Learning the Structure of Task-Driven Human–Human Dialogs

Srinivas Bangalore, *Senior Member, IEEE*, Giuseppe Di Fabbrizio, *Senior Member, IEEE*, and Amanda Stent, *Member, IEEE*

***Abstract*—With the availability of large corpora of spoken dialog, it is now possible to use data-driven techniques to build and use models of task-oriented dialogs. In this paper, we use data-driven techniques to build task structures for individual dialogs, and use the dialog task structures for: dialog act classification, task/subtask classification, task/subtask prediction, and dialog act prediction. We evaluate our approach using a corpus of customer/agent dialogs from a catalog service domain. This paper demonstrates the feasibility of using corpora of human–human conversation to learn dialog models suitable for human–computer dialog applications.**

***Index Terms*—Dialog systems, language generation, language understanding, prediction and classification.**

I. Introduction

IN ORDER to build a dialog system, it is essential to have a *dialog model* that captures information about the dialog partners' interaction patterns and about the task structure of the domain [1]. Some of the ways the dialog model can be used in a spoken dialog system include the following.

- Speech recognition—language models specific to particular dialog contexts can be used if the system can predict the user's next likely action(s).
- Interpretation of user utterances—task or dialog-act specific grammars can be used for spoken language understanding of user utterances.
- Dialog management—the task structure can be used to predict the actions that the system should perform.
- Response generation—the task structure can be used to determine the content of system utterance(s). In a template-based generator, templates may be selected using the current subtask and system dialog act(s).

Corpora of spoken dialog are now widely available and frequently come with annotations for tasks/games, dialog acts, named entities, and elements of syntactic structure. These types of information provide rich clues for building dialog models [2]. However, when dialog systems are built, the standard approach is still to hand-engineer a dialog model rather than to mine a corpus of existing dialogs.

Manuscript received June 29, 2007; revised May 14, 2008. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Bill Byrne.

The authors are with AT&T Labs-Research, Inc., Florham Park, NJ 07932 USA.

Digital Object Identifier 10.1109/TASL.2008.2001102

In this paper, we automatically build dialog models from corpora of human–human spoken dialog annotated for dialog acts and task/subtask information. We then apply these models to interpretation of user utterances and prediction of system actions. The contributions of this work include the following:

1) the combination of dialog act and task/subtask information in a single dialog model;
2) the comparison of two approaches to building task structures for dialogs: a flat, chunk-based model and a hierarchical, parse-based model;
3) the application of dialog modeling to two important dialog management tasks: interpretation of user utterances and prediction of system actions.

The outline of this paper is as follows: In Section II, we review related work in data-driven dialog modeling. In Section III, we describe the data we use in our experiments. In Section IV, we present our dialog models. In Section V, we describe our experimental method. In Section VI, we report experimental results using call center data from a catalog ordering service as well as publicly available corpora. The paper ends with a conclusions section (Section VII) describing current and future work.

II. Related Work

We use the phrase "dialog model" to refer to any structure or set of structures that abstractly represents important/interesting features of a set of dialogs. In this section, we compare three approaches to dialog modeling: plan-based, information state-based, and dialog act-based.

In the plan-based approach to dialog modeling, the process of a task-oriented dialog is treated as a special case of AI-style plan recognition (e.g., [1], [3]–[7]). Plan-based dialog models are used for both interpretation of user utterances and prediction of system actions.

- *Interpretation of user utterances*—Some researchers have built stochastic plan recognition models for interaction, including ones based on hidden Markov models [8], [9] and ones based on probabilistic context-free grammars [10], [11]. In recent work, Purver *et al.* take a first step towards detecting plans in meeting transcripts by using supervised learning to detect action items in meeting records [12], [13].
- *Prediction of system actions*—In recent work, Lewis and Di Fabbrizio [14], [15] used reinforcement learning in a planning-style dialog manager for system prompt selection.

We follow the plan-based approach in treating task structure as crucial to dialog modeling (see Section IV). However, for both user utterance interpretation and system action prediction we compare the performance of models that use task structure to those that do not (see Section V).

Another approach to dialog modeling is the information state-based approach, which focuses on the information to be communicated rather than on the task structure that leads to that communication (e.g., [16]–[18]). In recent years, there has been considerable research on how to automatically learn dialog models through reinforcement learning in an information state-based framework. These dialog models are used to predict system actions. Although most of this research uses user simulations to train dialog manager parameters ([19]–[23]), Henderson *et al.* have shown that it is possible to automatically acquire good dialog management strategies using reinforcement learning from automatically labeled dialog data [24]. They automatically annotated user utterances in Communicator for speech act and task as well as other information [25] and used this data for their reinforcement learning experiments. Their learned policy performed 37% better than the best performing Communicator system [24]. This performance was achieved using quite rich information state models, but Frampton and Lemon [26] have shown that the use of limited context information (the user's last dialog act) can still give considerable improvement over hand-designed dialog strategies. Unlike most work done on statistical dialog modeling in the information state-based approach, we are concerned with both determination of system actions *and* with interpretation of user utterances.

Finally, the dialog act-based approach to dialog modeling is based on tracking sequences of interactions between the dialog partners. Dialog act tags are used to abstract over interactions, and finite-state approaches are often used to abstract over interaction sequences [27]. There has been considerable research on statistical dialog act tagging. Corpora used for this task include corpora of human–human task-oriented dialog such as the TRAINS corpus [28], ATIS-like data [29], the VerbMobil corpus [30], [31], and the Maptask corpus [32]–[35]; the Switchboard corpus of human–human conversational dialog [36]–[38]; the ICSI MRDA meeting corpus of human–human multiparty interaction [39]–[41]; and human–computer spoken dialog (the Communicator multisystem corpora [42]).

Features used include speaker ID, speaker role (e.g., giver, follower), word *n-grams* from the current utterance, utterance length and position in the dialog, prosodic features, and contextual features (e.g., the dialog act of the previous utterance, the speaker of the previous utterance, and the task or dialog game of the current utterance). Best reported classification accuracy varies by dialog genre and tag set size: on VerbMobil, 75.12% [31] (163 dialogs, 18 tags); on Maptask, 73.91% [34]; on Switchboard, 71.29% [38]; on ICSI meeting data, 98.27% [41]; and on Communicator 2000, 98.5% [42], [43]. So far, dialog act-based dialog models have been used primarily for interpretation of user utterances and user utterance prediction [36], and for evaluation of dialog systems [44].

Our dialog models incorporate both dialog act and task/subtask information. As far as we know, only three other publications report work on statistical dialog modeling that makes extensive use of both dialog acts and task information. The first is by Poesio and Mikheev [32] who report that using game information from Maptask improves classification accuracy for dialog act tagging. However, they did not learn a model for task structure, but assumed that task/subtask information was given. The second is by Hardy and colleagues [45]. They used a large corpus of transcribed and annotated telephone conversations to develop the Amitiés dialog system. For their dialog manager, they trained separate task and dialog act classifiers on this corpus.

In their system, task information is used to label the type of the conversation (similar to call type labeling in a call routing task), while in our approach task/subtask information is used to classify user utterances and predict system actions. The third is by Chotimongkol [46]. She used unsupervised clustering and segmentation techniques to identify concepts and subtasks that occur in task-oriented dialogs but did not apply the automatically acquired dialog models to dialog tasks.

III. Data

Each type of dialog data (human–human, Wizard of Oz, human–computer) exhibits different behavior distributions. Corpora of human–computer dialog are highly structured, and contain speech recognition and understanding errors that do not typically occur in human–human dialog. Corpora of human–human dialog generally contain more variety in task structure and in language use, but do not exhibit the range of errors seen in human–computer dialog. Data collected with a Wizard of Oz setup have less variation in task structure and language use, as well as fewer (or at least more predictable) errors/simulated errors. In our research, we want to model the range of variation that occurs in task structure in dialog, so we chose to work with human–human corpora.[1]

As our primary data set, we used the CHILD corpus of 832 telephone-based customer-agent dialogs in a catalog-ordering domain. Each dialog was transcribed by hand; all customers' identifying numbers (telephone, credit card, etc.) were removed for privacy reasons. The average dialog lasts for 3.71 min and includes 61.45 changes of speaker. A single customer service representative might participate in several dialogs, but customers are represented by only one dialog each. Although the majority of the dialogs are on-topic, some are idiosyncratic, including: requests for order corrections; transfers to customer service; and long out-of-domain asides. Automatic annotations applied to these dialogs include: utterance segmentation (Section III-A), part of speech tagging and supertagging (Section III-B), and named entity annotation (Section III-C). We annotated the dialogs by hand for dialog acts (Section III-D) and tasks/subtasks (Section IV).

For comparison, we also used two other data sets: the Maptask corpus of task-oriented spoken dialog and the Switchboard corpus of conversational spoken dialog. The Maptask corpus consists of 128 dialogs that have been transcribed, segmented into utterances, and annotated for syntactic structure, map references (named entities that are all locations), dialog acts

[1]The Communicator corpus is the only large human–computer corpus of task-oriented dialog, and the task structure in that corpus exhibits little variation.

TABLE I
DIALOG ACT LABELS FROM THE CATALOG DOMAIN

Type	Dialog act labels
Ask	Info
Explain	Address, Catalog, CC_Rel, Discount, Email, Name, Online_Issue, Order_Info, Order_Problem, Order_Status, Payment_Rel, Phone_Number, Product_Info, Promotions, Related_Offer, Return_Rel, Shipping, Store_Info
Convers- -ational	Abandoned, Ack, Apology, Correction, DontKnow, EndOfTask, Goodbye, Hello, Help, Hold, No, Other, Not(Information), Repeat, Thanks, Yes, YourWelcome
Report	Code, Online_Issue, Order_Problem, Price_Problem
Request	Address, Call_Transfer, Catalog, CC_Rel, Store_Info, Change_Customer_Info, Change_Order, Conf, Credit, Discount, Email, Info, Make_Order, Name, Order_Info, Order_Status, Payment_Rel, Phone_Number, Product_Info, Promotions, Return_Rel, Shipping
YNQ	Address, Email, Info, Name, Order_Info, Order_Status, Product_Info, Promotions, Related_Offer, Shipping

TABLE II
TASK/SUBTASK LABELS FROM THE CATALOG DOMAIN

Type	Task/subtask labels
Call-level	call-forward, closing, misc-other, opening, out-of-domain, sub-call
Task-level	check-availability, contact-info, delivery-info, discount, order-change, order-item, order-problem, payment-info, related-offer, shipping-address, special-offer, summary

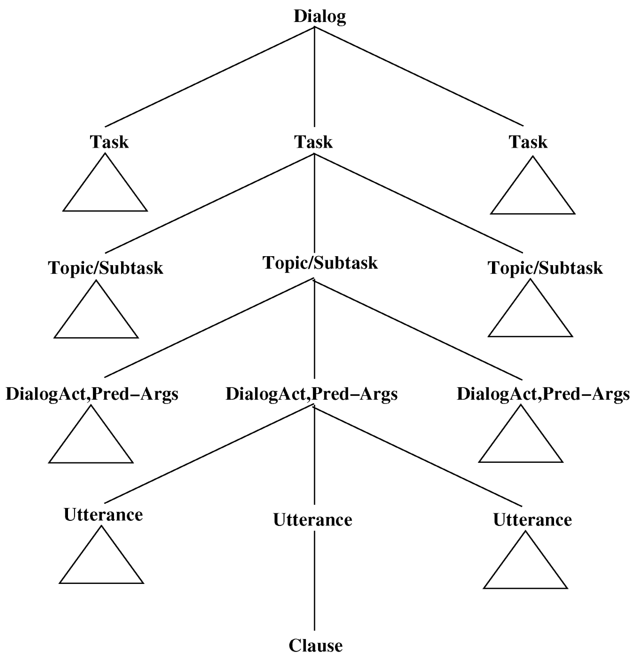

Fig. 1. Task- and dialog-act-based dialog model.

and dialog games (analogous to task/subtask annotations in our corpus). The Switchboard corpus consists of 1155 dialogs which have been transcribed, segmented into utterances, and annotated for syntactic structure and dialog acts. For consistency, we re-annotated these dialogs with part of speech tags and supertags using our automatic annotation tool. The CHILD corpus has the most fine-grained set of dialog act tags (75 domain-adapted tags versus 12 tags for Maptask and 49 clustered tags for Switchboard; see Table I) and the most specific set of task/subtask tags (see Table II). Maptask dialogs are annotated for game information rather than for task/subtask information; Switchboard dialogs (which are not task-oriented) are not annotated for task/subtask or conversational game.

A. Utterance Segmentation

The task of "cleaning up" spoken language utterances by detecting and removing speech repairs and dysfluencies and identifying sentence boundaries has been a focus of spoken language parsing research for several years (e.g., [47]–[50]). We use a system that segments the ASR transcribed output of a user's utterance into clauses. The system annotates an utterance for sentence boundaries, restarts and repairs, and identifies coordinating conjunctions, filled pauses, and discourse markers. These annotations are done using a cascade of classifiers, details of which are described in [51].

B. Predicate-Argument Annotation

We automatically annotate a user's utterance with supertags [52]. Supertags encapsulate predicate-argument information in a local structure. They can be composed with each other using the substitution and adjunction operations of Tree-Adjoining Grammars [53] to derive a dependency analysis of an utterance and its predicate-argument structure. We use a maximum entropy based supertagger with 4726 tags to annotate utterances with supertags.

C. Named Entity Annotation

We use a set of regular expressions to automatically label certain types of named entity, including person names, organization names, locations (addresses), dates/times, quantities, and phone numbers.

D. Dialog Act Tagging

We use a domain-specific dialog act tagging scheme based on an adapted version of DAMSL [28]. The DAMSL scheme is quite comprehensive, but as others have also found [36], the multidimensionality of the scheme makes the building of models from DAMSL-tagged data complex. Furthermore, the abstraction of the DAMSL tags reduces their utility for natural language generation. Other tagging schemes, such as the Maptask scheme [54], are also too general for our purposes. We were particularly concerned with obtaining sufficient discriminatory power between different types of statements (for generation), and to include an out-of-domain tag (for interpretation). Thus, we created our own set of tailored dialog act tags in Table I.

E. Task/Subtask Labeling

We used a domain-specific annotation scheme for labeling tasks and subtasks. The task/subtask labels are shown in Table II.

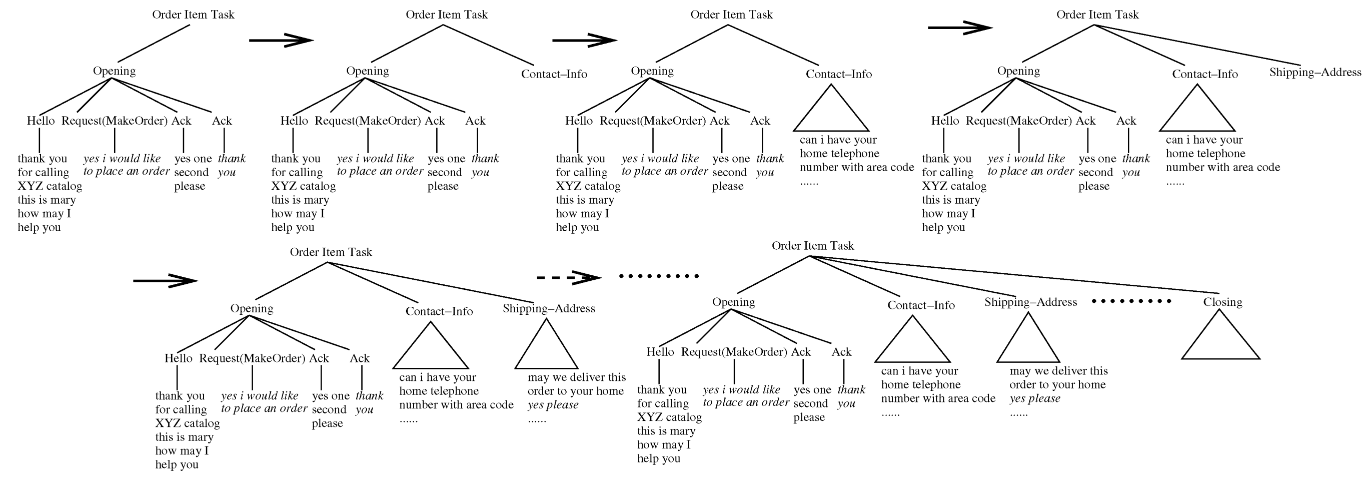
Order Item Task
Opening
Contact–Info
Shipping–Address
Closing
Hello Request(MakeOrder) Ack Ack
thank you for calling XYZ catalog this is mary how may I help you
yes i would like to place an order
yes one second please
thank you
can i have your home telephone number with area code ......
may we deliver this order to your home yes please ......

Fig. 2. Illustration of incremental evolution of dialog structure.

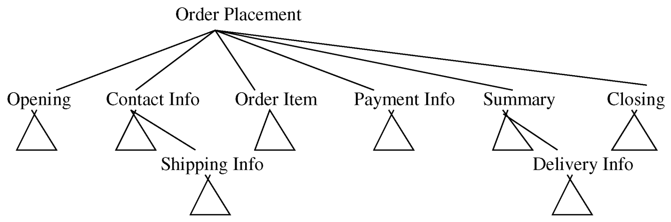

Fig. 3. Sample output (task/subtask structure) from a parse-based model for the catalog service domain.

IV. Dialog Models Based on Dialog Acts and Task Structure

We consider a task-oriented dialog to be the result of incremental creation of a shared plan by the participants [1]. The shared plan is represented as a single tree that encapsulates the *task* structure (dominance and precedence relations among tasks), *dialog act* structure (sequences of dialog acts), and *predicate-argument* structure of utterances (captured through inter-clausal relations and supertags in each clause), as illustrated in Fig. 1. In our representation, a task is comprised of a sequence of subtasks. A subtask is comprised of a sequence of dialog acts. Each dialog act corresponds to one clause, spoken by one speaker. A speaker's turn may consist of multiple clauses. A subtask frequently spans multiple turns, and may begin or end within a turn.

As the dialog proceeds, an utterance from a participant is accommodated into the tree in an incremental manner, much like an incremental syntactic parser accommodates the next word into a partial parse tree [10]. An illustration of the incremental evolution of dialog structure is shown in Fig. 2. With this model, we can tightly couple language understanding, dialog management, and response generation using a shared representation.

In this paper, we use our shared plan structure for interpretation of user utterances and prediction of system actions.

Fig. 3 shows the task structure for a sample dialog in our domain (catalog ordering). An *order placement* task is typically composed of the sequence of subtasks *opening, contact-information, order-item, related-offers, summary*. Subtasks can be nested; the nesting structure can be as deep as five levels in our data from the catalog domain. Most often, the nesting is at the left-most or right-most frontier of the subtask tree.

Table II shows the set of task/subtask labels with which the CHILD data is annotated.

We use the following notation in the presentation of these two models. The lexical, syntactic, and semantic information (e.g., words, part of speech tags, predicate-argument structures, and named entities) associated with speaker u's ith clause of is represented as c_i^u. For a speaker u, DA_i^u refers to the dialog act label of the ith clause, and ST_i^u refers to the subtask label to which the ith clause contributes. Where the speaker is not fixed, we drop the superscript.

Interpretation: In our model, each clause produced by the user is the realization of a user action, which is defined by: a subtask to which the clause contributes; the dialog act of the clause; and the named entities in the clause. Each user turn is recognized by the speech recognizer and split into clauses. Each clause is supertagged and labeled with named entities. So the input to the interpretation process (c_i^u) includes: the words from the clause produced by the user; the part of speech tags for the words; the supertags for the clause and the named entities mentioned in the clause.

We model the interpretation process in two stages. In the first stage, the dialog act of the clause is determined from the information about the clause and the previous dialog context as shown in (1). We use k previous utterances as the dialog context in (1). In the second stage, the subtask of the clause is determined from the lexical information about the clause, the dialog act assigned to the clause according to (1), and the dialog context, as shown in (2).

System Action Determination: Once we know the user's action, we must determine the action the system will take in response. Each system action is defined in terms of the subtask to which it contributes and the dialog act to be performed. The determination of the system action, therefore, also proceeds in two stages: prediction of subtask, and prediction of dialog act. In particular, we predict the subtask ST_i^a of c_i^a as shown in (3). Then, we predict the dialog act DA_i^a of c_i^a as shown in (4). The generation of c_i^a is not addressed in this paper, but preliminary results are reported in [55].

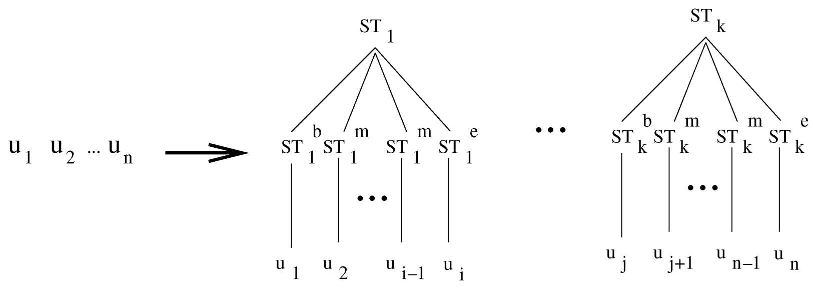

Fig. 4. Illustration of the chunk-based subtask label annotation.

TABLE III
EQUATIONS USED FOR MODELING DIALOG ACT AND SUBTASK LABELING OF AGENT AND USER UTTERANCES. c_i^u = THE WORDS, SYNTACTIC INFORMATION AND NAMED ENTITIES ASSOCIATED WITH THE ith UTTERANCE OF THE DIALOG, SPOKEN BY USER u. DA_i^u = THE DIALOG ACT OF THE ith UTTERANCE, SPOKEN BY USER u. ST_i^u = THE SUBTASK LABEL OF THE ith UTTERANCE, SPOKEN BY USER u. DA_{i-1}^{i-k} REPRESENTS THE DIALOG ACT TAGS FOR UTTERANCES $i-1$ TO $i-k$

Interpretation of a user's utterance:

$$DA_i^u = \underset{d^u \in D}{argmax}\ P(d^u | c_i^u, ST_{i-1}^{i-k}, DA_{i-1}^{i-k}, c_{i-1}^{i-k}) \quad (1)$$

$$ST_i^u = \underset{s^u \in S}{argmax}\ P(s^u | DA_i^u, c_i^u, ST_{i-1}^{i-k}, DA_{i-1}^{i-k}, c_{i-1}^{i-k}) \quad (2)$$

System action determination:

$$ST_i^a = \underset{s^a \in S}{argmax}\ P(s^a | ST_{i-1}^{i-k}, DA_{i-1}^{i-k}, c_{i-1}^{i-k}) \quad (3)$$

$$DA_i^a = \underset{d^a \in D}{argmax}\ P(d^a | ST_i^a, ST_{i-1}^{i-k}, DA_{i-1}^{i-k}, c_{i-1}^{i-k}) \quad (4)$$

We next present two ways of recovering subtask structure. In the *chunk-based* approach we simply group utterances into a subtask without positing any dominance relations among subtasks. In the *parse-based* approach, we attempt to recover the dominance relations as well. There is evidence in sentence analysis literature that the chunking approaches are more robust than parse-based approaches; however, they produce shallower representations compared to parse-based models.

A. Chunk-Based Approach

In the chunk-based approach, we simply label each utterance with the subtask label that it is to be assigned. The labels thus determine the span of a chunk. In this way, we recover the precedence relations (sequence) of the subtasks but not dominance relations (subtask structure) among the subtasks.

This representation of the chunked subtask labels is used to train models and to assign subtask labels for the agent and the user utterances using the equations shown in Table III.

B. Parse-Based Approach

As seen in Fig. 4, the chunk model does not capture dominance relations among subtasks, which are important for aspects such as resolving anaphoric references [2], among others. Also, the chunk model is representationally inadequate for center-embedded nestings of subtasks, which do occur in our domain, although less frequently than the more prevalent "tail-recursive" structures. In the parse-based model, we recover the complete task structure from the sequence of utterances as shown in Fig. 3. The subtask structure is recovered as a result of shift-reduce parsing process. The dialog subtask tree structure is converted into a binary branching tree. We construct feature vectors with the different contextual information and associate the vector with the appropriate parser action in order to train a classifier. The actions for the parser include *unary-reduce-X*, *binary-reduce-X*, and *shift*, where X is each of the nonterminals in the dialog tree. These feature vectors with the associated parser action is used to train a maximum entropy (MaxEnt) model [56]. This model is used during decoding to predict the structure of a new dialog.

C. Classifier

In order to estimate the conditional distributions shown in Table III, we use the general technique of choosing MaxEnt distribution that properly estimates the average of each feature over the training data [56]. This can be written as a Gibbs distribution parameterized with weights $\boldsymbol{\lambda}$, where V is the size of the label set. Thus

$$P(X = st_i | \boldsymbol{\Phi}) = \frac{e^{\boldsymbol{\lambda}_{st_i} \cdot \boldsymbol{\Phi}}}{\sum_{st=1}^{V} e^{\boldsymbol{\lambda}_{st} \cdot \boldsymbol{\Phi}}} \quad (5)$$

where X is ST_i^u, DA_i^u, ST_i^a, or DA_i^a and $\boldsymbol{\Phi}$ is a feature vector. We use the machine learning toolkit LLAMA [57] to estimate the conditional distribution using MaxEnt. LLAMA encodes multiclass classification problems using binary MaxEnt classifiers in order to increase the speed of training and to scale this method to large data sets. Each of the V classes is encoded as a bit vector such that, in the vector for class i, the ith bit is one and all other bits are zero. Then, V one-versus-other binary classifiers are used as follows:

$$P(y|\Phi) = 1 - P(\bar{y}|\Phi) = \frac{e^{\boldsymbol{\lambda}_y \cdot \Phi}}{e^{\boldsymbol{\lambda}_y \cdot \Phi} + e^{\boldsymbol{\lambda}_{\bar{y}} \cdot \Phi}} = \frac{1}{1 + e^{-\boldsymbol{\lambda}'_y \cdot \Phi}} \quad (6)$$

where $\boldsymbol{\lambda}_{\bar{y}}$ is the parameter vector for the *anti-label* $\bar{y}$ and $\boldsymbol{\lambda}'_y = \boldsymbol{\lambda}_y - \boldsymbol{\lambda}_{\bar{y}}$. We use the class independence assumption and require that $y_i = 1$ and for all $j \neq i\ y_j = 0$

$$P(st_i|\Phi) = P(y_i|\Phi) \prod_{j \neq i}^{V} P(y_j|\Phi).$$

D. Features

Offline natural language processing systems, such as part-of-speech taggers and chunkers, rely on both *static* and *dynamic* features. Static features are derived from the local context of the text being tagged. Dynamic features are computed based on previous decisions.

The static features we used in our experiments are as follows:

- the speaker ID for each utterance;
- unigrams, bigrams, and trigrams of the words in each utterance;
- unigrams, bigrams, and trigrams of the part of speech tags in each utterance;
- unigrams, bigrams, and trigrams of the supertags in each utterance;
- binary features indicating the presence or absence of particular types of named entity in each utterance (these features are not available for Switchboard).

The dynamic features we used in our experiments are as follows:

- the dialog act of each utterance;
- the task/subtask label of each utterance (these features are not available for Switchboard);
- for our parsing models, we also include as a dynamic feature the stack for the current utterance.

E. Decoder

The chunk- and parse-based subtask models depend on the dialog acts of agent and customer. In turn, the dialog act models depend on the subtask structure. Given this mutual dependency, we use a single decoder that predicts the dialog acts and subtask structure for agent and customer utterances. Also, since we want a left-to-right incremental decoder, we do not build a complete search network for the dialog (as is typically done for part-of-speech tagging, or syntactic parsing). The current implementation of the decoder is strictly *greedy*, by which we mean that one decision (dialog act or subtask label) is considered for each utterance. We are investigating potential extensions for the decoder which involve maintaining multiple local hypotheses for an utterance.

As mentioned above, we train different classifiers for dialog acts and subtask labels for the agent and customer utterances using a MaxEnt model. During decoding, the decoder simply queries the appropriate classifier (DA-customer, DA-agent, Subtask-agent, Subtask-customer) with a suitable feature vector constructed based on past decisions. The highest scoring prediction from the classifier is used as the greedy hypothesis for the current utterance.

The DA tagging model and the chunk-based structure prediction model assign a single tag for each utterance. For the tree-based structure prediction model, we use a shift-reduce parser to assign the structure for a dialog. This model is similar to the deterministic shift-reduce parsers proposed in recent literature for syntactic parsing of sentences [58]–[60] with the essential difference being that our parser is strictly incremental, that is, no information from the right context of the utterance is being used in the prediction of the parser action. Such a constraint is typically used in incremental shift-reduce syntactic parsing of sentences explored in the psycholinguistics literature [61].

V. Experiments

In this section, we report the results for dialog act and subtask classification and prediction on the Maptask, Switchboard and CHILD data sets. Maptask is a much smaller corpus than either Switchboard or CHILD. So for Maptask, we report results based on tenfold cross-validation. For the other two corpora, we report results based on a single 90%/10% training/testing split.

For each task (user utterance dialog act classification, user utterance task/subtask classification, system utterance task/subtask prediction, and system utterance dialog act prediction), we report three sets of results:

- **Static (S)**—results for that task using only static features;
- **Dynamic decoding (D)**—results for that task using static features and the values for dynamic features determined by dynamic decoding;
- **Oracle decoding (O)**—results for that task using static features and the true values for dynamic features (best possible).

For those corpora for which we have task/subtask labels, we build chunk- and parse-based task models and use them for static decoding (**SC**, **SP**), dynamic decoding (**DC**, **DP**), and oracle decoding (**OC**, **OP**). We also report **Baseline** performance, which in all cases is the performance that would be obtained by ignoring the input features and assigning to an utterance the most likely label from the training data.

In our experiments, we report results for different amounts of left-hand context: 0 utterances (the feature vector contains only features from the current utterance), 1 utterance (the feature vector contains only features from the current utterance and one previous utterance), 3 utterances, and 5 utterances. Since the average turn in our dialogs contains about 1.4 utterances (Maptask: 1.4 utterances/turn; Switchboard: 1.85 utterances/turn), 1 utterance of left-hand context is similar to capturing only the current turn; 3 utterances is similar to having one turn of left-hand context; and 5 utterances is similar to having two turns of left-hand context.

For each experiment, we report classification error rate.

VI. Results

A. User Utterance Dialog Act Classification

Fig. 5 shows results for user utterance dialog act classification. For all three corpora, the Static and Dynamic decoding models outperform Baseline. One utterance of left-hand context gives improved performance, while the addition of extra utterances does not help further.

The Static model achieves performance quite similar to Oracle decoding (best possible). Comparing the Static and Dynamic decoding models, we see that the inclusion of noisy task/subtask and dialog act information from previous utterances decreases performance compared to using utterance-level features alone. This is primarily due to the greedy approach adopted in the decoder for the dynamic model, where the highest scoring class from the classifier is used as the feature in the decoding of the next utterance. This results in a cascading of errors. The

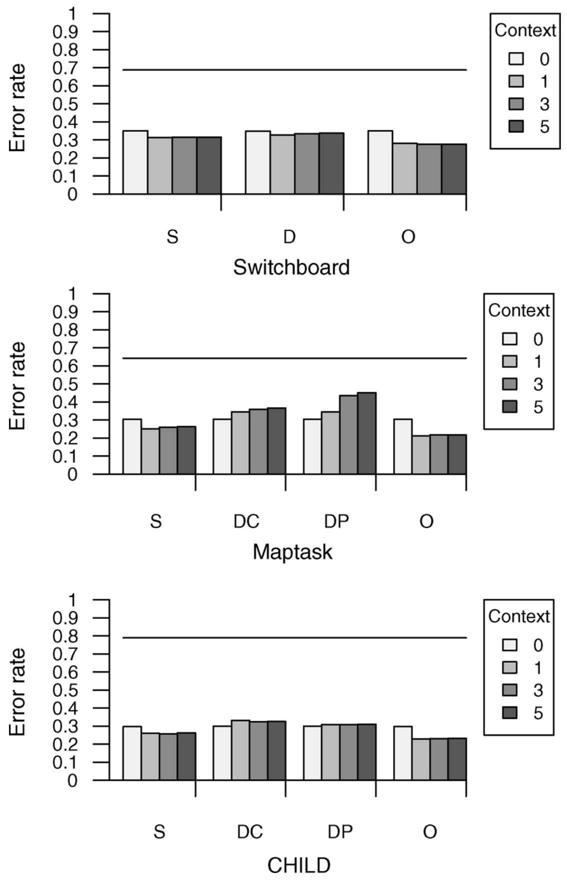

Fig. 5. Error rates for user dialog act classification. S= Static, D= Dynamic, DC= Dynamic Chunk-based, DP= Dynamic Parse-based, O= Oracle. Baseline shown as horizontal line on chart.

chunk-based model outperforms the parse-based model on dialog act classification because it is less susceptible to cascading errors.

For Maptask, our best-performing model (Static with one utterance of left-hand context) slightly outperforms the best-performing dialog act classifier for this corpus in the literature (error-rate of 25.1% compared to 26.1% [34]). For both Maptask and CHILD, the best performing models achieve 1-best error rates of under 30%.

We conclude that it is possible to achieve high performance on dialog act classification using only utterance-level features, with limited context.

Table IV shows part of the confusion matrix for the best-performing Static model for user dialog act classification for CHILD. Most of the classification errors are to be expected; however, we plan to improve the model with better discrimination between task-relevant dialog acts (e.g., *Explain*) and non-task-relevant ones (e.g., *Not(Information)*).

B. User Utterance Task/Subtask Classification

Fig. 6 shows results for user utterance task/subtask classification. Note that since Switchboard is not labeled with task/subtask information, there are no results for this domain. For both corpora, the Static and Dynamic Chunk-based decoding models

TABLE IV
MOST FREQUENTLY OCCURRING USER DIALOG ACT TAGS IN CHILD, WITH MISCLASSIFICATIONS THAT OCCUR TEN OR MORE TIMES

Tag	Most frequent errors
Acknowledge(622/747)	Yes (43), Explain(Product_Info) (22), Not(Information) (18)
Explain(Product_Info) (260/356)	Not(Information) (26), YNQ(Product_Info) (21), Explain(Order_Info) (14)
Yes (273/318)	Acknowledge (35)
Not(Information) (199/304)	Acknowledge (30), Explain(Product_Info) (28), Yes (13)
Explain(Order_Info) (106/164)	Explain(Product_Info) (19), Acknowledge(11), Not(Information) (11)
YNQ(Product_Info) (37/102)	Explain(Product_Info) (33), Not(Information) (11)
Abandoned (16/55)	Not(Information) (18)

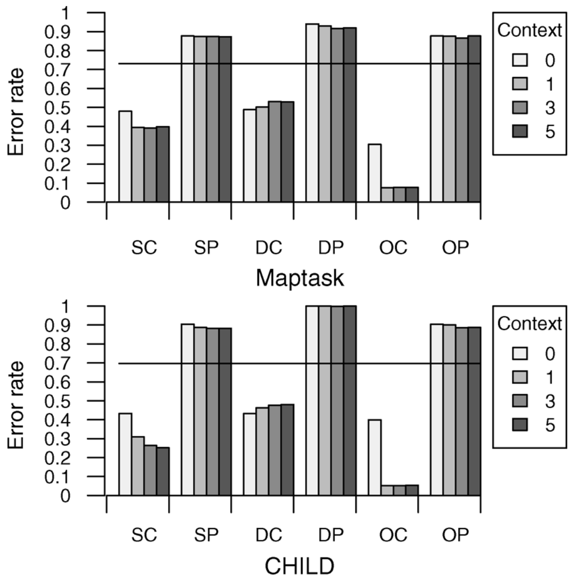

Fig. 6. Error rates for user task/subtask classification. SC= Static Chunk-based, SP= Static Parse-based, DC= Dynamic Chunk-based, DP= Dynamic Parse-based, OC= Oracle Chart-based, OP= Oracle Parse-based. Baseline shown as horizontal line on chart.

outperform Baseline. However, no model approaches the performance of Oracle Chunk-based decoding (best possible). In general, one utterance of left-hand context gives improved performance, while including additional utterances does not improve performance further. However, for CHILD for Static Chunk-based decoding, including additional utterances of context does improve performance. This may be because in CHILD, the task/subtask chunks tend to be longer than the "games" in Maptask.

Comparing the Static model to the Dynamic (Chunk-based) model, we can see that the inclusion of noisy task/subtask and dialog act information from previous utterances decreases performance. For both models, errors tend to occur around subtask boundaries.

For both corpora, the Parse-based models perform worse than Baseline. The parser tends to shift as long as possible, so many utterances end up with no task/subtask label. A model that accounts for the length (number of utterances) of a subtask is likely to alleviate this issue.

TABLE V
MOST FREQUENTLY OCCURRING USER TASK/SUBTASK LABELS IN CHILD, WITH MISCLASSIFICATIONS THAT OCCUR TEN OR MORE TIMES

Tag	Most frequent errors
order-item (1171/1264)	misc-other (36), order-problem (18)
contact-info (343/389)	order-item (29)
payment-info (243/265)	misc-other (11)
delivery-info (165/222)	summary (15)
misc-other (73/221)	delivery-info (16), discount (13), order-item (83)
out-of-domain (42/200)	delivery-info (10), misc-other (42), order-item (72), order-problem (10)

In general, models for task/subtask classification for CHILD perform better than those for Maptask. Task/subtasks in CHILD relate to the domain (see Fig. 3), while Maptask is labeled instead with dialog games, which relate more to the conversation. We conjecture that the task/subtask labels in CHILD are more constraining.

We conclude that it is possible to achieve high performance (<25% error rate) on user task/subtask classification using only utterance-level features with limited context. However, it may be possible to improve performance considerably if we can improve the performance of user dialog act classification.

Table V shows part of the confusion matrix for the best-performing Static model for user task/subtask classification for CHILD. Some misclassifications (e.g., *misc-other* for *out-of-domain*) are true misclassifications, while others (e.g., *order-item* for *out-of-domain*) are "boundary" misclassifications, i.e., the system is either skipping a very short subtask or mislabeling clauses at the edge of a subtask.

C. System Utterance Task/Subtask Prediction

The purpose of getting dialog act and task/subtask labels for user utterances is so that task-related information can be used to predict the system's next actions. In our framework, each clause in the system's next action is summarized using a task/subtask label and dialog act. Fig. 7 shows results for system utterance task/subtask prediction. Note that this task is different from user task/subtask classification, where the current utterance c_i is used for classification decisions. Hence, the prediction performance is worse than that for user task/subtask classification. However, for the CHILD corpus performance of the best performing prediction models approaches that of the best performing classification models. Apart from this notable difference, results are similar to those for user subtask classification.

We conclude from the Oracle results that it may be possible to achieve very good system task/subtask prediction performance (for CHILD, error rates of <20%) with simple chunk-based models. However, considerable improvement to dialog act classification and prediction modeling is necessary to achieve this goal.

Table VI shows part of the confusion matrix for the best performing Static model for system task/subtask prediction for CHILD. Many of the errors are "boundary" errors, where the system is "jumping ahead" or "hanging behind" in predicting a change of subtask (e.g., errors on *contact-info*, *payment-info* errors on *order-item*). Others are due to the system failing to predict an interjection of some kind (e.g., a *misc-other*). Of course, an actual dialog system would not produce utterances not related to a task (no *misc-other* or *out-of-domain* utterances). As with the system utterance dialog act prediction, there are some effects due to the use of human–human data.

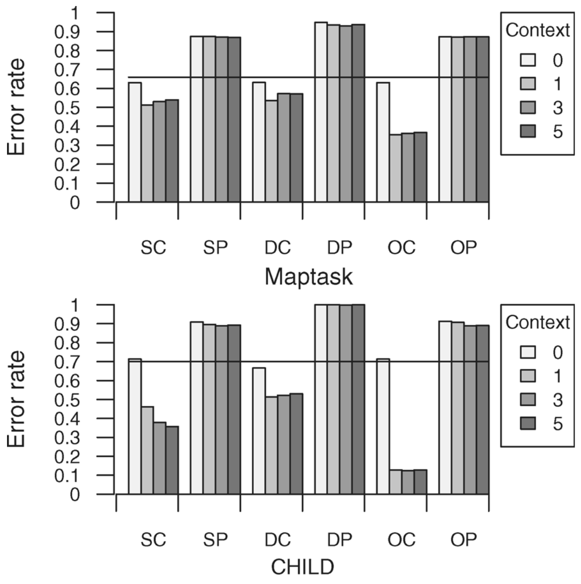

Fig. 7. Error rates for system task/subtask prediction. SC= Static Chart-based, SP= Static Parse-based, DC= Dynamic Chunk-based, DP= Dynamic Parse-based, OC= Oracle Chart-based, OP= Oracle Parse-based. Baseline shown as horizontal line on chart.

TABLE VI
MOST FREQUENTLY OCCURRING SYSTEM TASK/SUBTASK LABELS IN CHILD, WITH MISCLASSIFICATIONS THAT OCCUR 15 OR MORE TIMES

Tag	Most frequent errors
order-item (1188/1323)	misc-other (36), order-problem (15), payment-info (15)
contact-info (305/402)	opening (18), order-item (44), shipping-address (15)
payment-info (239/281)	order-item (20)
delivery-info (146/274)	order-item (43), summary (24)
misc-other (72/263)	delivery-info (16), order-item (104)

D. System Utterance Dialog Act Prediction

Fig. 8 shows results for system utterance dialog act prediction. Our first observation is that performance for system dialog act prediction is considerably worse than for user dialog act classification. This is because for dialog act classification we know the words and syntax of the current utterance, while for dialog act prediction we do not (c_i^a is missing).

For all three corpora, the Static model and the Chunk-based Dynamic decoding model outperform Baseline. As with dialog act classification, in general one utterance of left-hand context gives improved performance, while the addition of extra utterances does not help further.

Only in the case of Switchboard does the performance of the Dynamic decoding model approach that of the Oracle decoding model. The system needs more information about the global dialog state, and/or better predictive models for the task/subtask, in order to do better on task-based dialog.

The poor performance of the Parse-based model is due to cascading shift errors which lead to the subtask of many utterances being labeled as the empty string.

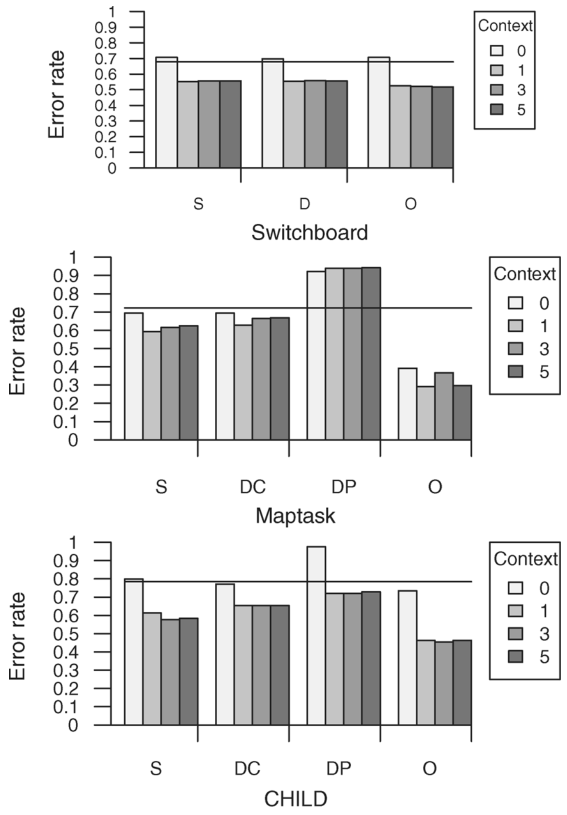

Fig. 8. Error rates for system dialog act prediction. S= Static, D= Dynamic, DC= Dynamic Chunk-based, DP= Dynamic Parse-based, O= Oracle. Baseline shown as horizontal line on chart.

We conclude from the Oracle results that it may be possible to achieve system dialog act prediction accuracy of around 60% overall (including both task-relevant utterances and non-task-relevant utterances). However, we need to improve task/subtask modeling to achieve this goal.

Table VII shows part of the confusion matrix for the best performing Static model for system utterance dialog act prediction for the CHILD corpus. In any prediction task, there may be several valid predictions; for example, following a request one may either acknowledge the request or may immediately answer the request. This confusion matrix shows that it may be better to model grounding contributions separately from task-related contributions. There may also be better evaluation methods for this task than a simple 1-best error rate.

E. Discussion

In general, in these experiments we found that one utterance of left-hand context (in addition to the current utterance for classification tasks) improves performance, but additional utterances do not give further improvements. Clearly, local dialog context does influence interpretation of the current utterance. However, we do not think that only the most immediate dialog context influences interpretation. Ideally, we would be able also to use global constraints on the dialog task structure, such as the order of subtasks in the training data. This would help particularly with task/subtask prediction and classification at task/subtask boundaries.

Our chunk- and parse-based models do a much better job of determining when the subtask is staying the same than of determining when it is changing. The experiments reported in this paper show that a flat model of dialog structure (our chunk-based model) performs better than one type of hierarchical model (our shift-reduce parse-based model). However, the chunk-based model only performs better than the parse-based model at determining the lowest level of task structure—only the parse-based model can track how subtasks nest in each other. Also, in the chunk-based model, the classifier assigns task labels to utterances, building the task structure directly. In the parse-based model, by contrast, the classifier assigns stack operations to utterances, and task structure is the result of interpreting the stack operations. This may lead to more error propagation. More generally, although the use of structure information as *features* did not improve the models, the goal of this paper is to illustrate techniques that learn to *output* a dialog structure. We are experimenting with alternative parse-based models that may perform better than the straightforward one presented here.

For these experiments we had limited and noisy named entity annotations, and no annotations for domain concepts (other than those in the dialog act tags). To improve our prediction models in particular, we could use abstractions over the global dialog state along the lines of the information state-based approach (e.g., more information about which named entities have been recognized, or which task slots filled) [25].

VII. Conclusion

In this paper, we use a classification-based approach to automatically create task structures for task-oriented dialogs. We apply this to interpretation of user utterances and prediction of system actions. We use a dialog modeling approach that tightly couples dialog act and task/subtask information. We show that the following.

- Our approach to dialog act classification of user utterances is highly successful, producing results that exceed those previously published with the use of limited context information and utterance-only features.
- Our chunk-based approach to task/subtask classification of user utterances is highly successful, but that there is considerable room for improvement;
- Best-possible performance of our approaches to system task/subtask and dialog act prediction indicates promise; however, improvements are needed.
- The use of highly constraining dialog acts and task/subtask labels leads to improved performance for both interpretation and generation tasks.

We are currently working on improvements to our models that better incorporate information about named entities and task structure. In future work, we plan to address aspects of interaction peculiar to human–computer dialog, and conduct an end-to-end evaluation in a working dialog system.

Acknowledgment

The authors would like to thank B. Hollister and her team for their effort in annotating the dialogs for dialog acts and subtask structure and P. Haffner for providing them with the LLAMA

TABLE VII
MOST FREQUENTLY OCCURRING SYSTEM DIALOG ACT TAGS IN CHILD, WITH MISCLASSIFICATIONS THAT OCCUR 15 OR MORE TIMES

Tag	Most frequent errors
Acknowledge (620/905)	Explain(Product_Info) (38), Not(Information) (37), Yes (36), Request(Product_Info) (27), Explain(Order_Info) (18)
Explain(Product_Info) (210/386)	Acknowledge (51), Not(Information) (33), Request(Product_Info) (15), YNQ(Product_Info) (15)
Not(Information) (54/303)	Acknowledge (121), Explain(Product_Info) (36)
Yes (120/242)	Acknowledge (79)
Explain(Shipping) (51/141)	Acknowledge (43)
Explain(Order_Info) (43/128)	Acknowledge (27), EndOfTask (19)
Request(Product_Info) (46/116)	Acknowledge (38), Explain(Product_Info) (17)
YNQ(Product_Info) (10/105)	Acknowledge (30), Explain(Product_Info) (27)
Request(Order_Info) (33/101)	Acknowledge (28)

machine learning toolkit used in our experiments. They would also like to thank A. Conkie, M. Gilbert, N. Gupta, and B. Stern for discussions during the course of this work and the anonymous reviewers whose comments have helped reshape this paper.

REFERENCES

[1] K. Lochbaum, "A collaborative planning model of intentional structure," *Comput. Linguist.*, vol. 24, no. 4, pp. 525–572, 1998.

[2] B. Grosz and C. Sidner, "Attention, intentions and the structure of discoursep," *Comput. Linguist.*, vol. 12, no. 3, pp. 175–204, 1986.

[3] C. Sidner, "Plan parsing for intended response recognition in discourse," *Comput. Intell.*, vol. 1, no. 1, pp. 1–10, 1985.

[4] D. Litman and J. F. Allen, "A plan recognition model for subdialogs in conversations," *Cogn. Sci.*, vol. 11, no. 2, pp. 163–200, 1987C. Rich and C. Sidner, "COLLAGEN: When agents collaborate with people," in *Proc. 1st Int. Conf. Autonomous Agents*, 1997, pp. 284–291.

[5] C. Rich and C. Sidner, "COLLAGEN: When agents collaborate with people," in *Proc. 1st Int. Conf. Autonomous Agents*, 1997, pp. 284–291.

[6] S. Carberry, "Techniques for plan recognition," *User Modeling and User-Adapted Interaction*, vol. 11, no. 1–2, pp. 31–48, 2001.

[7] D. Bohus and A. Rudnicky, "RavenClaw: Dialog management using hierarchical task decomposition and an expectation agenda," in *Proc. Eurospeech*, 2003, pp. 31–48.

[8] H. Bui, "A general model for online probabalistic plan recognition," in *Proc. Int. Joint Conf. Artif. Intell.*, 2003.

[9] N. Blaylock and J. F. Allen, "Hierarchical instantiated goal recognition," in *Proc. AAAI Workshop Modeling Others From Observations*, 2006, pp. 1309–1318.

[10] J. Alexandersson and N. Reithinger, "Learning dialogue structures from a corpus," in *Proc. Eurospeech*, 1997, pp. 8–15.

[11] D. Pynadath and M. Wellman, "Probabilistic state-dependent grammars for plan recognition," in *Proc. 16th Conf. Uncertainty Artif. Intell.*, 2000, pp. 2231–2234.

[12] M. Purver, P. Ehlen, and J. Niekrasz, "Detecting action items in multi-party meetings: Annotation and initial experiments," in *Proc. Joint Workshop Multimodal Interaction Related Mach. Learn. Algorithms*, 2006, pp. 507–514.

[13] W. Morgan, P. Chang, S. Gupta, and J. Brenier, "Automatically detecting action items in audio meeting recordings," in *Proc. SIGdial Workshop Discourse Dialogue*, 2006, pp. 200–211.

[14] C. Lewis and G. Di Fabbrizio, "Prompt selection with reinforcement learning in an AT&T call routing application," in *Proc. 9th Int. Conf. Spoken Lang. Process.*, 2006, pp. 96–103.

[15] C. Lewis and G. D. Fabbrizio, "Dialogue strategy optimization with reinforcement learning in an AT&T call routing application," in *Proc. AAAI Workshop Statist. Empirical Approaches Spoken Dialogue Syst.*, 2006, pp. 1770–1773.

[16] S. Larsson and D. Traum, "Information state and dialogue management in the TRINDI dialogue move engine toolkit," *Natural Lang. Eng. Special Issue Best Practice Spoken Lang. Dialogue Syst. Eng.*, vol. 6, no. 3–4, pp. 323–340, 2000.

[17] J. Bos, E. Klein, O. Lemon, and T. Oka, "DIPPER: Description and formalisation of an information-state update dialogue system architecture," in *Proc. SIGdial Workshop Discourse and Dialogue*, 2003, pp. 232–340.

[18] O. Lemon and A. Gruenstein, "Multithreaded context for robust conversational interfaces: Context-sensitive speech recognition and interpretation of corrective fragments," *ACM Trans. Comput.–Human Interaction*, vol. 11, no. 3, pp. 115–124, 2004.

[19] E. Levin and R. Pieraccini, "A stochastic model of computer–human interaction for learning dialogue strategies," in *Proc. Eurospeech*, 1997, pp. 241–267.

[20] K. Scheffler and S. Young, "Automatic learning of dialogue strategy using dialogue simulation and reinforcement learning," in *Proc. Human Lang. Technol. Conf.*, 2002, pp. 1883–1886.

[21] S. Singh *et al.*, "Optimizing dialogue management with reinforcement learning: Experiments with the NJFun system," *J. Artif. Intell. Res.*, vol. 16, pp. 105–133, 2002.

[22] J. Williams, P. Poupart, and S. Young, "Partially observable Markov decision processes with continuous observations for dialogue management," in *Proc. SIGdial Workshop Discourse and Dialogue*, 2005, pp. 105–133.

[23] J. Williams and S. Young, "Partially observable Markov decision processes for spoken dialog systems," *Comput. Speech Lang.*, vol. 21, no. 2, pp. 393–422, 2007.

[24] J. Henderson, O. Lemon, and K. Georgila, "Hybrid reinforcement/supervised learning for dialogue policies from COMMUNICATOR data," in *Proc. 4th IJCAI Workshop Knowledge and Reasoning in Practical Dialogue Syst.*, 2005, pp. 393–422.

[25] K. Georgila, O. Lemon, and J. Henderson, "Automatic annotation of COMMUNICATOR dialogue data for learning dialogue strategies and user simulations," in *Proc. DIALOR*, 2005, pp. 68–75.

[26] M. Frampton and O. Lemon, "Reinforcement learning of dialogue strategies using the user's last dialogue act," in *Proc. 4th IJCAI Workshop Knowledge and Reasoning in Practical Dialogue Syst.*, 2005, pp. 61–68.

[27] M. McTear, "Modelling spoken dialogues with state transition diagrams: Experiences with the CSLU toolkit," in *Proc. ICSLP*, 1998, pp. 83–90.

[28] M. Core, "Analyzing and predicting patterns of DAMSL utterance tags," in *Proc. AAAI Spring Symp. Applying Mach. Learn. Discourse Process.*, 1998, pp. 1223–1226.

[29] J. Chu-Carroll, "A statistical model for discourse act recognition in dialogue interactions," in *Proc. AAAI Spring Symp. Applying Mach. Learn. Discourse Process.*, 1998, pp. 18–24.

[30] N. Reithinger and M. Klesen, "Dialogue act classification using language models," in *Proc. Eurospeech*, 1997, pp. 12–17.

[31] K. Samuel, S. Carberry, and K. Vijay-Shanker, "Computing dialogue acts from features with transformation-based learning," in *Proc. AAAI Spring Symp. Applying Mach. Learn. Discourse Process.*, 1998, pp. 2235–2238.

[32] M. Poesio and A. Mikheev, "The predictive power of game structure in dialogue act recognition: Experimental results using maximum entropy estimation," in *Proc. Int. Conf. Spoken Lang. Process.*, 1998, pp. 90–97.

[33] H. W. Hastie, M. Poesio, and S. Isard, "Automatically predicting dialogue structure using prosodic features," *Speech Commun.*, vol. 36, no. 1–2, pp. 63–79, 2002.

[34] R. Serafin and B. D. Eugenio, "FLSA: Extending latent semantic analysis with features for dialogue act classification," in *Proc. Meeting Assoc. Comput. Linguist..*, 2004, pp. 63–79.

[35] D. Surendran and G.-A. Levow, "Dialog act tagging with support vector machines and hidden Markov models," in *Proc. Int. Conf. Spoken Lang. Process.*, 2006, pp. 692–699.

[36] D. Jurafsky *et al.*, "SwitchBoard discourse language modeling project report," Center for Speech and Lang. Process., Johns Hopkins Univ., Baltimore, MD, Tech. Rep. Res. Note 30, 1998.

[37] A. Stolcke *et al.*, "Dialogue act modeling for automatic tagging and recognition of conversational speech," *Comput. Linguist..*, vol. 26, no. 3, pp. 339–373, 2000.

[38] N. Webb, M. Hepple, and Y. Wilks, "Dialogue act classification based on intra-utterance features," in *Proc. AAAI 2005 Workshop Spoken Lang. Understanding*, 2005, pp. 339–373.

[39] J. Ang, Y. Liu, and E. Shriberg, "Automatic dialog act segmentation and classification in multiparty meetings," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2005, pp. 46–51.

[40] G. Ji and J. Bilmes, "Backoff model training using partially observed data: Application to dialog act tagging," in *Proc. Joint Meeting North Amer. Chapt. Assoc. Comput. Linguist.. Human Lang. Technol. Conf.*, 2006, pp. 1061–1064.

[41] D. Verbree, R. Rienks, and D. Heylen, "Dialogue-act tagging using smart feature selection: Results on multiple corpora," in *Proc. IEEE/ACL Workshop Spoken Lang. Technol.*, 2006, pp. 280–287.

[42] R. Prasad and M. Walker, "Training a dialogue act tagger for human–human and human–computer travel dialogues," in *Proc. SIGdial Workshop Discourse and Dialogue*, 2002, pp. 70–73.

[43] M. A. Walker, R. Passonneau, and J. E. Boland, "Quantitative and qualitative evaluation of DARPA Communicator spoken dialogue systems," in *Proc. Meeting Assoc. Comput. Linguist.*, 2001.

[44] H. W. Hastie, R. Prasad, and M. Walker, "Automatic evaluation: Using a DATE dialogue act tagger for user satisfaction and task completion prediction," in *Proc. Lang. Resources Eval. Conf.*, 2002, pp. 641–648.

[45] H. Hardy *et al.*, "Data-driven strategies for an automated dialogue system," in *Proc. Meeting of the Association for Comput. Linguist..*, 2004, pp. 71–78.

[46] A. Chotimongkol, "Learning the structure of task-oriented conversations from the corpus of in-domain dialogs," Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, PA, 2008.

[47] J. Bear, J. Dowding, and E. Shriberg, "Integrating multiple knowledge sources for detection and correction of repairs in human–computer dialog," in *Proc. Meeting Assoc. Comput. Linguist.*, 1992, pp. 56–63.

[48] S. Seneff, "A relaxation method for understanding spontaneous speech utterances," in *Proc. Speech Natural Lang. Workshop*, San Mateo, CA, 1992, pp. 299–304.

[49] E. Shriberg *et al.*, "Prosody-based automatic segmentation of speech into sentences and topics," *Speech Commun.*, vol. 32, no. 1–2, pp. 127–154, 2000.

[50] E. Charniak and M. Johnson, "Edit detection and parsing for transcribed speech," in *Proc. Conf. North Amer. Chapt. Assoc. Comput. Linguist.*, 2001, pp. 1–9.

[51] S. Bangalore and N. Gupta, "Extracting clauses in dialogue corpora: Application to spoken language understanding," *J. Traitement Automatique des Langues (TAL)*, vol. 45, no. 2, pp. 159–181, 2004.

[52] S. Bangalore and A. K. Joshi, "Supertagging: An approach to almost parsing," *Comput. Linguist..*, vol. 25, no. 2, pp. 237–265, 1999.

[53] A. K. Joshi, "An introduction to tree adjoining grammars," in *Math. Lang.*, A. Manaster-Ramer, Ed. Amsterdam, The Netherlands: John Benjamins, 1987.

[54] J. Carletta *et al.*, "The reliability of a dialog structure coding scheme," *Comput. Linguist..*, vol. 23, no. 1, pp. 13–31, 1997.

[55] A. Stent, S. Bangalore, and G. D. Fabrizzio, "Where do the words come from? Learning models for lexical selection and word ordering from spoken dialog corpora," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2008, pp. 5037–5040.

[56] A. Berger, S. Pietra, and V. Pietra, "A maximum entropy approach to natural language processing," *Comput. Linguist..*, vol. 22, no. 1, pp. 39–71, 1996.

[57] P. Haffner, "Scaling large margin classifiers for spoken language understanding," *Speech Commun.*, vol. 48, no. IV, pp. 239–261, 2006.

[58] J. Nivre and M. Scholz, "Deterministic dependency parsing of English text," in *Proc. COLING*, 2004, pp. 64–70.

[59] K. Sagae and A. Lavie, "A classifier-based parser with linear run-time complexity," in *Proc. Int. Workshop Parsing Technol.*, 2005, pp. 125–132.

[60] H. Yamada and Y. Matsumoto, "Statistical dependency analysis with support vector machines," in *Proc. Int. Workshop Parsing Technol.*, 2003, pp. 195–206.

[61] F. Keller, S. Clark, M. Crocker, and M. Steedman*,* in *Proc. ACL Workshop Incremental Parsing: Bringing Eng. Cognition Together*, 2004, Assoc. Comput. Linguist..

Srinivas Bangalore (SM'06) received the Ph.D. degree in computer science from the University of Pennsylvania, Philadelphia, in 1997.

He is currently a Principal Member of Technical Staff in the Voice and IP Services Laboratory, AT&T Labs-Research, Florham Park, NJ, working on speech and language processing including spoken language translation, multimodal understanding, and language generation. He has authored over 120 research publications in international journals, conferences, and workshops.

Dr. Bangalore's dissertation on "Supertagging" was awarded the Morris and Dorothy Rubinoff Award for Outstanding Dissertation that has resulted in or could lead to innovative applications of computer technology. He has been awarded the AT&T Outstanding Mentor Award in recognition of his support and dedication to the AT&T Labs Mentoring Program. He was a coeditor of the special issue of *Speech Communication* journal on Spoken Language Understanding in Conversational Systems. He is co-chair for the IEEE Workshop on Spoken Language Technologies 2008 in Goa, India. He has served as an editorial board member of the *Computational Linguistics Journal* and a program committee member for a number of ACL and IEEE Speech and Language Conferences. He is currently a member of the IEEE Speech and Language Technical Committee.

Giuseppe Di Fabbrizio (SM'01) received the M.S.E.E. degree in electrical engineering from the Politecnico di Torino, Turin, Italy, in 1990.

From 1990 to 1995, he was a Senior Researcher with Telecom Lab Italia (formerly CSELT), Turin. In January 1996, he joined AT&T Labs-Research, Florham Park, NJ, where he is currently a Lead Research Scientist in the IP and Voice Services Research Laboratory. During his career, he has conducted research mainly on spoken dialog systems, multimodal and speech architectures, and speech services, publishing more than 40 conference and journal papers on these subjects. He was instrumental in the development and deployment of the AT&T VoiceTone Natural Language Dialog Automation product for AT&T business enterprise customers.

Amanda Stent (M'07) received the B.A. degree in mathematics and music from Houghton College, Houghton, NY, in 1996 and the M.S. and Ph.D. degrees in computer science from the University of Rochester, Rochester, NY, in 1998 and 2001, respectively.

She is currently with AT&T Labs-Research, Florham Park, NJ. Her research interests include spoken dialog systems, natural language and multimodal generation, and corpus linguistics.