

Detecting Problematic Dialogs With Automated Agents

Alexander Schmitt¹, Carolin Hank¹, and Jackson Liscombe²

¹ Institute of Information Technology, University of Ulm, Germany
alexander.schmitt@uni-ulm.de
carolin.hank@uni-ulm.de

² SpeechCycle Inc., New York City, USA
jackson@speechcycle.com

Abstract. We present a supervised machine learning approach for detecting problematic human-computer dialogs between callers and an automated agent in a call center. The proposed model can distinguish problematic from non-problematic calls after only five caller turns with an accuracy of over 90%. Based on a corpus of more than 69,000 dialogs we further employ the classifier's decision to given business models and present the cost savings that can be achieved by deploying classification techniques to Interactive Voice Response systems.

1 Introduction

Increasingly, companies are looking to reduce the costs of customer service and support via automation. With respect to telephone applications, we are witnessing a growing utilization of spoken dialog technology in recent call centers. Such Interactive Voice Response (IVR) systems are being used in various domains: in call routing, where the system serves as front-end to the actual human expert; as information retrieval systems (train schedules, package tracking, etc.); as transactional applications (money transfer, stock trading, hotel booking); or even as problem solvers, such as technical support automated agents, the most recent and complex application of IVRs. Most of those systems are based on touch-tone input, spoken language keywords or a combination of both. Less often, they are able to deal with natural language input. The intentions behind these automation trends are various. Call center automation

- reduces costs: automated services offer a high ROI (return on investment)
- unburdens the operators: routine requests such as answering frequently asked questions or finding the right contact person at the company's switchboard are left to the automated system
- lowers the holding time for customers: the caller is handled directly without waiting time
- provides constant quality: the caller receives a consistent service and uniform information

However, not all calls can be handled successfully by automated systems. Common problems are: the topic the customer talks about is out-of-domain (semantic problem),

the speech recognition does not work due to bad transmission quality, dialect or background noise (speech recognition problem) or the customer is not used to handling this emerging technology due to low media competence (usability problem).

However, effusive and cumbersome automation lowers customer satisfaction. Hence, an elegant trade-off between automated agents and human operators ultimately needs to be found. An approach that has been discussed in recent years is to let the caller start with the automated system, automatically detect problematic conversations, and change the dialog strategy of the system or bring in a human operator if and when a problem occurs.

2 Related Work

Some of the first models to predict problematic dialogs in IVR systems have been proposed by Walker et al. ([Walker et al., 2000b], [Walker et al., 2000a], [Walker et al., 2002]). Walker et al. employ RIPPER, a rule-learning algorithm, to implement a Problematic Dialogue Predictor forecasting the call-outcome of calls in the HMIHY (How May I Help You) call routing system from AT&T. The classifier is able to determine whether a call belongs to the class “problematic” or “not problematic” and this information is used to adapt the dialog strategy of the dialog manager and to repair problems. The accuracy of the classifier after two turns between the user and the system is 79.9% using automatic features; an improvement of 24.4% compared to the baseline, a random guess of the majority class (64.0%). [Levin and Pieraccini, 2006] combined a classifier with various business models to arrive at a decision to escalate a caller depending upon expected cost savings in so doing. The target application is that of a technical support automated agent. [Horvitz and Paek, 2007] consider the influence of an agent queue model on the call outcome and include the availability of human operators in their decision process. Similarly, [Kim, 2007] present a problematic/non-problematic classifier that is trained with 5-grams of utterances from callers that reaches an accuracy of 83% after 5 turns. Escalation is performed when the quality falls below a certain threshold.

Our approach accounts for both points of view considered in the aforementioned studies. First, we introduce a classification approach that can either serve as indicator to the dialog manager to change the dialog strategy or to escalate to a human agent. Second, we integrate the classification outcome with different business models to consider the expected cost savings.

3 Domain

The following experiments are based on data collected from an automated troubleshooting agent that is currently deployed for a large High Speed Internet provider. The automated agent troubleshoots problems related to billing, provisioning, outages, integration and equipment. Callers are encouraged to describe their problems in natural spoken English. The application belongs to a new generation of dialog systems that goes beyond classical FAQ IVRs in that it takes similar steps as a human agent would to solve the issue and fix the problem. Figure 1 shows a dialog that we refer to as problematic.

Table 1. Example dialog of a problematic call from an automated agent. Caller utterances are transcribed using the predicted (ASR) transcriptions.

Agent: (chime) This is our automated Internet troubleshooter. Please hold on while I look up your account.

Agent: Okay. I found it.

Agent: Are you calling because you've lost your Internet connection? Please say yes or no.

Caller: um sure thanks yes

Agent: All right. Lets get that fixed. I'm going to check your modem from my end; one moment please.

Caller: all ack [nomatch]

Agent: Oh! Sorry, I'll just say this again. I just need to check your modem.

Caller: all ack [nomatch]

Agent: Thanks for waiting. I do see your modem on our network, but I need to make sure it can communicate correctly.

Caller: have map [nomatch]

Agent: Oh! Sorry, I'll just say this again. I just wanted to say that I can see your modem but I understand you cannot connect.

Caller: have map [nomatch]

Agent: I'm going to reset it in a moment, but before I do I need to know... Are you calling using Acme Telephone Service, a Regional Phone Service, or an Internet-Based Telephone Service? In other words, is your phone plugged into a piece of computer equipment?

Caller: yes mam yes

Agent: Okay, Acme's name for this service is Acme Telephone. Do you use Acme's telephone service?

Caller: no no

Agent: And who is your Internet Based Telephone Service provider?

Caller: oh that is

Caller hangs up

The sample dialog has been captured during a conversation between a customer and an automated agent unable to troubleshoot the caller’s Internet problems. As we can see, the customer hung up after several turns because the conversation didn’t run smoothly. While the automated agent was attempting to ping the customer’s modem, the customer continued talking to the agent. Since the agent did not expect an answer from the customer at this point in time and the utterance did not match the activated standard grammars, the ASR module output nomatch-events. Evidently, the customer was not comfortable and/or experienced in dealing with IVRs. At the very least, the caller appeared to be uncooperative. Our aim is to detect such problematic dialogs and intervened before it is “too late.” We highlight two different views on classifying problematic dialogs. The first model enables a constant monitoring and re-rating of the call quality wherein the accuracy increases as the dialog progresses. The second view considers call quality monitoring from the point-of-view of a service provider that applies a defined business model. We link the classifier output to a cost function and show the degree of cost reduction a classifier brings.

4 SLIPPER

We employed SLIPPER, a fast and effective rule learner for classification purposes. SLIPPER stands for **S**imple **L**earner with **I**terative **P**runing to **P**roduce **E**rroR **R**eduction [Cohen and Singer, 1999] and is an instance of a supervised learning algorithm. During training it creates easily understandable and compact if-then-else-rules for classification.

Using SLIPPER provides several advantages: it is fast in classification and is thus ideally suited for integration in a real-time system such as a dialog manager. Furthermore, its learned rules are easy to understand and facilitate comprehensible hypotheses that enable good classification accuracies. SLIPPER is based on AdaBoost, a special form of boosting algorithm that combines a number of weak classifiers to arrive at a precise predictor for classification.

For classifying between two classes, SLIPPER creates a strong hypothesis $H(x)$ comprising combination of weak hypotheses $h_t(x)$ that are weighted (α) in an iterative process depending on their significance to the overall outcome. Given a set of features x the classifier outputs “-1” for class 1 and “+1” for class 2 (*sign*-function).

$$H(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right) \quad (1)$$

This final hypothesis $H(x)$ contains all weighted weak rules that make up the classification algorithm.

In the call classification domain those weak rules for detecting a bad call can be, for example:

- if the average recognition accuracy from the ASR is below 60%
- if we observe the words “operator” or “agent” at least two times
- if the user did not respond more than three times to a question by from the IVR

SLIPPER determines such rules automatically. The model is trained with a set of features and an according label.

5 Corpus and Features

The employed corpus (see Table 2) comprises 69,296 calls from a commercially-deployed recent call center recorded between December 3, 2007 and Dec 14, 2007. It consists of log data that has been captured during the conversation between the caller and the automated agent.

Table 2. Labeled corpus: non-automated calls are escalated (“E”), completely automated and partially automated are not escalated (“DE”)

| | E | DE | E+DE |
|-----------------------------------|--------|--------|--------|
| Number of calls | 31,398 | 37,898 | 69,296 |
| Average turns | 4.81 | 18.4 | 12.25 |
| Average duration (min) | 0:52 | 3:46 | 2:27 |
| Average duration of 3 turns (min) | 0:34 | 0:42 | 0:39 |

The corpus consists of three call types: completely automated, i.e. all caller problems were solved by the system; partially automated, i.e. some caller problems were solved by the system but others had to be handled by a human agent; or not automated, i.e. the system could not help, the caller hung up prematurely or asked for an agent. We labeled all automated and partially automated calls as “don’t escalate” (DE) and all not automated calls as “escalate” (E).

For training the model, features from a subset of 36,362 calls were used. The test set consists of 18,099 disjunct calls. We extracted the following features from each dialog turn:

ASR features Raw ASR transcription of the caller’s utterance (utterance). ASR confidence of returned utterance transcription, from 0–100 (confidence). Name of the grammar that returned the parse (triggeredgrammarname). Whether the caller communicated with speech or keypad (inputmode). Whether the speech recognizer returned a valid parse (‘Complete’) or not (‘No Input’ or ‘No Match’) (recognition status). Whether or not the caller began speaking before the prompt completed (barged in).

NLU features The semantic parse of the caller utterance as returned by the activated grammar in the current dialog module (interpretation). Number of system tries to retrieve parsable caller input (loopname). The first time the system prompts the caller, loopname has the value ‘Initial.’ Subsequent re-prompts can either be ‘Timeout’ or ‘Retry.’

Dialog Manager features The number of previous tries to elicit a valid response from the caller (roleindex). Whether the system requested substantive user input or confirmed previous substantive input (rolename). Type of dialog activity (activitytype), e.g. ‘Question’ or ‘Announcement.’ Duration of the dialog turn, in seconds (duration).

6 Classification

In order to determine the call outcome accurately, we decided to create a separate model for each position in the dialog. Model 1 was trained with features from turn 1, model 2 with features from turn 1+2, model 3 from turns 1+2+3, and so on. In so doing, we were sure to use an appropriate model for classification at each point in the dialog, one that had been trained only on the dialog features which had been observed so far. As shown in Table 3, classification accuracy increases as the number of observed turns increases. After five turns we observe a call outcome classification rate of 90% being 64% better than the baseline (54,7%).

Table 3. Accuracy of each classifier at turn n .

| Turns | Accuracy | Test Error Rate | Standard Deviation |
|-------|----------|-----------------|--------------------|
| 2 | 77.73% | 22.27% | 0.75% |
| 3 | 83.74% | 16.26% | 0.75% |
| 4 | 88.04% | 11.96% | 0.69% |
| 5 | 90.50% | 9.5% | 0.63% |
| 6 | 93.09% | 6.91% | 0.56% |
| 7 | 94.39% | 5.61% | 0.53% |
| 8 | 95.15% | 4.85% | 0.51% |
| 9 | 95.51% | 4.49% | 0.52% |
| 10 | 95.96% | 4.04% | 0.53% |
| 11 | 96.21% | 3.79% | 0.55% |
| 12 | 95.13% | 4.87% | 0.65% |

The decision of whether to escalate or not would ideally be determined dynamically at the earliest possible point in a dialog. For now, though, we choose a fixed position in the dialog where we decide to escalate or to continue with the automated system. Given that we observe high accuracy (83%) after only three turns, it seems appropriate to set this as our fixed point in order to minimize the time callers spend in calls that eventually get escalated. Furthermore, the choice of a fixed detection point after three turns is justified by the fact that the average number of turns in escalated calls in our corpus is rather short (see Table 2).

7 Revenue Modeling

So far, we have only considered the overall performance accuracy of the classifier, but have not analyzed the impact of its behavior on business costs and revenue. This section describes the impact of call escalation prediction on different IVR revenue models.

For comparison reasons, the same two business models presented in [Levin and Pieraccini, 2006] are employed, but instead of a dynamic decision point—where the evaluation of the call quality is performed—a fixed decision point is chosen. By “fixed” we mean that we consult the classifier after three dialog turns to estimate the call outcome. If the classifier predicts that the call will be escalated at a later point in time, escalation is performed directly. When the classifier predicts the call will not be escalated, i.e. a revenue for the automation is probable, we don’t escalate and let the call proceed without any further classifier-based escalation.

In these models, each call type has an associated cost per minute rate (*cpm*) and each call can be awarded a value based on some scaled factor of $m * cpm$. In other words, the cost per minute multiplied by the number of minutes the person was on the phone with the system plus a constant revenue for automating the call.

The two models have the following parameters:

- M1: $cpm = \$0.10$ and revenue = \$1.00 for an automated call
- M2: $cpm = \$0.05$ and revenue = \$0.70 for an automated call

We evaluated the costs of each call on the corpus of 18,099 randomly chosen calls. Note that there are two different labeling methods employed. Labeling method *L1* assumes that automated and partially automated calls are “good” and are thus labeled as “don’t escalate.” Non-automated calls get the label “escalate.” Levin and Pieraccini (labeling method *L2*) labeled the calls differently: automated calls get the label “don’t escalate,” partially automated calls and non-automated calls are labeled with “escalate” (their terms are “automated” and “not automated”). According to the latter method, the corpus has a different distribution (see Table 4).

Table 4. Corpus labeled according to Levin and Pieraccini: automated calls are escalated (“E”), partially automated and not automated calls are not escalated (“DE”).

| | E | DE | E+DE |
|-----------------------------------|--------|--------|--------|
| Number of calls | 57,673 | 11,623 | 69,296 |
| Average turns | 11.19 | 17.47 | 12.25 |
| Average duration (min) | 2:14 | 3:33 | 2:27 |
| Average duration of 3 turns (min) | 0:22 | 0:43 | 0:39 |

Which labeling method is employed depends again on the business model. If revenues for partially automated calls are granted, an early automation of those calls is not

recommendable. In the other case, i.e. there is no revenue granted for partially automated calls, an early escalation saves costs. The results for both labeling methods *L1* and *L2*, in combination with the two business models *M1* and *M2*, are shown in Table 5.

Table 5. Cost savings using different business models and labeling methods. L1: automated calls and partially automated calls are not escalated, non-automated calls are escalated; L2: automated calls are escalated, partially automated and non-automated calls are not escalated.

| Labeling method | Model | costs without classifier | costs with classifier | savings |
|-----------------|-------|--------------------------|-----------------------|-----------------|
| L1 | M1 | 9.95 | 9.01 | 0.94 (9.4%) |
| L1 | M2 | 1.32 | 1.01 | 0.31 (23.4%) |
| L2 | M1 | 9.95 | 6.1 | 3.85 (38.7%) |
| L2 | M2 | 1.32 | 3.1 | -1.78 (-138.5%) |

As we can see, the savings strongly depend on the particular business model and the a priori distribution of the call outcomes. While a classifier in combination with a business model can bring significant cost reductions (9.4%, 23.4%, and 38.7% for L1/M1, L1/M2, and L2/M1, respectively), its employment can also cause much lower revenue (-138.5% for L2/M2). [Levin and Pieraccini, 2006] reach cost savings of 5.4 cents per call for model L2/M1 compared to 3.85 cents with our corpus and classifier. This might be attributed to a different distribution of automated and non-automated calls in the corpus. A direct comparison could only be given with the same corpus.

8 Conclusion

We presented a powerful model able to detect problematic calls in an IVR system. After only three turns (in average, after 39 seconds) the classifier is able to detect escalated calls with an accuracy of over 83%. This information can then be employed to change the dialog manager’s behavior, i.e. it can lead to an adaption of the dialog strategy or lead to an escalation of the caller to a human operator. Both measures can bring significant cost reductions as well as higher customer satisfaction and caller experience. First, calls that would usually fail can be recovered by an intelligent, adaptive and problem-aware dialog manager, yielding a higher automation rate. Second, early escalation of customers that the system will probably not be able to help saves time fruitlessly spent with the automated system and, moreover, leads to per-minute savings. Furthermore, the impact of including a classifier in the dialog manager has been justified on the basis of various concrete business models. Although these results are corpus-specific, they can be generalized to IVRs with similar class-distributions, feature sets and classification methods and one could expect similar accuracies and cost savings as shown in this study.

Future work will include the extension of the feature space by the emotional state of the caller based on lexical and acoustic information; a further refinement of the classes,

i.e. a direct prediction of “automated,” “partially automated” and “not automated” instead of “escalate” and “don’t escalate”; and the employment of other supervised learning methods such as neural networks and support vector machines.

References

- [Cohen and Singer, 1999] Cohen, W. W. and Singer, Y. (1999). A simple, fast, and effective rule learner. In In Proceedings of the Sixteenth National Conference of Artificial Intelligence, 1999.
- [Horvitz and Paek, 2007] Horvitz, E. and Paek, T. (2007). Complementary computing: policies for transferring callers from dialog systems to human receptionists. User Modeling and User-Adapted Interaction, 17(1-2):159–182.
- [Kim, 2007] Kim, W. (2007). Online call quality monitoring for automating agent-based call centers. In Proceedings Interspeech 2007 ICSLP, Antwerp, Belgium.
- [Levin and Pieraccini, 2006] Levin, E. and Pieraccini, R. (2006). Value-based optimal decision for dialog systems. In Workshop on Spoken Language Technologies (SLT 06), Aruba.
- [Walker et al., 2000a] Walker, M., Langkilde, I., Wright, J., Gorin, A., and Litman, D. (2000a). Learning to predict problematic situations in a spoken dialogue system: Experiments with how may i help you. In Proceedings of the first conference on North American chapter of the Association for Computational Linguistics.
- [Walker et al., 2002] Walker, M., Langkilde-Geary, I., Wright, H., Wright, J., and Gorin, A. (2002). Automatically training a problematic dialogue predictor for a spoken dialogue system. In Journal of Artificial Intelligence Research, 16: 293-319.
- [Walker et al., 2000b] Walker, M., Wright, J., and Langkilde, I. (2000b). Using natural language processing and discourse features to identify understanding errors in a spoken dialogue system. In Proc. 17th International Conf. on Machine Learning, pages 1111–1118. Morgan Kaufmann, San Francisco, CA.