follow which other SAs[1], as describe below.

SA Classification with TBL

Each rule $Rule_i$ is composed of two parts - (1) $RuleLHS_i$ - A combination of features that should be checked for applicability to the current message (2) $RuleTAG_i$ - SA tag to apply, if the feature combination is applicable to the current message.

[1] It is possible to collect related clues from SVM with distribution of feature values and information gain although dependencies can be easily recognized in TBL rules.

$$Rule_i :: RuleLHS_i => RuleTAG_i$$

Where $RuleLHS_i = X_i$

$$X_i \in X; (X \subseteq F1 \times F2 \times F3 \times F4 \times F5 \times F6)$$

The $RuleLHS_i$ component can be instantiated from all the combination of the features F1, …,F6. $RuleTAG_i$ is any SA (single SA) chosen from a list of all the SA categories. An example rule used in Speech Act learning is shown below:

Rule1 :: IF cue-phrase = {"not", "work"}
& poster-info = Student
& post-length = Long
=> ISSUE

Rule1 means if the post contains two unigrams "not" and "work", the poster is a student, and the post length is long, then the Speech Act for the post is ISSUE.

We apply each rule in the potential rule set on all the posts in the training corpus and transform the post label if the post is applicable. The rule with highest improvement by F score is selected into the optimal rule set and moved from the potential rule set. The iteration continues until there is no significant improvement with any rule.

The training corpus was divided into 3 parts for 3-fold cross validation. The rules from 3 rule sets are merged and sorted by weighted Mean Reciprocal Rank (MRR) (Voorhees, 2001). For example, if we have 5 rules among 3 rule sets as follows,

Rule set 1 (0.85 on test): R1 R2 R3
Rule set 2 (0.88 on test): R2 R1 R4
Rule set 3 (0.79 on test): R1 R4 R5

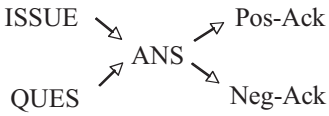
For R1, we calculate the weighted MRR as (0.85*1 + 0.88*(1/2) + 0.79*1) / 3. After sorting, we get top N rules from the merged rule set. Table 3 shows some of the rules learned.

Table 3. TBL rule examples

IF cue-phrase = {"?"} => QUES
IF cue-phrase = {"yes you can"} & poster-info = Instructor & post-length = Medium => ANS
IF cue-phrase = {"yes"} & cue-position = CP_BEGIN & prev-SA = QUES => ANS
IF cue-phrase = {"not know"}

& poster-info = student
& poster-change = YES => ISSUE

Based on the rules generated from TBL, we analyze dependencies among the SA categories for **F3** (previous message SAs). In TBL rules, ANS depends on ISSUE and QUES, i.e. some ANS rules have QUES and ISSUE for **F3**. Also Pos-Ack and Neg-Ack tend to follow ANS. Both SVM and TBL classifiers use this information during testing. That is, we apply independent classifiers first and then use dependent classifiers according to the dependency order as following:



Currently there is no loop in the selected rules but we plan to address potential issues with loops in SA dependencies.

SA Classification with SVM

Table 4. Some of the top selected features by Information Gain

SA Category	Top features
QUES	"?" POST_POSITION "_category_wh_ … ?" PREV_SA_FIRST_NONE "to … ?"
ANS	POST_POSITION PREV_SA_QUESTION "?" POSTER_INFO
ISSUE	POSTER_INFO "not … sure" POST_POSITION FEATURE_LENGTH "error"
Pos-Ack	PREV_SA_ANSWER POST_POSITION PREV_SA_FIRST_NONE "thanks" & cue-position = CP_BEGIN "ok" & cue-position = CP_BEGIN
Neg-Ack	"yes, " ", but" POST_POSITION ", but" "are … wrong"

Given all the combination of the features F1,…, F6, we use Information Gain (Yang and Pederson 1997) for pruning the feature space and selecting features. For each Speech Act, we sort all the features (lexical and non-lexical) by Information Gain and use the top N (=200) features. Table 4 shows the top features selected by Information Gain. The resulting features are used in representing a message in a vector format.

We did 5-fold cross validation in the training. RBF (Radial Basis Function) is used as the kernel function. We performed grid search to get the best parameter (C and gamma) in training and applied them to the test corpus.

Table 5. SA classification results

SA Category	SVM			TBL		
	Prec.	Recall	F score	Prec.	Recall	F score
QUES	0.95	0.90	0.94	0.96	0.91	0.95
ANS	0.87	0.80	0.85	0.83	0.64	0.78
ISSUE	0.65	0.54	0.62	0.46	0.76	0.50
Pos-Ack	0.57	0.44	0.54	0.58	0.56	0.57
Neg-Ack	0	0	0	0.5	0.38	0.47

Table 5 shows the current classification accuracies with SVM and TBL. The main reason that ISSUE, Pos-Ack and Neg-Ack show low scores is that they have relatively small number of examples (see statistics in Table 2). We plan to add more examples as we collect more discussion annotations. For thread classification described below, we use features with QUES, ANS, ISSUE and Pos_Ack only.

4 Identifying Discussions with Unanswered or Unresolved Questions: Thread Classification

Figure 2 shows typical patterns of interactions in our corpus. Many threads follow pattern (a) where the first message includes a question and the subsequent message provides an answer. In (b), after an answer, the student presents an additional question or misunderstanding (ISSUE), which is followed by another answer. Often students provide positive acknowledgement when an answer is sat-

isfying. Pattern (c) covers cases for when the question is unanswered.

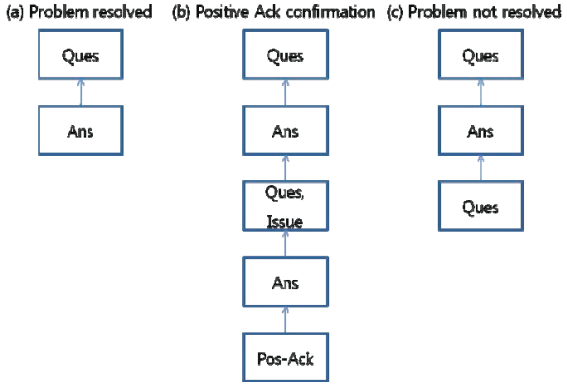


Figure 2. Example patterns in student discussion threads

We are interested in the following assessment questions.

(Q1) Were all questions answered? (Y/N)

(Q2) Were there any issues or confusion? (Y/N)

(Q3) Were those issues or confusions resolved? (Y/N)

There can be multiple questions, and Q1 is false if there is any question that does not have a corresponding answer. That is, even when some questions were resolved, it could still be False (not resolved) if some were not resolved. If Q2 is False (i.e. there is no issue or question), then Q3 is also False.

These questions are useful for distinguishing different interaction patterns, including threads with unanswered questions. In the second phase of learning, we use SA-based features. Our initial analysis of student interactions as above indicates that the following simple features can be useful in answering such questions:

(T-F1) Whether there was an [SA] in the thread

(T-F2) Whether the last message in the thread included [SA]

We used TBL rules for Pos-Ack and SVM classifiers for other SA categories because of relatively higher score of Pos-Ack from TBL and other categories from SVM. We use 8 (2 x 4) features created from T-F1 and T-F2. SVM settings are similar to the ones used in the SA classification.

Table 6 shows the thread classification results. We checked SVM classification results with human annotated SAs since they can show how useful SA-based features are (T-F1 and T-F2 in particular) in answering Q1—Q3. The results shown in Table 6-(a) indicate that the features (T-F1 and T-F2) are in fact useful for the questions.

When we used the SA classifiers and SVM in a pipeline, the system shows precisions (recalls) of 83%(84%), 77%(74%) and 68%(69%) for Q1, Q2, and Q3 respectively.

Table 6. Thread Classification Results

	Precision	**Recall**	**F score**
Q1	0.93	0.93	0.93
Q2	0.93	0.93	0.93
Q3	0.89	0.89	0.89

(a) Classification results with human annotated SAs

	Precision	**Recall**	**F score**
Q1	0.83	0.84	0.83
Q2	0.77	0.74	0.76
Q3	0.68	0.69	0.68

(b) SVM classification results with system generated SAs

The results with system generated SAs provide an average F score of 0.76. Although the ISSUE classifier has F score of 0.62, the score for Q2 is 0.76. Q2 checks one or more occurrences of ISSUE rather than identifying existence of ISSUE in a message, and it may become an easier problem when there are multiple occurrences of ISSUEs.

5 Direct Thread Classification without SAs

As an alternative to the SA-based two-phase learning, we crated thread classifiers directly from the features in discussion messages. We used SVM with the following features that we can capture directly from a discussion thread.

F1': cue phases and their positions in the thread: we use the same cue features in F1 but we use an optional thread level cue position: Last_message and Dont_Care. For example, for a given cue "ok", if it appears in the the last message of the thread, we generate two features, "ok"_Last_message and "ok"_Dont_Care.

Given a set of candidate features, we use Information Gain to select the top N (=200) features. The resulting features are used in creating vectors as described inS 3.3. Similar cross-validation and SVM settings are applied.

Table 7. Results from Direct Thread Classification

	Precision	Recall	F score
Q1	0.86	0.86	0.86
Q2	0.81	0.62	0.70
Q3	0.75	0.33	0.46

Table 7 shows the classification results. Although the direct learning approach can identify threads with unanswered questions well, SA based learning provides a little better results in identifying threads with issues (Q2) and unresolved issues (Q3). It seems that SA-based features may help performing more difficult tasks (e.g. assessment for ISSUEs in discussions) We need further investigation on different types of assessment tasks.

6 Related Work

Rhetorical Structure Theory (Mann and Thomson, 1988) based discourse processing has attracted much attention with successful applications in sentence compression and summarization. Most of the current work on discourse processing focuses on sentence-level text organization (Soricut and Marcu, 2003) or the intermediate step (Sporleder and Lapata, 2005). Analyzing and utilizing discourse information at a higher level, e.g., at the paragraph level, still remains a challenge to the natural language community. In our work, we utilize the discourse information at a message level.

There has been prior work on dialogue act analysis and associated surface cue words (Samuel 2000; Hirschberg and Litman 1993). There have also been Dialogue Acts modeling approaches for automatic tagging and recognition of conversational speech (Stolcke et al., 2000) and related work in corpus linguistics where machine learning techniques have been used to find conversational patterns in spoken transcripts of dialogue corpus (Shawar and Atwell, 2005). Although spoken dialogue is different from message-based conversation in online discussion boards, they are closely related to our thread analysis work, and we plan to investigate potential use of conversation patterns in spoken dialogue in threaded discussions.

Carvalho and Cohen (2005) present a dependency-network based collective classification method to classify email speech acts. However, estimated speech act labeling between messages is not sufficient for assessing contributor roles or identifying help needed by the participants. We included other features like participant profiles. Also our corpus consists of less informal student discussions rather than messages among project participants, which tend to be more technically coherent.

Requests and commitments of email exchange are analyzed in (Lampert et al., 2008). As in their analysis, we have a higher kappa value for questions than answers, and some sources of ambiguity in human annotations such as different forms of answers also appear in our data. However, student discussions tend to focus on problem solving rather than task request and commitment as in project management applications, and their data show different types of ambiguity due to different nature of participant interests.

There also has been work on non-traditional, qualitative assessment of instructional discourse (Graesser et al., 2005; McLaren et al., 2007; Boyer et al., 2008). The assessment results can be used in finding features for student thinking skills or level of understanding. Although the existing work has not been fully used for discussion thread analysis, we are investigating opportunities for using such features to cover additional discourse analysis capabilities. Similar approaches for classifying speech acts were investigated (Kim and Ravi 2007). Our work captures more features that are relevant to analyzing noisy student discussion threads and support a full automatic analysis of student discussions instead of manual generation of thread analysis rules.

7 Summary and Future Work

We have presented an approach for automatically classifying student discussions to identify discussions that have unanswered questions and need instructor attention. We applied a multi-phase learning approach, where the first phase classifies individual messages with SAs and the second phase classifies discussion threads with SA-based features. We also created thread classifiers directly from features in discussion messages. The preliminary results indicate that SA-based features may help difficult classification tasks. We plan to perform more analysis on different types of thread classification tasks.

We found that automatic classification of undergraduate student discussions is very challenging

due to incoherence and noise in the data. Especially messages that contain long sentences, informal statements with uncommon words, answers in form of question, are difficult to classify. In order to use other SA categories such as Neg-Ack and analyze various types of student interactions, we plan to use more annotated discussion data.

A deeper assessment of online discussions requires a combination with other information such as discussion topics (Feng et al., 2006). For example, classification of discussion topics can be used in identifying topics that participants have more confusion about. Furthermore, such information can also be used in profiling participants such as identifying mentors or help seekers on a particular topic as in (Kim and Shaw 2009). We are investigating several extensions in order to generate more useful instructional tools.

Acknowledgments

This work was supported by National Science Foundation, CCLI Phase II grant (#0618859).

References

Austin, J., *How to do things with words*. 1962. Cambridge, Massachusetts: Harvard Univ. Press.

Boyer, K., Phillips, R., Wallis M., Vouk M., Lester, J., Learner Characteristics and Feedback in Tutorial Dialogue. 2008. *ACL workshop on Innovative Use of NLP for Building Educational Applications*.

Brill, E. 1962. Transformation-based error-driven learning and natural language processing: a case study in part-of-speech tagging. *Comput. Linguist.,* 21(4).

Carvalho, V.R. and Cohen, W.W. 2005. On the collective classification of email speech acts. *Proceedings of SIGIR*.

Chang, C.-C. and Lin, C.-J. 2001. *LIBSVM: a library for support vector machines*.

Feng, D., Kim, J., Shaw, E., Hovy E., 2006. Towards Modeling Threaded Discussions through Ontology-based Analysis. *Proceedings of National Conference on Artificial Intelligence*.

Graesser, A. C., Olney, A., Ventura, M., Jackson, G. T. 2005. AutoTutor's Coverage of Expectations during Tutorial Dialogue. *Proceedings of the FLAIRS Conference*.

Hirschberg, J. and Litman, D. 1993. Empirical Studies on the Disambiguation of Cue Phrases", *Computational Linguistics*, 19 (3).

Kim, J., Chern, G., Feng, D., Shaw, E., and Hovy, E. 2006. Mining and Assessing Discussions on the Web through Speech Act Analysis. *Proceedings of the ISWC'06 Workshop on Web Content Miningwith Human Language Technologies* (2006).

Kim J. and Shaw E. 2009. Pedagogical Discourse: Connecting Students to Past Discussions and Peer Mentors within an Online Discussion Board, *Innovative Applications of Artificial Intelligence Conference*.

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

Mann, W.C. and Thompson, S.A. 1988. Rhetorical structure theory: towards a functional theory of text organization. Text: *An Interdisciplinary Journal for the Study of Text*, 8 (3).

McLaren, B. et al.,2007. Using Machine Learning Techniques to Analyze and Support Mediation of Student E‐Discussions, *Proc. of AIED 2007.*

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

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

Searle, J. 1969. *Speech Acts*. Cambridge: Cambridge Univ. Press.

Soricut, R. and Marcu, D. 2003. Sentence level discourse parsing using syntactic and lexical information. *Proceedings of HLT/NAACL-2003*.

Sporleder, C. and Lapata, M., 2005. Discourse chunking and its application to sentence compression. In *Proceedings of Human Language Technology conference – EMNLP*.

Stolcke, A. , Coccaro, N. , Bates, R. , Taylor, P. , et al., 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech, *Computational Linguistics*, v.26 n.3.

Shawar, B. A. and Atwell, E. 2005. Using corpora in machine-learning chatbot systems." *International Journal of Corpus Linguistics*, vol. 10.

Witten, I. H., and Frank, E. 2005. *Data Mining: Practical machine learning tools and techniques,* 2nd Edition, Morgan Kaufmann, San Francisco.

Yang, Y. and Pedersen, J. 1997. A Comparative Study on Feature Selection in Text Categorization. *Proc. International Conference on Machine Learning*.