AUTOMATIC PREDICTION OF PROBLEMATIC HUMAN-COMPUTER DIALOGUES IN ‘HOW MAY I HELP YOU?’

Irene Langkilde, Marilyn Walker, Jerry Wright, Allen Gorin, Diane Litman

AT&T Labs—Research
180 Park Avenue
Florham Park, NJ 07932-0971 USA

ABSTRACT

We apply machine learning methods to automatically identify and predict problematic human-computer dialogues in a corpus collected during a wizard override trial of the *How May I Help You* task. The results we report show that it is possible to automatically learn a problematic dialogue classifier, and that this classifier can predict problematic dialogues better than the baseline, even using only automatically acquired features from the first exchange in the dialogue. We find that a classifier restricted to task-independent automatic features performs almost as well. We also find that the automatic features can identify problematic dialogues nearly as well as the full set of features, including hand-labelled ones.

1. INTRODUCTION

How May I Help You (HMIHY) is a spoken dialogue system for customer care in a telecommunications environment developed at AT&T Labs Research [4]. AT&T has carried out several experimental trials of HMIHY on live customer traffic[6, 3]. The callers were real AT&T customers who were unaware that they were being tested with an experimental system. Because the system was experimental, each call was monitored by a human customer care agent serving as a wizard who could override the system if necessary. The system behaviors were automatically recorded in a log file, and later the dialogues were transcribed by humans and labelled with a semantic category representing the task that the caller was asking HMIHY to perform, on a per utterance basis. A sample dialogue from this experiment is shown in Figure 1.[1]

Analysis of the resulting corpus showed that, in some cases, the HMIHY system failed to complete the user’s task. In order to actually deploy HMIHY in the AT&T network, we wish to reduce or eliminate these situations. One way to do so would be to modify the system’s dialogue manager or spoken language understanding components to improve its ability to achieve TaskSuccess dialogues. An alternative solution would be to automatically identify those dialogues that are likely to be problematic and transfer them to a human customer care agent before they fail.

The problematic dialogues that we are interested in fall into 3 cases: (1) The human customer care agent intervened and took over the call (Wizard) as in Figure 2; (2) The user hung up (UserHangUp) as in Figure 3; (3) The system believed that it succeeded but it actually failed to complete the task successfully (TaskFailure) as in Figure 4. In our corpus of 4774 successful or problematic dialogues (from the corpus referred to as HM2 in [6]), approximately 64% of the dialogues were TaskSuccess dialogues. Ninety-seven percent consisted of 6 or fewer human-computer exchanges.

S1: AT&T How may I help you?
U1: I need to [uh] put a call on my calling card please
S2: May I have your card number, please?
U2: 7 5 5 2 5 0 0 4 4 0 8 0 8 3
S3: What number would you like to call?
U3: 8 1 4 6 7 3 4 8 7 2 (*misrecognized*)
S4: May I have that number again?
U4: 8 1 4 6 7 3 4 8 7 2
S5: Thank you.

Figure 1: Sample TaskSuccess Dialogue

S1: AT&T How may I help you?
U1: (*silence*)
S2: Sorry. Please briefly tell me how I may help you?
U2: I’m trying to call 1 8 hundred call A T T.
S3: Do you want to place a call?
U3: Through my calling card.
S4: May I have your card number, please?
U4: It’s 8 7 3 0 6 4 1 1 3 and then my pin number is 3 3 1 4. (*misrecognized*)
S5: Sorry. Please enter or say your card number again.
WIZARD STEPS IN

Figure 2: Sample Wizard Dialogue

S1: AT&T How may I help you?
U1: I need credit please.
S2: Okay. What is the reason for the credit?
U2: Miss, uh, Miss, different area code than I needed.
S3: Sorry. I need to know whether you need credit for a wrong number, bad connection or a call that was cut off.
U3: It is a wrong number. (*impatient humming*)
S4: What was the number that you dialed?
USER HANGS UP

Figure 3: Sample HangUp Dialogue

S1: AT&T How may I help you?
U1: (*silence*)
S2: Sorry. Please briefly tell me how I may help you?
U2: I wanta find out how much my bill’s worth.
S3: What phone number would you like the call billed to?
U3: *Um* the access number.
S4: Sorry. Could I have that phone number again?
U4: 1 8 hundred 4 4 5 6 0 6 3.
S5: What number would you like to call?
U5: 800 445 6063 (*touchtone input*)
S6: Thank you.

Figure 4: Sample TaskFailure Dialogue

The problem this paper addresses is whether we can learn to automatically predict that a dialogue will be problematic on the basis of information the system has: (1) early on in the dialogue; and (2) in real time. We can then use this prediction in HMIHY to identify situations where the system should transfer the caller to a customer care agent or modify its dialogue strategy. Our work builds on earlier research on learning to predict poor speech recognizer performance [5]. We show that we can learn to predict these problematic dialogues with an accuracy ranging from 72% to 87% depending on how much of the dialogue the system has seen so far, by training an automatic

[1] The phone numbers, card numbers, and pin numbers in the sample dialogues have been randomized, and are not real numbers.

classifer for predicting problematic dialogues from features that can be automatically extracted from the HMIHY HM2 corpus.

2. EXPERIMENTAL METHOD

Our experiments on the HMIHY corpus apply the machine learning program RIPPER [2] to automatically induce a "problematic dialogue" classification model. RIPPER (like other learning programs, e.g., C5.0 and CART) takes as input the names of a set of *classes* to be learned, the names and ranges of values of a fixed set of *features*, *training data* specifying the class and feature values for each example in a training set. Its output is a *classification model* for predicting the class of future examples. In RIPPER, the classification model is learned using greedy search guided by an information gain metric, and is expressed as an ordered set of if-then rules.

To apply RIPPER, the dialogues in the corpus must be encoded in terms of a set of classes (the output classification) and a set of input features that will be used as predictors for the classes. We start with the classes described above: TASKSUCCESS, USERHANGUP, WIZARD, TASKFAILURE. For our goal of developing algorithms for identifying problematic dialogues, we treat USERHANGUP, WIZARD and TASKFAILURE as equivalently problematic. Thus we will train the classifier to distinguish between two classes: TASKSUCCESS and PROBLEMATIC.

Next, we encode each dialogue in terms of a set of 240 features (58 features for each of the first four exchanges, and 8 features covering the whole dialogue) that are either automatically logged by one of the system modules, hand-labelled by humans, or derived from raw features. We will use the hand-labelled features to produce a TOPLINE, an estimation of how well a classifier could do that had access to perfect information. Because we are interested in the problem of *predicting* failures before they happen, we encode features for the first four exchanges of a dialogue. We then train classifiers that only have access to input features from exchange 1, or only the features from exchange 1 and exchange 2, in order to test how well we can *predict* problematic dialogues based on features that we have available early in the dialogue. We compare results for *predicting* problematic dialogues, with results for *identifying* problematic dialogues, when the classifier has access to features representing the whole dialogue. The system modules that contributed to the feature set were the acoustic processer/automatic speech recognizer [6], the natural language understanding module [4], and the dialogue manager [1].

The acoustic/automatic speech recognizer features were the output of the speech recognizer (*recog*), the number of words in the recognizer output (*recog-numwords*), the duration in seconds of the input to the recognizer (*asr-duration*), a flag for touchtone input (*dtmf-flag*), the input modality expected by the recognizer (*rg-modality*) (one of: none, speech, tt, speech+tt, tt-card, speech+tt-card, tt-date, speech+tt-date, or none-final-prompt), the grammar used by the recognizer (*rg-grammar*) [6], or the actual modality of the user utterance (*user-modality*) (one of: none, speech, tt, speech+tt, non-speech).

The features obtained or derived from the natural language understanding module are confidence scores for each of 15 possible calltypes [4], an intra-utterance measure of the inconsistency between services that the user appears to be requesting (*inconsistency*), a measure of the coverage of the utterance by salient grammar fragments (*salience-coverage*), a measure of the shift in context between utterances (*context-shift*), the calltype task with the highest confidence score (*top-calltype*), the calltype task with the second highest confidence score (*nexttop-calltype*), the value of the highest confidence score (*top-confidence*), the difference in values between the top and next-to-top confidence scores (*diff-confidence*).

- **Acoustic/ASR Features**
  - recog, recog-numwords, ASR-duration, dtmf-flag, rg-modality, rg-grammar, user-modality
- **NLU Features**
  - a confidence measure for all possible calltypes
  - salience-coverage, inconsistency, context-shift, top-calltype, nexttop-calltype, top-confidence, diff-confidence
- **Dialogue Manager Features**
  - utterance by utterance: sys-label, utt-id, rev-utt-id, prompt, reprompt?, confirmation?, subdial?
  - running tallies: num-reprompts, num-confirms, num-subdials, reprompt%, confirmation%, subdialogue%
  - whole dialogue: num-utts, num-reprompts, percent-reprompts, num-confirms, percent-confirms, num-subdials, percent-subdials, dial-duration.
- **Hand-Labelled Features**
  - tscript, human-label, age, gender, clean-tscript, cltscript-numwords,
  - match-numwords, ct-rest-numwords, rg-rest-numwords, %ct-match, %rg-match, match, ct-unmatched, rg-unmatched

Figure 5: Features for spoken dialogues.

The features from the dialogue manager can be categorized as per-utterance features, running tallies, and whole-dialogue features. The per-utterance features are the task-type label (*sys-label*), utterance id within the dialogue (implicit in the encoding), reverse-order utterance id (*rev-utt-id*), the name of the prompt played before the user utterance (*prompt*), and whether that prompt was a reprompt (*reprompt*), a confirmation (*confirm*), or a subdialogue prompt (a superset of the reprompts and confirmation prompts, plus a couple other prompts) (*subdial*). Running tallies were kept for the number of reprompts (*num-reprompts*), number of confirmation prompts (*num- confirms*), and number of subdialogue prompts (*num-subdials*), as well as for the percent of reprompts (*percent-reprompts*), percent of confirmation prompts (*percent-confirms*), and the percent of subdialogue prompts (*percent-subdials*). Similar to the running tallies were the features computed across the whole dialogue: (*num-utts*), (*num- reprompts*), (*percent-reprompts*), (*num- confirms*), (*percent-confirms*), (*num- subdials*), and (*percent-subdials*). In addition, we had the duration of the entire dialogue in seconds (*dial-duration*). The whole dialogue and the reprompt and confirm features were motivated by features used in earlier work [5]. Litman *etal.* proposed that features that were normalized by dialogue length, such as the *percent* features, are more likely to produce generalized predictors.

The features obtained via hand-labelling were human transcripts of each user utterance (*tscript*), a set of semantic labels (closely related to the system task-type labels) (*human-label*), age (*age*) and gender (*gender*) of the user, a cleaned transcript with non-word noise information removed (*clean-tscript*), the number of words in the cleaned transcript (*cltscript-numwords*), the number of words that occurred in both cleaned transcript and recognizer output, (*match-numwords*), the number of words that occurred only in the cleaned transcript (*ct-rest-numwords*), the number of words that occurred only in recognizer output (*rg-rest-numwords*), the percent of the cleaned transcript that matched recognizer output (*%ct-matched*), the percent of recognizer output that matched cleaned transcript (*%rg-matched*), the set of words that occurred in both the cleaned transcript and recognizer output (*match*), the set of words that occurred only in the cleaned transcript (*ct-unmatched*), and the set of words that occurred only in recognizer output (*rg-unmatched*).

The entire feature set is summarized in Figure 5, and an example

e2-recog:	can charge no one eight hundred call A T T		
e2-rg-modality:	speech-plus-tt	*e2-recog-numwords*:	10
e2-user-modality:	speech	*e2-dtmf-flag*:	0
e2-rg-grammar:	Reprompt-gram	*e2-asr-duration*:	6.68
e2-top-calltype:	dial-for-me	*e2-top-confidence*:	.81
e2-nexttop-calltype:	none	*e2-diff-confidence*:	.81
e2-salience-coverage:	0.000	*e2-calltype1*:	0
e2-inconsistency:	0.000	*e2-calltype2*:	0
e2-context-shift:	0.000	*e2-calltype3*:	0
e2-prompt:	top-reject-rep	*e2-calltype4*:	0
e2-reprompt:	reprompt	*e2-calltype5*:	0
e2-num-reprompts:	1	*e2-calltype6*:	.81
e2-percent-reprompts:	0.5	*e2-calltype7*:	0
e2-confirm:	not-confirm	*e2-calltype8*:	0
e2-num-confirms:	0	*e2-calltype9*:	0
e2-percent-confirms:	0	*e2-calltype10*:	0
e2-subdial:	subdial	*e2-calltype11*:	0
e2-num-subdials:	1	*e2-calltype12*:	0
e2-percent-subdials:	0.5	*e2-calltype13*:	0
e2-rev-utt-id:	-3	*e2-calltype14*:	0
e2-sys-label:	DIAL-FOR-ME	*e2-calltype15*:	0
e2-human-label:	no-info digitstr	*e2-no-info*:	1
e2-tscript:	*epr* I'm trying to call *uh* 1 8 hundred call A T T *nspn*		
e2-clean-tscript:	I'm trying to call 1 8 hundred call A T T		
e2-match:	1 8 hundred call A T T		
e2-ct-unmatched:	I'm trying to call	*e2-%ct-matched*:	.636
e2-rg-unmatched:	can charge no	*e2-%rg-matched*:	.7
match-numwords:	6	*e2-ct-rest-numwords*:	4
e2-cltscript-numwords:	11	*e2-rg-rest-numwords*:	3
e2-age:	adult	*e2-gender*:	male
	. . .		
num-subdials:	3	*num-utts*:	5
percent-subdials:	0.6	*dial-duration*:	51.32

Figure 6: Feature encoding for Wizard dialogue.

encoding of the second exchange of the WIZARD dialogue in Figure 2 is shown in Figure 6. In the example encoding, the prefix "*e2-*" designates the second exchange.

Because we are interested in predicting problematic dialogues automatically during runtime, we are particularly interested in the features that do not require hand-labelling, that would be available to the system. Specifically, these AUTOMATIC features would include all those in Figure 5 except the Hand-Labelled features and the reverse-utterance-id (among the Dialog Manager features).

We are also interested in generalizing our problematic dialogue predictor to other systems. Thus we also discuss below how well we can predict problematic dialogues using only features that are both automatically acquirable during runtime and independent of the HMIHY task. The subset of features from Figure 5 that fit this qualification are in Figure 7. We refer to them as the AUTO & TASK-INDEP feature set.

The output of each RIPPER experiment is a classification model learned from the training data. To evaluate these results, the error rates of the learned classification models are estimated using the resampling method of *cross-validation*. In 5-fold cross-validation, the total set of examples is randomly divided into 5 disjoint test sets, and 5 runs of the learning program are performed. Thus, each run uses the examples not in the test set for training and the remaining examples for testing. An estimated error rate is obtained by averaging the error rate on the testing portion of the data from each of the 5 runs.

- **Acoustic/ASR Features**
  - recog, recog-numwords, ASR-duration, dtmf-flag, rg-modality
- **NLU Features**
  - salience-coverage, inconsistency, context-shift, top-confidence, diff-confidence
- **Dialogue Manager Features**
  - utterance by utterance: utt-id, reprompt?, confirmation?, subdial?
  - running tallies: num-reprompts, num-confirms, num-subdials, reprompt%, confirmation%, subdialogue%

Figure 7: Automatic task independent features available at runtime.

Since we are interested in integrating the hypotheses learned by Ripper into the HMIHY system, we are also interested in the precision and recall performance of specific hypotheses. Because hypotheses from different cross-validation experiments cannot readily be combined together, we apply the hypothesis learned on one randomly selected training set (80% of the data) to that set's respective test data. This means that the precision and recall results that we report below are somewhat less reliable than the error rates we report from cross-validation.

3. RESULTS

Features Used		Accuracy	(SE)
BASELINE (majority class)		64.0 %	
EXCHANGE 1	AUTOMATIC	72.3 %	1.04 %
	AUTO, TASK-INDEP	71.6 %	1.05 %
EXCHANGES 1&2	AUTOMATIC	79.8 %	0.74 %
	AUTO, TASK-INDEP	78.4 %	0.26 %
FULL DIALOGUE	AUTOMATIC	87.0 %	0.72 %
	AUTO, TASK-INDEP	86.7 %	0.82 %
TOPLINE	ALL	88.5 %	0.81 %

Table 1: Results for predicting and identifying problematic dialogues (SE = Standard Error)

Our results are summarized in Table 1. The baseline shown on the first line of Table 3 represents the accuracy in prediction that could be achieved by always guessing the majority class, which is TASKSUCCESS in this experiment. The first EXCHANGE 1 row shows the prediction accuracy that can be achieved using only the AUTOMATIC features (as described earlier) from the first exchange. Our machine-learned classifier can predict problematic dialogues 8% better than the baseline after having seen only the first user utterance. Using only task-independent automatic features (Figure 7) the EXCHANGE 1 classifier can still do nearly as well.

The EXCHANGE 1&2 rows of Table 1 show the results for using features from the first *two* exchanges in the dialogue to predict the end status of the dialogue.[2] The additional exchange gives roughly an additional 7than the TOPLINE shown on the last row, in which problematic dialogues are simply identified post-hoc using the full set of features, including hand-labelled ones, for all exchanges in the dialogue.

The FULL DIALOGUE rows in Table 1 shows the ability of the classifier to *identify* problematic dialogues, rather than predict them, using either automatic or task-independent automatic features for the whole dialogue. Interestingly enough, automatic identification has an error rate less than 2% different from the TOPLINE result in the last row which include hand-labelled features. Just as notable is that in

[2] Since 23% of the dialogues consisted of only two exchanges, we exclude the second exchange features for those dialogues where the second exchange consists only of the system playing a closing prompt. We also excluded any features that indicated to the classifier that the second exchange was the last exchange in the dialogue.

all our experiments, the task-independent automatic features perform within 1.5% error of the automatic features. Below we show examples of rules learned for two of these feature sets.

Exchange 1, Automatic Features:

if (e1-top-confidence ≤ .924) ∧ (e1-dtmf-flag = '1') **then** *problematic*,

if (e1-diff-confidence ≤ .916) ∧ (e1-asr-duration ≥ 6.92) **then** *problematic*,

default is *tasksuccess*.

The performance of this hypothesis is summarized in Table 2. These results show that given the first exchange, this hypothesis predicts that 22% of the dialogues will be problematic, while 36% of them actually will be. Of the dialogues that actually will be problematic, it can predict 41% of them. Once it predicts that a dialogue will be problematic, it is correct 69% of the time.

Class	Occurred	Predicted	Recall	Precision
Success	64.1 %	78.3 %	89.44 %	73.14 %
Problematic	35.9 %	21.7 %	41.47 %	68.78 %

Table 2: Precision and Recall with Exchange 1 Automatic Features

Exchanges 1&2, Automatic Features:

if (e2-recog-numwords ≤ 0) **then** *problematic*.

if (e1-top-confidence ≤ .924) ∧ (e2-asr-duration ≥ 5.36) ∧ (e1-asr-duration ≥ 8.72) ∧ (e2-recog-numwords ≤ 6) **then** *problematic*.

if (e2-salience-coverage ≤ .889) ∧ (e1-asr-duration ≥ 13.64) ∧ (e2-asr-duration ≥ 5.16) **then** *problematic*.

if (e1-top-confidence ≤ .916) ∧ (e1-user-modality = nothing) ∧ (e2-user-modality = non-speech) ∧ (e1-salience-coverage ≤ .75) **then** *problematic*.

if (e1-salience-coverage ≤ .889) ∧ (e2-asr-duration ≥ 7.48) ∧ (e2-recog contains "I") **then** *problematic*.

if (e1-dial-for-me ≤ .9) ∧ (e1-recog-numwords ≥ 2) (e1-user-modality = nothing) ∧ (e1-salience-coverage ≥ .154) **then** *problematic*.

if (e1-salience-coverage ≤ .889) ∧ (e1-recog contains "help") **then** *problematic*.

default is *tasksuccess*.

Class	Occurred	Predicted	Recall	Precision
Success	64.1 %	73.9 %	91.42 %	79.26 %
Problematic	35.9 %	26.1 %	57.35 %	78.95 %

Table 3: Precision and Recall with Exchange 1&2 Automatic Features

The performance of this hypothesis is summarized in Table 3. These results show that given the first two exchanges, this hypothesis predicts that 26% of the dialogues will be problematic, while 36% of them actually will be. Of the dialogues that actually will be problematic, it can predict 57% of them. Once it predicts that a dialogue will be problematic, it is correct 79% of the time. Compared with the classifier for the first utterance alone, this classifier has a improvement of 16% in recall and 10% in precision, for an overall improvement in accuracy of 7%.

4. DISCUSSION AND FUTURE WORK

In summary, our results show that: (1) All feature sets significantly improve over the baseline; (2) Using automatic features from the whole dialogue, we can identify problematic dialogues 23% better than the baseline; (3) Just the first exchange provides significantly better prediction (8%) than the baseline; (4) The second exchange provides an additional significant (7%) improvement, (5) A classifier based on task-independent automatic features performs with less than 1% degradation in error rate relative to the automatic features; (6) The automatic feature set identifies problematic dialogues nearly as well as the complete feature set that includes human labels.

The research we report here is the second that we know of to attempt automatic analysis of entire dialogues for the purpose of identifying or predicting problematic situations. Litman et al. [5] also used a machine learning approach to analyze problematic dialogues, but focused on detecting poor speech recognition rather than prediction of problematic endings. Their features synthesized phenomenon across entire dialogues rather than on a per utterance basis, and thus the hypotheses learned could not be used directly for prediction during runtime. In contrast, our work uses primarily per-utterance features, some of which were suggested by their work. Also, their work analyzed a much smaller set of dialogues collected in controlled experiments from three research prototype systems. In contrast, the HMIHY data collection was done with live customer traffic.

In future work, we plan to develop additional features to improve the overall classifier. We also hope to integrate the hypotheses we learned into the HMIHY dialogue system in order to improve the system's overall performance.

5. REFERENCES

[1] A. Abella and A.L. Gorin. Construct algebra: An analytical method for dialog management. In *Proc. of the Association for Computational Linguistics*, 1999. reference for dialog component.

[2] William Cohen. Learning trees and rules with set-valued features. In *14th Conference of the American Association of Artificial Intelligence, AAAI*, 1996.

[3] A.L. Gorin E. Ammicht and T. Alonso. Knowledge collection for natural language spoken dialog systems. In *Proc. of EUROSPEECH 99*, 1999. reference for Wizard Over-ride.

[4] A.L. Gorin, G. Riccardi, and J.H. Wright. How may i help you? *Speech Communication*, 23:113–127, 1997. General reference for HMIHY.

[5] Diane J. Litman, Marilyn A. Walker, and Michael J. Kearns. Automatic detection of poor speech recognition at the dialogue level. In *Proceedings of the 37th Annual Meeting of the Association of Computational Linguistics , ACL99*, pages 309–316, 1999.

[6] G. Riccardi and A.L. Gorin. Spoken language adaptation over time and state in a natural spoken dialog system. *IEEE Transactions on Speech and Audio*, to appear. reference on dialog trials.