Statistical Shared Plan-Based Dialog Management

Amanda J. Stent, Srinivas Bangalore

AT&T Labs - Research, Inc., 180 Park Avenue, Florham Park, NJ 07932, USA
{stent,srini}@research.att.com

Abstract
In this paper we describe a statistical shared plan-based approach to dialog modeling and dialog management. We apply this approach to a corpus of human-human spoken dialogs. We compare the performance of models trained on transcribed and automatically recognized speech, and present ideas for further improving the models.

1. Introduction
In recent years, we have conducted a series of experiments aimed at modeling the task structure of human-human dialog for corpus analysis and dialog systems engineering [1, 2, 3]. In our previous work, we looked at dialog modeling and dialog act classification from transcribed speech. In this paper, we look at the impact on our model of using recognized speech as well as at how our model could be used to improve speech recognition (by predicting later user actions). The contributions of this work include: (1) the application of classification-based dialog models to interpretation of user utterances, prediction of system actions, and prediction of later user actions; and (2) a comparison of the performance of classification-based dialog models trained on transcribed vs. recognized speech.

2. Statistical Shared Plan-Based Dialog Management
For several years, we have been conducting experiments on statistical shared plan-based approaches to dialog modeling [1, 2, 3]. Other work on statistical dialog modeling tracks the information state shared between the dialog participants, typically as a set of slots whose values may have been requested, filled, confirmed, etc. (e.g. [4, 5, 6]). By contrast, we track the incremental creation of a shared plan by the dialog participants [7]. The shared plan is represented as a single tree that captures the task structure (dominance and precedence relations among tasks), dialog act structure (sequences of dialog acts), and predicate-argument structure of utterances (captured through supertags [8] in each clause), as shown in Figure 1. A task is a sequence of subtasks and dialog acts. Each dialog act corresponds to a single clause by one speaker. A turn may contain multiple clauses and participate in several subtasks. As the dialog proceeds, each utterance is accommodated into the tree in an incremental manner. With this model, we can tightly couple language understanding, dialog management and response generation using a shared representation.

The flow of processing within our dialog manager is illustrated in Figure 2 and involves five tasks. The dialog manager performs each task using a statistical model trained using three kinds of information for the i\textsuperscript{th} clause produced by speaker a: the lexical and syntactic features (e.g., words, part of speech tags, and supertags) associated with the clause (c\textsubscript{a}i); the dialog act label for the clause (DA\textsubscript{a}i); and the most immediate subtask label for the clause (ST\textsubscript{a}i) (cf. [1]). Our models use up to four previous utterances as local context.

Figure 1: Shared plan structure of a dialog

Figure 2: Dialog manager processing of three turns in a dialog.
\[
(DAC_i^u) : DAC_i^u = \arg \max_{d \in D} P(d|c_i^u, ST_{i-1}^u, DA_{i-1}, c_{i-1}, \ldots, ST_{i-k}^u, DA_{i-k}, c_{i-k})
\]

\[
(STC_i^u) : STC_i^u = \arg \max_{s \in S} P(s|DA_i^u, c_i^u, ST_{i-1}^u, DA_{i-1}, c_{i-1}, \ldots, ST_{i-k}^u, DA_{i-k}, c_{i-k})
\]

\[
(STP_{i+1}) : STP_{i+1} = \arg \max_{s \in S} P(s|ST_i^u, DA_i^u, c_i^u, ST_{i-1}^u, DA_{i-1}, c_{i-1}, \ldots, ST_{i-k+1}^u, DA_{i-k+1}, c_{i-k+1})
\]

\[
(DAP_{i+1}) : DAP_{i+1} = \arg \max_{d \in D} P(d|ST_{i+1}^u, ST_i^u, DA_i, c_i, \ldots, ST_{i-k+1}^u, DA_{i-k+1}, c_{i-k+1})
\]

\[
(DAP_{i+2}) : DAP_{i+2} = \arg \max_{d \in D} P(d|ST_{i+1}^u, DA_{i+1}, c_{i+1}, \ldots, ST_{i-k+2}^u, DA_{i-k+2}, c_{i-k+2})
\]

Table 1: Equations used for modeling dialog act and subtask labeling of user and system utterances.

In our dialog manager, interpretation of user input consists of: classifying the dialog act (DAC_u, Equation 1); and classifying the task/subtask of the user’s utterance (STC_u, Equation 2). This information and the user’s utterance are then added to the dialog history. Determination of system output consists of: predicting the task/subtask (STP_u, Equation 3); and predicting the dialog act of the system’s next utterance (DAP_u, Equation 4). After the system’s utterance has been produced, it is added to the dialog history. The dialog history is then used to predict the dialog act of the user’s next utterance, for improved speech recognition and language understanding (DAP_u, Equation 5).

The only other work on statistical dialog management that explicitly tracks the task/subtask structure of the dialog is that of Hardy and colleagues [9]. They used a large corpus of transcribed and annotated telephone conversations to develop the Amitis dialog system. For their dialog manager, they trained separate task and dialog act classifiers on that 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 an integral feature in the dialog model and dialog management strategy. Poesio and Mikheev, in one of the first papers on statistical dialog act classification [10], report that using dialog game information (analogous to task/subtask information) improves accuracy for dialog act classification. However, they did not learn a model for dialog game, but assumed this information was given. Best prior reported dialog act classification accuracy for the corpus we use in this paper, the Maptask corpus, is 73.91% [11].

3. Experiments

In this section, we describe the performance of shared plan-based dialog models for our five dialog management tasks. We compare performance of models trained on transcribed and automatically recognized speech, and of models that incorporate dynamic features to those that do not.

3.1. Data Preprocessing

For the experiments reported here, we used the Maptask corpus of human-human task-oriented spoken dialog [12]. We used this corpus because it is one of the largest task-oriented human-human dialog corpora for which speech data is publicly available. Each dialog in this corpus involves a direction giver and a direction follower who collaborate to resolve differences between and draw a route on maps they are given. In most dialog systems the system has the task initiative, so for our experiments we treat the giver as the system and the follower as the ‘user’. We chose this corpus because it is freely available and is already annotated for dialog acts and task/subtask information (‘games’). (However, this corpus does not involve a task typical of human-computer dialog, and the task structure of the dialogs is quite flat.) We performed all experiments using 10-fold cross validation because this corpus is quite small (128 dialogs total).

This corpus contains annotated transcripts and audio data. To get automatically recognized speech we used AT&T’s Watson speech recognizer [13]. For each fold, we adapted an existing acoustic model (trained on an improved version of the American English telephone speech model described in [14]), and trained a trigram language model, using the nine training folds. We then decoded the speech in the testing fold. The average word accuracy across all 27084 utterances in the corpus was 55.9%. For ‘follower’ utterances, it was 49.8%, while for ‘giver’ utterances, it was 58.7%. (Better recognition accuracy for Maptask was reported in [15]; however, our goal for this work was not to maximize speech recognition accuracy, but simply to explore the impact of using recognized speech.)

We ran each recognized and transcribed utterance through a supertagger [8]. We then extracted n-gram features from the following information for each utterance: speaker, recognized text, transcribed text, part of speech tag sequences for recognized and transcribed text, supertag sequences for recognized and transcribed text, dialog act, and most immediate task/subtask. The set of dialog acts is: acknowledge, align, check, clarify, explain, instruct, query-w, query-yn, ready, reply-n, reply-w, reply-y, uncodable. The set of task/subtasks (dialog games) is: align, check, explain, instruct, query-w, query-yn, uncodable.

3.2. Method

We trained maximum entropy models for each of the five dialog management tasks described above [16] (for more information about our features and method, see [3]). For each task we report results from two model training variations: Recognized Static (R) – model is trained on recognized speech; Transcribed – model is trained on transcribed speech. We also report results from three decoding variations: Static decoding (S) – the models do not include dialog act and task/subtask features; Dynamic decoding (D) – the values for dialog act and task/subtask features are determined at run-time from the classifier output; and Oracle decoding (O) – the true values for dialog act and task/subtask features are used (best possible). We report 1-best results for all experiments. (For dynamic decoding for this corpus, 3-best results were almost the same as 1-best results.) The baseline performance for each task (to always choose the most common label regardless of input) is as follows: DAC_u, acknowledge, 33.8; STC_u, check, 27.32; STP_u, instruct, 26.63; DAP_u, instruct, 27.65; DAP_u, acknowledge, 33.8.
3.3. Results and Discussion

The classification accuracy for each model variant for each of our five tasks for 0, 1, and 3 utterances of history are shown in Figure 3. The static models for each task (R-S and T-S) outperform the baseline for that task. (Accuracy for T-S with 1 utterance of history is: $DAC_u$, 74.79; $STC_u$, 59.96; $STP_u$, 45.2; $DAP_u$, 48.94; $DAP_{*}$, 40.51. Accuracy for R-S with 1 utterance of history is: $DAC_u$, 59.38; $STC_u$, 47.1; $STP_u$, 39.22; $DAP_u$, 36.62; $DAP_{*}$, 44.19). Some of the dynamic models, however, perform close to or below baseline ($DAP_{*}$ and $STP_{*}$). In general, having more than one utterance of history does not improve performance, perhaps because the task structure of the MapTask dialogs is fairly flat. Also, use of more domain-specific task knowledge (e.g. discussion status of each step in the route being discussed) would help.

For all cases except $DAC_u$, the use of recognition output does not hurt performance for oracle decoding (R-O). Even for dynamic decoding, accuracy with recognition output declines less than 10% for most tasks (compare R-D and T-D). With dynamic decoding, use of recognition output lowers performance more for classification tasks (dialog act classification and task/subtask classification) than for prediction tasks. Since the prediction models have fairly low performance, we can draw no general inferences from this observation.

Static models (T-S and R-S) give better performance than those that also use dynamic features (T-D and R-D) (for context ≥ 1). We think this is due to the high error rate for subtask prediction and the fairly high error rates for dialog act classification/prediction leading to cascading errors.

We also ran experiments where we trained on transcribed speech and tested on recognized speech. For oracle decoding, the results of these experiments were almost identical to those for T-O (accuracy for 1 utterance of history: $DAC_u$, 78.52; $STC_u$, 92.07; $STP_u$, 67.69; $DAP_u$, 66.12; $DAP_{*}$, 54.69). For 1-best static decoding, the results were not much better than those for R-S (accuracy for 1 utterance of history: $DAC_u$, 59.81; $STC_u$, 48.11; $STP_u$, 39.98; $DAP_u$, 37.27; $DAP_{*}$, 43.75).

Figure 4 shows a dialog extract from this corpus. For each utterance, the actual dialog act (DA) and task/subtask (ST), and those determined by the R-S model (DA* and ST*), are shown. Table 2 shows the most common labeling errors for each task. As illustrated in Figure 4, the actions align and acknowledge are similar. Also, acknowledge and align may be predicted instead of instruct because often a speaker will start a turn with a grounding utterance and then take some other action.

We conclude that it is possible to use recognized speech rather than transcribed speech for training our dialog models. We also identify the following as having potential to improve our models: use of a less flat model for task/subtask prediction and classification; use of more specific (possibly domain-specific) labels for dialog acts and tasks/subtasks; and separate modeling of grounding behavior (e.g. acknowledgments) and task-related utterances. Also, in order to achieve better performance overall we need to model the global dialog context as well as the local dialog context. We might borrow some features from the information state literature, such as slot values or named entities present in the utterance.
6. References


