ONLINE SUPERVISED LEARNING OF NON-UNDERSTANDING RECOVERY POLICIES

Dan Bohus, Brian Langner, Antoine Raux, Alan W Black, Maxine Eskenazi, Alex Rudnicky
{dbohus, blangner, antoine, awb, max, air}@cs.cmu.edu

Carnegie Mellon University,
Pittsburgh, PA, 15217

ABSTRACT

Spoken dialog systems typically use a limited number of non-understanding recovery strategies and simple heuristic policies[1] to engage them (e.g. first ask user to repeat, then give help, then transfer to an operator). We propose a supervised, online method for learning a non-understanding recovery policy over a large set of recovery strategies. The approach consists of two steps: first, we construct runtime estimates for the likelihood of success of each recovery strategy, and then we use these estimates to construct a policy. An experiment with a publicly available spoken dialog system shows that the learned policy produced a 12.5% relative improvement in the non-understanding recovery rate.

Index Terms— non-understandings, recovery strategy, recovery policy, spoken language interface

1. INTRODUCTION

Two types of understanding-errors commonly affect spoken dialog systems: misunderstandings and non-understandings. In a misunderstanding the system constructs an incorrect interpretation of the user's turn, while in a non-understanding the system fails altogether to construct an interpretation. Both types of understanding-errors typically stem from speech recognition problems, and both exert a significant negative impact on the overall quality and success of the interactions [1].

For misunderstandings, detection is the key issue [2]. The number of strategies that can be used to recover is fairly small (e.g. explicit and implicit confirmation) and the tradeoffs between these strategies have been studied and are relatively well understood [3]. In contrast, for non-understandings detection is generally trivial, but the set of strategies that can be used to recover is significantly larger. For instance, the system could ask the user to repeat; it could ask the user to rephrase; it could simply notify the user that a non-understanding happened; it could ignore the non-understanding altogether and try a different dialog plan; it could provide various types of help messages, and so on. The relative tradeoffs between these strategies are less well understood. Moreover, these tradeoffs might be domain- and task-dependent. As a consequence, designing a policy for choosing between different strategies is a non-trivial task. Most spoken dialog systems use a limited number of non-understanding recovery strategies in conjunction with simple heuristic rules for engaging them. A typical example is the so-called "three strikes and you're out" approach [4]: repeat the system question after the first non-understanding, provide help after the second one, and transfer the user to a human operator if a third consecutive non-understanding occurs.

In this paper we address the problem of designing a non-understanding recovery policy over a large set of non-understanding recovery strategies (9 in our case). We present a supervised, online-learning method for this task. The approach, discussed in detail in Section 2, consists of two steps: first, we construct runtime estimates for the likelihood of success of each individual strategy, together with confidence bounds. Then, we use these estimates to choose between the strategies, and construct a recovery policy.

We implemented and evaluated the proposed approach in a telephone-based spoken dialog system that provides bus route and schedule information in the greater Pittsburgh area. The system learned a non-understanding recovery policy online, throughout a period of 5 weeks. Our experiments indicate that the learned policy led to a 12.5% relative improvement in the non-understanding recovery rate. Furthermore, the improvement was attained quickly, in only 10 days from the beginning of the learning period.

2. METHOD

The starting point for the proposed approach is the intuition that certain non-understanding recovery strategies are more likely to succeed under certain circumstances. For instance, if the source of the non-understanding is an out-of-vocabulary word, asking the user to repeat is less likely to help than asking the user to rephrase. However, if the non-understanding is caused by a transient noise, asking the user to repeat might be a more appropriate course of action. If we could estimate the likelihood of success for each strategy, an optimal policy would be easy to construct: simply pick the strategy with the highest likelihood of success. The method we are proposing works therefore in two steps: first, we use a supervised learning approach to construct predictors for the likelihood of success of each individual recovery strategy. Then, we use these predictors at runtime to select which strategy to engage.

2.1. Learning Predictors for Strategy Success

To predict the likelihood of success for each recovery strategy, we use logistic regression models. One separate model is constructed for each strategy. Its goal is to predict whether or not the strategy has successfully recovered, i.e. put the dialog back on track following a given non-understanding. We consider that a strategy has successfully recovered if the following user turn is correctly understood by the system. For training and evaluation purposes, this information is manually annotated. In fact, since the system already

[1] By *recovery strategy* we denote a single, one-turn action the system takes to recover from an error (e.g. asking the user to repeat, asking the user to rephrase, etc). By *recovery policy* we denote a method for choosing between different recovery strategies at runtime.

knows when non-understandings occur, a semi-automatic approach can be used to create the recovery labels: all the non-understandings followed by another non-understanding are automatically labeled as not-recovered; the remaining non-understandings are inspected and labeled by a human annotator. The features (i.e. the dependent variables in the regression model) capture various aspects of the last non-understanding (e.g. the number of words, acoustic, language modeling and goodness-of-parse scores, etc.), as well as information about the current dialog state and about the history of the dialog so far (e.g. number of previous non-understandings, previous recovery actions taken.)

Logistic regression models [5] present a number of advantages over other machine learning techniques in this task. In contrast with other discriminative approaches, logistic regression generally produces well-calibrated class posterior probability scores [6, 7]. In other words, the model predictions accurately reflect the probability of success (e.g. a strategy will be successful in x% of the cases when the model predicts that the likelihood of success is x). This is an important property since we plan to use the model scores as probability estimates. Secondly, logistic regression models can automatically provide the confidence intervals for these predictions, a prerequisite for the strategy selection method described in the next subsection. Furthermore, logistic regression is sample efficient. This is another desirable property since we plan to learn one separate model for each strategy and therefore a relatively small number of data-points will be available for training each predictor. Last but not least, logistic regression models can be constructed in a stepwise manner. This allows us to consider a very large number of features; the relevant features will be automatically included in the model.

2.2. Highest Upper Bound Strategy Selection

Once we can predict the likelihood of success for each strategy, we are left with choosing the method for selecting between the strategies. Ideally, we should choose the strategy with the highest likelihood of success. However, we are interested in developing an approach in which the system learns a recovery policy on-line, through experimentation. As a result, we are faced with an exploration-exploitation tradeoff. We need to strike a balance between using strategies we know to be successful (exploitation) and gathering more training data for the strategies about which we are still unsure (exploration).

We address this tradeoff by always selecting the strategy that has the highest upper bound on the estimated probability of success. This selection method, also known as the interval estimation, was initially proposed by Kaelbling in [8], and has been shown empirically to perform very well in various exploration-exploitation tasks.

Intuitively, by selecting the strategy with the highest upper bound, we either choose a strategy that has a high likelihood of success (Figure 1.a), or we choose a strategy that has a wide confidence interval (Figure 1.b1). In the first case, we are exploiting, in the second we are exploring. After a strategy is engaged, we obtain a new data-point for training the predictor for that strategy. As more and more data becomes available for a strategy, the corresponding confidence interval will shrink. As a result, another strategy will have the highest upper bound, and the system will switch to exploring that strategy (Figure 1.b1 → 1.b2).

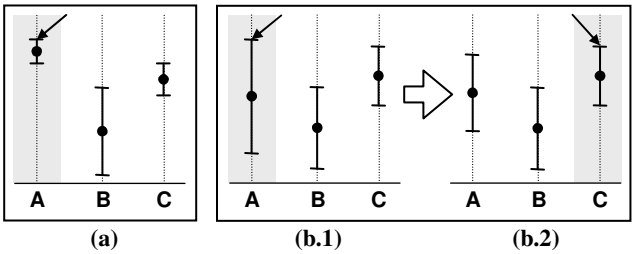

Figure 1. Highest-upper-bound selection between 3 fictitious strategies (A, B and C)

3. EXPERIMENTAL SETUP

3.1. System

The experiments described below were performed in the context of Let's Go! [9], a telephone-based spoken dialog system which provides access to bus route and schedule information for Pittsburgh Port Authority busses. Since March 2005, the system has been available to the general public via the Port Authority customer service line during non-business hours (i.e. 7pm-7am on weekdays and 6pm-8am on weekends and holidays). Throughout this time, the system has serviced over 23,000 calls. On average, the system receives about 50 calls per night. Given this density of calls, this system provided an excellent basis for our experiment.

3.2 Strategies

During the first year of operation, the system used 5 non-understanding recovery strategies in conjunction with a simple heuristic policy (for more details, see [9]). Prior to starting the policy learning experiment, we redesigned the set of non-understanding recovery strategies. The final set is shown in Table 1.

Additionally, we also designed a set of rules to restrict the circumstances under which each strategy can be used. Here are some examples: don't ask the user to repeat more than twice in a row; don't ask the user to give shorter answers unless the number of words is greater than 5; don't ask the user to rephrase if the number of words is below 3; don't give up unless the number of turns is above 30 and the ratio of non-understandings so far above 80%. These rules encapsulate prior expert knowledge. They are used to ensure that the system never uses an unreasonable policy as well as to constrain the search space for the policy learning algorithm. In effect, they implement a heuristic strategy selection policy, which, instead of selecting one strategy, selects a set of valid strategies at each non-understanding. Given these heuristic constraints, the average number of strategies available to the system was 4.2, with a minimum of 1 and a maximum of 9.

3.3. Features

We identified a large set of features (294) which carry potentially relevant information for predicting the likelihood of successful recovery. Due to space constraints, we only provide a brief outline of the feature set (the full set is available online [10]):

- **features describing the current non-understanding**: speech recognition features (e.g. acoustic and language model scores, speech rate, signal and noise levels, clipping information), lexical features (e.g. number of words, presence and absence

Name	Description
HLP	Give a help message indicates how users might answer the current system question
HLP_R	Same as above, but also tell users that they can say start-over to restart the dialog
RP	Repeat the previous system prompt
AREP	Ask user to repeat what they said
ARPH	Ask user to rephrase what they said
MOVE	Ignore the current non-understanding and back-off to an alternative dialog plan (this strategy task-specific and was only available when the system requested the departure stop; when engaged, the strategy would first request the departure neighborhood, then a departure stop in that neighborhood)
ASA	Ask user for a shorter answer
SLL	Ask user to speak less loud
IT	Give general interaction tips to the user
ASO	Ask user if he/she would like to start over
GUP	Give up dialogue and hang up

Table 1. 11 non-understanding recovery strategies in the Let's Go! Public Bus Information system

of confirmation markers), grammar features (e.g. various goodness-of-parse scores, number of grammar slots), timing information (e.g. barge-ins and timeouts), the non-understanding type (e.g. no-parse vs. rejection);

- **features describing the current non-understanding segment:** the length of the current non-understanding segment; information about which recovery strategies were already taken in the current non-understanding segment, etc.;
- **features describing the current dialog state and the dialog history:** we encoded the 22 dialog states with a set of 22 binary variables; additionally, we computed history features such as the ratio of non-understandings, and running averages of confidence scores, goodness-of-parse scores, acoustic and language model scores, etc.

3.4. Learning

We started the experiment on March 11th 2006. First, we constructed a baseline by running the system for 2 weeks with the new set of 11 recovery strategies. During this time, the system randomly chose a recovery strategy whenever a non-understanding happened, while obeying the set of constraints described in Section 3.2. In effect, the system was using a heuristic, stochastic non-understanding recovery policy.

After 2 weeks, on March 26th, 2006 we started the online policy learning algorithm. During this learning phase, we excluded the last two strategies shown in Table 1, due to their incompatibility with our local definition of successful recovery. The first excluded strategy (ASO) notifies the user that a non-understanding occurred and asks if the user would like to start over. This generally elicits a yes/no type answer from the user. Although this answer might be correctly understood by the system in most cases, a correct understanding does not really represent a successful recovery from the previous non-understanding. Similarly, when the give up (GUP) strategy is engaged, the system apologizes, asks the user to call during normal business hours, and hangs up. No recovery is therefore possible in this case.

Throughout the learning phase, we retrained the models on a daily basis. Each morning the data from the previous night was semi-automatically labeled with non-understanding recovery information; the models for predicting the likelihood of success for each strategy were retrained and introduced in the system for the following night. We allowed the system to learn in this manner for 5 weeks, until May 5th, when we stopped the experiment.

4. RESULTS

To evaluate the proposed approach we computed the average non-understanding recovery rate (ANRR) throughout the baseline and learning periods. The presence of two extra strategies (ASO and GUP) during the baseline period could confound the results. To make the comparison fair, we excluded all the sessions from the baseline period in which these strategies were engaged (27 out of 524). In fact, this approach artificially inflates the baseline performance of the system. The reason is that, because of the heuristic constraints, the ASO and GUP strategies were only available during sessions with large numbers of non-understandings. ASO was available when the non-understanding ratio was >50%, GUP when this ratio was >80%. By eliminating any session that contained one of these strategies we are therefore also eliminating a significant number of unrelated non-understandings, which were not recovered. This artificially inflates the baseline performance. Nevertheless, we can still detect a learning effect.

The resulting daily and weekly averages for the non-understanding recovery rate (ANRR) throughout the baseline and learning periods are illustrated in Figure 2. Despite fairly wide daily fluctuations, a comparison of average recovery performance between the last and first two weeks reveals a statistically significant improvement from 33.6% to 37.8% (a 12.5% relative improvement, p=0.0309).

To better understand the learning process we fitted a learning curve to the data, described by the following equation:

$$ANRR \leftarrow A + B \cdot \frac{e^{C \cdot n + D}}{1 + e^{C \cdot n + D}}$$

The curve describes a temporal learning process (n is the number of days elapsed) that starts from the baseline $ANNR=A$ and asymptotes at $ANRR=A+B$. The learning rate is captured by the C parameter. The D parameter allows for a shift in the starting point for the learning. The resulting fit is also illustrated in Figure 2. The coefficients were A=0.3385, B=0.0470, C=0.5566, D=-11.44. The

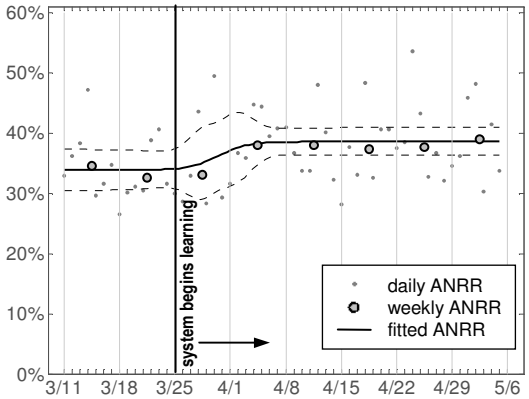

Figure 2. Improvement in average non-understanding recovery rate (ANRR)

fitting process recovered our baseline (A=33.85%) and indicates