Complementary Computing: Policies for Transferring Callers from Dialog Systems to Human Receptionists

Eric Horvitz and Tim Paek

Microsoft Research
Redmond, WA USA 98052
{horvitz, timpaek}@microsoft.com

Abstract

We describe a study of the use of decision-theoretic policies for optimally joining human and automated problem-solving efforts. We focus specifically on the challenge of determining when it is best to transfer callers from an automated dialog system to human receptionists. We demonstrate the sensitivities of transfer actions to both the inferred competency of the spoken-dialog models and the current sensed load on human receptionists. The policies draw upon probabilistic models constructed via machine learning from cases that were logged by a call routing service deployed at our organization. We describe the learning of models that predict outcomes and interaction times and show how these models can be used to generate expected-utility policies that identify when it is best to transfer callers to human operators. We explore the behavior of the policies with simulations constructed from real-world call data.

Keywords: Spoken dialog systems, machine learning, human-machine systems, probabilistic user modeling, complementary computing

1. Introduction

Machine learning and reasoning methods promise to introduce new efficiencies into the world. However, a number of attempts to field fully automated reasoning methods, such as spoken dialog systems, have been associated with disappointment.[1] In many cases, frustrating situations come as intermittent failures in otherwise valuable and competent services. A promising path to fielding computational intelligence, even when such methods are not fully competent, is to exploit statistical methods to identify valuable couplings of human and machine automation—and to bring human effort to bear when such efforts will be most useful in bridging gaps and deficiencies in automated reasoning. We refer to methods that mesh together the intelligences of reasoning systems and people via explicit policies about when and how to engage people as *complementary computing*. Although automated systems are often designed to default to human service providers when failures occur, complementary computing is more than a back-off strategy. It considers the most efficacious way to leverage both human and machine resources, allowing for the possibility that some callers may not even encounter the automated system at all.

The importance of developing computing solutions that are designed to complement human intelligence and resources has been highlighted in the past within the realm of personal computing [6,7]. Methods that attempt to interleave the efforts of people and machines at multiple places during a dialog or problem-solving challenge, including situations that involve fast-paced shifts of contribution, are often referred to as performing *mixed-initiative interaction*. However, while complementary computing includes activities referred to in typical uses of the phrase *mixed initiative*, it more generally encompasses methods that attempt to optimize how automated reasoning and human resources should be best coupled in the provision of services. The approach includes the identification of ideal patterns of initiative, flows of analysis, and configurations within problem-solving systems composed of both human and computational components. We shall focus in this paper on a class of complementary computing problems that center on the development of effective principles, and machinery with the ability to understand how and when the skills of people should be called upon to take over or to bolster imperfect automated reasoners to solve a problem at hand.

We report, as a representative analysis and case study, the use of machine learning to enhance the coupling of human receptionists and an automated spoken dialog system for handling calls within the Microsoft Corporation. We focus on the use of machine learning and expected-value decision making to

[1] See [4] for a reflection from the business community about the failure to date of automated speech recognition systems to penetrate widely.

decide when a user's interaction with an automated dialog system should be transferred to a human receptionist. We take the ideal-transfer challenge as an example that alludes to a larger space of opportunities with balancing computational competencies and human resources in the provision of services. In the case of the call transfer challenge, receptionists are drawn from a team of personnel that faces a changing call load, leading to callers waiting in queues with durations that vary from moment to moment. As we shall see, the ideal time to transfer a user to a human operator for assistance depends on the current availability of operators as well as on the probability distributions over the ultimate outcomes and durations of the conversation that is currently in progress.

Rather than building a dialog system from scratch, we explored how machine learning can be used to overlay a decision-theoretic policy for call-transfer on top of a legacy automated dialog system. We learned probabilistic models from a log of real-world cases that predict the ultimate outcomes and the durations of interactions with the legacy dialog system. We show how we can employ the models to generate real-time policies for transferring people interacting with an automated call routing system to a human operator, based on an analysis of a call session as the dialog progresses, and the current load on the staff of human operators. The study highlights the promise of using probabilistic techniques to better mesh human resources with automated methods that have competencies and policies that can be characterized via machine learning.

In many real-world applications, the competencies of systems may change with ongoing training or with shifts in the distribution of challenges being seen. Thus, it can be important for such methods to employ ongoing learning about the competency of automated components to optimize the best way to employ computing and human resources. As we shall see, the approach allows for the ongoing re-optimization of the complementary computing policies as new operators are added to the staff and as the competencies of an automated dialog system change over time. By continuing to collect data about how a dialog system performs, adjustments can be made in the transfer policies. For example, the speech recognition component of the dialog system might be enhanced, such as through the integration of an updated language model associated with greater recognition accuracies for some or all situations. The speech recognizer's accuracy may also degrade over time. This was the case for the system deployed at our organization. The recognition accuracy for names of people at the company was dropping for a period of time with the churn of people at the organization. When the system was initially fielded, experts were employed to tune the acoustic model of the recognition system, and to provide hints to the system about the common pronunciations of first and last names for a large number of employees at the company. However, such manual tuning effort was not regularly performed. System operators noticed that

recognition rates would drop during lulls in the maintenance of the acoustic models, as employees left the company and new employees were hired.

The methodology we describe adapts in an elegant manner to changes in the competencies of the core speech recognition component of the automated dialog system. Thus, the system can take into consideration the changing competency of the speech recognition component, shifting gracefully depending on the most recent analysis of the recognizer's accuracy in different situations.

2. A Status-Quo Call Handling System

Organizations have been turning away from touch-tone routing systems for call routing, and have been turning to automated dialog systems that employ speech recognition and natural language processing to assist users. There is evidence that such a policy is warranted, based on studies of callers' reactions to touch-tone routing [14]. Dialog systems utilize automatic speech recognition (ASR) to facilitate requests in natural language, which customers appear to favor over touch-tone menus [13].

For several years at the Microsoft Corporation, an automated dialog and call routing system named *VoiceDialer* has fielded all internal directory assistance calls. Using speech recognition, VoiceDialer attempts to uniquely identify one of over twenty thousand name entries in the company's global address book. In building telephony applications for task-oriented domains, system designers can choose from a wide array of approaches to dialog representation, such as finite state controllers, slot-filling templates, and rule-based models [1]. The dialog flow for the VoiceDialer legacy system was fully specified by a finite state controller. Although we centered our studies on this specific dialog system, we point out that the overall approach of applying expected-value decision making to identify complementary-computing solutions is agnostic about the underlying dialog representation. System designers can choose a dialog representation that best characterizes their domain and still employ the methods we describe to consider the costs and benefits of transferring control to human operators.

To assess the overall performance of the system, we obtained over 250 megabytes of data logs covering a period of roughly one year. The log contained approximately 60,000 transcriptions of individual sessions with VoiceDialer, capturing key system and caller actions for each call. We have performed machine learning on the logs of sessions to build models that can predict outcomes and durations of interactions. The distinct outcomes and respective prevalencies are as follows:

- **Success.** The system eventually recognizes the name spoken by the caller as a name in the directory and transfers the call to that person (45%).
- **Operator transfer-name unavailable**: The system infers that the person requested is not in the directory, and routes the call to an operator (6%).
- **Operator transfer-maximum mistakes**: The system reaches the maximum number of allowed mistakes, and routes the call to an operator (12%).
- **Operator request:** The user requests assistance by pressing '0, #, or *' (25%).
- **Hang up**: The user simply hangs up during the session (13%).

For the legacy policy in force in the VoiceDialer system, sessions with the automated dialog system are allowed to progress for at most four steps or until a maximum tolerated number of mistakes is reached. If we examine the cases where callers engage the system to completion, removing from consideration the 38% of cases where a user either hangs up or requests an operator, we find that the VoiceDialer system has a success rate of only 66%.

We shall focus on the use of probabilistic machine learning and decision analysis to identify ideal actions with regard to the best time to transfer a call to a human operator. As we shall see, the policy takes into consideration the real-time stream of evidence, gathered by the automated dialog system over the course of a call session, and the current load on a team of human operators. The analysis highlights the promise of integrating, in a graceful manner, human and computational intelligence in the form of dynamic decision policies that take into consideration the changing evidence about the progression of interactions with an automated dialog system, and potentially fast-paced changes in load on human operators. Over longer periods of time in the course of the evolution of technology, the approach provides a means for ideally harnessing automated dialog systems as their competency grows with improvements in the underlying recognition technologies—or diminishes with the increasing size or scope of the problems faced.

3. Policies for Transfer from Machine to Human Operators

To highlight key concepts, we shall focus on a time-centric utility model for guiding the construction of policies for transferring a caller from the automated dialog system to a human operator. With this preference model, we consider time as a measure of cost, and examine methods for minimizing the time required for the appropriate routing of a call. That is, we assume that the utility of a call-handling action is captured by the total time required for a caller to be routed to a target telephone number, and we

consider situations where callers attempt to work with the automated dialog system or request routing to an operator, rather than hanging up. We will explore a more detailed utility model in Section 6, and introduce there a consideration of such real-world issues as frustration with dialog errors, with the time spent in a quiet queue waiting for human attention, and the cost to a business of losing customers via people hanging up in frustration [12].

In our approach to complementary computing for a system composed of an automated dialog system and human operators, we employ machine learning to build models that predict the ultimate outcomes of a session. We consider inferences about the outcomes and durations of interactions over sessions as key building blocks of decision-theoretic call-transfer policies. We seek to develop predictive models that can be applied anytime in a call session, and report, based on observations seen so far, the probability distribution over the outcomes and overall durations of the interaction. Such models could provide predictions at each step of a dialog, via an analysis of observations gathered up to the current time during the interaction.

We use $p(H|E,\xi)$ as the probability distribution over the ultimate outcomes H of an interaction with an automated dialog system for call handling, given observational evidence E and background information, ξ. We wish to learn models that can be used to infer the likelihoods of different outcomes and the expected duration of the interactions, conditioned on the outcomes at hand. The expected duration of a caller's interaction with the automated portion of the call handling, t^a, based on an observed stream of evidence is,

$$t^a = \sum_i p(H_i \mid E,\xi) \int_t p(t \mid H_i, E, \xi) t dt \qquad (1)$$

That is, to compute the expected time of a caller's session with the automated dialog system at any point in a dialog with the automated system, we consider the expected duration of the session conditioned on each ultimate outcome H_i of the session and weight these times by the likelihood of each outcome. We integrate over time to compute the mean time expected for the interaction with the automated dialog system until an outcome is reached, conditioned on the occurrence of each outcome, H_i. For simplicity, we shall rewrite this mean time as $<t| H_i.>$, which we refer to as the *mean conditional time* for each outcome state. Rewriting Equation 1 with mean conditional times $<t| H_i>$, we have

$$t^a = \sum_i p(H_i \mid E,\xi) < t \mid H_i > \qquad (2)$$

We now focus on learning predictive models that provide the probabilities of different outcomes, as well as the expected durations of the remaining times of interaction, conditioned on observations gleaned from the history of the dialog session at hand and the wait time for an operator. We will apply these probabilistic models to generate policies that drive real-time decisions about transferring the user from the automated spoken dialog system to a human operator.

Let us consider the experience with continuing to engage the status-quo automated dialog system. Callers who choose to work with the automated dialog system, versus hanging up in frustration, will eventually either be provided with the information they need from the automated system, take the initiative to transfer themselves to an operator manually (if they understand how to engage the system with touchtone commands), or be transferred automatically to an operator.

We will consider the total expected time required to receive routing assistance as including both the time required for the automated component and the wait for an operator, should a transfer to an operator occur. We can decompose outcomes into situations that lead eventually to a transfer to an operator, H^o, and those that are eventually successful via automation without any operator attention, H^a. Beyond the time t^a working with the automated dialog system, we also consider the expected duration of time t^o that a caller will spend in a queue waiting for a human operator and then being assisted by an operator, should the caller be transferred to a human operator for assistance. The mean wait time for an operator can be sensed directly at any moment by monitoring queues in a call center. We will assume that an operator can relay immediate, accurate assistance to the user.

Putting everything together, the total expected time associated with engaging the legacy system call routing system, $t^{a,o}$, is

$$t^{a,o} = \sum_i p(H_i^o \mid E, \xi)(< t \mid H_i^o > + t^o) \; + \sum_i p(H_i^a \mid E, \xi) < t \mid H_i^a > \qquad (3)$$

That is, the total time for the interaction with the status-quo system is the expected time required by the automated and potential human-assisted aspects of the call-handling session, in cases where people are routed to human operators with the legacy system.

Let us now dive deeper into the machine learning and reasoning to infer ideal call-handling policies for minimizing the expected time for the interaction with the combined human—computer call-handling system. Rather than rely on the legacy fixed transfer policy, predictive models can provide a continuing stream of forecasts about the total time that is expected to be required with the use of the default policy encoded in a status quo call-handling system. We use such estimates in an ongoing comparison of the

expected time until reaching a goal, provided by Equation 3, with the expected time after making an immediate, courteous automated transfer into the queue for an operator.

At the crux of computing the expected time for the overall process of call routing is the construction of models that can provide the probabilities of successful call handling by the automated system and of transfers to human operators, and the conditional expected durations for the different outcomes. Given the availability of inferences about these probabilities, and observations about the current load of operators, we can make decisions about if and when to execute an automated transfer to a human operator.

In operation, we continue to check the load on operators and test to see if the expected time for continuing the automated dialog with the user is greater than the time spent waiting in the queue and then being serviced by a human operator, testing if $t^{a,o} - t > t^{o.}$ where t is the amount of time already invested in interacting with the system. If the expected wait time at any point in the automated dialog becomes greater than continuing to engage the user with the automated system, we immediately transfer the user to the queue for a human operator. As we are doing point-wise checking, the approach can be viewed as a greedy approximation to a solution invoking a more complex look-ahead strategy.

We note that the transfer policy employed at an organization can change the numbers of people being transferred into a queue, thus influencing the wait times. We have assumed in this section that any single transfer does not influence the wait time significantly in a large-scale system; we simply continue to measure the overall result of a transfer policy across the organization directly via direct inspection of the wait time. We could extend this model by including a term that increments the wait time with each transfer. In Section 7, we will discuss the opportunity for modeling the influence of a transfer policy being executed across a large organization on the overall wait times experienced by people being transferred to operators at different times of day. Such analyses, employing queuing theory for modeling loads on the overall system, promise to be useful for guiding offline decisions about the ideal number of human operators to employ, given the competency of an automated dialog system.

4. Constructing Probabilistic Models to Predict Dialog Outcomes

With the goal of developing dynamic ideal transfer policies, we seek to construct probabilistic models from a database of session logs. We wish to harness sets of observations drawn from traces encoding the timing of actions and recognitions associated with a user's interaction with VoiceDialer's spoken dialog system. At the core of this challenge is gaining an understanding of the discriminatory power of

observations for making inferences about different outcomes, and of session durations for different outcomes.

Let us first consider the observations available for building case libraries for machine learning. Rather than applying special feature-selection methodologies for choosing specific features a priori, we compiled a set of features that we could engineer from the available log data with relative ease, and then sought to identify the most discriminatory features through Bayesian structure learning, as we describe below. The log data includes *n*-best list hypotheses, a list of top candidates identified by the speech recognizer after every caller utterance. A number of the features that we provided to the learning algorithm were derived from these recognition distinctions. The features we used to learn predictive models can be generally classified into four broad categories (see Appendix I for a complete list of all of the features we used to build predictive models):

- **System and user actions**: *System and user* observations represent all prior actions taken by the system or the user. Such evidence includes the observation that the dialog system asked the user to confirm between its top two guesses of names based on the user's utterance, or to spell the last name of the intended person, and the observation that the user has pressed a touchtone key rather than providing an utterance.

- **Session summary evidence**: *Session summary* observations summarize the overall statistics of events within the session once it has finished. Such observations include the number of attempts by the user to specify the name of the person being sought, the number of *n*-best lists that were generated by the logging system, and the overall duration of session.

- ***N*-best list evidence**: *N-best list* observations refer to features output by the speech recognition system including the range of confidence scores assigned by the speech recognition subsystem, the mode of the scores, the maximum consecutive score difference, and the count of the most frequent first/last/full names that appears among the hypotheses. We sought to derive as many speech-related features as possible from the *n*-best list and to conduct feature selection later using our learning procedure for model selection.

- **Generalized temporal evidence**: *Generalized temporal* features capture trends across the multiple *n*-best lists generated during a dialog, such as whether the top name hypothesis is the same between two *n*-best lists or the maximum number of times any name occurs in multiple lists.

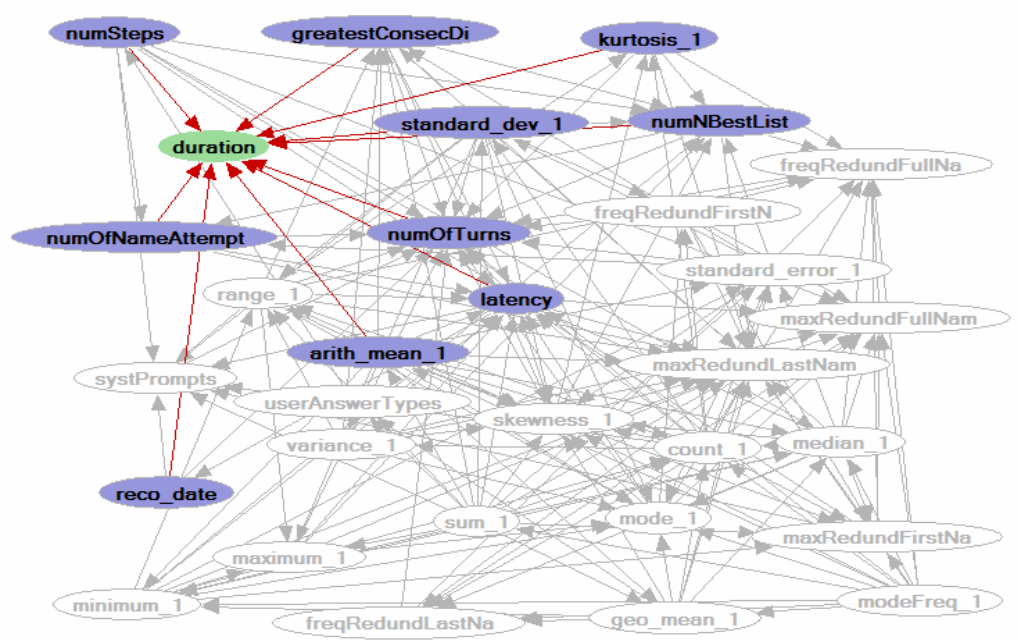

Figure 1. Graphical model learned from logged training data for predicting duration of interaction with a dialog system, conditioned on the ultimate outcome being a successful transfer. Variables directly influencing the *duration* variable are highlighted with shaded fill.

We pursued the construction of models that could be used to infer the likelihood that a session with the VoiceDialer system would ultimately handle the whole session successfully, ending in a successful transfer to the right person, or fail to autonomously address the caller's goal. Failures for automated handling includes sessions where 1) the caller is ultimately transferred to an operator by the legacy policy, given a failure to match the name it recognizes; 2) the caller is transferred to the operator after the maximum number of mistakes tolerated by the system has been reached; 3) the user hangs up prematurely, and 4) the user requests an operator via a touchtone command.

We constructed two models for predicting outcome. The first considers all outcomes, including cases where callers disengage via a hang up. A second model, which focused on predictions for engaged callers who stick with the system, was developed for use in studies of the behavior of call-transfer policies based on time-minimization for callers. The engaged model removes consideration of the situation and cases where callers hang up.

We now turn to the learning procedure, the models constructed, and the evaluation methods used to test predictive accuracy. Given the enumerated features, we employed Bayesian structure learning to build Bayesian networks for predicting session outcomes. We used methods developed by Chickering, *et al.*

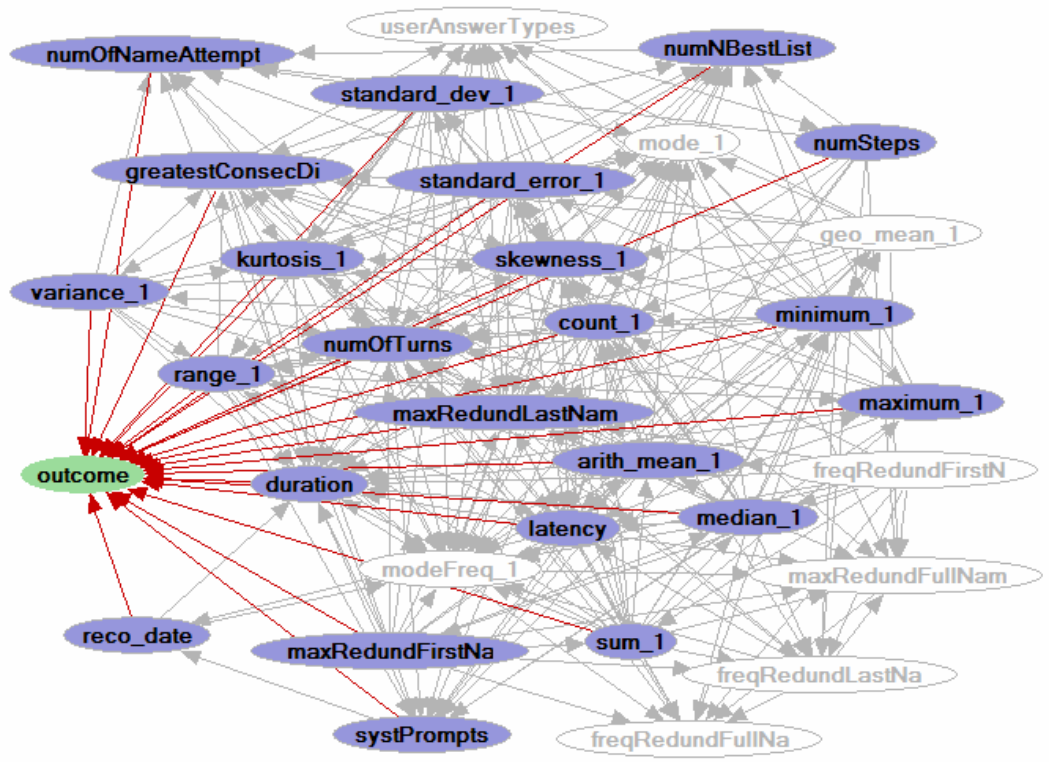

Figure 2. Graphical model learned from logged training data for predicting the ultimate out of an interaction with a dialog system. Variables directly influencing the *outcome* variable are highlighted with shaded fill.

[3] and by Friedman and Goldszmidt [5] to construct the models. These methods employ heuristic search over feasible probabilistic dependency models guided by a Bayesian score to rank the candidate models, each represented as a candidate Bayesian network of the set of random variables under consideration, and proposed structure of directed arcs between the nodes. In constructing a Bayesian network, the system builds and explores a search tree where each node represents a Bayesian network with a different dependency structure. Each node generated during the search procedure is assigned a Bayesian score, which is an estimation of the likelihood of the data given the proposed dependency structure.

The Bayesian structure search method we use employs both global and local search. For each variable, the method creates a tree containing a multinomial distribution at each leaf, exploiting the local structure-search methods described in Friedman and Goldszmidt [5]. The method provides a graphical view of the constructed model, enabling us to inspect multiple variables and influences.

We constructed two classes of models from the training data. One class of models predicts the ultimate outcome of a session. The other class predicts the expected duration of an interaction for each ultimate outcome. We constructed a set of distinct Bayesian network models for each class for each step of a dialog with the legacy system, where *steps* refer to places in the dialog where processing of callers' utterances took place. These are points in the dialog associated with the generation of an *n*-best list by

the speech recognizer. Based on the configuration of the legacy system, dialog sessions did not last more than four steps.

We employed a ten-fold cross-validation methodology for constructing and evaluating the models from a library of cases, where each case includes the outcome, duration, and step-by-step observations made during engagements by a caller with the legacy dialog system. The ten-fold cross validation is executed as follows: For each dialog step, we select a set of ten different folds of training and test data from the total number of cases for that step. The folds are selected by randomly segmenting case libraries into ten equal-sized sets of cases. Ten predictive models are constructed for each prediction of interest. The training data for each model is composed of nine of the ten folds. The performance of each learned model is then tested with cases contained in the tenth fold, which had been held out from the training. For predictions of the dialog outcomes, classification accuracies are computed for each of the folds and a mean and standard deviation of the accuracies for the ten folds are reported. We also performed ten-fold cross validation for building and testing models of duration at each step, conditioned on different ultimate outcomes. We made sure to manage the folds downstream such that training cases used to build the outcome classification models were not used to test outcomes of the duration models. For predictions of durations, we report results as means and standard deviations in errors in time predicted for the durations of calls versus the actual durations across the ten folds, for each outcome and step.

Table 1.

Classification accuracies and lifts of predictive models for ultimate outcomes by dialog step.

	Step 1	**Step 2**	**Step 3**	**Step 4**
All				
Call outcome	0.86 (0.01)	0.77 (0.01)	0.69 (0.02)	0.81 (0.02)
Lift	0.17 (0.01)	0.15 (0.02)	0.19 (0.01)	0.23 (0.04)
Engaged				
Call outcome	0.90 (0.01)	0.80 (0.01)	0.75 (0.02)	0.82 (0.02)
Lift	0.17 (0.01)	0.13 (0.01)	0.21 (0.03)	0.23 (0.04)

Table 2.

Mean duration errors and lifts of predictive models by dialog step.

	Step 1	**Step 2**	**Step 3**	**Step 4**
Duration | Automation success				
Mean error	4.30 (0.21)	5.17 (0.21)	6.13 (0.30)	6.39 (0.88)
Lift	9.79 (0.32)	10.18 (0.25)	9.76 (0.44)	9.95 (1.25)
Duration | Name unrecognized				
Mean error	9.78 (0.59)	9.24 (0.41)	8.79 (0.65)	6.40 (0.72)
Lift	10.63 (0.43)	10.98 (0.61)	11.40 (0.75)	13.01 (0.66)
Duration | Max errors				
Mean error	10.39 (0.80)	12.24 (0.97)	14.54 (2.69)	14.58 (5.51)
Lift	11.31 (0.61)	12.64 (1.01)	13.28 (1.55)	12.91 (2.54)
Duration | Hang up				
Mean error	7.37 (1.55)	8.41 (1.88)	10.00 (5.07)	10.38[1]
Lift	8.26 (0.89)	8.17 (0.87)	11.06 (3.24)	0.00
Duration | Operator requested				
Mean error	5.63 (0.84)	7.043 (1.52)	9.19 (3.03	5.97[1]
Lift	10.55 (0.64)	7.19 (1.18)	7.80 (2.78)	0.00

In summary, we learned predictive models for outcomes and expected durations for each outcome for each of the four maximum tolerated steps of the dialog. We constructed, for each step of the dialog, Bayesian networks that predict the likelihood of each of the ultimate outcomes under consideration. The full outcome model considers five ultimate outcomes and the engaged model considers four outcomes, bypassing consideration of the hang-up cases. We additionally constructed Bayesian networks for predicting the durations conditioned on each of the five ultimate outcome states for each of the four

[1] These outcomes are based on a 70/30 split given a sparcity of cases for the outcomes; see discussion in paper for details.

dialog steps. At each step, the probabilistic models perform inference from observations gathered in previous steps of a session, including features seen in the current and all earlier steps.

Table 1 displays the classification accuracies of predictions by the models that predict the ultimate outcomes of dialog sessions, for both the complete and the engaged models, for each of the four dialog steps. Table 2 displays the errors in predicted durations for each of the ultimate outcomes, based on observations made in each dialog step. The tables report the means and standard deviations of the accuracies over each of the ten test sets held out during the cross validation. In addition, means and standard deviations are reported on the *lift* associated with each of the predictive models over the respective marginal model. The lift captures the difference in the predictive power of the learned models and the marginal models—classifiers that employ the background statistics for predictions, and select the most likely outcome based on these statistics. For two of the conditional duration outcomes, both occurring in the fourth dialog step, a low number of cases for the hang-up and operator-request cases (37 hang-up cases and 12 user operator request cases respectively) made performing a tenfold cross validation inappropriate. For these two outcomes, we performed a single 70/30 split; that is, we trained predictive models for these cases on 70% of the data and tested on the remaining 30%. As we have a single fold and test analysis for each, we do not report a standard deviation on these results. Overall, we saw significant lifts over marginal models in all cases except for the two data-sparse outcomes. We believe that additional data would lead to enhanced predictive power for the two latter outcomes.

Beyond predictions, learning graphical probabilistic models can be used to gain insights into the influences among variables, and the overall sensitivities of predictions to observations. We inspected the Bayesian networks to seek a deeper understanding of the domain. Figures 1 and 2 display two of the learned graphical models. Nodes are random variables and arcs represent learned probabilistic dependencies among the variables. Definitions of the model variables are contained in Appendix I.

Figure 1 shows the Bayesian network learned for predicting the duration of the interaction with VoiceDialer given an ultimate outcome of *successful transfer*, when evaluated at the first step of dialog. We show the model with best performance on test data, drawn from the ten models constructed during the cross-validation procedure for this dialog step. Variables showing significant influence for this model include the greatest difference in confidence scores between any two name hypotheses (greatestConsecDi), the duration of the interaction so far (latency), number of name attempts detected so far (numOfNameAttempt), the time of day that the log was recorded (reco_date), and various statistics generated from the *n*-best list generated by the speech recognition system. The latter include such

statistics as the arithmetic mean of all the confidence scores in the first recognition (arith_mean_1) and the kurtosis of the score distribution (kurtosis_1). Figure 2 displays the graphical structure of the best performing model at the first step of a dialog for predicting the ultimate outcome of the dialog session. This model was selected from the ten models learned for this dialog step as part of cross validation process. As highlighted in the figure, similar features have influence in the prediction of the long-term outcomes of dialog sessions.

As for some general reflections about the learned Bayesian networks, we found that the graphical models showed that inferences about the durations and the ultimate outcomes of sessions are influenced by observations drawn from multiple classes of evidence. We found that generalized temporal features, such as the maximum number of times a first or last name identified during the first dialog turn appears in successive turns, tend to have discriminatory power. Other influential observations include subtle characteristics of the distribution of confidence scores reported by the VoiceDialer's speech recognition system such as the skewness and kurtosis of the distributions of these scores. We also found that the number and nature of relevant features for predicting outcomes and durations differed depending on the dialog step.

5. Exploration with Simulations using Real-World Cases

In use, we substitute inferences about the probability distributions over outcomes and durations generated by the learned Bayesian networks into Equation 3 and transfer engaged callers to human operators when that transfer is associated with a lower expected time than continuing on with the automated dialog with the legacy system. To test the value of overlaying the decision-theoretic call-transfer policy on the legacy system, we constructed a simulation system. The simulator steps through traces of real-world calls drawn from the test case library, and allows us to explore the influence of using the decision-theoretic policies at different steps. For explorations, we apply models to cases held out from the training of models for each fold of the cross validation.

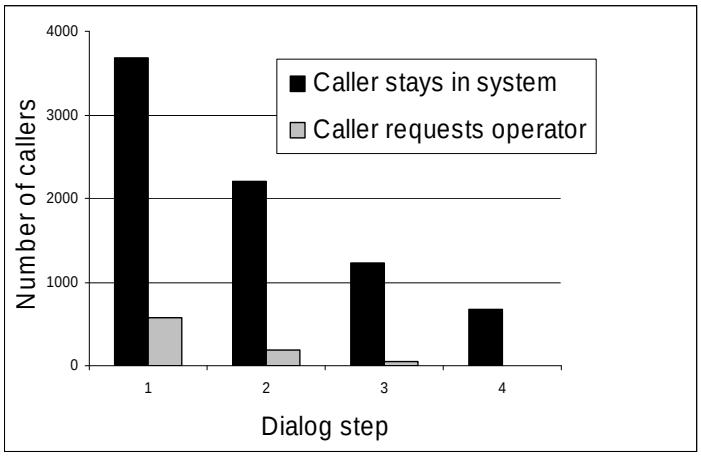

Figure 3. Total numbers of callers staying in legacy system (black bars) versus requesting a transfer to an operator (gray bars) at each step in the dialog, from case library of logged data.

For each call, we have access to a log that contains timing information as well as the series of recognitions and related statistics of the interaction. We also have the ultimate outcome and duration of the interaction. When exploring the influence of the policies that minimize expected total time, the simulator examines the log of test calls and executes at each step, the appropriate model for ultimate outcome of the interaction, and the set of models for duration conditioned on each outcome—considering the observations available at the dialog step under consideration. That is, at each step, the outcome model is used to compute the probability distribution over the ultimate outcomes of the interaction with the automated dialog system. The four models for duration provide inferences about the durations of the interaction with the system, conditioned on each ultimate outcome. Decisions about the automated transfer are made in accordance with the policy described in Section 3 and can be compared with the legacy outcomes.

As background, Figure 3 shows, for the calls in the case library, the portion of callers remaining in the system at each of four steps in the dialog. The figure also shows, at each step, the quantity of callers that manually request operator assistance. Figure 4 through 6 display the results of several analyses with the simulator.

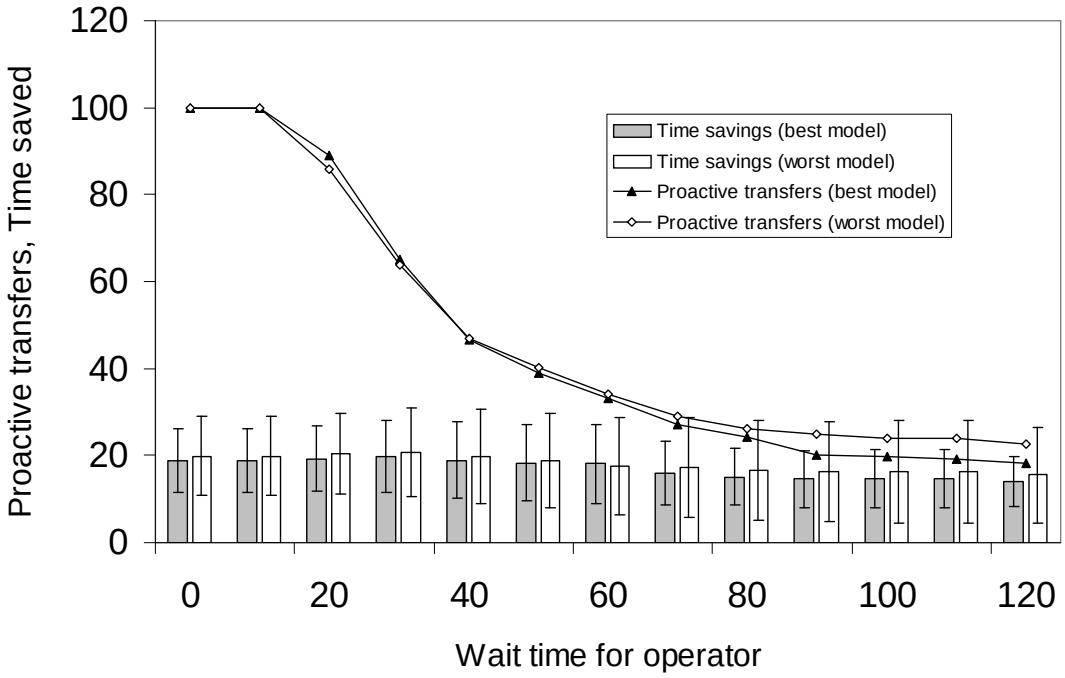

Figure 4. Simulation examining the percentage of callers in the case library who would have been transferred proactively by the decision-theoretic policy as a function of varying the assumed wait time for operator attention. The two curves result from the use of models and corresponding test sets associated with the best and worst folds for predicting ultimate outcome. Means and standard deviations of the savings per call are displayed for different wait times for each fold under

Figure 4 summarizes the results of a study exploring the percentage of sessions that would have been shunted to a human operator by the decision-theoretic policy in advance of callers manually requesting an operator. We consider test cases where callers in reality took a manual action to request a transfer to a human operator somewhere in their call session, and note cases, within and across dialog steps, where the callers would have been routed by the decision-theoretic policy to the operator in advance of their manual action. At each step of call sessions, the simulation computes the expected duration of sessions, by calling predictive models with available evidence to infer probability distributions over outcomes and call durations conditioned on the outcomes. The simulation recommends making a transfer to the operator when this action is associated with a lower overall expected duration. We considered the behavior of the system on the test cases for different assumed waiting times in the queue for operator assistance. Specifically, we note the percentage of callers who would have been transferred proactively

for waiting times starting at zero and successively growing by 10 seconds up to 120 seconds. For each wait time, we compute the mean decrease in the amount of time per session required to be routed successfully.

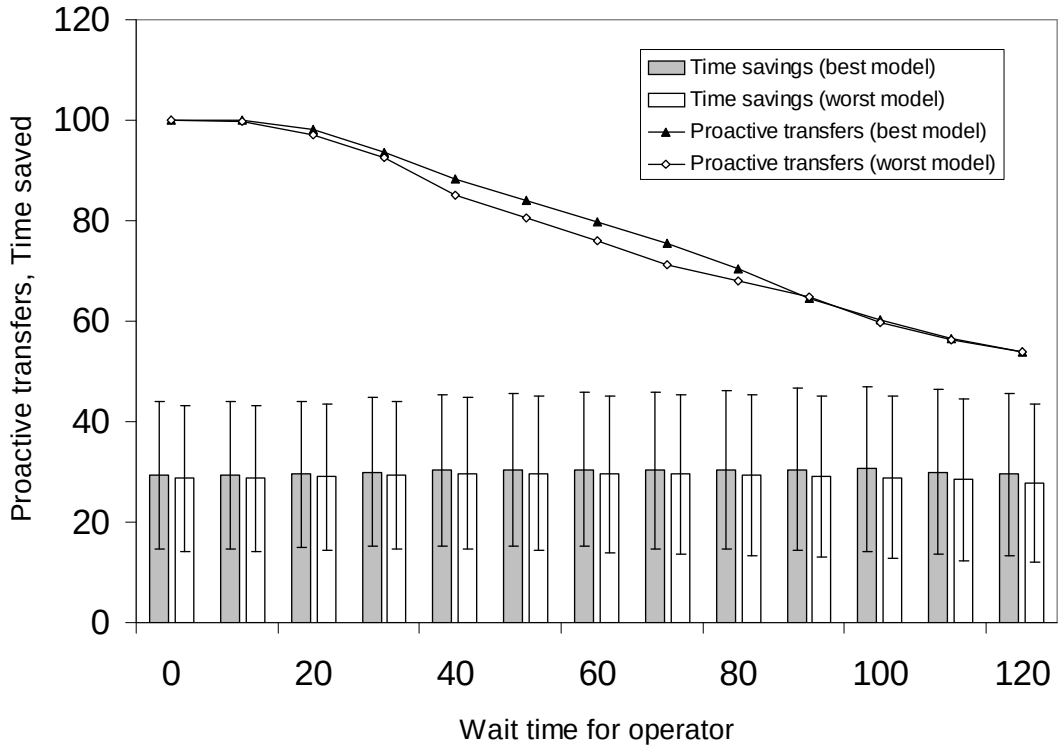

Figure 5. Callers who ultimately received operator assistance in the legacy system who would have been transferred proactively by the expected-time minimization policy as a function of assumed wait time for human assistance. The curves indicate the percentage of callers who would have been transferred to operators in advance of being transferred via a manual request or via automation within the legacy system. The bars show the mean savings in time per session.

To explore the sensitivity of the results to varying the quality of models, we explored the behavior of the decision-theoretic policies within the simulation for the folds associated with the best and worst outcome classification models. Manual transfers were seen in the legacy dialog system for 257 sessions within the test library associated with the best model and for 303 sessions of the test libraries associated with the worst model. Figure 4 includes, for each set of test cases, the mean and standard deviation of the savings per session for the different waiting times. We found that the percentage of proactive transfers to be

similar for the best and worst models and their associated libraries of test cases. For both models, all calls are transferred proactively when waiting time is zero. With increasing wait times for an operator, the percentage of proactive transfers falls in a sigmoid manner. The mean savings for the transferred calls is similar for the worst and best modes, but we note that the variance around the savings becomes larger with increasing wait times for the less accurate model.

Figure 5 displays the results of another simulation, exploring proactive transfers coming in advance of *all* transfers to operators within the legacy system. In this simulation, we move beyond consideration of manual requests for operators to consider all transfers to the human operator seen in the legacy system. Paths to the ultimate receipt of operator assistance include (1) manual transfers to a human operator, as covered above, and automated transfers by the legacy dialog system that occur when (2) the legacy system decides that it has heard correctly and that the requested name is not contained in its lexicon, and (3) when the system has reached its maximum tolerated errors. An ultimate routing to an operator was seen in the legacy dialog system for 2,115 sessions within the library for the best model and for 2,217 sessions of the case library for the worst performing model. Figure 5 shows the changing percentages of proactive transfers to an operator for all of these operator-assistance outcomes. The mean reductions in session times and associated standard deviations achieved for these proactive transfers are also displayed. We note that the fall off in the percentage of callers proactively transferred with increasing waiting times is still sigmoid but is significantly less steep over the range of waiting times plotted. Mean savings in time per session are greater but have higher variance than in the case considering only proactive transfers occurring before manual requests for an operator.

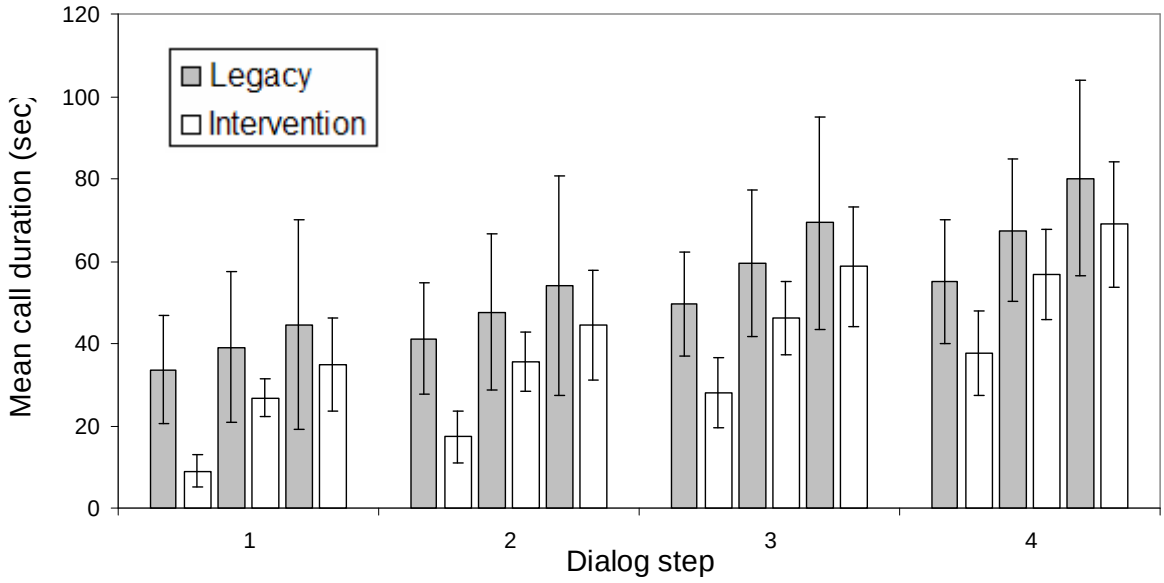

Figure 6. Mean session durations associated with interventions at each step of the dialog system for different assumed waiting times for human assistance. Clusters at each dialog step show mean session durations for the legacy system (dark bars) and decision-theoretic policies (white bars) for 20, 40, and 60 second waiting times (left to right in each cluster).

To further probe the behavior of the decision-theoretic policy relative to the legacy system we explored the value of making interventions with the time-minimization policy at each of the dialog steps for different assumed waiting times. The results of these simulations for wait times of 20, 40, and 60 seconds are displayed in Figure 6. We note that the interventions are associated with a reduction of the call durations and a tightening of the standard deviations around the mean durations.

The simulations demonstrate the potential value of modifying the legacy system with a utility-directed coupling of the system with human operators. The methods can endow automated dialog systems with the ability to shift more calls to people as human resources become available, and, conversely, of relying more on automation as human resources become scarce—where it becomes increasingly valuable to gamble on the prospect that users might have a successful outcome with the automated system. Such resource-sensitive decision policies, guided by the predictive models and a measure of the monitored current wait time for human assistance, allow for systems of people and automated dialog systems to evolve effectively in light of changing levels of human resources and automated dialog competencies.

6. Toward More Expressive Utility Models

For clarity, we have so far investigated concepts in complementary computing with the use of a straightforward time-minimizing model. A richer preference model takes into consideration a more comprehensive measure of utility. We shall now review a more general handling of utility for guiding the transfer of callers from automated systems to human assistance. The model highlights the potential richness of preference considerations, and can guide future data collection and preference assessment.

In a more general consideration of preferences, we move beyond a consideration of the total time required for achieving a goal, to consider the nature of the interaction steps. In one generalization, we would like to consider the differences in the cost of time associated with engaging with an automated system versus that of waiting in a queue. The cost of engaging with a dialog system can be influenced greatly by details of the experience. For example, it may be dominated by the number or density of errors over the course of a dialog rather than just the clock time associated with interaction with the automated system. Such factors can be folded into a cost-benefit analysis of routing actions under uncertainty, considering the number and nature of each step in a dialog. Beyond the number of turns and wait times, the utility of an interaction for a caller may be influenced by other subtle factors. For example, callers may simply have a negative emotional reaction to working with an automated system versus a human operator. Also, a more general preference model considers the nature and preferences of the owner or *principal agent* of the decision making of the overall complementary computing solution. For example, we can consider decision makers at the organizations hosting the automated dialog systems—and employing human receptionists—as the principal agents of the actions, and consider the utilities of the host decision makers, and consider multiple additional economic factors within the overall solution.

In a richer utility model, we move from a general notion of time as a cost function into finer distinctions about a caller's effort and frustration. We distinguish the time that a user engages with an automated dialog system, t^E, and the time that the user waits in a queue for a human operator, t^W. We introduce cost functions, C^E and C^W, that map the times of engagement and waiting to dollar values, where $C^E(t^E)$ describes the dollar value cost with interacting with the automated dialog system and $C^W(t^W)$ maps the time waiting in a queue for a human operator to a dollar value. Such functions are monotonically increasing and potentially non-linear functions. There is opportunity for working with callers to assess such cost functions. Such assessment might be simplified by the assumption of simple parametric models like linear or sigmoid models. For example, for simplicity, one might attempt to map assessments of the functions into constant rates, C^W and C^E, of accruing cost, as dollars per minute. Beyond simple

considerations of the cost of time with engaging in a system, $C^E(t^E)$ can be formulated to capture the frustration experienced by callers with errors of recognition and intention. In such a formulation, the function is designed to map a dollar-value cost to the number and nature of errors, and of such statistics as the density of errors over a set of steps. Creating such a function would rely on careful studies of callers' frustration with errors experienced in working with an automated dialog system, including decision-analytic assessments of "willingness to pay" to avoid such experiences.

Moving to other economic considerations, the decision makers hosting an automated dialog system at an organization may have concerns that extend beyond the costs of the interaction to callers. Let us assume that, from the perspective of a decision maker at an organization, the cost of handling a caller with an automated dialog system is the cost of maintaining the automated dialog system, amortized per call, C^S, and that a transfer to a human operator costs C^o for each call. Also, as the decision maker has some economic goal in having calls handled appropriately, we assess and represent the cost of losing callers via early disconnections via a hang up. We shall refer to this as the cost of disconnection, C^D.

Considering these factors, we start with a basic expression of utility represented as the expected cost of the complementary computing solution as follows:

$$\textit{Expected Cost} \quad = \sum_i p(H_i \mid E, \xi) C_i(H_i) \tag{4}$$

That is, the expected cost with using the legacy system is computed as the probability distribution over the ultimate results of sessions, where situations H include the ultimate outcomes and durations, and the costs C associated with each. Let us further expand this utility model for the spoken dialog problem, by enumerating the outcomes and associated durations, and the costs associated with each of the situations. We consider the following outcomes as separate contributions to the overall cost of a session:

- H^A: Caller has success working with the automated dialog system.
- H^{AO}: Caller is transferred from spoken dialog system to human operator and waits until operator is available.
- H^{AOD}: Caller is transferred from spoken dialog system to human operator but disconnects before the operator is available.
- H^D: Caller hangs up while working with the automated dialog system.

Summarizing the cost considerations, we consider the following terms, accessible as assessments from the principal agent of the decision making of the complementary computing system, yielding dollar value costs:

- C^S: Cost of maintaining the spoken-dialog system (all costs are dollar values)
- C^O: Cost of human operator handling a call.
- C^D: Cost of losing a caller to a premature disconnection via hang up.
- $C^E(t^E)$: Cost function that maps time engaging with spoken dialog system to a dollar value cost.
- $C^W(t^W)$: Cost function that maps time waiting in a queue for a human operator to a dollar value cost.
- *W:* The current sensed length of wait in the queue for a human operator.

We generalize mean conditional time to the mean conditional cost of time and use $<C(t)|H>$ to refer to the expected cost associated with an outcome. The mean conditional costs are computed by summing the cost over the time, weighting the costs by the probability of each of the times.

Putting all of these terms together, we have as the expected utility of the legacy system for handling each call as

$$
\begin{aligned}
&Expected\,Cost = \\
&\quad C^S + p(H^A \mid E,\xi) < C^E(t^E) \mid H^A > \\
&\qquad + \sum_i p(H_i^{A,O} \mid t^E, t^W, E, \xi)(< C^E(t^E) \mid H_i^{A,o} > + C^W(W) + C^O) \\
&\qquad\quad + \sum_i p(H_i^{A,O,D} \mid t^E, t^W, E, \xi)(< C^E(t^E) \mid H_i^{A,o} > + < C^W(t^w) \mid H_i^{A,o}, W > + C^D) \\
&\qquad\qquad + p(H^D \mid t^E, E, \xi)(< C^E(t^E) \mid H^D > + C^D)
\end{aligned}
\tag{5}
$$

Assessments of functions for the costs and the probabilistic models we described in Section 4 can be plugged into this richer equation. The remaining missing predictive models required in the richer utility model (required for the two inner terms of Equation 5) are inferences about the probabilities that callers will hang up as they wait in a queue for an operator, and the probability distribution over the time they will spend waiting in a queue before disconnecting. As indicated in Equation 5, these are likely to be functions of the history, and the length of time that they must wait in a queue for the operator.

In decision making, we compare the expected cost associated with the use of the legacy system as computed with Equation 5 and the expected cost of making an immediate transfer to an operator or the

queue for the operator if the queue is non-zero. The cost of the session, following such an immediate transfer during the engagement of the user by the automated system is,

$$\begin{aligned} Expected\ Cost \quad &= C^E(t^E) \\ &+ p(Wait \mid t^E, W)(C^W(W) + C^O) \\ &+ (1 - p(Wait \mid t^E, W))(< C^W(t^w \mid t^E, W > + C^D) \end{aligned} \tag{6}$$

As with the use of the simpler time-minimization policy described in Section 3, we compare the cost of these two policies, and execute a transfer when the expected cost of the immediate transfer is smaller than the expected cost of sticking with the legacy system. This policy is myopic, and thus, may be made more accurately with additional lookahead.

Our discussion of the richer utility model is intended to demonstrate how the basic decision-theoretic policy that we introduced in Section 3 can be expanded to consider additional costs and uncertainties. The particular costs and uncertainties will differ for different applications depending on characteristics of the domain and dialog system, but the principled approach to transferring control from one computational or human resource to another based on the consideration of evidence and expected-value decision making about the best interleaving of resources remains the same.

7. Discussion

We have explored a methodology of collecting evidence from an automated dialog system about competency and progress, learning predictive models, and then using the models, within an expected utility framework, to guide the transfer of control from the automated system to more competent human operators. Key contributions of the methodology include the abstraction of a dialog system into a set of stages and the construction and use of predictive models that leverage observations about progress to infer, at any of the stages, the overall long-term outcomes of the situation, based on evidence that is currently available.

The approach relies on the reduction of complexity via abstraction of a dialog into a representation of the ultimate outcomes and effort required by the user, and the construction of models that predict the outcomes and effort. Complexity is managed by abstracting detailed interactions into stages or key branches of a dialog that capture progress, and the leveraging of sets of features at the stages that provide updates about ultimate outcomes. Such evidence includes indications of successes (*e.g.*, confirmations) versus failures (*e.g.*, repeats, other signs of frustration) to proceed successfully.

We decomposed the dialog of the VoiceDialer system into four stages, representing successive depths in the dialog tree. Each step is associated with a caller's utterance and the associated analysis of the

attempted recognition. We found this to be an efficient and useful decomposition of the dialog and logged data for constructing and reasoning with probabilistic models. For our domain, it was straightforward to build models for each depth of the tree, as the maximum depth was four.

We believe that the methodology is applicable to systems that perform more complex dialogs, including dialog systems that seek to fill multiple slots, such as systems designed to book travel plans [17]. We are optimistic that similar decompositions can be identified and applied with success in performing predictive modeling in more complex dialog systems. For any dialog system, it is possible to generate a tree of outcomes and durations, where the leaves and nodes of the tree represent system actions. For every node and leaf, statistics can be maintained on how often paths are visited, capturing outcomes where the system reached particular nodes from the root. For more complicated domains, it may be useful to build predictive models in a selective manner, focusing on modeling progress at major branches of the tree—and to access predictions for the models when the system reaches these landmark locations during real-time dialog. The discriminatory power of predicting outcomes and durations of a dialog session with models constructed from data gathered at key branches or stages, will likely depend on the details of the system, domain, and the particular formulation of abstractions of the dialog into landmarks or phases that capture notions of progress through the dialog.

In the work we described, we collected data and constructed predictive models for the ultimate outcomes and durations at successive major steps in the dialog, and also increased the size of the feature space to include distinctions observed in prior steps. The construction of models that predict the ultimate outcome at each step may be unnecessary. We believe that useful predictors about outcome may be constructed by limiting the analysis to the last *n* steps of a dialog at each point in a dialog, or within the set of steps within a well-defined subdialog. Such moving windows of analysis may be combined with more global models that look at densities and burstiness of failures to make progress. We look forward to additional research on the feasibility of using such limited moving windows of analysis.

Our goals were to explore the use in dialog systems of models that can predict the ultimate outcomes of sessions as well as the durations expected before the outcomes are reached and, to investigate how such predictive modeling could be used to guide decisions about transferring calls from automated dialog systems to human operators. We employed a particular form of learning, Bayesian structure search, that centers on the construction of probabilistic graphical models. The method allows us to visually inspect inferred probabilistic relationships among variables. We found that the learning methodology provided useful predictive accuracies in our studies. Other learning methodologies, as well as marginal statistics,

could be used in place of the Bayesian structure search in the complementary-computing approach presented here.

We note that we have assumed a challenge from a specific family of complementary computing challenges—the class of problems where we seek to introduce automation to reduce the cost of expert human assistance, in a context where humans are considered to have the ability to solve challenges accurately and efficiently and where preliminary automation may be prone to errors. History is rife with examples of the maturation of automation, where early, preliminary solutions that do not perform as well as human experts, evolve into approaches that provide equivalent or even better performance than that associated with human intelligence. Thus, more general approaches to solving complementary-computing problems involve considering the set of resources, including computational agents and groups of experts, and reasoning about the ideal of flow of analysis to solve a challenge.

We note that complementary-computing methods extend beyond real-time decision making about the flow of control in solving problems. We conditioned the analysis of call-transfer policies on a fixed staff of operators and fixed technology, and take as inputs the current wait time for gaining access to human assistance. Moving beyond such a fixed-staff assumption, the policies can be used offline in a design setting to inform decisions about ideal expenditures for personnel and technology. As an example, it is feasible to couple expected-value analyses of ideal couplings of people and computing systems with queue-theoretic simulations that provide estimates of the potential waits for callers to receive assistance from operators [12]. Simulations with queuing models can elucidate such factors as the influence of a transfer policy and number of operators on the length of queues at peak call times. These simulations can allow decision makers to examine how changing the number of operators on staff or shifting or updating the dialog technology would influence the expected cost of handling calls.

Although simpler, heuristic strategies for joining human and computing resources into effective services might work well in particular cases, we believe that the principled methodology that we have discussed has broad applicability to a spectrum of complementary computing challenges. We found that the principled approach does not impose a great deal of overhead to implement and execute, and that it can provide insights into the relationships and tradeoffs among key control variables. The decision-theoretic models allow for optimizations that would likely be difficult to discover through experimentation with a few approximate designs.

8. Related Research

Other research teams have explored the use of statistical methods to enhance various aspects of spoken dialog systems. The closest related research centers on studies of methods for predicting potential problems with a user's interactions with a spoken-dialog system. Most efforts in this realm have focused on identifying when users are experiencing poor speech recognition behavior [10]. In the TOOT system, decisions to employ alternate dialog strategies, such as whether to tightly direct users versus allowing users to have input or *initiative* in the flow of a dialog, are based on a user's responses. These policies are represented as rules generated from a classification analysis of "good" and "bad" dialogs trained over dialog sessions [11]. Unlike the decision-theoretic approach that we have presented, the investigators employed deterministic policies as a function of the output of classifiers.

Models that move beyond identification and predict where problematic situations are likely to occur in a call-handling context have been previously explored within the AT&T *How May I Help You* (HMIHY) system [9,15,16]. The HMIHY system considered sets of evidence from a speech recognition system, a natural language understanding component, a dialog manager, and sets of hand-labeled features. Classifiers were trained to predict failures before they might occur based on observations available to the system after different steps of a dialog. Our work extends prior efforts in several ways. We also learn and reason explicitly about both outcomes and durations to generate the decision-theoretic call-transfer policies, and we employ statistical modeling and prediction within an expected-value decision making framework that seeks to ideally use the changing availability of human resources to work with people.

Finally, the approach that we take is similar to decision-theoretic planning using fully-observable or partially-observed Markov decision processes (MDP) [8]. In recent work, an application that makes use of MDP models for providing care to patients with dementia explored the inclusion of a fall-back option to a person in its action set [2]. It is feasible to represent the decision-making task of transferring to an operator as an MDP. However, using an MDP would require the overhead of formulating a stochastic transition model, assuming a Markov assumption on the state space, and decomposing the objective function into local rewards, mapped to each state. In contrast to the MDP approach, we have learned rich models at several successive stages of a dialog, where the models predict the ultimate long-term outcome and expected durations of the session.

9. Summary and Conclusion

We focused on the use of the predictive models in an expected-time analysis to identify the best time to transfer a caller automatically to a human operator at different points in callers' interactions with a legacy automated call-handling system. We presented the case study as an example of a larger space of opportunities in the realm of complementary computing, pursuing the development of ideal configurations, patterns of initiative, and workflows within systems composed of people and computational components.

We discussed the abstraction of the flow of interaction of an automated dialog system into a set of conversational steps, the collection of competency and progress-related data as callers progressed through the dialog steps, and the construction of predictive models via machine learning from data that had been logged by a voice routing system in use for several years at our organization. We demonstrated the construction of models that can predict, at each step within a dialog, the ultimate outcomes of interactions with an automated dialog system, and the expected durations of time of the session, conditioned on each outcome. Then, we presented policies that transfer calls away from a legacy system, based on an objective function that seeks to minimize the overall interaction times for callers. We discussed how the policies can be executed, relying on the inferences from the learned models about the ultimate outcomes and durations under uncertainty. We tested the behavior of the policies within a simulation environment that uses as test cases real-world calls that had been logged by the legacy automated dialog system. We examined recommended transfer actions, conditioned on different assumed wait times for accessing a human operator. The studies with the time minimization policy showed that the decision-theoretic policies, driven by the learned models, can save callers time. After investigating the time-minimization policies, we reviewed a more detailed preference model that represents multiple dimensions of cost and value in a complementary-computing solution. The extended model highlights directions in utility assessment, data collection and modeling for complementary computing solutions.

The methods and studies demonstrate specifically the value of employing the decision-theoretic policies for transferring callers to human operators. More generally, the methodology demonstrates how we can use machine learning to characterize an automated service, and then apply inference with learned models to build a more reflective service that can reflect about the best times to engage human resources to assist with solving problems.

We hope that the methods and case study we presented will stimulate additional interest in opportunities for employing machine learning and expected-value decision making to weave computational and human resources together into effective composite systems. We believe that there is a large space of opportunities with employing analogous learning and reasoning in other realms to guide the design, fielding, and testing of complementary-computing solutions that optimize the way people and machines work together to deliver solutions and services.

Appendix I.

Observational evidence used in models

Dialog status log

- **numSteps**: The number of steps as defined by the logging system; the number does not necessarily match the number of utterances.
- **systPrompts**: The prompt type sequences such as 'Greeting with Operator Option' followed by 'Confirm Top Choice,' followed by 'Spell First Name,' etc.
- **userAnswerTypes**: User answer sequences such as 'Name' followed by 'Spelling'.
- **numOfTurns**: The number of turns, defined as the number of user utterances.
- **numOfNameAttemptsDetected**: The number of name attempts detected by the recognizer. At times, for a single utterance, there may be two name attempts detected, depending on whether the recognizer goes through a second pass.
- **numNBestList**: The number of the *n*-best lists generated by the recognizer. This number does not necessarily match numOfNameAttemptsDetected.
- **reco_date**: The date of the interaction. As employees leave and new employees are hired, the efficacy of the language model for different requests may change over time.

Speech recognition features-base level

For any observed feature_*i*, the index *i* represents the *i*th utterance.

- **maxRedundFirstNames:** Maximum number of times a first name is repeated in the n-best list; *i.e.*, the cardinality of the most frequently occurring first name.
- **maxRedundFirstNames:** Maximum number of times a first name is repeated in the *n*-best list.
- **maxRedundLastNames:** Maximum number of times a last name is repeated in the *n*-best list.
- **maxRedundFullNames:** Maximum number of times a full name is repeated in the *n*-best list.

- **freqRedundFirstName**s: Number of distinct first names that have one or more repetitions in the list*; i.e.*, the cardinality of distinct names that are found to be repeated
- **freqRedundLastNames**: Number of distinct last names that have one or more repetitions in the list
- **freqRedundFullNames**: Number of distinct full names that have one or more repetitions in the list
- **count**: The number of items in the current *n*-best list.
- **sum**: The sum of all the confidence scores.
- **maximum**: The maximum confidence score.
- **minimum**: The minimum confidence score.
- **range**: The difference between the maximum and minimum confidence scores.
- **median**: The median confidence score if any.
- **arith_mean**: The arithmetic mean of the confidence scores.
- **geo_mean**: The geometric mean of the confidence scores.
- **greatestConsecDiff**: The greatest difference between any two consecutive confidence scores, if there are two or more confidence scores.
- **variance**: The variance of the confidence scores.
- **standard_dev**: The standard deviation of the confidence scores.
- **standard_error**: The standard error of the confidence scores.
- **mode**: The mode of the confidence scores.
- **modeFreq**: The frequency of the mode.
- **skewness**: The skewness of the distribution of confidence scores.
- **kurtosis**: Kurtosis of the distribution of confidence scores.

Speech recognition feature-combinations

- For any observed feature_*i*_*j*, the index *i* and *j* represent the *i*th and *j*th utterance respectively.

- **maxRedundFirstNamesBtw_i_j**: The maximum number of times any first name is repeated between the *i*th and the *j*th *n*-best lists; *i.e.*, the cardinality of the most frequently occurring first name between lists.
- **maxRedundLastNamesBtw_i_j**: The maximum number of times any last name is repeated between the *i*th and the *j*th *n*-best lists.
- **maxRedundFullNamesBtw_i_j**: The maximum number of times any full name is repeated between the *i*th and the *j*th *n*-best lists.
- **freqRedundFirstNamesBtw_i_j**: The number of distinct first names that have one or more repetitions in both the *i*th and *j*th *n*-best lists; i.e., the cardinality of distinct first names that repeat between lists.
- **freqRedundLastNamesBtw_i_j**: The number of distinct last names that have one or more repetitions in both the *i*th and *j*th *n*-best lists.
- **freqRedundFullNamesBtw_i_j**: The number of distinct full names that have one or more repetitions in both the *i*th and *j*th *n*-best lists.
- **getsBetterBtw_i_j**: Whether the average confidence score is higher in the *j*th utterance than in the *i*th utterance
- **topScoreDiffBtw_i_j**: Difference between the *i*th and *j*th top confidence scores

Acknowledgments

We thank Johnson Apacible for support with the data extraction and analysis and Robert Moon for assistance with gaining access to the VoiceDialer logs.

References

[1] Biermann, A. W., Inouye, R. B., McKenzie, A. (2005). Methodologies for Automated Telephone Answering, *Proceedings of ISMIS*, Saratoga Springs, NY, pp 1-13.

[2] Boger, J., Poupart, P., Hoey, J., Boutilier, C., Fernie, G., Mihailidis, A. (2005). A Decision-Theoretic Approach to Task Assistance for Persons with Dementia. *Proceedings of IJCAI*, Edinburgh, Scotland, pp. 1293-1299.

[3] Chickering, D.M., Heckerman, D. and Meek, C. A (1997). Bayesian approach to learning Bayesian networks with local structure. In *Proceedings of UAI*, Providence, RI, Morgan Kaufmann, pp. 80-89.

[4] D'Agostino, D. (2005) Weak Speech Recognition Leaves Customers Cold, CIO Insight, Ziff-Davis, December 29, 2005. (http://www.cioinsight.com/print_article2/0,1217,a=168124,00.asp)

[5] Friedman, N. and Goldszmidt, M. (1996). Learning Bayesian networks with local structure. In *Proceedings of UAI*, Portland, OR, Morgan Kaufmann, pp. 252–262.

[6] Hearst, M.A. Trends & Controversies: Mixed-initiative interaction (1999). *IEEE Intelligent Systems* 14(5), IEEE Computer Society, pp. 14-23.

[7] Horvitz, E. (1999). Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI*, Pittsburgh, PA, ACM Press, pp. 159-166.

[8] Kaelbling, L.P., Littman, M.L., and Moore, A.P. (1996). Reinforcement Learning: A Survey, *Journal of Artificial Intelligence Research*, 4:237-285.

[9] Langkilde, I., Walker, M., Wright, J., Gorin, A., and Litman, D. (1999). Automatic Prediction of Problematic Human-Computer Dialogs in How May I Help You? *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, Keystone, CO, pp. 369-372.

[10] Litman, D., Walker, M., and Kearns, M. (1999). Automatic Detection of Poor Speech Recognition at the Dialog Level, In *Proceedings of ACL 1999*, College Park, MD, pp. 309-316.

[11] Litman, D., and Pan, S. (2002). Designing and Evaluating an Adaptive Spoken Dialog System, *User Modeling and User-Adapted Interaction*, 12(2/3), pp. 111-137.

[12] Paek, T. and Horvitz, E. (2004). Optimizing call routing by integrating queuing models with spoken dialog models. In *Proceedings of HLT/NAACL 2004*, Boston, MA, pp. 41-48.

[13] Suhm, B., Bers, J., McCarthy, D., Freeman, B., Getty, D., Godfrey, K., and Peterson, P. (2002). A Comparative Study of Speech in the Call Center: Natural Language Call Routing vs. Touch-Tone Menus, In *Proceedings of SIGCHI*, MN, pp. 283-290.

[14] Tatchell, G.R. (1996). Problems with the existing telephony customer interface: The pending eclipse of touch-tone and dial-tone, In *Proceedings of SIGCHI*, Vancouver, BC, pp. 242-243.

[15] Walker, M., Langkilde, I., Wright, J., Gorin, A., and Litman, D. (2000). Learning to Predict Problematic Situations in an automated dialog system: Experiments with HMIHY? In *Proceedings of ANLP-NAACL-2000*, Seattle, WA, pp. 210-217.

[16] Walker, M., Langkilde-Geary, I., Hastie, H., Wright, J., and Gorin, A. (2002). Automatically Training a Problematic Dialog Predictor for the HMIHY Spoken Dialog System, *Journal of Artificial Intelligence Research* 16: 293 – 319.

[17] Xu, W. and Rudnicky, A. (2000). Task-based dialog management using an agenda. *Proceedings of ANLP-NAACL 2000 Workshop on Conversational Systems*, Seattle, WA, pp. 42-47.

Eric Horvitz

Microsoft Research, One Microsoft Way, Redmond WA, USA 98052

Eric Horvitz is Research Area Manager and Principal Researcher at Microsoft Research. He has pursued basic and applied research in decision making under uncertainty, machine learning and reasoning, information retrieval, user modeling, and human-computer interaction. He is President-elect and Fellow of the American Association for Artificial Intelligence (AAAI). He received PhD and MD degrees at Stanford University. The work with Tim Paek described in this article stems from a research program on developing mixed-initiative and complementary computing solutions that enable people and computers to engage in dialog and to jointly solve problems. More information is at http://research.microsoft.com/~horvitz.

Tim Paek

Microsoft Research, One Microsoft Way, Redmond, WA, USA 98052

Tim Paek is a researcher in the Machine Learning and Applied Statistics group at Microsoft Research. Tim received his M.S. in Statistics and Ph.D. in Cognitive Psychology from Stanford University, and his B.A. in Philosophy from the University of Chicago. His primary research focus is on spoken dialogue systems. With a keen interest in enhancing deployed systems, he has pursued research in the following areas: dialogue management, user modeling, personalization, machine learning and human-computer interaction.