# A Tutorial Dialogue System for Real-Time Evaluation of Unsupervised Dialogue Act Classifiers: Exploring System Outcomes

Aysu Ezen-Can and Kristy Elizabeth Boyer

Department of Computer Science, North Carolina State University
{aezen,keboyer}@ncsu.edu

**Abstract.** Dialogue act classification is an important step in understanding students' utterances within tutorial dialogue systems. Machine-learned models of dialogue act classification hold great promise, and among these, unsupervised dialogue act classifiers have the great benefit of eliminating the human dialogue act annotation effort required to label corpora. In contrast to traditional evaluation approaches which judge unsupervised dialogue act classifiers by accuracy on manual labels, we present results of a study to evaluate the performance of these models with respect to their performance within end-to-end system evaluation. We compare two versions of the tutorial dialogue system for introductory computer science: one that relies on a supervised dialogue act classifier and one that depends on an unsupervised dialogue act classifier. A study with 51 students shows that both versions of the system achieve similar learning gains and user satisfaction. Additionally, we show that some incoming student characteristics are highly correlated with students' perceptions of their experience during tutoring. This first end-to-end evaluation of an unsupervised dialogue act classifier within a tutorial dialogue system serves as a step toward acquiring tutorial dialogue management models in a fully automated, scalable way.

**Keywords:** Dialogue Act Classification, Unsupervised Machine Learning, Tutorial Dialogue Systems

## 1 Introduction

Today's tutorial dialogue systems are effective [22], yet they still aspire to improve by supporting the flexible natural language interactions of the most effective human tutors [9, 1]. However, improving natural language interactions is a challenging task because there is extensive engineering effort required to build a full natural language dialogue pipeline [4]. We have seen an upsurge of interest in improving natural language understanding in tutorial dialogue for a variety of domains such as physics (AutoTutor [17], Why2Atlas [23], Andes [24], ITSPOKE [16] and Rimac [13]), the circulatory system (CIRCSIM-Tutor [6]), electricity and electronics (BEETLE-II [4]), and programming (ProPL [14], iList [8]).

Dialogue act classification is one of the most useful mechanisms for understanding student utterances. Dialogue acts aim to capture the "act" underlying an utterance such as asking a question, making a statement, and acknowledgement [19]. Classifying student dialogue acts accurately may support more effective tutoring, as the whole pipeline of dialogue management depends on them. For example, a negative feedback from the student such as "I am not following" may be followed by remedial help from the tutor, in contrast to a statement of plan such as "I am not working on that method yet" which may be followed by an acknowledgment from the tutor.

The task of automatic dialogue act classification has been studied extensively in the literature, mostly within supervised machine learning [19, 3]. However, supervised classification is labor-intensive as it requires engineering dialogue act taxonomies and labeling corpora before training classifiers. As an alternative, unsupervised classifiers have gained attention recently. These models build groupings of student utterances directly from the data. However, to date, no deployed dialogue system has utilized an unsupervised dialogue act classifier; rather, researchers have evaluated the performance of unsupervised models as standalone components by comparing to manual labels [18, 7]. The downside of this approach is that expecting a fully data-driven model to replicate a human annotation scheme may not capture how well that data-driven model will perform in an end-to-end deployment. Therefore, evaluating unsupervised dialogue act models without comparing to manual annotations and within their usage environment is of the utmost importance.

We have implemented a tutorial dialogue system that can be utilized to compare the performance of two different dialogue act classifiers within a real-time system. We trained an unsupervised dialogue act model on a corpus of human tutorial dialogue in the domain of introductory computer science, and for comparison we trained a supervised model. We hypothesized that the unsupervised dialogue act model, which relies on hierarchical clustering and uses no manual labels during model training, would support equal or better student learning than the supervised dialogue act model, which relies on a decision tree classifier and represents a state-of-the-art, highly accurate dialogue act classifier that agrees with human annotations 89.6% of the time. Experimental results with 51 students show that unsupervised and supervised dialogue act models indeed achieved similar performance for supporting learning gains and user satisfaction.

Additionally, we conduct a PARADISE dialogue analysis in which the relative contribution of various factors to a system's overall performance is investigated [25]. We conduct regression analyses to determine factors affecting the outcomes of the system. The results show that students' perceptions are significantly associated with system outcomes such as how involved students become during the tutoring session and how difficult students feel that the tasks are.

## 2 System Design

The primary goal of the study is to evaluate an unsupervised dialogue act classifier in its intended usage environment and to compare it to a supervised classifier within a tutorial dialogue system. This section describes two versions of a tutorial dialogue system: one that implements an *unsupervised* dialogue act classifier and one that uses a *supervised* dialogue act classifier. A screenshot is shown in Fig. 1.

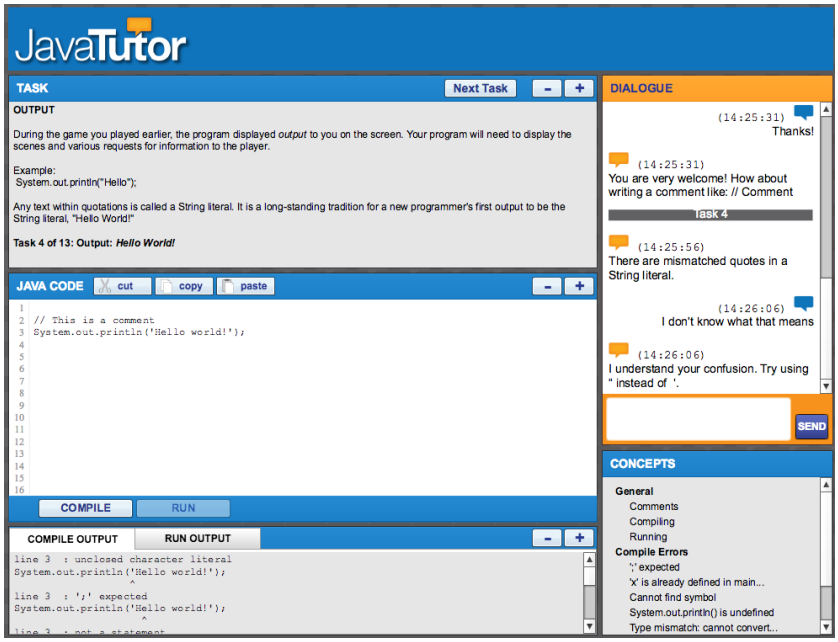


**Fig. 1:** Screenshot from the tutorial dialogue system.

### 2.1 System Architecture Overview

Both versions of the tutorial dialogue system depend on the same pipeline. First, dialogue act classification takes place to interpret student's words. Then, a code analysis module utilizes regular expressions to identify errors in student's code. The output of these modules is used in the generation module, where the tutor move is determined. Fig. 2 presents the architecture diagram of the dialogue system.

### 2.2 Dialogue Act Classification

In this section, we explain how the dialogue act classification task is handled within the system. Both dialogue act classifiers were trained on a corpus of 2,417 student utterances collected in a computer-mediated environment for teaching Java programming language within a study which has been detailed in previous

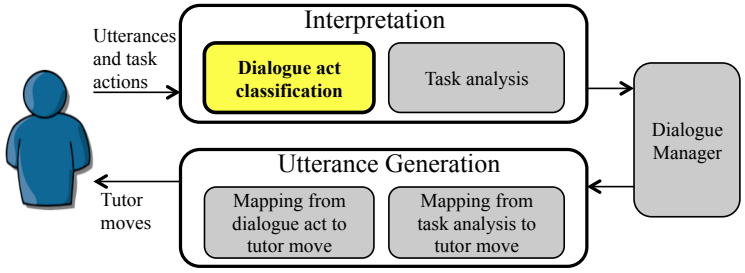


**Fig. 2:** System architecture diagram.

publications [10]. First, the features are extracted, which are then used as inputs to the dialogue act classifiers (supervised or unsupervised).

**Features.** For classification of dialogue acts, we extract two sets of features: textual and task-related. These are the same set of features extracted both for training (building the dialogue act classifiers) and testing (real-time classification of dialogue acts). The textual features are solely extracted from the student utterances. These include unigrams and bigrams of both tokens (words and punctuation) and part-of-speech tags.

While experimenting with the dialogue interactions, students are asked to complete the tasks provided. To satisfy the requirements, the students write and test their programming code. The task-related features are extracted from the task events that occur in real-time throughout the tutoring. There are three task-related features used within the system. Two of them are utilized within dialogue act classification, and one of them is used to provide remedial help. We use the latest task action (compile, run, writing a message to the tutor) and its result (success, error, begin, stop) to improve the dialogue act classification task. In addition, we use regular expressions to compare the student's code to the solutions of previous students to understand whether the student's code has an error.

**Supervised Dialogue Act Classification.** For supervised dialogue act classification, we use an off-the-shelf decision tree classifier from Weka [11] and train it on dialogue act tags [20]. This tagging scheme consists of 18 student dialogue act labels for the portion of the task that the system implements (*Greeting, Extra-Domain Question, Ready Question, Confirmation Question, Direction Question, Information Question, Observation, Correction, Understanding Feedback, Not Understanding Feedback, Explanation, Other, Yes-No Answer, Extra-Domain Answer, Answer, Positive Feedback, Ready Answer, Acknowledgement*), with Cohen's Kappa of $\kappa = 0.87$ (89.6 % agreement) showing high reliability [20].

**Unsupervised Dialogue Act Classification.** As for the unsupervised dialogue act classification, we utilize a hierarchical clustering approach that assigns utterances to individual clusters initially and merges the most similar two clusters in each iteration until the hierarchy is completed by having one large cluster. By examining the whole hierarchy, we qualitatively chose the stopping point where the groupings of utterances make sense. The number of clusters determined at this stopping point would have been the number of clusters to

be used if no comparison with a supervised classifier were to take place. However, our goal is to provide similar conditions to both of the supervised and unsupervised classifiers. In order to make sure that the number of clusters are not different from the number of manual labels, we merged the clusters that are sparse and used the same number of clusters (18) as the number of manual tags. The details of this unsupervised framework are beyond the scope of this paper due to space limitations but have been fully described in a prior publication [7].

### 2.3 Tutorial Policies and Utterance Generation

Having obtained the machine-learned models, we authored policies which govern system moves given the output of the classifier. For the supervised version, we authored moves for each manual label (e.g., Question; Negative Feedback) and for the unsupervised version, we crafted tutor moves to each cluster (with clusters interpreted qualitatively). These policies were created to be as similar as possible while providing contextually appropriate responses to the dialogue act.

When students interact with the system, it calls upon the trained dialogue act classifier to classify each new student utterance. Then based upon the dialogue policy, the system chooses tutor moves on the fly. Additional features provided for generating tutorial moves include the output of automated code analysis using regular expressions, which compares student's code to previous correct student solutions to understand if there are any errors. The output of this code analysis is used to fill slots within the tutorial move templates. Example tutor moves are depicted in Fig. 3.

In addition to the dialogue act classifiers' support, the system takes initiative to provide feedback when needed; that is, it is not constrained to respond to student requests [10]. This task-based feedback is the same across both the supervised and unsupervised versions of the system.

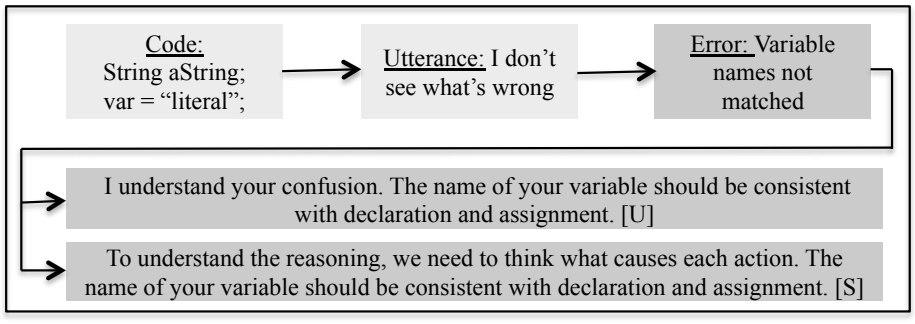


**Fig. 3:** Sample tutor moves with sample code and utterance. [U] indicates the unsupervised dialogue act classifier move and [S] indicates the supervised classifier move.

In cases where there are multiple code errors returned by the regular expression module, we apply prioritization of errors. We provide the tutor move that

corresponds to the latest error, which is determined by the line number of the error. In addition, less priority is given to the task errors, such as variable declaration not found although the task description asks for a variable description, than syntax errors such as trying to declare a variable but could not successfully do so syntactically.

In addition to the classifiers, we incorporate a hybrid approach which makes use of simple rules. The motivation for utilizing rules is that for some phenomena such as greetings and thanking, the responses are clear; therefore the classifiers do not need to be run. If a student utterance falls into one of these categories, the approach returns the corresponding moves from the rules. For more complex utterances, we run the dialogue act classifiers.

## 3 Evaluation

We have hypothesized that our unsupervised dialogue act classifier will support equal or better student learning and satisfaction compared to a state-of-the-art supervised dialogue act classifier. To test this hypothesis, we conducted a study with two conditions: 24 participants were in the supervised condition and 27 participants were in the unsupervised condition. The students (12 female, 39 male) were drawn from a university-level first-year-engineering class and participated as part of an in-class activity.

The students were randomly assigned to the different versions of the system. Their interactions with the system were logged. They took a pre-test and pre-survey, and after tutoring completed a post-survey and post-test identical to the pre-test. The pre-survey consisted of several widely used measures including goal orientation [21], general self-efficacy [2], and confidence in learning computer science and programming [15]. We investigate contributions of these measures to system outcomes in Section 4.

Students in both conditions received a statistically equivalent number of tutor messages: 35.2 in the supervised condition and 38.7 in the unsupervised condition. To compare the effectiveness of the two systems, we use multiple metrics from the surveys and tests. First, we consider usability as indicated by a set of ten items on the post-survey (e.g., 'The tutoring system was knowledgeable about programming.', 'The tutoring system was supportive.'). There was no significant difference between two versions of the system with respect to usability, with both sets of users rating the system 2.99 out of 5 ($p$=0.5; stdev=0.8). The second metric we compare is students' perception of how effective the tutor feedback was. This measure is taken from fourteen post-survey questions (e.g., 'It was easy to learn from the tutor's feedback.','I paid attention to the tutor's feedback.'). Similar to the usability questions, the tutor feedback ratings in supervised and unsupervised versions were not significantly different ($p$=0.4; stdev=0.9).

Finally, we compare the systems in terms of learning gain. The learning gains were significantly positive in both conditions ($mean_{sup}$=0.12, $p$=0.05, stdev=0.21; $mean_{unsup}$=0.14, $p$=0.0009, stdev=0.18) and these means were not

significantly different ($p$=0.3). As hypothesized, the results indicate that the unsupervised dialogue act classifier supported statistically equivalent learning gain and user satisfaction as the state-of-the-art supervised models, while the unsupervised model required only a small fraction of the manual labor (for interpreting clusters) compared to the supervised model (for which extensive labeling of the corpus was required). Next we build descriptive regression models to examine the relationships between pre-measures and post-measures.

## 4 Evaluating System Outcomes

Since the unsupervised and supervised dialogue act classifiers produced comparable tutorial dialogue interactions, we aggregated the data from the two conditions in order to explore the factors affecting the outcome of the system. We leveraged the dialogue system evaluation framework PARADISE [25] and built multiple regression to reveal relationships between student characteristics and fine-grained logs from the tutorial dialogue interactions (the predictors), and students' perceptions of the tutorial dialogue (the response variables).

We built one multiple regression model for predicting each post-measure of interest from the surveys or tests (endurability, curiosity, felt involvement, focused attention, task difficulty and post-test score), with the same set of independent variables each time that include pre-survey, pre-test, number of utterances written by the student, number of total logged activities, number of compile/run events, number of program content changes logged, number of compile errors and number of tutor messages received. The goal was not to obtain accurate predictive models necessarily, but to investigate *descriptive* models that indicate the aspects of the learners or of the interactions that are significantly associated with outcomes.

As shown in Fig. 4, the outcomes of the system were related not only to the effectiveness of the tutoring, but also heavily to incoming perceptions of the students. We present features with significance $p < 0.05$ and the post-measures they predict. The endurability category had four post-survey items which measure the extent to which the students considered the tutoring session worthwhile and rewarding. The felt involvement category consisted of three survey items measuring how much the students were involved in the task, and the focused attention category involved seven questions about how much the students were focused on the task. Finally, the curiosity category measured how interested the students were with the system using three questions.

The results show that some predictors were correlated with multiple system outcomes. For example, the confidence that students have in computer science was significantly predictive of endurability, curiosity, felt involvement and task difficulty. Similarly, the extent to which students find computer science useful was significantly associated with all presented post-measures except task difficulty, highlighting the fact that the perceptions of students are related to how they feel about the tutoring system. Learning goal orientation (mastery vs. performance) was measured with twelve pre-survey questions, and the models showed that

students that were willing to face challenging tasks also felt more involved with the learning task. Finally, students who aimed to score higher than other students felt that the tasks were more difficult.

**Descriptive Linear Regression Models**

| | |
|---|---|
| **Endurability** | = 0.4028 * CS confidence |
| *My learning experience was rewarding.* | + 0.4279 * CS usefulness<br>- 0.7784 * self-efficacy |
| **Curiosity** | = 0.5480 * CS confidence |
| *The content of the tutoring system incited my curiosity.* | + 0.3681 * CS usefulness |
| **Felt involvement** | = 0.4724 * CS confidence |
| *I was really drawn into my learning task.* | + 0.3171 * CS usefulness<br>- 1.5409 * self-efficacy$^{\dagger}$<br>+ 0.7083 * learning goal orientation |
| **Focused attention** | = 0.4744 * CS usefulness$^{\dagger}$ |
| *I blocked out things around me when I was working.* | |
| **Task difficulty** | = -9.6884 * CS confidence |
| *How mentally demanding was the task?* | + 9.0752 * achievement goal orientation$^{\dagger}$<br>- 0.1358 * number of compile/run events<br>+ 0.4956 * number of tutor moves$^{\dagger}$ |

**Fig. 4:** Multivariate linear regression analyses for describing the outcomes of the system using measures from pre-survey and from tutorial interaction. CS stands for computer science, $^{\dagger}$ represents $p < 0.005$, all others are $p < 0.05$. Task difficulty has a range from 0-100, all others 1-5.

In addition to these student characteristics or attitudes, several aspects of the tutorial interaction were correlated with outcomes. The number of logged activities, code changes and compile/run events were all positively correlated with the number of utterances written by students throughout the session. One might argue that these measures are correlated with the outcomes because they are also correlated with pre-test scores, which affects the system outcome. However, pre-test score was not significantly correlated with any of the predictors in the regression analyses.

The results suggest that the perceptions of students even before starting the tutoring session are indicative of the outcomes of the system. Such observations are harmonious with those from prior studies [12, 5]. The findings highlight the importance not only of honing the tutorial dialogue within interaction to be as effective as possible, but to adapting to the characteristics and attitudes that students bring in to the tutoring session. In fact, an important limitation of this work arises from students' attitudes and preferences: namely, as students participated as part of an in-class activity and knew that their final product would not be graded for quality, they typically tried to finish as quickly as possible and made fewer than desired utterances to the system. Accounting for and mitigating these types of issues with tutorial dialogue systems is a crucial area for the field as we aim to provide one-on-one adaptive tutoring to very

broad populations of students with varying levels of intrinsic motivation toward the task.

## 5 Conclusion

Tutorial dialogue systems have traditionally required tremendous hand authoring. If we can acquire effective unsupervised dialogue act classifiers from the increasingly vast corpora available, we can transform the way tutorial dialogue systems are built. This paper has described a tutorial dialogue system that relies on a fully unsupervised dialogue act model, and the results demonstrate that it supports student learning and satisfaction as well as a comparable system that relies on a state-of-the-art supervised dialogue act model.

Motivated by the promise of unsupervised models, both in suggesting fully data-driven classification schemes and eliminating human labor, it is very promising to explore them further for tutorial dialogue systems. Another important area of research involves integrating more sophisticated natural language generation with these dialogue act models to increase the flexibility of system utterances. Finally, because incoming motivation of students has such a strong correlation with outcomes, adaptive systems that adjust their strategies according to student motivation are a promising direction for improving tutorial dialogue systems.

## References


1. B. S. Bloom. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, pages 4–16, 1984.
2. G. Chen, S. M. Gully, and D. Eden. Validation of a new general self-efficacy scale. *Organizational Research Methods*, 4(1):62–83, 2001.
3. L. Chen and B. D. Eugenio. Multimodality and dialogue act classification in the robohelper project. In *Proceedings of the Annual SIGDIAL Meeting*, pages 183–192, 2013.
4. M. Dzikovska, N. Steinhauser, E. Farrow, J. Moore, and G. Campbell. BEETLE II: Deep natural language understanding and automatic feedback generation for intelligent tutoring in basic electricity and electronics. *IJAIED*, 24(3):284–332, 2014.
5. S. DMello, C. Williams, P. Hays, and A. Olney. Individual differences as predictors of learning and engagement. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, pages 308–313, 2009.
6. M. W. Evens, R.-C. Chang, Y. H. Lee, L. S. Shim, C. W. Woo, Y. Zhang, J. A. Michael, and A. A. Rovick. CIRCSIM-Tutor: An intelligent tutoring system using natural language dialogue. In *Proceedings of Applied Natural Language Processing*, pages 13–14, 1997.
7. A. Ezen-Can and K. E. Boyer. Combining task and dialogue streams in unsupervised dialogue act models. In *Proceedings of the Annual SIGDIAL Meeting*, pages 113–122, 2014.
8. D. Fossati, B. Di Eugenio, C. Brown, and S. Ohlsson. Learning linked lists: Experiments with the iList system. In *Proceedings of ITS*, pages 80–89, 2008.

9. A. C. Graesser, N. K. Person, and J. P. Magliano. Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied Cognitive Psychology*, 9(6):495–522, 1995.
10. E. Y. Ha, J. F. Grafsgaard, C. M. Mitchell, K. E. Boyer, and J. C. Lester. Combining verbal and nonverbal features to overcome the 'information gap' in task-oriented dialogue. In *Proceedings of the Annual SIGDIAL Meeting on Discourse and Dialogue*, pages 247–256, 2012.
11. M. Hall, H. National, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.
12. G. T. Jackson, A. C. Graesser, and D. S. McNamara. What students expect may have more impact than what they know or feel. In *Proceedings of AIED*, pages 73–80, 2009.
13. P. Jordan, P. Albacete, M. J. Ford, S. Katz, M. Lipschultz, D. Litman, S. Silliman, and C. Wilson. Interactive event: The Rimac Tutor-a simulation of the highly interactive nature of human tutorial dialogue. In *Proceedings of AIED*, pages 928–929. Springer, 2013.
14. H. C. Lane and K. VanLehn. Teaching the tacit knowledge of programming to novices with natural language tutoring. *Computer Science Education*, 15(3):183–201, 2005.
15. C. Lee and P. Bobko. Self-efficacy beliefs: Comparison of five measures. *Journal of Applied Psychology*, 79(3):364, 1994.
16. D. Litman and S. Silliman. ITSPOKE: An intelligent tutoring spoken dialogue system. In *Demonstration Papers at HLT-NAACL 2004*, pages 5–8, 2004.
17. B. D. Nye, A. C. Graesser, and X. Hu. AutoTutor and family: A review of 17 years of natural language tutoring. *IJAIED*, 24(4):427–469, 2014.
18. V. Rus, C. Moldovan, N. Niraula, and A. C. Graesser. Automated discovery of speech act categories in educational games. In *Proceedings of EDM*, pages 25–32, 2012.
19. A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. Van Ess-Dykema, and M. Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–373, 2000.
20. A. K. Vail and K. E. Boyer. Identifying effective moves in tutoring: On the refinement of dialogue act annotation schemes. In *Proceedings of ITS*, pages 199–209, 2014.
21. D. VandeWalle, W. L. Cron, and J. W. Slocum Jr. The role of goal orientation following performance feedback. *Journal of Applied Psychology*, 86(4):629, 2001.
22. K. VanLehn, A. C. Graesser, G. T. Jackson, P. Jordan, A. Olney, and C. P. Rosé. When are tutorial dialogues more effective than reading? *Cognitive Science*, 31(1):3–62, 2007.
23. K. VanLehn, P. W. Jordan, C. P. Rosé, D. Bhembe, M. Böttner, A. Gaydos, M. Makatchev, U. Pappuswamy, M. Ringenberg, A. Roque, et al. The architecture of Why2-Atlas: A coach for qualitative physics essay writing. In *Proceedings of ITS*, pages 158–167, 2002.
24. K. VanLehn, C. Lynch, K. Schulze, J. A. Shapiro, R. Shelby, L. Taylor, D. Treacy, A. Weinstein, and M. Wintersgill. The Andes physics tutoring system: Lessons learned. *IJAIED*, 15(3):147–204, 2005.
25. M. A. Walker, D. J. Litman, C. A. Kamm, and A. Abella. PARADISE: A framework for evaluating spoken dialogue agents. In *Proceedings of the European Chapter of the Association for Computational Linguistics*, pages 271–280, 1997.