Dialogue Act Modeling in a Complex Task-Oriented Domain

Kristy Elizabeth Boyer | **Eun Young Ha** | **Robert Phillips*** | **Michael D. Wallis*** | **Mladen A. Vouk** | **James C. Lester**

Department of Computer Science, North Carolina State University
Raleigh, North Carolina, USA

*Dual affiliation with Applied Research Associates, Inc.
Raleigh, North Carolina, USA

{keboyer, eha, rphilli, mdwallis, vouk, lester}@ncsu.edu

Abstract

Classifying the dialogue act of a user utterance is a key functionality of a dialogue management system. This paper presents a data-driven dialogue act classifier that is learned from a corpus of human textual dialogue. The task-oriented domain involves tutoring in computer programming exercises. While engaging in the task, students generate a task event stream that is separate from and in parallel with the dialogue. To deal with this complex task-oriented dialogue, we propose a vector-based representation that encodes features from both the dialogue and the hierarchically structured task for training a maximum likelihood classifier. This classifier also leverages knowledge of the hidden dialogue state as learned separately by an HMM, which in previous work has increased the accuracy of models for predicting tutorial moves and is hypothesized to improve the accuracy for classifying student utterances. This work constitutes a step toward learning a fully data-driven dialogue management model that leverages knowledge of the user-generated task event stream.

1 Introduction

Two central challenges for dialogue systems are interpreting user utterances and selecting system dialogue moves. Recent years have seen an increased focus on data-driven techniques for addressing these challenging tasks (Bangalore et al., 2008; Frampton & Lemon, 2009; Hardy et al., 2006; Sridar et al., 2009; Young et al., 2009). Much of this work utilizes dialogue acts, built on the notion of speech acts (Austin, 1962), which provide a valuable intermediate representation that can be used for dialogue management.

Data-driven approaches to dialogue act interpretation have included models that take into account a variety of lexical, syntactic, acoustic, and prosodic features for dialogue act tagging (Sridhar et al., 2009; Stolcke et al., 2000). In task-oriented domains, recent work has approached dialogue act classification by learning dialogue management models entirely from human-human corpora (Bangalore et al., 2008; Chotimongkol, 2008; Hardy et al., 2006). Our work adopts this approach for a corpus of human-human dialogue in a task-oriented tutoring domain. Unlike the majority of task-oriented domains that have been studied to date, our domain involves the separate creation of a persistent artifact, in our case a computer program, by the user during the course of the dialogue. Our corpus consists of human-human textual dialogue utterances and a separate, parallel stream of user-generated task actions. We utilize structural features including task/subtask, speaker, and hidden dialogue state along with lexical and syntactic features to interpret user (student) utterances.

This paper makes three contributions. First, it addresses representational issues in creating a dialogue model that integrates task actions with hierarchical task/subtask structure. The task is captured within a separate synchronous event stream that exists in parallel with the dialogue. Second, this paper explores the performance of dialogue act classifiers using different lexical/syntactic and structural feature sets. This comparison includes one model trained entirely on lexical/syntactic features, an important step toward robust unsupervised dialogue act tagging

(Sridhar et al., 2009). Finally, it investigates whether the addition of HMM and task/subtask features improves the performance of the dialogue act classifiers. The findings support this hypothesis for three student dialogue moves, each with important implications for tutorial dialogue.

2 Related Work

A variety of modeling approaches have been investigated for statistical dialogue act classification, including sequential approaches and vector-based classifiers. Sequential approaches typically formulate dialogue as a Markov chain in which an observation depends on a finite number of preceding observations. HMM-based approaches make use of the Markov assumption in a doubly stochastic framework that allows fitting optimal dialogue act sequences using the Viterbi algorithm (Rabiner, 1989; Stolcke et al., 2000). Like this work, the approach reported here adopts a first-order Markov formulation to train an HMM on sequences of dialogue acts, but the prediction of this HMM is subsequently encoded in a feature vector for training a vector-based classifier.

Vector-based approaches, such as maximum entropy modeling, also frequently take into account both lexical/syntactic and structural features. Lexical and syntactic cues are extracted from local utterance context, while structural features involve longer dialogue act sequences and, in task-oriented domains, task/subtask history. Work by Bangalore et al. (2008) on learning the structure of human-human dialogue in a catalogue-ordering domain (also extended to the Maptask and Switchboard corpora) utilizes features including words, part of speech tags, supertags, and named entities, and structural features including dialogue acts and task/subtask labels. In order to perform incremental decoding of dialogue acts and task/subtask structure, they take a greedy approach that does not require the search of complete dialogue sequences. Our work also accomplishes left-to-right incremental interpretation with a greedy approach. Our feature vectors differ from the aforementioned work slightly with respect to lexical/syntactic features and notably in the addition of a set of structural features generated by a separately trained HMM, as described in Section 4.2.

Recent work has explored the use of lexical, syntactic, and prosodic features for online dialogue act tagging (Sridhar et al., 2009); that work explores the notion that structural (history) features could be omitted altogether from incremental left-to-right decoding, resulting in computationally inexpensive and robust dialogue act classification. Although our textual dialogue does not feature prosodic cues, we report on the use of lexical/syntactic features alone to perform dialogue act classification, a step toward a fully unsupervised approach.

Like Bangalore et al. (2008), we treat task structure as an integral part of the dialogue model. Other work that has taken this approach includes the Amitiés project, in which a dialogue manager for a financial domain was derived entirely from a human-human corpus (Hardy et al., 2006). The TRIPS dialogue system also closely integrated task and dialogue models, for example, by utilizing the task model to facilitate indirect speech act interpretation (Allen et al., 2001). Work on the Maptask corpus has modeled task structure in the form of conversational games (Wright Hastie et al., 2002). Recent work in task-oriented domains has focused on learning task structure with unsupervised approaches (Chotimongkol, 2008). Emerging unsupervised methods, such as for detecting actions in multi-party discourse, also implicitly capture a task structure (Purver et al., 2006).

Our domain differs from the task-oriented domains described above in that our dialogues center on the user creating a persistent artifact of intrinsic value through a separate, synchronous stream of task actions. To illustrate, consider a catalogue-ordering task in which one subtask is to obtain the customer's name. The fulfillment of this subtask occurs entirely through the dialogue, and the resulting artifact (a completed order) is produced by the system. In contrast, our task involves the user constructing a solution to a computer programming problem. The fulfillment of this task occurs partially in the dialogue through tutoring, and partially in a separate synchronous stream of user-driven task actions about which the tutor must reason. The stream of user-driven task actions produces an artifact of value in itself (a functioning computer program), and that artifact is the subject of much of the dialogue. We propose a representation that integrates task actions and dialogue acts from these streams into a shared vector-based representation, and we investigate the use of the resulting structural, lexical, and syntactic features for dialogue act classification.

3 Corpus and Annotation

The corpus was collected during a controlled human-human tutoring study in which tutors and students worked through textual dialogue to solve an introductory computer programming problem. The dialogues were effective: on average, students exhibited significant learning and self-confidence gains (Boyer et al., 2009).

The corpus contains 48 dialogues each with a separate, synchronous task event stream as depicted in Excerpt 1 of the appendix. There is exactly one dialogue (tutoring session) per student. The corpus captures approximately 48 hours of dialogue and contains 1,468 student utterances and 3,338 tutor utterances. Because the dialogue was textual, utterance segmentation consisted of splitting at existing sentence boundaries when more than one dialogue act was present in the utterance. This segmentation was conducted manually by the principal dialogue act annotator.[1]

The corpus was manually annotated with dialogue act labels and task/subtask features. Lexical and syntactic features were extracted automatically. The remainder of this section describes the manual annotation.

3.1 Dialogue Act Annotation

The dialogue act annotation scheme was inspired by schemes for conversational speech (Stolcke et al., 2000) and task-oriented dialogue (Core & Allen, 1997). It was also influenced by tutoring-specific tagsets (Litman & Forbes-Riley, 2006). Inter-rater reliability for the dialogue act tagging on 10% of the corpus selected via stratified (by tutor) random sampling was κ=0.80. The dialogue act tags, their relative frequencies, and their individual kappa scores from manual annotation are displayed in Table 1.

3.2 Task Annotation

All task actions were generated by the student while implementing the solution to an introductory computer programming problem in Java. These task actions were recorded as a separate event stream in parallel with the dialogue corpus. This stream included 97,509 keystroke-level user task events, which were manually aggregated into task/subtask event clusters and annotated for subtask structure and then for correctness. A total of 3,793 aggregated student subtask actions were identified through manual annotation. The task annotation scheme is hierarchical, reflecting the nested nature of the subtasks. A subset of this task annotation scheme is depicted in Figure 1. In the models reported in this paper, the 66 leaves of the task/subtask hierarchy were encoded in the input feature vectors.

Table 1. Student dialogue acts

Student Dialogue Act	Rel. Freq.	Human κ
ACKNOWLEDGMENT (ACK)	.17	*.90*
REQUEST FOR FEEDBACK (RF)	.20	*.91*
EXTRA-DOMAIN (EX)	.08	*.79*
GREETING (GR)	.04	*.92*
UNCERTAIN FEEDBACK WITH ELABORATION (UE)	.01	*.53*
UNCERTAIN FEEDBACK (U)	.02	*.49*
NEGATIVE FEEDBACK WITH ELABORATION (NE)	.01	*.61*
NEGATIVE FEEDBACK (N)	.05	*.76*
POSITIVE FEEDBACK WITH ELABORATION (PE)	.02	*.43*
POSITIVE FEEDBACK (P)	.09	*.81*
QUESTION (Q)	.09	*.85*
STATEMENT (S)	.16	*.82*
THANKS (T)	.05	*1*

Each group of task events that occurred between dialogue utterances was tagged, possibly with many subtask labels, by a human judge. The judge aggregated the raw task keystrokes and tagged the task/subtask hierarchy for each cluster. (Please see Excerpt 1 in the appendix.) A second judge tagged 20% of the corpus in a reliability study for which one-to-one subtask identification was not enforced, an approach that was intended to give judges maximum flexibility to cluster task actions and subsequently apply the tags. All unmatched subtask tags were treated as disagreements. The resulting kappa statistic at the leaves was κ= 0.58. However, we also observe that the sequential nature of the subtasks within the larger task produces an ordinal relationship between subtasks. For example, in Figure 1, the "distance" between subtasks *1-a* and *1-b* can be thought of as "less than" the distance between subtasks *1-a* vs. *3-d* because those subtasks are farther from each other within the larger task. The weighted Kappa statistic (Artstein & Poesio, 2008) takes into account such an ordinal relationship and its implicit distance function. The weighted Kappa is

[1] Automatic segmentation is a challenging problem in itself and is left to future work.

$\kappa_{weighted}$=0.86, which indicates acceptable inter-rater reliability on the task/subtask annotation.

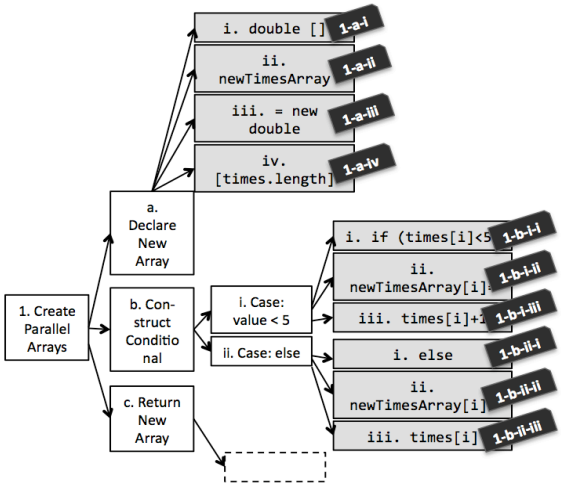

Figure 1. Portion of task annotation scheme

Along with its tag for hierarchical subtask structure, each task event was also judged for correctness according to the requirements of the task as depicted in Table 2. The agreement statistic for correctness was calculated for task events on which the two judges agreed on subtask tag. The resulting unweighted agreement statistic for correctness was κ=0.80.

Table 2. Task correctness labels

Label	Description
CORRECT	Fully satisfying the requirements of the learning task. Does not require tutorial remediation.
BUGGY	Violating the requirements of the learning task. Often requires tutorial remediation.
INCOMPLETE	Not violating, but not yet fully satisfying, the requirements of the learning task. May require tutorial remediation.
DISPREFERRED	Technically satisfying the requirements of the learning task, but not adhering to its pedagogical intentions. Usually requires tutorial remediation.

4 Features

The vector-based representation for training the dialogue act classifiers integrates several sources of features: lexical and syntactic features, and structural features that include dialogue act labels, task/subtask labels, and set of hidden dialogue state prediction features.

4.1 Lexical and Syntactic Features

Lexical and syntactic features were automatically extracted from the utterances using the Stanford Parser default tokenizer and part of speech (*pos*) tagger (De Marneffe et al., 2006). The parser created both phrase structure trees and typed dependencies for individual sentences. From the phrase structure trees, we extracted the top-most syntactic node and its first two children. In the case where an utterance consisted of more than one sentence, only the phrase structure tree of the first sentence was considered. Typed dependencies between pairs of words were extracted from each sentence. Individual word tokens in the utterances were further processed with the Porter Stemmer (Porter, 1980) in the NLTK package (Loper & Bird, 2004). The *pos* features were extracted in a similar way. Unigram and bigram word and *pos* tags were included for feature selection in the classifiers.

4.2 Structural Features

Structural features include the annotated dialogue acts, the annotated task/subtask labels, and attributes that represent the *hidden dialogue state*. Our previous work has found that a set of hidden dialogue states, which correspond to widely accepted notions of dialogue modes in tutoring, can be identified in an unsupervised fashion (without hand labeling of the modes) by HMMs trained on manually labeled dialogue acts and task/subtask features (Boyer et al., 2009). These HMMs performed significantly better than bigram models for predicting human *tutor* moves (Boyer et al., 2010), which indicates that the hidden dialogue state leveraged by the HMMs has predictive value even in the presence of "true" (manually annotated) dialogue act labels. Therefore, we hypothesized that an HMM could also improve the performance of models to classify student dialogue acts. To explore this hypothesis, we trained an HMM utilizing the methodology described in (Boyer et al., 2009) and used it to generate hidden dialogue state predictions in the form of a probability distribution over possible user utterances at each step in the dialogue. This set of stochastic features was subsequently passed to the classifier as part of the input vector (Figure 2).

4.3 Input Vectors

The features were combined into a shared vector-based representation for training the classifier. As depicted in Table 3, the components of the

feature vector include binary existence vectors for lexical and syntactic features for the current (target) utterance as well as for three utterances of left context (this left context may include both tutor and student utterances, which are distinguished by a separate indicator for the speaker). The task/subtask and correctness history features encode the separate stream of task events. There is no one-to-one correspondence between these history features and the left-hand dialogue context, because several task events could have occurred between a pair of dialogue events (or vice versa). This distinction is indicated in the table by the representation of dialogue time steps as [*t, t-1,…*] and task history steps as [*task(t), task(t-1),…*]. In total, the feature vectors included 11,432 attributes that were made available for feature selection.

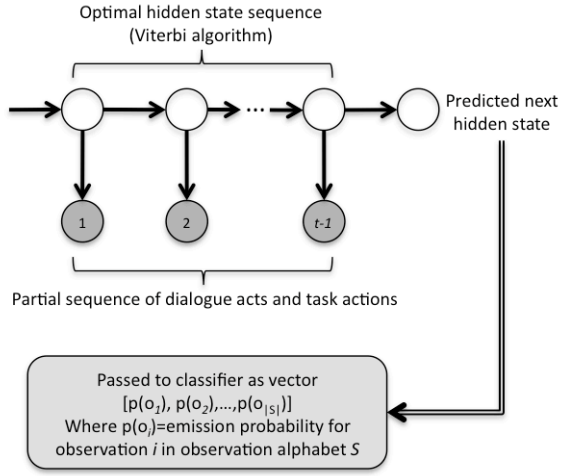

Figure 2. Generation of hidden dialogue state prediction features

5 Experiments

This section describes the learning of maximum likelihood vector-based models for classification of user dialogue acts. In addition to investigating the accuracy of the overall model, we also performed experiments regarding the utility of feature types for discriminating between particular dialogue acts of interest.

The classifiers are based on logistic regression, which learns a discriminant for each pair of dialogue acts by assigning weights in a maximum likelihood fashion.[2] The logistic regression models were learned using the Weka machine learning toolkit (Hall et al., 2009). For feature selection, we performed attribute subset evaluation with a best-first approach that greedily searched the space of possible features using a hill climbing approach with backtracking. The prediction accuracy of the classifiers was determined through ten-fold cross-validation on the corpus, and the results below are presented in terms of prediction accuracy (number of correct classifications divided by total number of classifications) as well as by the kappa statistic, which adjusts for expected agreement by chance.

Table 3. Feature vectors

Feature vector *f*	Description
$[w_{t,1},\ldots w_{t,\|w\|}, p_{t,1},\ldots,p_{t,\|p\|}, d_{t,1},\ldots,d_{t,\|d\|}, s_{t,1},\ldots,s_{t,\|s\|}]$	Binary existence vector for word unigrams & bigrams, *pos* unigrams & bigrams, dependency types, and syntactic nodes for current target utterance *t*
$[w_{t-k,1},\ldots w_{t-k,\|w\|}, p_{t-k,1},\ldots,p_{t-k,\|p\|}, d_{t-k,1},\ldots,d_{t-k,\|d\|}, s_{t-k,1},\ldots,s_{t-k,\|s\|}]$ where $k=1,\ldots,3$	Binary existence vector for word unigrams & bigrams, *pos* unigrams & bigrams, dependency types, and syntactic nodes for three utterances of left context
$[p(o_1),\ldots,p(o_{\|S\|})]$	Probability distribution for emission symbols in predicted next hidden state as generated by HMM
$[da_{t-1}, da_{t-2}, da_{t-3}]$	Dialogue act left context
$[sp_{t-1}, sp_{t-2}, sp_{t-3}]$	Speaker label left context
$[tk_{task(t-1)}, tk_{task(t-2)}, tk_{task(t-3)}]$	Three steps of subtask history (each level of hierarchy represented as a separate feature)
$[c_{task(t-1)}, c_{task(t-2)}, c_{task(t-3)}]$	Three steps of task correctness history
pt	Indicator for whether the target utterance was immediately preceded by a task event

5.1 Overall Classification Task

The overall dialogue act classification model was trained to classify each utterance with respect to the thirteen dialogue acts (Table 1). For this task, the feature selection algorithm selected 63 attributes including some syntax, dependency, *pos*, and word attributes as well as dialogue act, speaker, and task/subtask features. No hidden dialogue state features or task correctness attributes were selected. The overall classification accuracy was 62.8%. This accuracy constitutes a 369% improvement over baseline chance of 17% (the relative frequency of the most frequently occurring dialogue act, ACK). An alternate nontrivial baseline is a bigram model on true dialogue acts (including speaker tags); this model's accuracy was 36.8%. The

[2] In general, the model that maximizes likelihood also maximizes entropy under the same constraints (Berger et al., 1996).

overall kappa for the full classifier was κ=.57. The confusion matrix for this model is depicted in Figure 3.

In addition to the classifier described above, we experimented with classifiers that used only the lexical and syntactic features of each utterance. This approach is of interest in part because it avoids the error propagation that can happen when a model relies on a series of its own previous classifications as features. The classifier that used only the set of lexical and syntactic features achieved a prediction accuracy of 60.2% and κ=.53 using 85 attributes.

GR	N	P	S	RF	Q	T	ACK	Ex	NE	PE	L	LE	
50	0	0	1	0	0	0	1	0	0	0	0	0	GR
0	19	6	13	2	2	0	5	2	1	0	6	2	N
0	5	52	37	1	1	0	18	3	0	1	1	1	P
0	6	21	145	3	9	0	15	7	0	4	4	0	S
0	2	1	11	232	23	0	6	5	0	1	0	1	RF
1	2	2	13	60	46	0	1	5	0	1	0	0	Q
0	0	1	3	0	0	60	3	1	1	0	0	1	T
0	0	7	19	4	0	2	195	4	0	0	2	0	ACK
0	1	4	24	12	3	1	16	40	0	1	0	0	Ex
0	1	1	9	0	1	1	0	1	0	3	1	0	NE
0	2	4	13	0	2	0	1	1	1	4	2	2	PE
0	6	4	2	3	0	0	0	0	1	3	6	1	L
0	3	0	5	2	2	1	0	1	0	0	0	1	LE

Figure 3. Confusion matrix

5.2 Binary Dialogue Act Classification

In tutoring, some student dialogue acts are particularly important to identify because of their implications for the tutor's response or for the student model. For example, a student's REQUEST FOR FEEDBACK requires the tutor to assess the condition of the task, rather than to query the in-domain factual knowledge base. UNCERTAIN FEEDBACK is another dialogue act of high importance because identifying it allows the tutor to respond in an affectively advantageous way (Forbes-Riley & Litman, 2009).

To explore which features are useful for classifying particular dialogue acts, we constructed binary dialogue act classifiers, one for each dialogue act, by preprocessing the dialogue act labels from the set of thirteen down to TRUE or FALSE depending on whether the label of the utterance matched the target dialogue act for that specialized classifier. Table 4 displays the features that were selected for each binary classifier, along with the percent accuracy and kappa for each model. Note that for some dialogue acts the chance baseline is very high, and therefore even a model with high prediction accuracy achieves a low kappa.

As depicted in Table 4, for several dialogue act models, the feature selection algorithm retained subtask and HMM features.

Table 4. Binary DA classifiers

DA	# Features Selected	% Correct	Model κ
ACK 51	Lexical/syntax, HMM, DA history (preceding=S), speaker history (preceding=Tutor)	.933	*.75*
RF 42	Lexical/syntax, DA history, preceded by subtask	.905	*.72*
EX 57	Dependency, *pos*, word, HMM, DA history (preceding=EX), subtask	.939	*.45*
GR 11	Syntax, *pos*, word, DA (previous=EMPTY), speaker, subtask	.998	*.97*
UE 21	Dependency, *pos*, word, subtask	.991	*.33*
U 63	Syntax, dependency, *pos*, word, HMM, subtask	.979	*.21*
NE 44	Dependency, *pos*, word, HMM, DA history (2 ago=UNCERTAIN), subtask	.987	*0*
N 83	Lexical/syntax, DA history, subtask	.966	*.76*
PE 90	Dependency, *pos*, word, HMM, subtask	.976	*.10*
P 110	Dependency, *pos*, word, HMM, DA history (previous=REQUEST FEEDBACK)	.945	*.58*
Q 43	Syntax, dep, *pos*, word, HMM, subtask	.940	*.60*
S 92	Syntax, *pos*, word, HMM, DA history (previous=EMPTY or Q)	.901	*.57*
T 29	Syntax, *pos*, word, DA history (previous=POSITIVE) (3 ago=POSITIVE)	.992	*.92*

In an experiment to quantify the utility of these features, it was found that for many dialogue acts, a binary dialogue act classifier that was trained using only lexical and syntactic features achieved the same or better classification accuracy than the model that was given all features (Figure 4). For comparison, the modified baseline model used the last three true dialogue acts (with speaker tags); this model achieved better than chance for four dialogue acts and achieved nearly as well as the full model for GREETING (GR). The models that were given all possible features for selection outperformed the lexical/syntax-only model for seven of the thirteen dialogue acts (GREETING (GR), REQUEST FOR FEEDBACK (RF), POSITIVE FEEDBACK (P), POSITIVE ELABORATED FEEDBACK (PE), UNCERTAIN ELABORATED FEEDBACK (UE), NEGATIVE FEEDBACK (N), and EXTRA-DOMAIN (EX)); however, it should be noted that none of these differences in performance is statistically reliable at the p=0.05 level.

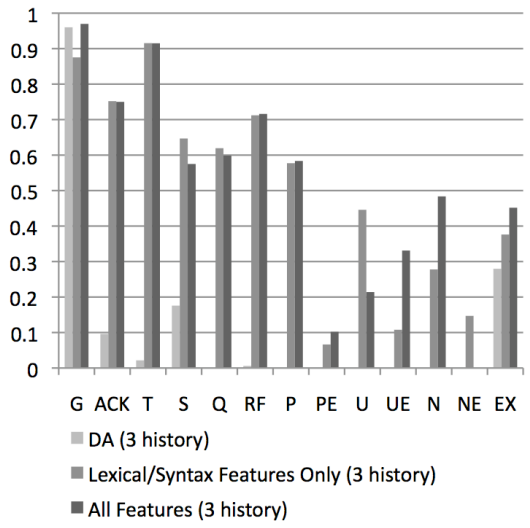

Figure 4. Kappa for binary DA classifiers by features available for selection

6 Discussion

We have presented a maximum likelihood classifier that assigns dialogue act labels to user utterances from a corpus of human-human tutorial dialogue given a set of lexical, syntactic, and structural features. Overall, this classifier achieved 62.8% accuracy in ten-fold cross-validation on the corpus. This performance is on par with other automatic dialogue act tagging models, both sequential and vector-based, in task-oriented domains that do not feature complex, user-driven parallel tasks.

In a catalogue ordering domain with an integrated task and dialogue model, Bangalore et al. (2009) report 75% classification accuracy for user utterances using a maximum entropy classifier, a 275% improvement over baseline. Poesio & Mikheev (1998) report 54% classification accuracy by utilizing conversational game structure and speaker changes in the Maptask corpus, an improvement of 170% over baseline. Recent work on Maptask reports a classification accuracy of 65.7% using local utterance (such as lexical/syntactic) features alone, with prosodic cues yielding further slight improvement (Sridhar et al., 2009). This classifier is analogous to our lexical/syntactic feature model, which achieved 60.2% accuracy.

The results of these models demonstrate that, consistent with the findings in other task-oriented domains, lexical/syntactic features are highly useful for classifying student dialogue moves in this complex task-oriented domain. Models trained using those lexical/syntactic features performed almost universally better (with the exception of the binary classifier for GREETING) than models that were given the same left context of true dialogue act tags.

It was hypothesized that leveraging both the hidden dialogue state and hierarchical subtask features would improve the performance of the classifiers. There is some evidence that the subtask structure was helpful for the overall classifier; however, no HMM features were kept during feature selection for the overall model. Of the binary models, approximately half performed better than the overall model in terms of true positive rate; of those, three did so by including HMM or task/subtask features among the selected attributes to differentiate different tones of student feedback. However, this difference in performance was not statistically reliable. This finding suggests that, given lexical and syntactic features which are strong predictors of dialogue acts, the hidden dialogue state as captured by an an HMM may not contribute significantly to the dialogue act classification task.

7 Conclusion and Future Work

Dialogue modeling for complex task-oriented domains poses significant challenges. An effective dialogue model allows systems to detect user dialogue acts so that they can respond in a manner that maximizes the chance of success. Experiments with the data-driven classifiers presented in this paper demonstrate that lexical/syntactic features can effectively classify student dialogue acts in the task-oriented tutoring domain. For POSITIVE, NEGATIVE, and UNCERTAIN ELABORATED student feedback acts, which play a key role in tutorial dialogue system, the addition of hidden dialogue state features (as learned by an HMM) and task/subtask features (annotated manually) improve classification accuracy, but not statistically reliably.

The overarching goal of this work is to create a data-driven tutorial dialogue system that learns its behavior from corpora of effective human tutoring. The dialogue act classification models reported here constitute an important step toward that goal, by integrating the dialogue stream with a parallel user-driven task event stream. The next generation of data-driven systems should leverage models that capture the rich interplay between dialogue and task. Future work will focus on data-driven approaches to task recognition and tutorial planning. Additionally, as dialogue system research addresses

increasingly complex task-oriented domains, it becomes increasingly important to investigate unsupervised approaches for dialogue act classification and task recognition.

Acknowledgements. This work is supported in part by the North Carolina State University Department of Computer Science and the National Science Foundation through a Graduate Research Fellowship and Grants CNS-0540523, REC-0632450 and IIS-0812291. Any opinions, findings, conclusions, or recommendations expressed in this report are those of the participants, and do not necessarily represent the official views, opinions, or policy of the National Science Foundation.

References

Allen, J., Ferguson, G., & Stent, A. (2001). An architecture for more realistic conversational systems. *Proceedings of the IUI,* 1-8.

Artstein, R., & Poesio, M. (2008). Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4), 555-596.

Austin, J. L. (1962). *How to do things with words*. Oxford: Oxford University Press.

Bangalore, S., Di Fabbrizio, G., & Stent, A. (2008). Learning the structure of task-driven human-human dialogs. *IEEE Transactions on Audio, Speech, and Language Processing, 16*(7), 1249-1259.

Berger, A. L., Pietra, V. J. D., & Pietra, S. A. D. (1996). A maximum entropy approach to natural language processing. *Comp. Ling., 22*(1), 71.

Boyer, K. E., Phillips, R., Ha, E. Y., Wallis, M. D., Vouk, M. A., & Lester, J. C. (2009). Modeling dialogue structure with adjacency pair analysis and hidden markov models. *Proceedings of NAACL-HLT, Short Papers, 49-52.*

Boyer, K. E., Phillips, R., Ha, E. Y., Wallis, M. D., Vouk, M. A., & Lester, J. C. (2010). Leveraging hidden dialogue state to select tutorial moves. *Proceedings of the 5th NAACL HLT Workshop on Innovative use of NLP for Building Educational Applications,* Los Angeles, California.

Chotimongkol, A. (2008). *Learning the structure of task-oriented conversations from the corpus of in-domain dialogs.* (Unpublished Ph.D. Dissertation). Carnegie Mellon University School of Computer Science.

Core, M., & Allen, J. (1997). Coding dialogs with the DAMSL annotation scheme. *AAAI Fall Symposium on Communicative Action in Humans and Machines,* 28–35.

De Marneffe, M. C., MacCartney, B., & Manning, C. D. (2006). Generating typed dependency parses from phrase structure parses. *Proceedings of LREC,* Genoa, Italy.

Forbes-Riley, K., & Litman, D. (2009). Adapting to student uncertainty improves tutoring dialogues. *Proceedings of AIED,* 33-40.

Frampton, M., & Lemon, O. (2009). Recent research advances in reinforcement learning in spoken dialogue systems. *The Knowledge Engineering Review, 24*(4), 375-408.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. (2009). The WEKA data mining software: An update. *SIGKDD Explorations, 11*(1)

Hardy, H., Biermann, A., Inouye, R. B., McKenzie, A., Strzalkowski, T., Ursu, C., Webb, N., & Wu, M. (2006). The Amitiés system: Data-driven techniques for automated dialogue. *Speech Comm., 48*(3-4), 354-373.

Litman, D., & Forbes-Riley, K. (2006). Correlations between dialogue acts and learning in spoken tutoring dialogues. *Natural Language Engineering, 12*(2), 161-176.

Loper, E., & Bird, S. (2004). NLTK: The natural language toolkit. *Proceedings of the ACL Demonstration Session,* Barcelona, Spain. 214-217.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program, 14*(3), 130-137.

Purver, M., Kording, K. P., Griffiths, T. L., & Tenenbaum, J. B. (2006). Unsupervised topic modelling for multi-party spoken discourse. *Proceedings of the ACL,* Sydney, Australia. *, 44*(1) 17.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE, 77*(2), 257-286.

Sridhar, V. K. R., Bangalore, S., & Narayanan, S. (2009). Combining lexical, syntactic and prosodic cues for improved online dialog act tagging. *Computer Speech & Language, 23*(4), 407-422.

Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., Martin, R., Van Ess-Dykema, C., & Meteer, M. (2000). Dialogue act modeling for automatic tagging and recognition of conversational speech. *Comp. Ling., 26*(3), 339-373.

Wright Hastie, H., Poesio, M., & Isard, S. (2002). Automatically predicting dialogue structure using prosodic features. *Speech Communication, 36*(1-2), 63-79.

Young, S., Gasic, M., Keizer, S., Mairesse, F., Schatzmann, J., Thomson, B., & Yu, K. (2009). The hidden information state model: A practical framework for POMDP-based spoken dialogue management. *Computer Speech and Language, 24*(2), 150-174.

Appendix

Time Stamp	Dialogue Stream	Task Stream	
2008-04-11 18:23:45	Student: so do i have to manipulate the array this time? [Q]		
2008-04-11 18:23:53	Tutor: this time, we need to do two things [S]		
2008-04-11 18:24:02	Tutor: first, we need to create a new array to hold the changed values [S]		
2008-04-11 18:24:28		i	
2008-04-11 18:24:28		n	**1-a-i**
2008-04-11 18:24:28		t	**BUGGY**
2008-04-11 18:24:28		\sp	
2008-04-11 18:24:35		\del	
2008-04-11 18:24:36		\sp	
2008-04-11 18:24:36		d	
2008-04-11 18:24:36		o	
2008-04-11 18:24:36		u	
2008-04-11 18:24:36		b	**1-a-i**
2008-04-11 18:24:37		l	**CORRECT**
2008-04-11 18:24:37		e	
2008-04-11 18:24:37		\sp	
2008-04-11 18:24:39		[]	
2008-04-11 18:24:40		\sp	
2008-04-11 18:24:42		n	
2008-04-11 18:24:42		e	
2008-04-11 18:24:42		w	
2008-04-11 18:24:43		\sp	
2008-04-11 18:24:44		\del	
2008-04-11 18:24:45		T	
2008-04-11 18:24:46		\del	**1-a-ii**
2008-04-11 18:24:54		T	**CORRECT**
2008-04-11 18:24:54		i	
2008-04-11 18:24:54		m	
2008-04-11 18:24:54		e	
2008-04-11 18:24:54		s	
2008-04-11 18:24:55		3	
2008-04-11 18:24:57		;	
2008-04-11 18:25:11	Student: good? [RF]		
2008-04-11 18:25:14	Tutor: good so far, yes [PF]		
2008-04-11 18:25:29	Student: so now i have to change parts of the times array right? [Q]		
2008-04-11 18:25:34	Tutor: not quite [LF]		
2008-04-11 18:25:57	Tutor: So, when you create a new object, like a String for example, you'd say something like String s = new String() [S]		
2008-04-11 18:25:59	Tutor: right? [AQ]		
2008-04-11 18:26:06	Student: right [P]		
2008-04-11 18:26:14	Tutor: arrays are similar [S]		

Excerpt 1. Parallel synchronous dialogue and task event streams with annotations. (Note tutor dialogue acts: AQ=ASSESSING QUESTION, LF=LUKEWARM FEEDBACK, PF=POSITIVE FEEDBACK)