# A Principled Approach for Rejection Threshold Optimization in Spoken Dialog Systems

*Dan Bohus, Alexander I. Rudnicky*

Computer Science Department
Carnegie Mellon University, Pittsburgh, PA, 15213
dbohus@cs.cmu.edu, air@cs.cmu.edu

## Abstract

A common design pattern in spoken dialog systems is to reject an input when the recognition confidence score falls below a preset rejection threshold. However, this introduces a potentially non-optimal tradeoff between various types of errors such as misunderstandings and false rejections. In this paper, we propose a data-driven method for determining the relative costs of these errors, and then use these costs to optimize state-specific rejection thresholds. We illustrate the use of this approach with data from a spoken dialog system that handles conference room reservations. The results obtained confirm our intuitions about the costs of the errors, and are consistent with anecdotal evidence gathered throughout the use of the system.

## 1. Introduction

One of the major problems in today's spoken dialog systems is their brittleness when faced with understanding errors. The problem stems mostly from the unreliability of the speech recognition process and is exacerbated by the conditions under which spoken dialog systems typically operate: spontaneous speech, large vocabularies, large and diverse user populations, noisy phone lines, etc. In the absence of robust mechanisms for assessing the reliability of decoded results, recognition errors exert a significant negative impact on the quality and success of the interaction [1, 2].

To perform this reliability assessment, spoken dialog systems typically use recognition confidence scores. A common design pattern is to reject an input when its confidence score falls below a preset *rejection threshold.* The use of a rejection threshold introduces a tradeoff between the number of misunderstandings and false rejections. As the threshold increases the system becomes more conservative: it accepts only inputs that have high confidence, and as a result the number of misunderstandings will decrease. This happens however at the cost of increasing the number of false rejections (see Figure 1(a)). Alternatively, we can also think about this tradeoff in terms of correctly and incorrectly transferred concepts. In each utterance, the user tries to convey one or more concepts to the system. If the confidence is below the rejection threshold, the system rejects the utterance and no concept is transferred. On the other hand, if the system accepts the utterance, some concepts will be transferred correctly, while others might be misrecognized. Ideally, we would like to maximize the number of correctly transferred concepts and minimize the number of incorrectly transferred ones. However, as we raise the rejection threshold to lower the number of incorrectly transferred concepts, the number of correctly transferred ones will also decrease (as shown in Figure 1(b)).

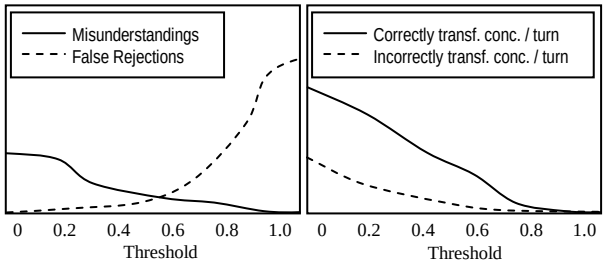


*Figure 1.* Typical rejection tradeoff curves
(a) misunderstandings and false rejections.
(b) correctly and incorrectly transferred concepts per turn

The question we are addressing in this paper is: *given the existence of this tradeoff, what is the optimal value for the rejection threshold?*

The paper is organized as follows: we start by reviewing current practices for setting rejection thresholds in the next section. Then, in Section 3, we describe a data-driven method for deriving the costs of various types of understanding errors, and for fine-tuning the rejection threshold. In Section 4 we briefly discuss a data collection effort which provided the corpus used in these experiments. Section 5 presents the results we obtained and Section 6 concludes the paper.

## 2. Related Work

Oftentimes rejection thresholds are set either at some arbitrary point (e.g. 0.3), or according to various rules-of-thumb and existing recommendations (e.g. find the break-even point). For instance, the Nuance speech recognition engine [3] has a default value of 45 for the rejection threshold (the range is 0-100), and a number of spoken dialog systems which use this recognizer do not change this value. We believe that an a-priori, fixed rejection threshold is likely to be suboptimal for a number of reasons. Confidence annotators included in off-the-shelf speech recognition engines are typically optimized with respect to word-error, while in the context of a spoken dialog system concept-error is more important. Even if the annotator captures semantic confidence, the relative costs of rejections and misunderstandings are likely to vary across different systems, and even across different dialog states within the same system. As a consequence, the tradeoff profile between these types of errors, and hence the optimal rejection threshold is likely to be different in each of these conditions.

Another common approach is to empirically construct the tradeoff curves and optimize the threshold based on rules-of-thumb regarding the costs of misunderstandings and rejections [4, 5, 6]. A frequently used rule is to assume that these costs are equal and therefore minimize the total sum of

errors. Other systems assume that misunderstandings are twice more costly than rejections (as they seem harder to repair), and find the optimal rejection threshold accordingly.

In [7], we proposed a more principled, data-driven technique that allows us to derive actual costs for various types of errors made by a confidence annotator. The costs we obtained from a model build on data from the CMU Communicator system [8] confirmed the intuition that misunderstandings are more costly than rejections. When using the costs to optimize the threshold we obtained however an unexpected result: for a wide range of rejection threshold values, the model predicted that the overall cost would stay roughly constant.

In this work, we start from the intuition that the costs for different types of understanding errors might vary across different dialog states. We expand on the models proposed in [7] by introducing state distinctions. The new models allow us to capture the costs of various types of errors at different points in the dialog, and to find state-specific optimal values for the rejection threshold.

## 3. Method

We now describe the proposed method for deriving costs for errors and finding the optimal rejection threshold. For purposes of clarity we will start by illustrating the approach with an example.

Assume we are interested in optimally balancing the average number of correctly and incorrectly transferred concepts per turn. Let's call these variables *CTC* and *ITC*. Note that these quantities vary with the rejection threshold: as the threshold increases, the average number of incorrectly transferred concepts per turn will decrease, but so will the average number of correctly transferred concepts (see Figure 1(b)). In addition, let's assume we want to optimize for task success – *TS*, modeled as a binary variable. We can determine the relative costs of correctly and incorrectly transferred concepts with respect to task success by fitting a logistic regression model [9] using *CTC* and *ITC* as predictors and *TS* as the dependent variable. Each data-point in the regression model corresponds to an entire dialog session. Let's assume we obtain a good fit for the model, represented by the following regression equation:

$$\text{logit}(TS) = 0.21 + 2.14 \cdot CTC - 4.12 \cdot ITC$$

This equation would tell us that on average an incorrectly transferred concept has a cost (-4.12) about twice as high as the utility (+2.14) a correctly transferred concept. With these costs and the CTC and ITC profiles in hand, we can now easily find the threshold which maximizes the overall utility: $2.14 \cdot CTC - 4.12 \cdot ITC$.

The method can therefore be summarized in 4 steps:

1. identify a set of variables $A, B, \ldots$ involved in the rejection tradeoff (e.g. *CTC* and *ITC*)
2. choose a global dialog performance metric $P$ to optimize for (e.g. *TS* - task success);
3. fit a model $m$ which relates the tradeoff variables to the chosen global dialog performance metric:
$$P \leftarrow m(A,B)$$
4. find the threshold which maximizes the performance:
$$th^* = \arg\max_{th}(P) = \arg\max_{th}(m(A(th), B(th)))$$

In general, the performance metric $P$ (i.e. the dependent variable in the regression model) can be any objective or subjective dialog performance metric. Candidates include: task success (measured either as a binary variable or using the Kappa agreement statistic [2]), task duration, user satisfaction, etc. The type of this variable dictates the type of the regression model. For instance, binary task completion should be modeled using logistic regression, while task duration (expressed as the number of turns) can be modeled as a Poisson variable in a generalized linear model [9].

Similarly, any variables affected by the rejection tradeoff can be used as predictors in the model. In the example discussed above, we used the average number of correctly and incorrectly transferred concepts per turn. In Section 4 we show how, by using the same methodology with different predictor variables, we can develop finer-grained models that allow us to determine state-specific costs, and therefore find state-specific rejection thresholds.

## 4. Data

The cost models discussed in this paper were built using data collected in a user study in which 46 participants interacted with RoomLine [10], a phone-based, mixed-initiative spoken dialog system for making conference room reservations.

The system has information about the availability and characteristics (e.g. size, location, A/V equipment, etc.) of 13 conference rooms in 2 buildings on campus. To make a room reservation, the system finds the list of available rooms that satisfy an initial set of user-specified constraints, and engages in a follow-up negotiation dialog to present this information to the user and identify which room best matches the user's needs. Throughout the data collection experiment, the system used a fixed rejection threshold of 0.3.

Each participant attempted a maximum of 10 scenario-based interactions with the system, within a set time period of 40 minutes. The scenarios were designed to cover all important aspects of the system's functionality and had different degrees of difficulty. To avoid lexical entrainment, they were presented to the users in a graphical manner.

The user speech data was transcribed orthographically by a human annotator. The transcriptions were subsequently checked by a second annotator. Each dialog session was labeled as successful or not. Each user turn was labeled with the number of correctly and incorrectly transferred concepts. If a turn contained at least one incorrectly transferred concept, it was labeled as a misunderstanding.

The corpus contains a total of 449 dialog sessions and 8278 user turns. For an in-depth description of the data collection experiment, as well as other corpus statistics and existing annotations, the interested reader is directed to [11].

## 5. Cost Models and Threshold Optimization

Following the methodology presented in Section 3, we constructed models for estimating the relative costs of correctly and incorrectly transferred concepts. We started from the hypothesis that these costs would vary across dialog states, and hence used as predictor variables the average number of correctly (and incorrectly) transferred concepts per turn for each of the dialog states considered. The model becomes:

$$P \leftarrow m(CTC_1, ITC_1, CTC_2, ITC_2, \ldots)$$

where $CTC_i, ITC_i$ are the average number of correctly and incorrectly transferred concepts per turn in state *i*.

The state-space for the RoomLine system subsumes 71 states. We clustered these states into 3 classes which we suspected would exhibit different characteristics in terms of the rejection tradeoff: *open-request*, in which the system asks an open question (e.g. "How may I help you?"); *request(bool)*, in which the system requests a yes/no answer from the user (e.g. "Do you want a reservation for this room?"); and *request(non-bool)*, in which the system requests a concept with more than 2 possible values from the user (e.g. "Starting at what time do you need the room?"). We believe that finer distinctions could be made if larger amounts of training data were available.

### 5.1. Optimizing for Task Success

In a first model we used task success as the target for optimization. The predictor variables were the average number of correctly and incorrectly transferred concepts per turn, for each of the 3 dialog states discussed above.

Given the binary representation of task success, we fitted a logistic regression model. The model showed a good fit, increasing the average log-likelihood of the data from a baseline of -0.4655 to -0.2952 ($p<10^{-4}$ in a likelihood ratio test). The 10-fold cross-validation average log-likelihood was -0.3059, indicating a robust fit. In a hard metric evaluation, the model is able to predict task success with an error rate of 11.75% (the majority baseline was at 17.62%).

Table 1 shows the resulting regression coefficients together with their corresponding standard errors and p-values (the null hypothesis is that the coefficient is 0). The regression coefficients reflect the impact on task success of correctly and incorrectly acquired concepts. As expected, the coefficients for correctly transferred concepts are positive, while the coefficients for incorrectly transferred concepts these are negative. Note also that the ratio of the costs for correctly and incorrectly transferred concepts ranges widely between the 3 different states, confirming our initial hypothesis.

Next, we used the costs obtained in regression to find optimal thresholds for each of the 3 dialog states, in light of the CTC / ITC tradeoff curves. Figure 2 shows how CTC, ITC and the overall utility function (UTIL) vary with the rejection threshold. For the *open-request* and *request(bool)* states the maximum utility is attained when the rejection threshold is at zero (i.e. always accept). If for the *open-request* state, 0 is clearly the maximizing point, for the *request(bool)* state the utility profile has a large plateau indicating that for a certain range of threshold values (0 – 0.6), the utility stays roughly constant. Finally, in the *request(non-bool)* state the utility function has again a clear maximum which is reached for a rejection threshold value of 0.61.

| Variable | Coefficient | S.E. | p-value |
|---|---|---|---|
| Const | -2.34 | 1.15 | 0.0416 |
| CTC / open-req | 0.55 | 0.29 | 0.0619 |
| ITC / open-req | -0.40 | 0.46 | 0.3801 |
| CTC / req(bool) | 3.31 | 1.00 | 0.0010 |
| ITC / req(bool) | -0.59 | 1.30 | 0.6491 |
| CTC / req(non-bool) | 2.55 | 0.81 | 0.0017 |
| ITC / req(non-bool) | -3.44 | 1.10 | 0.0018 |

*Table 1.* Regression coefficients (i.e. costs)

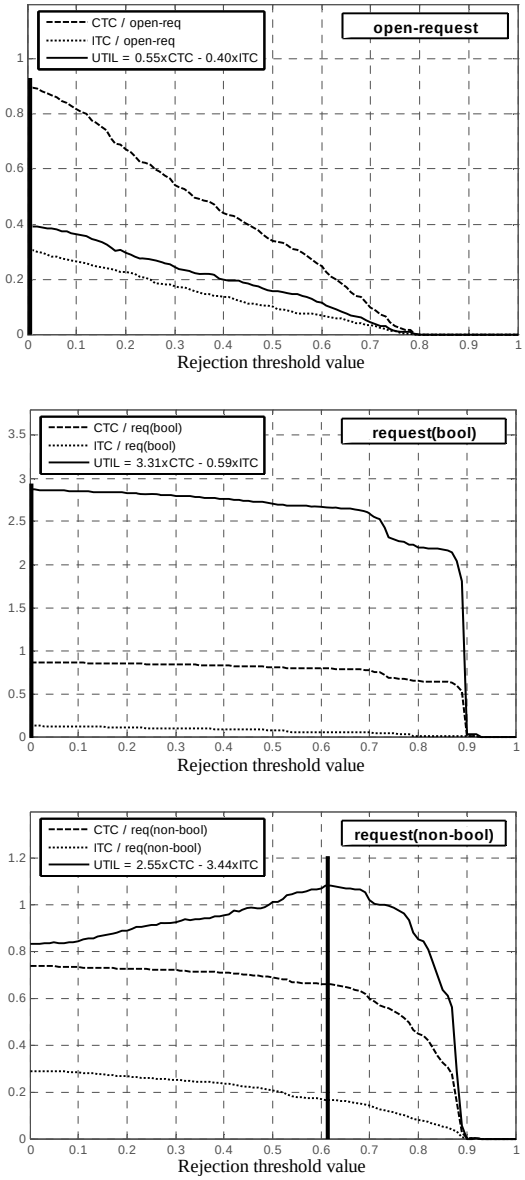


*Figure 2.* Threshold optimization for the 3 dialog states

As the standard errors on the coefficients raised some concerns, we performed an additional robustness check. We split the data into halves, built separate models on each half and compared the results. The variations in the cost coefficients were minor, the resulting utility profiles had very similar shapes for all states across the models, and the maximizing points were at the same locations.

The resulting threshold values are consistent with anecdotal evidence gathered throughout the data collection experiment. As experiment was under way, we noticed that long utterances (which were very frequent after the initial "How may I help you?" prompt) would generally have low confidence scores and therefore would be rejected by the system even though they were correctly recognized. This was confirmed by a later analysis which showed that in the *open-request* state the proportion of falsely rejected utterances was much larger than in the other 2 states (17.4% as opposed to 2.0% and 1.7%). We conjecture that this behavior was caused by a mismatch between the data encountered throughout the

experiment and the data used to train the confidence annotator. The confidence training data consisted of calls to the live version of the system; in this data a number of users were exploring the boundaries of the system, and long utterances generally led to misunderstandings. This pattern was picked up by the confidence annotator, but it was no longer valid throughout the controlled experiment. The model we developed indicates that the cost for incorrect concepts is not very high in this state (we are at the beginning of the dialog). Taking into account the potentially larger number of correctly transferred concepts in this state (due to longer utterances), the model correctly indicates that the system should use a threshold of zero (i.e. accept everything). This result shows how the proposed model can indeed compensate for potential mismatches between a given confidence annotator and the characteristics of the domain in which it is used.

The large plateau in the resulting cost profile for the *request(bool)* state is consistent with the fact that most responses in this state are simple yes/no answers, which generally have high confidence scores. As a result, for low values of the threshold, very few utterances are rejected, and the numbers of correctly and incorrectly transferred concepts, as well as the overall cost stay roughly constant.

Finally, the last question we asked was: what changes should we expect as we move the threshold from its current default position at 0.3 to the new values? The results are shown in Table 2. For the *open-request* state, we expect an average increase of 0.35 correctly transferred concepts per turn, at the expense of a 0.15 increase in incorrectly transferred ones. For the *request(bool)* state the situation stays roughly the same. For the *request(non-bool)* state we expect a small decrease in both the incorrectly and the correctly transferred concepts per turn. Finally, we can also query the constructed logistic regression model for the expected value of the overall task success rate at the new average CTC / ITC values. The model predicts a 4.4% absolute increase in task success rate. This however is an estimate of the expected value of task success, and remains to be validated empirically.

| State | Variable | Current | New | Delta |
|---|---|---|---|---|
| open request | CTC | 0.54 | 0.89 | +0.35 |
| | ITC | 0.16 | 0.31 | +0.15 |
| request bool | CTC | 0.84 | 0.86 | +0.02 |
| | ITC | 0.09 | 0.12 | +0.03 |
| request non-bool | CTC | 0.72 | 0.66 | -0.06 |
| | ITC | 0.25 | 0.17 | -0.08 |
| **Task success rate** | | 82.75% | 87.16% | +4.41% |

*Table 2.* Estimated changes in CTC, ITC and Task Success

### 5.2. Optimizing for Task Duration

In a second model we used task duration (for successful tasks) as the target for optimization. We kept the same predictor variables as for the task success model.

In this case, given that task duration was expressed as the number of turns, we modeled it as a Poisson response variable in a generalized linear model [9]. We normalized for the inherent differences in durations between the 10 different scenarios by introducing the mean duration for the scenario as an offset variable in the regression model.

The model showed a good fit ($p<10^{-4}$). The resulting utility profiles for the *open-request* and *request(bool)* states were very similar to those obtained when optimizing for task success, indicating again an optimal rejection threshold of 0 for these states. For the *request(non-bool)* state, the optimal threshold was again 0.61, but in this case the utility profile had a less pronounced maximum.

## 6. Conclusion

We have described a data-driven approach for determining the costs of various types of understanding errors involved in the rejection tradeoff in a spoken dialog system. The method determines these costs by relating the errors to a global dialog performance metric. Once the costs are available, they can be used to optimize utterance rejection thresholds in a principled manner. We have shown that the proposed method can be used to derive state-specific costs of errors and determine state-specific utterance rejection thresholds.

The results we obtained by using the method with data from a spoken dialog system for conference room reservations confirmed our expectations, and were consistent with other anecdotal evidence gathered through use of the system. In the future, we plan to empirically validate the predicted gains in performance.

Generally, the proposed method can be used to bridge the potential mismatch between an existing confidence annotator and the particular characteristics of the dialog system and domain in which it is used.

## 7. References


[1] Sanders, G., Le, A., and Garofolo, J., *Effects of Word Error Rate in the DARPA Communicator Data During 2000 and 2001*, in Proc. of ICSLP'02, Denver, CO, 2002.

[2] Walker, M., Litman, D., Kamm, C., and Abella, A.– *Evaluating Spoken Dialogue Systems with PARADISE: Two Case Studies*, in Computer Speech and Language, 12-3, 2002.

[3] Nuance Speech Recognition – www.nuance.com

[4] Raymond, C., Bechet, F., De Mori, R., Damnati, G., *On the use of confidence for statistical decision in dialogue strategies*, in Proc. of SIGdial-2004, Boston, USA.

[5] Kawahara, T., and Komatani, K., *Flexible mixed-initiative dialogue management using concept-level confidence measures of speech recognizer output* , in Proc. of COLING, Saarbrucken, Germany, 2000.

[6] Cuayahuitl, H., and Serridge, B., *Out-of-Vocabulary Word Modeling and Rejection for Spanish Keywork Spotting Systems*, in Proc. of the 2nd Mexican International Conference on Artificial Intelligence, 2002.

[7] Bohus, D., and Rudnicky, A., *Modeling the Cost of Misunderstandings in the CMU Communicator Spoken Dialogue System*, in Proc. of ASRU-2001, Italy, 2001

[8] Rudnicky, A., Thayer, E., Constantinides, P., Tchou, C., Stern, R., Lenzo, K., Xu, W., and Oh, A. *Creating natural dialogs in the Carnegie Mellon Communicator System*, in Proc. of Eurospeech, 1999, pp 1531-1534

[9] Myers, R., Montgomery, D., and Vining, G. *Generalized Linear Models: With Applications in Engineering and the Sciences*, Wiley-Interscience, New York, 2002

[10] RoomLine – www.cs.cmu.edu/~dbohus/RoomLine/

[11] Bohus, D., and Rudnicky, A., *An Empirical Analysis of Non-understandings and Recovery Strategies in Spoken Dialog Systems*, to be submitted to SIGDial-2005, Lisbon, Portugal