IEEE/ACL 2006 Workshop on Spoken Language Technologies (SLT 06), Aruba, Dec. 10-13, 2006

VALUE-BASED OPTIMAL DECISION FOR DIALOG SYSTEMS

Esther Levin[1] *and Roberto Pieraccini*[2]

[1]City College of New York –138th Street & Convent Avenue, New York, NY, 10031
[2]SpeechCycle, Inc. – 535 W 34th Street, New York, NY, 10001

ABSTRACT

In this paper we address the problem of optimizing the decisions taken by a spoken dialog system based on a given business model. Although the model can be applied to other types of decisions, we refer to the choice on whether to continue a session or to escalate it to a human agent. We consider a family of business models based on costs and savings that are dependent on the decision taken and the automation outcome of each session. Based on a corpus of more than 61,000 dialogs, we show that an optimal classifier can be selected to minimize the cost for a given business model.

1. INTRODUCTION

This paper is about optimizing interaction decisions taken by commercially deployed automated dialog systems with respect to a defined business model[1]. Although the methods described here can be generalized to other actions, the focus of this study is about the decision on whether to continue the interaction or to escalate the call to an agent.

The cost of customer care is a serious issue for providers of services such as broadband internet, cable TV, wireless, and VoIP telephony. Automated telephony based spoken dialog systems are a viable cost reduction solution. However, the current technology is not able to automate 100% of the calls, especially for complex tasks like technical support. Yet, state of the art automation rates in the range of 20% to 40% are showing compelling cost savings. When system automation fails the call is escalated to a human agent, but often callers have to go through a lengthy dialog with the system before that decision is taken. This situation has a negative impact at several levels. First, it is a bad caller experience that affects the service provider and sheds a negative light on the whole spoken dialog technology. Second, often there is a significant cost associated to long non automated calls. For instance, for systems hosted by third parties, there is a hosting cost proportional to the duration of each call. Additionally, depending on the adopted business model, a non-automated call corresponds to a missed opportunity for savings or missed revenue.

If a dialog system could predict the likelihood that a call would eventually result in a missed automation, it could use this information to decide whether it would be more business effective to continue the call or to escalate it to an agent right away. Previous works [5] [6] [7] [8] [9] address the problem of predicting the outcome of dialog. Since any prediction comes with a certain margin of error, it is very important, in order to optimize the decision, to take into account the correct business objectives, and integrate them with the likelihood of such errors. Differences in the business rules and costs can change the optimal decision policy.

In this paper we extend previous work on the prediction of dialog outcome in order to take into account quantifiable business objectives with the goal of making optimal decisions. We applied the proposed methods to real data collected from a high volume technical support application in the area of cable TV. The results of the experiments described in this paper show that it is possible to build a modeling tool that can help optimize the escalation decision for any given business objective parameters.

2. RELATED RESEARCH

Previous work on predictions of problematic situations in dialog has been carried out in the context of the AT&T *How May I Help You* (HMIHY) call routing system [5] [6] [7]. A Problematic Dialog Predictor was trained to predict failure in call automation based on acoustic, NLU, and dialog features extracted after the first few exchanges in the dialog.

Similarly [8] explored the problem of predicting the time until various outcomes were likely to occur in order to minimize call duration. In [9] an agent queue model for predicting caller wait time was integrated in the optimization process. This work, based on actual call center data, tried to determine a balance between keeping customers in the automated system versus transferring them to a human agent.

In this paper we extend the work described in [9] by training a set of classifiers to predict automation failure. We show

[1] Patent pending.

that an optimal decision policy can be selected from this set by systematically exploring the precision/recall curve through the application of a parametrically quantified business model.

3. TECHNICAL SUPPORT APPLICATIONS

The experiments described here are based on live data collected from a technical support application deployed for a large cable TV provider. The application provides support to callers who are experiencing one of a number of problems that can affect analog and digital cable TV service. Examples of the problems are the complete absence or poor quality of images and sound, error codes appearing on the cable box, missing or frozen channels, problems with the remote control, or with ordering pay-per-view or on demand events.

While simpler technical support applications resort to playing recorded messages extracted from a set of FAQs (Frequently Asked Questions), the application described here performs the same steps as a human agent would. The problem is identified from the symptom expressed by the caller in natural language. The identification of the symptom is carried out by using statistical language understanding (SLU), similar to that used for automated call routing [5], followed, if necessary, by further disambiguation

Once the problem is identified, one of several possible resolutions is selected and the caller is instructed on how to proceed. Typical resolution may go from rebooting the cable box, to checking the connectivity and the continuity of the cables, to correcting problems in the configuration of other devices, such as VCR, DVR, and DVD players.

Once the resolution is completed, the caller may acknowledge the solution of the problem. In this case, the call is considered fully *automated*. In case the problem is not solved, and other resolution steps are not available, the call is escalated to a human agent along with a report of what was done and indications on what needs to be done next (e.g. arrange the visit of a technician, ship a new cable box, etc.). This is considered a *partially automated* call. A third category—i.e. *non automated* calls—occurs when the caller hangs up before the call is completed. An early and accurate prediction of calls that will result in partial or no automation may lead to significant savings in time and costs.

4. BUSINESS MODEL PARAMETERIZATION

One of the current business models associated to the commercialization of customer care automation is based on the actual value provided by each call handled by the system. As we saw in the previous section, each individual call can provide full, partial, or no automation, and a different monetary value can be associated with each one of the three situations.

In the remainder of this paper we consider a simplified model consisting only of two categories: the call is automated (A) or not automated (NA), the latter category consisting of both partially and non automated calls.

A positive value, resulting from associated savings or revenue, can be associated with a call that belongs to category A. Conversely every call, regardless of the above categorization, may incur in a cost that accounts for hosting and commercial arrangement (e.g. licensing) of the system. The precise definition of the above categories and the associated saving and cost values depends on several factors, such as the cost of human agents and the particular commercial agreement between the dialog system developer and the service provider. Thus the optimal decision policy depends on the actual parameters of the business model—i.e. costs and savings—and the precise definition of what constitutes a fully automated call.

5. DECISION OPTIMIZATION

The parameters of a generic value-based business model with respect to the different decisions taken by the system (e.g. continue or escalate) can be expressed by the two cost matrices: $\mathbf{C}^N$ and $\mathbf{C}^T$. Matrix $\mathbf{C}^N$ represents the per-call costs. Each element $c^N_{i,j}$ is the cost associated to a call in category i (i = A, NA in the experiments reported here), regardless of its duration, when action j (j = escalate (E), or continue (C) in the experiments reported here) is performed. Matrix $\mathbf{C}^T$ accounts for the costs that are proportional to the duration of each call. Each element $c^T_{i,j}$ is the per-minute cost associated to a call in category i (e.g. i = A, NA), when action j (e.g. j = C, E) is performed. For instance, the following matrices:

$$\mathbf{C}^N = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix} \quad \mathbf{C}^T = \begin{bmatrix} 0 & 0.1 \\ 0 & 0.1 \end{bmatrix},$$

represent a business model where each automated call brings a saving (negative cost) of \$1 and each non escalated call (regardless of its level of automation) carries a cost of \$0.1 per minute.

With the above sample model, the cost of each action (E or C) can be computed as

$$C_E = 0; \qquad C_C = \begin{cases} -1 + 0.1 \cdot d & \text{if A} \\ 0.1 \cdot d & \text{if NA} \end{cases},$$

where d is the duration of the call after the decision point.

The problem of the optimal action to perform at any given point in dialog can be cast in terms of Bayesian decision theory by choosing the action that minimizes the *expected* cost. Thus, for any given A/NA classifier, the optimal action choice depends not only on the cost matrices $\mathbf{C}^N$ and $\mathbf{C}^T$, but also on the precision/recall performance of the classifier, and general call statistics.

Let's denote by *a* and *na* the events that a classifier predicts a given call being in category A or NA respectively. The precision of a particular classifier, for instance with respect to the NA decision, is an estimate of $P(NA|na)$, i.e. the probability that a call predicted as *na* by the classifier is indeed a non-automatable call. Precision determines whether the prediction is useful for reducing cost for given business model parameters.

For the above example, the action of escalation should only take place if the associated expected cost is smaller than that of continuing the call, i.e., given that the classifier prediction is *na*, only if

$$\bar{C}_C = P(NA|na)\cdot\bar{d}\cdot c^T_{NA,C} + P(A|na)\cdot(\bar{d}\cdot c^T_{A,C} + c^N_{A,C}) > 0$$

If the precision of the classifier is below the limit specified by the above equation, its prediction should be ignored and the call continued. Given a value of precision satisfying the above condition, the expected savings resulting from escalating the call following the *na* prediction is:

$$\bar{S} = (\bar{d}\cdot c^T_{NA,E} + (1 - P(NA|na))\cdot c^N_{A,C})\cdot P(na)$$

A possible approach to the cost minimization problem is to train a classifier to minimize the expected cost. Disadvantages of this approach are: a) most of the available machine-learning algorithms are designed to minimize an overall error rate and therefore cannot be used to minimize the expected cost as a function of classifier's performance, and b) the parameters of the business model can change over time and from customer to customer.

In the approach described here we perform optimization in two stages. In the first stage we generate a large set of classifiers, each one of them providing different precision/recall values. In the second stage we use the parameters of the business model to select the classifier that minimizes the expected cost.

5. DATA, FEATURE REPRESENTATION AND CLASSIFICATION MODELS

The experiments reported here were based on a corpus (Table 1) of over 61,000 logs from dialog sessions, of which 15,000, and 10,000 were used as development and training sets respectively. All results reported below are for classifiers trained using features extracted right after the prompt the instruct callers to describe their problem.

Similarly to [7] we extracted three types of features from the logs:

1. **Acoustic/ASR Features** including recognition status (rejection, no-input, and recognition), recognized utterance, utterance duration, number of words, recognition confidence, and DTMF input.
2. **SLU Features** including SLU hypotheses and corresponding confidence measures.
3. **Dialog Features,** including cumulative counts of re-prompts, confirmations, etc.

All features were extracted automatically, and each session was automatically labeled as A or NA.

	A	NA	A+NA
Number of sessions	14256	47275	61531
Average duration (min)	4.2	3.2	3.4
Average turns	13.9	10.6	11.4
Average duration after the decision point	3.4	2.4	2.6

Table 1: Characterization of the corpus

We used the rule-learning system SLIPPER[2] based on confidence-rated boosting [4] which is an extension of .the widely-used RIPPER [2]. SLIPPER, like other supervised learning systems, takes training pairs as input—i.e. a set of feature values and the associated class—and generates a classifier based on an ordered set of if-then-else rules. There are several advantages in using SLIPPER for the dialog outcome prediction problem. First, the generated if-then-else rules are human readable [1] [2], they can be easily integrated into the dialog manager, and can provide useful insight into the data. Second, SLIPPER is asymptotically faster than other competitive rule learning algorithms. This is especially important since the goal is that of generating a large number of classifiers, each trained on a relatively large training set. Third, SLIPPER supports both continuous, symbolic and textual (e.g. bags of words) features. Finally it allows controlling the resulting classification precision and recall by appropriately weighting training samples of different classes.

6. EXPERIMENTAL RESULTS

We created a set of 100 classifiers by running SLIPPER on the training set using 100 different values for the weight of training samples representing NA calls. The range of

[2] http://www.cs.cmu.edu/~wcohen/slipper/

weights was chosen in such a way to have 0% and 100% values of recall and precision at each end. After the classifiers were trained on the training set, development data was used to select the best classifier, which was later tested on the test set. Figure 1 shows the saving in cost per dialog session estimated on the development set for each one of the 100 classifiers, assuming that a *na* prediction of each classifier results in call escalation. The two curves in Figure 1 correspond to 2 different business models. The first (M1), corresponds to a 0.1 per minute cost and 1.0 saving per automated call. The best classifier achieves an average cost reduction of 5.4 cents on the development set as compared to the case where a classifier is not used. The same classifier achieved a cost reduction of 5.7 cents per call on the test set. The second curve (M2) corresponds to a 0.05 cost per minute and a 0.7 saving for each automated call. Savings are not as significant for this model, with the best classifier producing 0.35 cents per call saving on the development set, and 0.4 on the test set. The classifier that performs best is different for each business models.

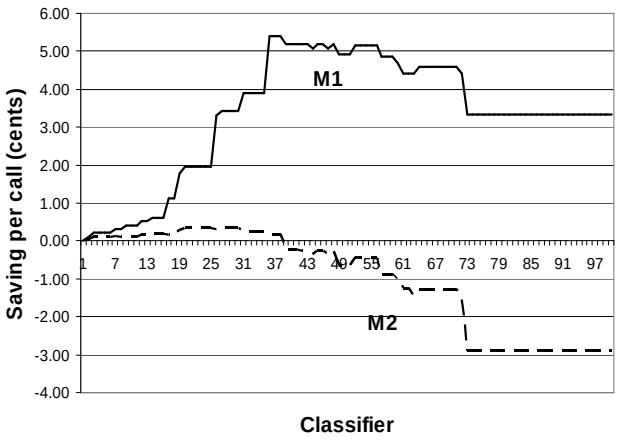

Figure 1: Savings per call (in cents) achieved on the development set by each classifier under two different business models (M1, M2)

7. CONCLUSIONS

In this paper we have discussed the problem of performing decisions in a dialog system in order to optimize a given arbitrary business model. We have introduced a general business model parameterization based on costs per minute and costs per session that depend on both the action performed by the system and the outcome of the session. We considered two possible actions, i.e. the continuation of a call and the escalation to a live agent, while each call can potentially result in a full automation or not. We used more than 61,000 logs of calls from a deployed technical support application in order to estimate and test a number of call outcome classifiers. The precision recall characteristics of each classifier are used to determine, based on a given business model, which is the one that optimizes the average cost of a call. We show that the procedure described in this paper can lead to cost savings. The amount of cost saving achieved, and the choice of the best classifier depend strictly on the given business model.

8. REFERENCES

[1] Catlett, J., "Megainduction: A test flight," Proc. of the 8th International Conference on Machine Learning. Morgan-Kaufmann, 1991.

[2] Cohen, W, "Fast effective rule induction," Proc. of the 12th International Conference on Machine Learning. (ICML95), 1995.

[3] Cohen, W. ,"Learning trees and rules with set-valued features,", Proc. of the 13th Conference of the American Association of Artificial Intelligence, (AAAI96), 1996.

[4] Cohen, W., Singer, Y, "A Simple, Fast, and Effective Rule Learner,"Proc. of 16th National Conference on Artificial Intelligence, 1999.

[5] Langkilde, I., Walker, M., Wright, J., Gorin, A., and Litman, D, "Automatic Prediction of Problematic Human-Computer Dialogues in How May I Help You?" Proc. of ASRU 1999.

[6] 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?" Proc. of *NAACL-2000*, pp. 210-217.

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

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

[9] Paek, T., Horvitz E., "Optimizing Automated Call Routing by Integrating Spoken Dialog Models with Queuing Models." Proc. of HLT-NAAC 2004, pp. 41-48.