Optimizing Automated Call Routing by Integrating Spoken Dialog Models with Queuing Models

Tim Paek & Eric Horvitz
One Microsoft Way
Redmond, WA 98052
{timpaek|horvitz}@microsoft.com

Abstract

Organizations are increasingly turning to spoken dialog systems for automated call routing to reduce call center costs. To maintain quality service even in cases of failure, these systems often resort to ad-hoc rules for dispatching calls to a human operator. We present a principled procedure for determining when callers should be transferred to operators based on a cost-benefit analysis. The procedure integrates models that predict when a call is likely to fail using spoken dialog features with queuing models of call center volume and service time. We evaluate how the procedure would have performed on cases drawn from logs of interactions with a legacy spoken dialog system.

1 Introduction

Automated call handling systems have provided organizations with an opportunity to reduce the cost of handling incoming calls. The most common systems utilize touch-tone or dial-tone interaction, which many callers find difficult to use and frustrating. Callers in fact frequently seek assistance from a live operator at the first opportunity (Suhm et al., 2002). To improve user experience, many companies have been turning to spoken dialog systems. These systems utilize automatic speech recognition (ASR) to facilitate requests in natural language, which customers overwhelmingly favor over touch-tone menus (Suhm et al., 2002). While spoken dialog systems purportedly reduce operation costs, when they fail, they not only waste the caller's time but also potentially damage customer relations by frustrating users. Moreover, failures jeopardize the return on investment (ROI) in deploying these systems. On the other hand, while human operators generally provide better customer service, they are much more expensive.

In attempting to get the best of both automation and live customer service, spoken dialog systems often resort to ad-hoc rules for dispatching calls. Most commonly, the rule is to simply dispatch a call when the system fails. Even when these rules are tuned from data, the decision to transfer a call typically does not take into consideration the real-time stakes, such as the cost of customer time and the loss of ROI. We present a principled procedure and systems design framework, based on modeling techniques from decision analysis and queuing theory, for determining when callers should be dispatched so as to minimize support costs. The procedure integrates models that predict when a call is likely to fail based on spoken dialog features with queuing models of call center volume and service time.

In the first part of the paper, we describe and evaluate predictive models learned from the session logs of a legacy call routing system deployed at Microsoft. Then, we present the formal details of the optimization procedure and evaluate how it would have performed on the real-world outcomes of the legacy system.

2 Related Research

The prediction of problematic situations in a spoken dialog system has links to research on the *identification* of when users are experiencing poor speech recognition performance (Litman et al., 1999). In the TOOT system, dialog strategies, such as taking system or mixed initiative, are adapted to user responses (Litman & Pan, 2002) based on rules from a classifier of "good" and "bad" dialogs trained over whole dialog sessions. Unlike the cost-benefit approach that we propose, the investigators employed deterministic policies as a function of the output of the classifiers.

Models that move beyond identification and actually predict where problematic situations in a call handling context are likely to occur have been previously explored with the AT&T *How May I Help You* (HMIHY)

call routing system (Langkilde et al., 1999; Walker et al., 2002; Walker et al., 2000). The HMIHY system provided over a dozen services. Using features from the speech engine, the natural language understanding component, and dialog manager, as well as hand-labeled features, classifiers were trained to predict failures *before* they occurred based on observations available to the system after the first exchange, second, third, etc.

In similar spirit to the HMIHY work, we explored the use of online models that could predict the duration of time until various outcomes with an automated dialog system were likely occur, in support of decision-theoretic procedures for minimizing a caller's time (Horvitz & Paek, 2003). This work was initially performed in the context of static call handling resources, taking as input a variable wait time for a human operator. In this paper, we relax this constraint with the introduction of a queue-theoretic model for predicting operator load and wait time based on actual call center data. The queue-theoretic model provides a modeling and simulation capability that can be used for exploration and design of call center staffing, as informed by the performance of a spoken dialog system. By considering call center costs, such as the cost of a caller's time, as well as predicted times, such as time waiting for an operator or before a failure occurs, the procedure we describe endeavors to strike a balance as a call progresses between keeping customers in the automated system versus transferring them to a live operator so as to minimize overall support costs.

3 Predicting Dialog Failure

In this section, we first describe the legacy call routing system that has been deployed at Microsoft. Then we summarize the logged data obtained from the system and review details of the spoken dialog features we extracted to construct case libraries for training and testing. Finally, we evaluate the performance of the models we learned for predicting whether a call is likely to ultimately fail or succeed.

3.1 Legacy Call Routing System

In the past, the Microsoft Corporation fielded a commercially available spoken dialog system called *VoiceDialer* for handling directory assistance requests. Using speech recognition, *VoiceDialer* attempted to identify one of over 20,000 name entries in a global address book. We analyzed nearly 60,000 session logs collected over 11 months. We found that the system succeeded in correctly identifying the proper name in only 45% of the sessions. The success rate jumped to 69% when sessions in which the caller did not even attempt to engage the system were removed; in such "no name attempt" sessions, accounting for roughly one out of every three sessions, users completely bypassed interaction with the spoken dialog system by either immediately transferring to an operator with a touchtone command or hanging up. On top of this alarming failure rate and immediate bypass of the system, longitudinal trends revealed that a growing percentage of callers were requesting the operator. The urgency of these findings motivated the need to devise a procedure for optimizing dispatch to an operator based on predictive models constructed from features drawn from the session logs.

3.2 Extracting Cases

In seeking to extract cases for building predictive models, we analyzed the logs generated by *VoiceDialer*. The session logs included transcriptions of all system actions as well as the main output of the speech recognizer; namely, *n*-best lists of hypotheses for the first and last names with their corresponding confidence scores. The definitions and distribution of final outcomes for sessions, as labeled by the system, were as follows:

- *SpeakFound* (45%): System finds the correct name in the directory, as confirmed by a transfer.
- *OperatorRequest* (23%): Caller presses '0' for an operator.
- *HangUp* (13%): Caller hangs at some point in the session.
- *MaxErrors* (12%): System reaches threshold of allowed misrecognitions and routes the call to an operator.
- *SpeakNotFound* (6%): System concludes that the name is not in the directory and routes the call to an operator.
- *Undefined* (1%): Caller presses other numeric keys.
- *HelpRequest* (<1%): Caller requests help by pressing '*' or '#'.
- *NotReady* (<1%): System is temporarily out of service.

We built models to predict these outcomes from a set of observational features encoded in the logs. We were particularly interested in inferring the likelihood of the eventual success versus failure of the spoken dialog interaction. For our cost-benefit modeling, we also sought to infer probability distributions over the duration of time until success or failure of the system.

The spoken dialog features we extracted from the session logs fell into four broad categories (listed along with the total number of features for each category):

- *Sequences of system and user action types* (3): Sequences of system actions, such as asking the

Target	Recognition 1		Recognition 2		Recognition 3		Recognition 4	
	First Split	Nodes (Depth)	First Split	Nodes (Depth)	First Split	Nodes (Depth)	First Split	Nodes (Depth)
Outcome	skew_1	18 (8)	gcdiff_2	27 (9)	max_3	12 (6)	gcdiff_4	1 (2)
Failure	skew_1	18 (8)	gcdiff_2	23 (9)	max_3	12 (6)	gcdiff_4	13 (6)
Duration	skew_1	19 (10)	skew_2	16 (7)	min_3	13 (6)	max_4	7 (4)

Table 2. Summary of decision trees for the primary target variables.

user to repeat first/last/full name, sequences of user actions such as pressing a key, etc.

- *Whole dialog features* (2): Outcome, total completion time or duration
- *ASR features* (22): Number of hypotheses in the *n*-best list, range of confidence scores, mode, greatest consecutive score difference (gcdiff), skewness of the scores (skew), maximum score (max), minimum score (min), etc.
- *Pairwise ASR features* (8): Number of recurring first/last/full names that match in consecutive n-best lists, whether the maximum score increased or decreased, etc.

Note that all these features can be observed in real-time as a dialog session progresses, with the exception of whole dialog features, which constitute the target variables.

3.3 Incremental Data Sets

Because the optimization procedure harnesses models for real-time decision making over the course of a dialog, we decomposed the data into sets of features that are revealed incrementally with dialog progression. We take as the fundamental unit of time each posting by the speech engine of ASR outputs. We segmented the data by detected ASR outputs for two reasons: first, ASR features constituted the vast majority of automatically extractable real-time features, and second, alternative dialog units (*e.g.*, "moves," "adjacency pairs" or exchanges) were either unavailable in the transcriptions or provided insufficient discriminatory features.

Four data sets were created with incrementally growing number of features, as summarized in Table 1. The data was split 70/30 for training and testing. Having no ASR output was not included as a data set since not enough features could be extracted, rendering it functionally equivalent to using the marginal distribution in lieu of an inference. Furthermore, data sets with greater than 4 ASR outputs were dropped since, out of nearly 60,000 session logs, we only found 6 such cases.

	Features	Train	Test	Total
Recognition 1	32	27587	11824	39411
Recognition 2	62	13556	5811	19367
Recognition 3	93	5625	2411	8036
Recognition 4	122	2579	1106	3685

Table 1. Summary of incremental data sets.

3.4 Building Predictive Models

For model building, we learned Bayesian networks employing decision trees to encode local conditional probability distributions within variables using a tool that performs Bayesian structure search (Chickering, 2002). Decision trees can be learned for both discrete and continuous variables, where splits in the trees are made through greedy search guided by a Bayesian scoring function (Chickering et al., 1997). We learned Bayesian networks not only to perform inference over the joint distributions needed for the decision-theoretic procedure, but also to determine what spoken dialog features would comprise the local structure of three primary variables of interest: *Outcome*, as previously defined, *Failure*, a binary recoding of *Outcome* with *Speak-Found* as '1' and '0' for everything else, and finally, *Duration*, the expected completion time of the session. Note that *Duration*, a continuous variable for which the decision tree learned a Gaussian distribution, is required since cost is oftentimes a function of time.

Table 2 shows a summary of the decision trees that were learned for the target variables in all four data sets. In all cases, the first splits in the decision trees, which represent the feature with the strongest dependency, were ASR features for the most recent *n*-best list. For example, the decision tree for *Outcome* trained on all available features after the fourth ASR output depended most on the greatest consecutive difference between any two hypotheses for just the last *n*-best list, and not on any pairwise features between the third and fourth, or the second and third *n*-best lists, though pairwise features and action sequences were included as dependencies. We were surprised to find that *Outcome* after the fourth recognition only depended on that feature.

3.5 Evaluation

Table 3 displays the classification accuracies of the decision trees for *Outcome* and *Failure* on the test data.

	Recognition 1	Recognition 2	Recognition 3	Recognition 4
Marginal Outcome	68.7%	61.8%	48.3%	56.8%
Outcome (Lift)	71.5% (2.9%)	67.0% (5.2%)	62.9% (14.6%)	77.1% (20.3%)
Marginal Failure	68.7%	61.8%	48.3%	61.8%
Failure (Lift)	75.9% (7.2%)	75.1% (13.3%)	71.3% (23.0%)	82.3% (20.4%)

Table 3. Classification accuracies for predicting dialog outcome and failure, and their relative improvement.

	Recognition 1	Recognition 2	Recognition 3	Recognition 4
Marginal Duration	-0.38	-0.28	-0.15	-0.09
Duration (Lift)	-0.05 (0.33)	-0.06 (0.22)	0.04 (0.19)	0.11 (0.20)
Marginal Duration + Outcome	-0.71	-0.72	-0.65	-0.49
Duration + Outcome (Lift)	-0.50 (0.21)	-0.52 (0.20)	-0.48 (0.18)	-0.25 (0.24)
Marginal Duration + Failure	-0.50	-0.47	-0.42	-0.38
Duration + Failure (Lift)	-0.30 (0.20)	-0.30 (0.18)	-0.25 (0.17)	-0.15 (0.23)

Table 4. Log posterior scores for the learned Bayesian network models and their corresponding lifts above the marginal model.

The baselines represent their marginal distributions. Not surprisingly, the *Failure* models outperformed the *Outcome* models, with a maximum accuracy of 82% for the data after the fourth recognition. Consistent with intuition, the lift above the marginal gradually rises, with the highest gain relative to the baseline at 23%. Looking at the classification accuracy for *Outcome* after the fourth recognition, despite the fact that there was only one dependency, as stated previously, the model performed 20% better than the baseline. While the lifts for the third and fourth recognitions seem impressive, the baselines are dismally low, attesting to the poor performance of the legacy spoken dialog system.

The log posterior of the data given the models are reported in Table 4. To evaluate the relative performance of the learned Gaussians for *Duration* over the marginal log score, models were trained using just that as the target variable. Thereafter, models included either *Outcome* or *Failure* as the second target variable. Note that positive log scores reflect a non-normalized Gaussian density function. The maximum lift above the marginal was 0.24, though unlike the classification accuracies, the lifts for the log scores do not exhibit a trend upward. On the other hand, the log scores in general do seem to improve with more features.

4 Optimization

The purpose of building models that can predict the likely outcome of a call with a spoken dialog system using real-time, extractable features is to employ them in a modeling and design setting as well as for optimal decision making. Optimization can be approached from with different objective functions. Given that the goal of optimizing dispatch to a live operator is to minimize support costs at the enterprise level, while at the same time exploiting real-time likelihoods, we combine probability and utility within the framework of decision theory according to the principle of maximum expected utility (MEU), which states that we should select the action $A = a$ that maximizes its expected utility, $EU(a|\xi)$. If ξ denotes all background information and H represents all possible states of the world, then we select actions guided by the following optimization:

$$\arg\max_a EU(a \mid \xi) = \arg\max_a \sum_h P(H = h \mid \xi) u(a, h) \quad (1)$$

where $u(a,h)$ expresses the utility of taking action a when the state of the world is h. Note that cost is simply negative utility.

Relating the MEU principle to the process of optimizing dispatch, let d denote the action of dispatching a call. Furthermore, let S denote the possible outcomes of the call routing system, which, for the sake of simplicity, corresponds to the binary variable *Failure*. Since transferring a call when the operator is busy poses a problem, let O denote the state of the operators, which may or may not be busy. Rewriting (1) to include O and S, we obtain the following optimization procedure:

Dispatch Procedure: Dispatch a call to an operator only when the expected utility of d, given the state of the operator O and call routing system S, exceeds that of any dialog action a. That is,

$$\forall_{a \neq d} \left[EU(d \mid S, O) > EU(a \mid S, O) \right] \quad (2)$$

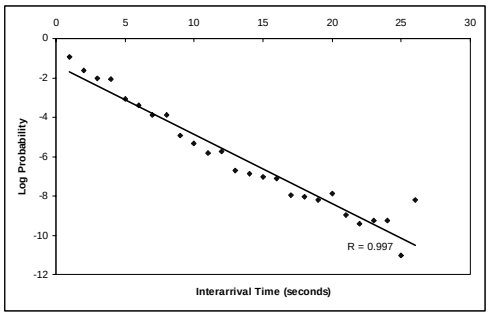

Figure 1. Linearity of the log transformed inter-arrival times.

In particular, for dispatch *d*, applying the definition of conditional probability, we obtain:

$$EU(d) = \sum_S \sum_O P(S,O \mid d) u(d,S,O)$$

$$EU(d) = \sum_S \sum_O P(O \mid S,d) P(S) u(d,S,O) \quad (3)$$

Calculating the expected utility of a dispatch, as delineated in (3), involves three components: first, $P(O|S,d)$, or the likely state of the operator queue; second, $P(S)$, or the likely outcome of the call routing system; and third, $u(d,S,O)$, or the utility of dispatching a call when the operators may or may not be busy and the system may or may not be failing. The first component requires stochastic modeling of the call center queue. The second component was discussed previously. And finally, the third component involves the application of standard cost functions in operations research. We elucidate the first and last components in detail below.

It is important to note that the only dialog action that affects *O*, whether or not the operator is busy, is *d* since a transfer increases the number of callers waiting to be serviced by the operator. The effect of all other dialog actions taken by a call routing system remain within the system, and as such:

$$P(O \mid S, \neg d) P(S) = P(O \mid \neg d) P(S) \quad (4)$$

In other words, *O* and *S* are probabilistically independent for all other dialog actions, though for simplicity, we only consider the action of keeping someone engaged with the spoken dialog system.

4.1 Modeling the Call Center Queue

While modeling the call center may require more than simple parameter estimation, calculating whether or not the operators are busy, once the queuing models have been fit, entails either closed-form solutions or can be estimated using Monte Carlo methods. Since the calculations depend on what kind of queuing models evince the best goodness-of-fit, we first describe how we fit the models using data collected from the call center at Microsoft, and then discuss the calculations.

4.1.1 Fitting a Poisson Process

Many queues, and in particular, call centers, are governed by two parameters: λ, the average arrival rate into the call center, and μ, the average service time to complete a call once an operator receives it. If the counts of inter-arrival and service times follow an exponential distribution, which exhibits the *memoryless* property of being probabilistically independent of any previous call, the process over time is called a *Poisson process* (Gross & Harris, 1998).

Since a Poisson process is mathematically well-characterized, we sought to ascertain whether the call center at our organization could be modeled as such. For that effort, we obtained about two months of call center data for weekday working hours. Over 1700 calls were received with varying service completion times. The average rates, which represent both the maximum likelihood estimate and method of moments estimate for the exponential distribution, were 4.41 seconds (λ) between calls and 28.22 seconds (μ) to dispense a call. To make sure that the distributions were indeed exponential, we performed a log transformation of the empirical distributions and fit regression lines to estimate the correlation coefficients. Figure 1 shows the linear regression fit for the inter-arrival times. The correlation for the arrival rate ($r=.997$) was significant ($t(26)=63.06$, $p<.001$), and likewise, the correlation for service rate ($r=.710$) was significant ($t(24)=4.69$, $p<.001$). Hence, the fit arrival and service processes were quite reasonably Poisson.

4.1.2 Calculating Operator Load

Given that the arrival and service rates for the call center were Poisson processes, we used the estimated λ and μ parameters to model an M/M/s queue, which denotes a queue in which the arrival process is "memory-less," as well as the service process, and the number of servers is *s*, though we use *z* to avoid confusion with *S*, the state of the system. The call center at Microsoft employs 10 operators or servers.

Returning to the first component of (3), calculating the likelihood that all the operators are busy in a call center if a call is dispatched for an M/M/z queue is:

$$P(O \mid S,d) = P(n \geq z)$$

$$P(O \mid S,d) = \left(1 - \sum_{n=0}^{z} P_n\right) \quad (5)$$

where *n* represents the number of callers, and P_n is a closed form solution for multiple servers (Gross & Har-

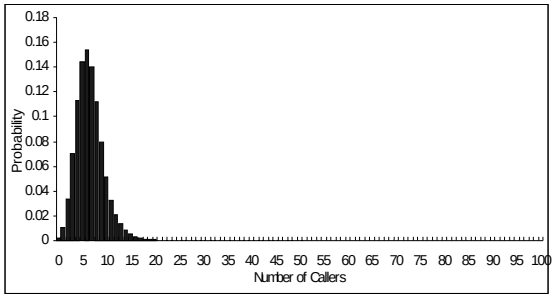

Figure 2. Distribution of the likely number of callers in the call center.

ris, 1998). According to (5), an operator is busy when the number of callers in the queue, including the dispatch call from the automated system, exceeds the number of operators at the call center.

Figure 2 displays the distribution over the likely number of callers in the queue for an M/M/z queue using the fitted parameters for the call center data and 10 operators. According to this distribution, the likely number of callers at any given time is around 7. To verify the appropriateness of our queuing model, we checked all statistical implications with supervisors of the call center, including total waiting time, and found that indeed the M/M/z queue made accurate predictions regarding call center statistics.

4.2 Assessing Costs

Having delineated how to calculate the likelihood of the operators being busy, and how to predict the likelihood of success using spoken dialog features, we now turn to the last component of the optimization in (3), assessment of the utility of dispatching a call given the state of the operators and the spoken dialog system. In many respects, the utilities drive the optimization in that ultimately a company has to defray the expense of maintaining a call center. This is what distinguishes the procedure: it bases its decisions on the overall cost and benefit of running a call center with both an automated call routing system and a staff of human operators, weighted by the efficiency and performance of both. Since queuing models are integrated with dialog models, the cost-benefit analysis allows call center managers to determine if less operators are needed as the automated system improves its performance, and vice versa. In other words, the integration provides a framework for optimizing call center design with support costs playing the principal role.

The utility of a dialog action a given the operator state O and system state S can be approximately decomposed as follows:

$$u(a,S,O) \cong u(a,S) + u(a,O) \quad (6)$$

Suppose $a = d$, or dispatch to an operator. Then, (6) can be further decomposed into the general cost function (note c instead of u):

$$c(d,S,O) \cong CustCost \cdot (t+W) + OpCost \cdot z \quad (7)$$

where t is the time a caller has already spent in the automated system, and W is the predicted "dwell time," which includes both the time waiting in line and the time being served (Gross & Harris, 1998). W is derived from the M/M/z queuing model. According to (7), the cost of dispatching a call is simply the cost of a caller's time with the automated system in the call center, plus the cost of employing z operators for that call. When the state of O is "Not Busy," we can drop out the W. When the state of S is "System Failing," we require another term in (7); namely, the return on investment (ROI). The ROI in this context is the amount of capital that the organization would have saved in *not* hiring more operators.

In order to calculate ROI for the call center, we first modeled the inter-arrival rate of calls entering the spoken dialog system as a Poisson process. The average arrival rate was 1.51 seconds, and a linear regression line fit to the log transformation of the empirical distribution revealed a significant correlation ($r=.997$, $t(30)=65.9$, $p<.001$), indicating an exponential distribution. Since the average service time of the operators remains the same, we used an M/M/z queuing model and (7), without the variable t, which is unknown, to determine the optimal number of the operators. This optimization is common in operations research (Ashley, 2002). The optimal number of operators needed to field the calls that would have been handled by the automated system is 25, as shown in Figure 3. Note that this is a conservative estimate since operators not only handle those calls dispatched from the automated system but also direct operator lines. ROI is then simply *Op-*

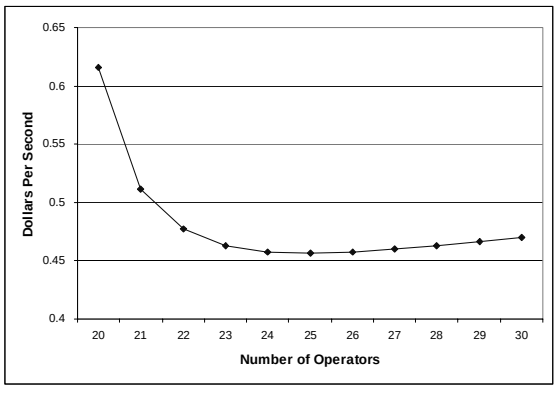

Figure 3. Optimizing the number of operators that would have been required to handle calls to the automated call routing system.

	Recognition 1			Recognition 2			Recognition 3			Recognition 4		
Outcome	Total	Marg	DT	Total	Marg	DT	Total	Marg	DT	Total	Marg	DT
Failure	31.3%	48.4%	1.5%	38.2%	55.7%	30.9%	51.7%	61.2%	30.7%	61.8%	88.9%	52.6%
Success	68.7%	11.6%	6.8%	61.8%	12.9%	2.9%	48.3%	17.9%	2.5%	38.2%	28.4%	4.7%

Table 5. Percentage of failure and success calls that would have been dispatched by the decision theoretic (DT) procedure and by an alternative procedure using the marginal (Marg)

Cost·25. We also performed the same optimization using the inter-arrival rate for the call center, and found that the optimal number of operators was 11, one more than the call center was currently employing, though apparently 2 supervisors are always on hand to take calls to avoid overly lengthy queues. This was yet another validation of queuing model.

When the dialog action is to not dispatch, $\neg d$, but to keep a caller in the automated spoken dialog system, the second term of (7) drops out since we only need to consider the cost of a caller's time. Furthermore, if the state of *S* is "System Not Failing," then having kept the caller in the system would have saved ROI. Hence, ROI needs to be subtracted from the cost function. Finally, we need to consider the future cost of remaining in the system for an estimated amount of time longer than *t*, the equivalent of *W* in the queuing model. Here we apply the Gaussian models we learned from the session logs in the previous section to estimate this duration.

4.3 Evaluation

Putting everything together, we can combine the spoken dialog models we learned in the first section, with the fitted M/M/z queuing models, along with the cost functions for both d and $\neg d$ for every cross product of *S* and *O* within the decision-theoretic procedure for optimizing dialog actions delineated in (6). To assess the effectiveness of the procedure, we used the same test data for evaluating the spoken dialog models to examine how many calls the procedure would have dispatched after each ASR output given the incrementally growing number of spoken dialog features. As a comparison, we considered an alternative baseline procedure of using the marginal distribution to decide whether to dispatch: namely, if the most likely state of *S*, or the binary variable *Failure*, is "System Not Failing," then keep the call in the system; otherwise, dispatch the call. Note that in using the marginal distribution, the alternative procedure represents a more intelligent way of transferring a call than simply dispatching when the system reaches a failure, the most prevalent procedure in automated call routing. The marginal procedure was also selected because it allows us to compare how the system would have performed without consideration of the queue.

Table 5 displays the results of the analysis on each test set comparing the decision theoretic (DT) procedure against the marginal procedure (Marg). Looking at the second row, which corresponds to the "false positive" cases; that is, incorrectly dispatching a call that was actually ultimately successful, the DT procedure is fairly conservative about making false positives, consistently dispatching less than 7% of the calls. On the other hand, the marginal procedure continually increases its rate of false positives, reaching 28% by the last ASR output. One way to interpret these results is to say that using just the predictive models, as in the case of the marginal procedure, is not enough. Optimal performance benefits from the incorporation of both a model of the operator queue and the stakes involved.

Looking at the first row, corresponding to those "true positive" cases in which the calls that would have ultimately failed were dispatched correctly, the DT procedure starts off conservatively dispatching cases, whereas the marginal procedure immediately transfers close to half of the calls, which makes sense given that these are the calls that ultimately failed. The reason why the DT procedure does not transfer as many calls is because the loss of ROI is heavy in the beginning but is eventually outweighed by the cost of failure. In other words, the DT procedure is attempting to save the ROI. As more recognition results are received and the percentage of calls that ultimately fail in the system increase, the DT procedure gradually lifts the percentage of dispatch, even transferring over half of the failed calls by the fourth ASR output. Although the marginal procedure also demonstrates a similar trend, it does so in a much more aggressive fashion.

4.4 Approximate Cost Savings

To better appreciate how the DT procedure balances the tradeoffs between dispatching a call to an operator and keeping it within the automated system, we can approximately monetize the average cost savings for each test data set as follows:

$$AverageSavings = \frac{\sum_{n} \left| ec(d) - ec(\neg d) \right|}{n} \tag{8}$$

The intuition behind (8) is that if the expected cost of dispatching is greater than not dispatching and we decide to stay in, we would cut costs. Conversely, if the

	Average Savings	Dispatch Only
Recognition 1	$0.1952	$0.0010
Recognition 2	$0.1399	$0.0065
Recognition 3	$0.1565	$0.0101
Recognition 4	$0.1973	$0.0288

Table 6. Average cost savings in dollars per second using the decision-theoretic procedure.

expected cost of staying in is greater than dispatching and we decide to transfer the call, we would again cut costs. (8) approximates the average amount of cut costs saved for both decisions.

Table 6 displays the average cost savings using the DT procedure for all four data sets. The procedure cuts more costs in the first recognition, where it keeps calls in the system at a point when the probability of success is highest, and again in the fourth recognition, where it dispatches calls to the call center at a point when the probability of success is lowest. Looking only at the average savings when the expected cost of dispatch is lower than the expected cost of keeping a caller in the automated system, we can see a gradually increasing trend of cost savings, culminating at roughly 3 cents.

5 Discussion and Future Research

In this paper, we presented a decision-theoretic procedure for determining when callers should be dispatched to a live operator so as to minimize support costs at the enterprise level. The procedure integrated learned models of call failure and success based on extractable real-time spoken dialog features with queuing models of call center volume and service time. The spoken dialog models predicted failure with a maximum accuracy of 82%, a 20% relative lift above the baseline. Using an M/M/z queue fit to call center data, we evaluated the procedure against making decisions from the marginal distribution. The procedure was shown to be conservative about making false positives. For true positives, the procedure reliably dispatched more calls as the probability of success diminished.

One limitation in the evaluation we performed of the procedure was that we did not account for the increase in average arrival rate into the call center as the procedure dispatched more calls. In a live system, we would monitor the average rate of dispatch, and adjust the average rate of arrival in the queuing models of the call center accordingly. Although we have not implemented a live system, we are exploring the possibility of integrating the decision-theoretic procedure with a new call routing system that has recently been deployed.

Finally, while this paper investigated the problem of optimizing dispatch at the enterprise level, the same framework can be applied to optimizing different costs, such as user frustration. We discuss different kinds of utility models including those that consider the principal agent of call-handling decisions to be the caller, versus the hosting organization in (Horvitz & Paek, 2003).

References

Ashley D. 2002. "An Introduction to Queuing Theory in an Interactive Text Format," *INFORMS Transactions on Education*, 2(3).

Chickering, D. 2002. "The WinMine Toolkit." *Microsoft Technical Report, MSR-TR-2002-103*.

Chickering, D., Heckerman, D., and Meeks, C. 1997. "A Bayesian Approach to Learning Bayesian Networks with Local Structure," *UAI-97*, pp. 80-89.

Gross, D. and Harris, C. 1998. *Fundamentals of Queuing Theory*. Wiley-Interscience.

Horvitz, E. and Paek, T. 2003. "Utility-Directed Coupling of Spoken Dialog Systems and Human Operators for Call Routing." *MSR Technical Report 2003-77*.

Langkilde, I., Walker, M., Wright, J., Gorin, A., and Litman, D. 1999. "Automatic Prediction of Problematic Human-Computer Dialogues in How May I Help You?" *Proceedings IEEE Workshop on Automatic Speech Recognition and Understanding*.

Litman, D., Walker, M., and Kearns, M. 1999. "Automatic Detection of Poor Speech Recognition at the Dialogue Level," *ACL-99*, pp. 309-316.

Litman, D., and Pan, S. 2002. "Designing and Evaluating an Adaptive Spoken Dialogue System," *User Modeling and User-Adapted Interaction*, 12(2/3): 111-137.

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," *TOCHI*, 4(1).

Tatchell, G. 1996. "Problems with the Existing Telephony Customer Interface: The Pending Eclipse of Touch-Tone and Dial-Tone," *CHI-96*.

Walker, M., Langkilde, I., Wright, J., Gorin, A., and Litman, D. 2000. "Learning to Predict Problematic Situations in a Spoken Dialogue System: Experiments with HMIHY?" *NAACL-2000*, pp. 210-217.

Walker, M., Langkilde-Geary, I., Hastie, H., Wright, J., and Gorin, A. 2002. "Automatically Training a Problematic Dialog Predictor for the HMIHY Spoken Dialogue System," *Journal of Artificial Intelligence Research*.