Student Speech Act Classification Using Machine Learning

Travis Rasor, Andrew Olney, Sidney D'Mello

Institute for Intelligent Systems, 202 Psychology Building, University of Memphis, Memphis TN, 38152, USA
tlrasor@memphis.edu, aolney@memphis.edu, sdmello@memphis.edu

Abstract

Dialogue-based intelligent tutoring systems use speech act classifiers to categorize student input into answers, questions, and other speech acts. Previous work has primarily focused on question classification. In this paper, we present a complimentary speech act classifier that focuses primarily on non-questions, which was developed using machine learning techniques. Our results show that an effective speech act classifier can be developed directly from labeled data using decision trees.

Introduction

Intelligent tutoring systems (ITS) are artificially intelligent computer programs that seek to be as effective instructors as human tutors (Sleeman & Brown, 1982; VanLehn, 2006; Woolf, 2009). Within this larger research program, a group of researchers have attempted to make ITS interactions more naturalistic and conversational. In order to accomplish this goal, researchers have analyzed corpora of human-human tutorial dialogues to better understand both individual dialogue acts and patterns of acts that occur in human tutoring (Graesser & Person, 1994; Graesser, Person, & Magliano, 1995; Litman & Forbes-Riley, 2006; Person, Lehman, & Ozbun, 2007; Boyer et al., 2009; Chi, Roy & Hausmann, 2008; Chi, Siler & Jeong, 2001; Lepper & Woolverton, 2002). The linguistic unit of analysis in these studies is a speech act or dialogue act which abstracts away from the content of an utterance to its underlying communicative function, e.g. question, assertion, or directive (Searle, 1969).

We are currently engaged in building an ITS, called Guru, to emulate an expert human tutor for biology (D'Mello, Olney, & Person, 2010). To date we have collected 50-hours of expert human tutorial dialogues, which we have transcribed, coded into dialogue acts, and analyzed (D'Mello et al., in press). Our analyses revealed that state transition networks constructed from sequences of tutor and student dialogue acts can capture a large portion of the observed behavior in our expert human tutoring corpus. The collaborative dialogue patterns in our corpus reveal a rich interleaving of initiative between student and tutor, yielding a finer coding of non-question dialogue acts. These state transition networks can be used as the backbone for an ITS, as long as the student's utterance can first be classified to the same dialogue act coding scheme. In this paper we outline a machine learning based approach to dialogue act classification for ITS.

Related Work

A substantial amount of research has addressed dialogue act tagging over the past two decades (Fisel, 2007; Olney, Graesser, & Person, 2010; Samuel, Carberry, & Vijay-Shanker, 1998; Stolcke et al., 2000; Verbree, Rienks, & Heylen, 2006; Sridhar, Bangalore, & Narayanan, 2009; Di Eugeno et al., 2010). The plurality of taxonomies, the differences amongst available features, and the techniques used have yielded a variety of approaches. Verbee et al. (2006) examined the features used by 16 dialogue act tagging studies and identified 24 features that have been previously used. While an extensive discussion of these features is outside the scope of the present paper, the features fall loosely into four categories: word based (e.g. cue phrase), acoustic (e.g. prosody), surface form (e.g. sentence length), and context (e.g. previous dialogue act). Not all 24 features are meaningful for all applications; for example, Sridhar et al. (2009) make use of acoustic features in speech input, e.g. prosody, which are not available in the present text-based system.

Olney et al., (2003) developed a speech act classifier (SAC) that focuses primarily on question classification according to the scheme developed by Graesser and colleagues (Graesser, & Person, 1994). That SAC contained 16 categories for questions, two categories for metacommunicative acts, and one for an assertion. The SAC's emphasis on question classification was highly aligned to the dialogue model of the AutoTutor system in which it was embedded (Graesser, Olney, Haynes, & Chipman, 2005) In that system, dialogue is tutor-driven, such that the most natural student dialogue act at any time is a statement in response to a question posted by the tutor. Questions therefore represent a major shift in initiative, and students' contributions must be correctly classified in order to discriminate between questions and other speech acts and respond to the right category of question, e.g. causal question rather than judgmental question. Assertions in the AutoTutor system are not differentiated, i.e. there is only one type of statement in this coding scheme.

The choice of classification method is also important and directly related to the features selected. Fisel (1997) looked at several different methods, including decision-trees, Hidden-Markov models and n-grams, Bayesian classifiers, neural networks, transformation-based learning and more. Although many of these techniques could be effective, inspection of our tutorial dialogue transcripts (described below) led us to conclude that many of the distinguishing characteristics of our dialogue acts were word-based and semantically-based, rather than driven by the surrounding dialogue acts. Hidden Markov models and n-grams appear less suitable for our purpose as frequently their use focuses on sequences of dialogue acts. Instead we focus on the following five techniques from machine learning that have been widely effective in a number of applications (Wu et al., 2007): ZeroR, NaiveBayes, LogitBoost, J48, and IBk. These machine learning techniques and their parameters for the present study are further discussed in the Methods section.

Corpus

Our expert tutoring corpus was created by collecting observations of naturalistic one-to-one expert tutoring (Olney et al., 2010; Person et al., 2007). Ten expert math and science tutors were recruited to participate in the project. The following criteria were used to define expertise: all have a minimum of five years of tutoring experience, a secondary teaching license, and a degree in the subject that they tutor. All of the students in our study had genuine need of tutoring and were either sought tutoring or were recommended for tutoring by school personnel. The content of the corpus came from tutoring sessions from a number of different math and science courses. Guru (the ITS we are developing) is designed specifically for biology tutoring rather than a variety of science courses.

Fifty, one-hour, one- on-one tutoring sessions with these expert tutors were videotaped, transcribed, and coded into dialogue move categories. Taxonomies, or coding schemes, were developed to classify every tutor and student dialogue move (D'Mello et al., 2010), but only the student scheme will be described here, since only student moves are unknown and require classification. A student dialogue move was a dialogue act, an action, or a qualitative contribution made by a student. A taxonomy using 16 categories was developed for classification of all student dialogue moves. Some move categories capture the quality of student dialogue move, e.g., correctness of answer, while others capture types of questions, conversational acknowledgments, and student actions, such as reading aloud. The Student Dialogue Move Scheme is presented in Table 1.

Table 1. Student Dialogue Moves

Student Move	Example
Acknowledgment	Yes, ma'am.
Common Ground Q.	The dipoles?
Correct Answer	Dipoles have two poles.
Error Ridden Answer	Poles.
Gripe	I might as well not pay attention.
Knowledge Deficit Q.	Well, what's a Dipole?
Metacomment	I don't know.
Misconception	So dipoles have no polarity.
No Answer	
Offtopic Conversation	It could be.
Partial Answer	Dipoles have more than one pole
Read Aloud	
Social Coordination A.	Hand me the calculator.
Student Works Silently	
Think Aloud	Uh, dipoles are like little magnets which are from before
Vague Answer	Yeah, polarity

Note. Q = Question. A. Action

Four trained judges coded the 50 transcripts on the dialogue move schemes, for which Cohen's Kappas were computed to determine the reliability of their judgments. A Kappa score of .88 was obtained for the Student Move Scheme; this is indicative of excellent reliability. In all, 47,296 dialogue moves were coded. Most classification methods require thousands of data-points to create robust decision models (Hämäläinen & Vinni, 2006). By having access to such a large database, we are not limited by these constraints.

As mentioned above, not every coded student dialogue act is a speech act. Some are nonverbal student behaviors, such as "Student Works Silently" and "No Answer." Prosodic features are likewise difficult to interpret from text input. Therefore the number of categories needed to implement a runtime ITS are fewer than those required to code the face-to-face interaction in our tutoring corpus. Likewise, the intent of an ITSs SAC is different from that of corpus coding scheme. An ITS could use the SAC as a dispatch module, routing the problem of handling the student's input to the relevant module. For a question category that relevant module would be a question answering module, and for a statement/answer category the relevant module would be an answer assessment module. Under this conceptualization, a coarser scheme is sufficient in an ITS because other components will further refine the categories. Thus the five categories of interest from our original scheme are Metacomment (representing

838 dialogue acts), Common Ground Question (1060 dialogue acts), Knowledge Deficit Question (515 dialogue acts), Acknowledgement (5794 dialogue acts), and Answer (5129 dialogue acts). Note that Answer represents a collapsing of answer types of varying quality such as correct or partially-correct answers, vague or error-ridden answers, and no-answers.

Our five category scheme is highly complementary to the previous work presented in (A. Olney et al., 2003). That work makes finer distinctions in question classification, essentially partitioning the present "Knowledge Deficit Question" into 16 subtypes of question. Similarly, the present "Metacomment" category is represented in the (A. Olney et al., 2003) scheme as "Metacognitive" and "Metacommunicative." Conversely, the present classifier extends that previous work by subdividing "Contribution" into "Answer" and "Acknowledgement."

Method

WEKA (Hall et al., 2009) is a tool kit for various machine learning and visualization algorithms written in Java. WEKA is widely used in multiple applications, from bioinformatics to network security (Frank, Hall, Trigg, Holmes, & Witten, 2004; Panda & Patra, 2008), and is extremely easy to use with dialogue analysis due to its included packages. Because of the large memory requirements of our data set, this project was run on a 64-bit build of WEKA.

Many of the classifications algorithms used in machine learning require numerical or categorical input and cannot accept text-strings. WEKA has numerous packages for pre-processing data, including string-to-numeral transformation filters needed to use most classifying algorithms using the StringToWordVector package. This filter can be configured to do tf-idf weighting. The tf-idf (term frequency–inverse document frequency) weight is commonly used in text mining applications due to its good performance and is a statistical measurement to determine how important a word is in a corpus. We used tf-idf to generate features from student utterances based on our hypothesis that word level features are the most discriminating features for this dataset.

The WEKA filter transformed the text into 2300 weighted numerical features for use with its classifier packages. This data set was evaluated with five different classification algorithms, representing several different, common approaches to classification, each of which was validated using 10-fold cross-validation. First, ZeroR, a simple rule-based classifier that classifies all dialogue moves into the most prevalent classification. Second, NaiveBayes, which assumes feature independence and uses a Bayes rule for classification. Third, IBk, a classifier that uses a version of the k-nearest neighbor (k-NN) algorithm which was configured with *k=10*. j48, an open source implementation of the C4.5 algorithm, was also used. It builds decision trees designed to give the maximum discrimination between data in a training set. j48 is particularly interesting because its decision-tree allows manual inspection and simpler visualization. Lastly, we used LogitBoost, which uses a boosting algorithm to create a strong learner out of a collection of small weak learners. These weak learners used a decision stump for their classifiers.

In addition to the tf-idf feature-based models, we constructed another model based on the best tf-idf model but with features suggested by the analysis in (Verbee et al., 2006). For our corpus, the most relevant features were sentence length (expressed in characters), the presence of a question mark, and the previous tutor speech act. Results for the five basic models and the augmented model are presented in the next section.

Results

The overall results for the five different classifiers using tf-idf features are presented in Table 2. The numbers presented show the percent of correctly classified instances aggregated across all categories.

Table 2. Classifier results

Classifier	Percent Correct
ZeroR	45.72
NaiveBayes	66.51
IBk	78.49
j48	79.33
LogitBoost	77.84

The ZeroR classifier, a knowledge poor baseline, classified all dialogue moves into Acknowledgment (45% of the corpus). Three of the five classification methods, IBk, j48 and LogitBoost, had moderate results, with j48 performing the best by a small margin. Thus these three models have almost twice the accuracy of the majority class baseline. The naïve Bayes classifier's performance was intermediate. The success of the three classification models suggests that groupings of features are needed to produce good results. NaiveBayes, in comparison, considers each feature as independent from other features. This independence assumption is violated because this grouping of features required for good classification demonstrates that the features in the data are not independent, leading to the poor performance of the classifier.

Class-level performance for j48, the best classifier, is presented in Table 3. j48 is proficient at classifying Acknowledgments and Answers, with moderate accuracy, precision and recall rates. The other three, however, are not as accurately classified using this approach, thereby leaving room for some improvement. The scores for Common Ground and Knowledge Deficit Questions are particularly low, with TP-rates of only 13% and 4% respectively. Additionally, unlike the other categories, their precision and recall rates are imbalanced as well, with substantially higher precision than recall in both cases.

Table 3. Accuracy statistics for the j48 classifier

Category	**TP Rate**	**FP Rate**	**Precision**	**Recall**
Common Ground Question	0.039	0.006	0.344	0.039
Knowledge Deficit Question	0.125	0.005	0.452	0.125
Metacomment	0.520	0.025	0.586	0.520
Acknowledgment	0.949	0.111	0.878	0.949
Answer	0.862	0.197	0.705	0.862
Weighted Avg.	**0.778**	**0.122**	**0.741**	**0.778**

LogitBoost and IBk both also classify Answers and Acknowledgements reasonably well, but their classification of the other three categories is worse than j48. The overall Kappa statistic for the j48 classifier was 0.685.

The j48 classifier produced a large decision tree with over 400 leaves, so a complete description of the tree is not practical due to space constraints. However, three leaves account for 51% of the total mass (each leaf has over a thousand classifications), and an inspection of the branches leading to these leaves yields some insight into the decision making process of the tree. In a nutshell, the decision tree that j48 creates looks to see if certain words are contained in the dialogue move. For each of the classifications, therefore, a list of excluded and included words can be created. Table 4 presents the excluded words that the first heavy leaf (representing 4068 Answers) uses to classify a dialogue move. In other words, if a move is lacking all of these words, it is classified as an Answer. Table 4 also includes the frequency with which each word appears in the corpus.

Table 4. List of excluded words in the first heaviest leaf of the j48 decision-tree, term-frequency included

Excluded Words (word term-frequency)			
oh (869)	know (259)	how (105)	is (759)
mmm (115)	right (1158)	where (60)	are (137)
sure (47)	sure (47)	wait (47)	you (913)
hmm (2019)	mm (1998)	if (151)	really (64)
i (1675)	ok (1836)	got (130)	yup (21)
want (40)	why (49)	dont (292)	which (107)
makes (38)	maam (224)	do (387)	like (470)
me (48)	yeah (1157)	thing (76)	huh (186)
wouldnt (56)	alright (529)	was (188)	yes (442)
isnt (42)	okay (335)	height (76)	does (53)

An important characteristic of this list it is mostly devoid of domain-specific content words (e.g., “speed”, “cell”, “mitosis”, “hydrogen”). The only exception in this leaf was the word “height” from math and physics tutoring. This trend of few or no content words continues as we traverse through the tree, a beneficial result

The second heaviest leaf (representing 1669 Acknowledgments) relies on both the exclusion and inclusion of words to determine its classification. According to this branch, the single included word, “OK,” is very diagnostic of an Acknowledgement. The words excluded by this branch are also very distinct. The words excluded show that an Acknowledgment is recognized by the exclusion of personal pronouns, such as “I” in this leaf, Metacognitive words, “know” and “remember,” and question initiators, such as “what.” Table 5 presents the word list for the second heaviest leaf.

Table 5. List of words used by second heaviest leaf of the j48 decision-tree

Category	Word (term-frequency)
Excluded Words	know (259), remember (86), what (302), mm (1998), i (1675), so (1164)
Included Words	ok (1836)

The third heaviest leaf (representing 1982 Acknowledgments) again uses both the inclusion and exclusion of certain words. Interestingly, the word list for this leaf shares most of its words with the prior leaf. For this leaf, though, the text “mm”, previously on the excluded list, is now included. Since these latter two leaves represent around a quarter of the total number of classifications, the dual roles of words like “mm” suggest that the surrounding words in a dialogue move can greatly affect its classification. Additionally, neither of these leaves have any content words, continuing to suggest that the features necessary for good classification of most categories are largely independent of domain. Table 6 displays the word list for this leaf.

Table 6. List of words used by third heaviest leaf of the j48 decision-tree

Category	Word (term-frequency)
Excluded Words	know (259), remember (86), what (302)
Included Words	mm (1998)

Our second set of results is based on the best performing model, j48. Recall that this augmented model includes the tf-idf features from the first set, plus sentence length, previous tutor move, and the presence of a question mark. Classification accuracy for the augmented model increases to 85%, a 6% increase, with a somewhat higher Kappa statistic of 0.775. Table 7 contains the statistics for this new classifier.

Table 7. Accuracy statistics for the revised j48 classifier

Category	TP Rate	FP Rate	Precision	Recall
Common Ground Question	0.650	0.031	0.620	0.650
Knowledge Deficit Question	0.375	0.011	0.559	0.375
Metacomment	0.509	0.014	0.718	0.509
Acknowledgment	0.945	0.053	0.940	0.945
Answer	0.896	0.097	0.839	0.896
Weighted Avg.	0.858	0.063	0.853	0.858

It is notable that the overall percent correct increased by about 6% without reducing classification accuracy in the most accurate categories (See Table 3). Thus this augmented model showed very similar results for the Acknowledgements, Metacomments, and Answers categories, but provided a much better model for the previously less accurate classifications of Common Ground Questions and Knowledge Deficit Questions. This performance improvement is most dramatic in the Common Ground Question category where classification performance was 16 times better than in the prior model.

The differences in results between the first and second j48 models suggests that one of our initial hypothesis, that word based features were highly diagnostic of dialogue acts, was true for some dialogue act categories but not others. Common Ground Questions, in particular, appear to be marked to a greater extent by superficial features such as the presence of a question mark and a shorter sentence length, e.g. “This one?” Knowledge Deficit questions are more difficult to categorize, however they do share some of the same features of Common Ground Questions.

These results look similar to those in other efforts, such as those of Sridhar et al. (2009); however, the corpus used in our research is different than those used previously. In the cases of corpora that necessarily include features unavailable to our system (prosody, etc.), these differences are largely insurmountable. Other studies, such as the use of Latent Semantic Analysis (LSA) and an IBk classifier by Di Eugeno et al. (2010), are more similar and we are interested in future possible performance comparisons.

Conclusions

We presented a machine learning approach to constructing a speech act classifier from an annotated tutorial dialogue corpus. Although the overall accuracy of the classifier is moderately high, the accuracy for the less frequent categories is somewhat lower. For a runtime ITS, the accuracy for all categories would ideally be in the 80-90% range. Our approach focusing primarily on non-questions is complementary to that of Olney and colleagues (2003) which was more focused on questions and an area of future exploration is in the combination these classifiers.

Some possible limitations of this study relate to the transferability of the features used to an ITS. Since the features used in this study were derived from transcripts of human-to-human tutorial dialogue, it may be the case that the transcribed speech of students differs significantly from the typed input of students using the ITS. Furthermore, the punctuation and word length features used by the augmented model may not be properly calibrated for typed student input. In future work we will analyze student sessions with the Guru ITS to determine if the accuracy of the augmented j48 classifier is preserved.

Additionally, future work should determine what the cost is to the system for incorrectly classifying a speech act to one of the other categories. These costs may differ depending on the categories in question, e.g. is it worse to misclassify an acknowledgement as a question or an answer? This cost could be a valuable tool for fine tuning the precision/recall curve for our classifier.

Acknowledgments

The research reported here was supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A080594 to the University of Memphis. The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Department of Education.

References

Boyer, K., Young, E., Wallis, M., Phillips, R., Vouk, M., & Lester, J. (2009). Discovering Tutorial Dialogue Strategies with Hidden Markov Models. In V. Dimitrova, R. Mizoguchi, B. Du Boulay & A. Graesser (Eds.), *Proceedings of the 14th International Conference on Artificial Intelligence in Education* (pp. 141 - 148). Amsterdam: IOS Press.

Chi, M., Roy, M., & Hausmann, R. (2008). Observing tutorial dialogues collaboratively: Insights about human tutoring effectiveness from vicarious learning. *Cognitive Science, 32*(2), 301-341.

Chi, M., Siler, S., Jeong, H., Yamauchi, T., & Hausmann, R. (2001). Learning from human tutoring. *Cognitive Science, 25*(4), 471-533.

D'Mello, S., Hays, P., Williams, C., Cade, W., Brown, J., & Olney, A. (2010). Collaborative Lecturing by Human and Computer Tutors In J. Kay & V. Aleven (Eds.), *Proceedings of 10th International Conference on Intelligent Tutoring Systems* (pp. 609-618). Berlin / Heidelberg: Springer.

D'Mello, S., Olney, A. M., & Person, N. (2010). Mining Collaborative Patterns in Tutorial Dialogues. *Journal of Educational Data Mining*, *2*(1), 1-37.

Di Eugenio, B., Xie, Z., & Serafin, R. (2010). Dialogue act classification, instance-based learning, and higher order dialogue structure. *Dialogue & Discourse, 1*(2), 81-124. doi: 10.5087/dad.2010.00

Fisel, M. (2007). Machine learning techniques in dialogue act recognition. *Estonian Papers in Applied Linguistics*, *3*, 117-134.

Frank, E., Hall, M., Trigg, L., Holmes, G., & Witten, I. H. (2004). Data mining in bioinformatics using Weka. *Bioinformatics*, *20*(15), 2479 -2481. doi:10.1093/bioinformatics/bth261

Graesser, A. C., Olney, A. M., Haynes, B. C., & Chipman, P. (2005). AutoTutor: A Cognitive System That Simulates a Tutor That Facilitates Learning Through Mixed-Initiative Dialogue. In C. Forsythe, M. L. Bernard, & T. E. Goldsmith (Eds.), *Cognitive Systems: Human Cognitive Models in Systems Design* (pp. 177-212). Mahwah, NJ: Erlbaum.

Graesser, A. C., & Person, N. K. (1994). Question Asking during Tutoring. *American Educational Research Journal*, *31*(1), 104–137.

Graesser, A. C., Person, N. K., & Magliano, J. P. (1995). Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied Cognitive Psychology*, *9*, 1-28.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software. *ACM SIGKDD Explorations Newsletter*, *11*(1), 10. doi:10.1145/1656274.1656278

Hämäläinen, W., & Vinni, M. (2006). M.: Comparison of machine learning methods for intelligent tutoring systems. *Proceedings of the 8th International Conference in Intelligent Tutoring Systems.* 525-534. doi:10.1007/11774303_52

Lepper, M., & Woolverton, M. (2002). The wisdom of practice: Lessons learned from the study of highly effective tutors. In J. Aronson (Ed.), *Improving academic achievement: Impact of psychological factors on education* (pp. 135-158). Orlando, FL: Academic Press.

Litman, D., & Forbes-Riley, K. (2006). Correlations between dialogue acts and learning in spoken tutoring dialogues. *Nat. Lang. Eng.*, *12*(2), 161–176.

Olney, A., Louwerse, M., Mathews, E., Marineau, J., Hite-Mitchell, H., & Graesser, A. (2003). Utterance Classification in AutoTutor. *Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing* (pp. 1-8). Philadelphia: Association for Computational Linguistics.

Olney, A., Graesser, A. C., & Person, N. K. (2010). Tutorial Dialog in Natural Language. In R. Nkambou, J. Bourdeau, & R. Mizoguchi (Eds.), *Advances in Intelligent Tutoring Systems*, Studies in Computational Intelligence (Vol. 308, pp. 181-206). Berlin: Springer-Verlag.

Panda, M., & Patra, M. R. (2008). A Comparative Study of Data Mining Algorithms for Network Intrusion Detection. *2008 First International Conference on Emerging Trends in Engineering and Technology* (pp. 504-507). Presented at the 2008 1st International Conference on Emerging Trends in Engineering and Technology (ICETET), Nagpur, Maharashtra, India. doi:10.1109/ICETET.2008.80

Person, N. K., Lehman, B., & Ozbun, R. (2007, July). *Pedagogical and Motivational Dialogue Moves Used by Expert Tutors*. Presented at the 17th Annual Meeting of the Society for Text and Discourse. Glasgow, Scotland.

Samuel, K., Carberry, S., & Vijay-Shanker, K. (1998). Dialogue act tagging with Transformation-Based Learning. *Proceedings of the 17th international conference on Computational linguistics.* (pp. 1150-1156). doi:10.3115/980432.980757

Searle, J. R. (1969). *Speech acts: an essay in the philosophy of language.* Cambridge University Press.

Sridhar, V. K. R., Bangalore, S., & Narayanan, S. (1998). Combining lexical, syntactic and prosodic cues for improved online dialog act tagging. *Computer Speech & Language,* 23(4), 407-422.

Sleeman, D., & Brown, J. (Eds.). (1982). *Intelligent tutoring systems*. New York: Academic Press.

Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., et al. (2000). Dialogue Act Modeling for Automatic Tagging and Recognition of Conversational Speech. *Computational Linguistics*, *26*(3), 339-373. doi:10.1162/089120100561737

VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education., 16*(3), 227-265.

Verbree, D., Rienks, R., & Heylen, D. (2006). Dialogue-act tagging using smart feature selection: results on multiple corpora. *IEEE Spoken Language Technology Workshop, 2006.* 70-73. doi: 10.1.1.77.3645

Woolf, B. (2009). *Building intelligent interactive tutors*. Burlington, MA: Morgan Kaufmann Publishers.

Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., et al. (2007). Top 10 algorithms in data mining. *Knowledge and Information Systems*, *14*(1), 1–37. doi:http://dx.doi.org/10.1007/s10115-007-0114-2