Reducing Recognition Error Rate based on Context Relationships among Dialogue Turns

Hsu-Chih Wu† *and Stephanie Seneff*‡

† Industrial Technology Research Institute (ITRI), Hsinchu, Taiwan
‡ MIT CSAIL Laboratory
32 Vassar Street, Cambridge, MA 02139
seneff@csail.mit.edu, hcw@itri.org.tw

Abstract

We have recently been conducting research on developing spoken dialogue systems to provide conversational practice for a learner of a foreign language. One of the most critical aspects of such a system is speech recognition errors, since they often take the dialogue thread down a wrong turn that is very confusing to the student and may be irrecoverable. In this paper we report on a machine learning technique to assist the process of selection from a list of N-best candidates based on a high-level description of the semantics of the preceding dialogue. In a user simulation experiment, we show that a significant reduction in sentence error rate can be achieved, from 29.2% to 23.6%. We have not yet verified that our techniques hold for real user data.

Index Terms: dialogue modeling, user simulation, confidence scoring, machine learning

1. Introduction

For some time, we have been developing spoken dialogue systems to allow a student to practice conversation in a foreign language [6]. While most of the past research in dialogue systems has centered around information access applications, our interest is in choosing a dialogue topic that is more approriate for a first-year student of a foreign language. To this end, we have designed a dialogue "game" in which the computer and student role play different personas, and they are jointly tasked with finding a suitable time in the near future to jointly engage in an activity that they both "like." One interesting aspect of this dialogue interaction is its symmetry: once a computer dialogue manager has been designed that can role play the system half of the conversation, it can easily be configured to simulate the user half of the conversation as well, since both sides share a common goal. Each conversational partner (whether it be a human or a computer) is assigned a distinct *persona*, with particular preferences and a specified future schedule of events. A further advantage is that the simulated user can also role-play a tutor, advising the student on what to say next.

Since our intent is to allow students of a foreign language to speak with the system, we are confronted with a difficult task of recognizing heavily accented speech. Thus, while the game vocabulary is currently quite small (212 words), the hesitant and accented speech will be difficult for the computer to understand. Thus we need to exploit as much information as possible to help with the selection of the most promising recognition hypothesis. This paper describes our experiments designed to exploit user simulation and machine learning to utilize context relationships among dialogue turns. The goal is to both provide a confidence score and to reduce recognition error, in a dialogue system intended for use as a mechanism for conversational practice for a student learning English.

In the remainder of this paper, we will first discuss related research, with respect to machine learning and exploiting dialogue context. We will then describe our application domain and our use of user simulation experiments to obtain our models. In Section 5 we describe the Learning Classifier System (LCS) model and our decisions about parameterizing the models. We follow with a description of our experiments and results. The final section summarizes and discusses future plans.

2. Related Research

It is well known that, by taking into account dialogue context information, speech understanding performance can be improved. Information that has been exploited includes acoustic information, word-level information, sentence-level information, and dialogue-level information. Furthermore, different features can be integrated to achieve better results. For example, Jonson [3] used many features, including utterance features, immediate context feature, dialogue context features, and list features to rerank speech recognition hypotheses.

Machine learning approaches used include the memory based learner TiMBL [3, 1] and the rule induction learner RIPPER [4, 1]. Below are two examples of typical learned rules:

$$\begin{array}{ll} If(f_1 = C_1, f_2 < C_2, f_3 < C_3) & \text{then Reject} \\ If(f_1 = C_1, f_2 > C_4, f_3 < C_5) & \text{then Accept} \end{array}$$

I.e., selected system parameters (f_i) are evaluated against simple arithmetic conditions involving constants (C_i) optimized by some training method.

A potential limitation of the memory-based learner and the rule induction learner is that thus far only two kinds of dialogue-level information have been considered: dialogue context information involving the previous one or two turns, or statistical information measured over the entire dialogue (e.g., overall dialogue error rate or number of dialogue turns). In our analysis, we have found that context information integrated over a longer window (even over turns that are not consecutive) can be taken into account to improve recognition performance. We used a rule representation that can model the relationships among a sequence of N preceding dialogue turns. We used LCS, a massive evolutionary parallelism approach, to train the rules. LCS [2] is a well-known machine learning algorithm, closely related to reinforcement learning and genetic algorithms. The trained rules can be used both to assign a confidence score and to rerank an N-best list in order to improve ASR performance.

Sys: Are you free tomorrow afternoon?
Usr: No, I am going to go shopping tomorrow afternoon.
Sys: I like shopping.
Usr: Would you like to join me?
Sys: That would be great!
Usr: Good bye now.

Figure 1: A typical dialogue in our domain.

Pattern Length	# Abnormal Patterns	Longest Abnormal Subsequence: L=2	L=3	L=4
2	7	7	n/a	n/a
3	32	18	14	n/a
4	38	16	16	6

Table 1: Analysis of data from simulation runs focused on patterns that were associated with recognition errors. Right three columns provide counts for shorter subsequences that also detect the error. See text for discussion.

3. User Simulation Dialogues

A machine learning algorithm requires a large number of example patterns to train the model statistics. It is very costly to acquire examples through real-user interactions, so it is important to be able to simulate the user to automatically generate appropriate training dialogues. Due to the dialogue symmetry in our domain, the user simulator and the dialogue manager are identical, except that each one randomly generates a different set of preferences and scheduled events. A typical dialogue in the domain is shown in Figure 1. In a configuration of our system that includes a real user, we still include the simulated user, whose role is to act as a tutor to propose something the student could say next. This proposal can be presented in their native language or in English, depending on the difficulty level. At the highest difficulty level, the tutor only provides help if asked by the user.

In order to simulate the effect of recognition errors, we include an extra synthesize and recognize cycle for the "user" half of the dialogue turn in user simulations. That is, the text of the simulated user's turn is processed through a speech synthesizer (Dectalk) and then the speech is recognized by a speech recognizer. While we think the error rate is not as high as it would be for a student of the language, still it serves as a useful starting point for training the LCS model.

4. Dialogue Studies

In an initial analysis, our goal was to confirm that recognition error can be detected by analyzing dialogue turn sequences. We define the context information of a dialogue turn in terms of its high level speech act, taking on values such as "*V*erify," "*Q*uery," "*S*tate," "*A*ffirm," "*D*eny," and "*C*larify." While we only make use of speech acts as features in this paper, our framework could be extended to include other relevant features from the dialogue context, such as user-specified or system-specified attributes. Our interest is in demonstrating a technique to model more sophisticated and long-term relationships among dialogue turns.

We define a "dialogue turn pattern" as a sequence of consecutive speech acts. For example, the dialogue below is represented as the dialogue turn pattern, *VDVA*:

Usr[*V*]: Do you like baseball?
Sys[*D*]: No, I don't like baseball.
Usr[*V*]: Do you like tennis?
Sys[*A*]: Yes, I do like tennis.

Our procedure involves the following four steps:

1. Run the user simulation on a large number of dialogues in text mode. The resulting dialogue log, Log1, contains no recognition errors.
2. Run the user simulation including a synthesize-and-recognize cycle to introduce recognition errors. Call this Log2.
3. Analyze Log1 and Log2, transforming them into dialogue turn patterns.
4. Tabulate occurrences of each combination of strings with various lengths.

Through this analysis, we discovered that there were 38 abnormal strings of length 4 showing up only in the runs that included the recognizer, only 16 of which can be detected if we only consider length 2 subsequences. For example, the pattern "*S*tate-*A*ffirm-*Q*uery" is odd in our domain, but the sub-patterns "*S*tate-*A*ffirm" and "*A*ffirm-*Q*uery" are common.

Table 1 shows some results when string length is 2, 3, and 4. If we examine the strings whose length is 2, then we find that all of the 7 strings occurring in log2, but not in log1, are associated with recognition error. For example, QQ occurs in log2, but it is not reasonable to respond to a wh-question with another wh-question.

We next examined strings with length 3, of which there are 32 occurring in log2 but not in log1. Among these 32, 18 contain unique substrings that occur in the case of length=2. For example, QQV is an abnormal string with length=3, while QQ is abnormal in length 2. Thus all of these 18 abnormal strings would have been detected if we had examined only strings with length=2. However, there are an additional 14 abnormal strings that happen only when length=3. For example, *S*tate-*A*ffirm-*Q*uery contains substrings *SA* and AQ. Both *SA* and *AQ* are normal in the case of length=2. But *SAQ* is abnormal in case of length=3, as illustrated in the abnormal dialogue example below:

Usr[*S*]: "Let's get together to play tennis this afternoon."
Sys[*A*]: "Yes, I would be delighted to play tennis."
Usr[*Q*]: "What did you do this morning?" [error?]

This means that, if we only consider the case length=2, then, among the 32 abnormal string patterns, only 18 can be detected. Similarly, there are 38 abnormal patterns with length=4, 6 of which would have been missed if we only looked at length 3 patterns.

From the result we can see that the more previous turns are taken into account, the more cases of recognition error can be detected. Furthermore, it is intuitive to think that there may be relationships among dialogue turns that are not consecutive. Thus, we adopted a model to represent such relationships, exploiting '#' to symbolize "don't-care," e.g., *DDQ#* ⟶ A.

5. *The LCS model*

A block diagram of our learning procedure is shown in Figure 2. We use LCS to derive a rule set of dialogue turn patterns, for the following reasons:

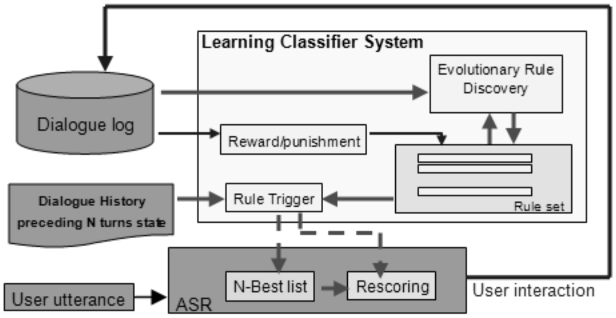

Figure 2: LCS Model

1. Consecutive relationships can be encoded as a concatenated string, which is a simple but convenient representation in LCS.
2. Relationships among dialogue turns may be too complex to be represented by a single rule. By using LCS, many rules can be applied simultaneously to determine a dialogue turn.

5.1. Representation of rules

Rules in LCS are represented as:

$$M_1M_2M_3...M_n : R, C \tag{1}$$

Where:

- $M_1M_2M_3...M_n$: the consecutive sequence of dialogue turn context information. M_n can include predefined context information and the special symbol '#' (don't care).
- R: the resulting speech act in the current user turn.
- C: the score (confidence measure) of this rule.

For example, suppose there is a rule: *VDVA*: *V*, **20**, and the following sequence of dialogue turns is observed:

Usr[*V*]:	Do you like baseball?
Sys[*D*]:	No, I don't like baseball.
Usr[*V*]:	Do you like tennis?
Sys[*A*]:	Yes, I do like to play tennis.
Usr[*V*]:	Would you like to play tennis with me?

Since the previous 4 dialogue turns match the "if-part" of the rule, and the last dialogue turn matches the "result part," the score of the rule is increased by 1.

5.2. Training the Rule Set

In the evolutionary rule discovery module, the rule set is trained by an evolutionary computation approach, as shown in Figure 3.

1. Initially there are P randomly generated rules in the rule set. Dialogue logs that do not have any recognition error, namely logs_NoError, are used to train the LCS.
2. At each generation, the rule set will be processed by three evolutionary operators: rule variation, rule evaluation and rule selection.

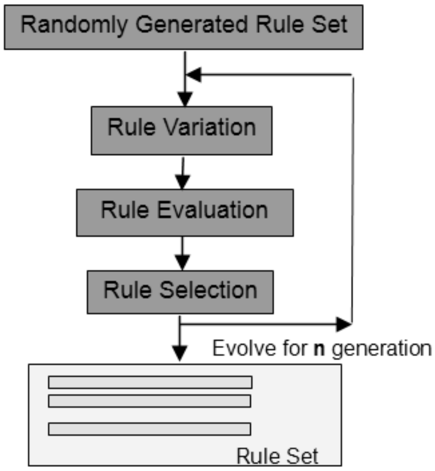

Figure 3: Evolutionary rule discovery procedure.

   - Rule variation: rule variation contains two kinds of operators: combination and mutation. In combination, two rules are combined to generate a new rule. For example, VS##:S and ##SQ:S can be combined to generate the new rule: VSSQ:S. In mutation, one dialogue turn is mutated into another kind of context information. For example, VD#V:S can be mutated into VD##:S for a more general rule, or VDSV:S for a more specific rule.
   - Rule evaluation: each rule's score is computed according to the number of times it matches dialogue turn patterns in logs_NoError.
   - Rule selection: rules with low score are replaced by other randomly generated rules or rules in the rule set with a higher score.
3. The LCS is trained for a large number of generations (in our case 10,000) in a simulation run.
4. The LCS is run N times, and P distinctive rules with the highest scores are extracted. Multiple runs can mitigate the effect of premature convergence, should the LCS fall into a local maximum. In our experiments, N=5, and P=300. Each run takes about ten minutes of computer time.

   The extraction step involves the following considerations:

   - The most general rules, e.g., "####: S," are ignored. Since, although their score is relatively high, they contain no useful information.
   - We must keep the rule set as specific as possible. For example, if there are two rules, e.g.,: "##QC: S" and "##Q#: S" with the same score, then "##Q#: S" would be skipped since it is the more general one. If this were not done, then the score of other similar rules, such as "##QC: S," would be incomparable, since they would not benefit from such an artificial bonus.

5.3. Detecting Recognition Error

The trained rule set is applied to dialogue logs that have recognition errors, to validate their usefulness, according to the fol-

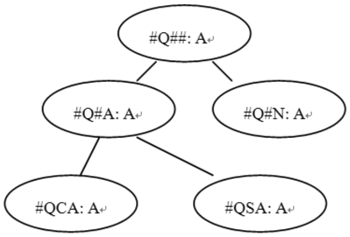

Figure 4: Graph showing five rules and their dependencies.

	Baseline	After	Relative
		Rescoring	Reduction
Overall	42 (29.2%)	34 (23.6%)	19%
Critical	20	12	40%
Non-critical	22	22	0%

Table 2: Recognition results, in terms of Sentence Error Rate (SER), comparing the incorporation of the LCS model with a baseline that simply picks the 1-best recognizer hypothesis. Results are reported separately for two subsets of the data, based on whether the errors are critical or not.

lowing algorithm:

1. For each dialogue move in the dialogue, apply all the rules to it, and the confidence measure of that turn is computed as the sum of scores of all matching rules.

   When matching the rule, it is important to match the most specific rule, not all rules that match the move, to increase the accuracy of the score. For example, suppose there are five rules in the rule set, and their set-subset relation is as shown in the graph in Figure 4. It should be noted that their scores are all different. (If two of them were the same, then the more general rule would have been deleted during the rule extraction phase).

   If the dialogue move is: #QVA:A, then it matches two rules: #Q##:A and #Q#A:A. Only the score of #Q#A:A is added, keeping the more specific rule.

2. Analyze each possible hypothesis in the N-best list (N =10), to find its context pattern (CP). Identify the highest scoring CP, CP_{max}, appearing in the N-best list.

3. Among the hypotheses associated with the pattern CP_{max}, choose the hypothesis with the highest combined parse and acoustic score.

6. Experimental Results

Table 2 shows the results for a simulated dialogue log containing 144 turns in total. The overall sentence error rate was reduced from 29.2% to 23.6%, a relative improvement of 19%. Some of the errors are more critical than others. For example, “I do like swimming,” was incorrectly recognized as “Do you like go swimming?”, which will disrupt the continuation of the dialogue. A much less disruptive error occurs when “Okay, we will get together to dance,” is incorrectly recognized as “Okay, let’s get together to dance.” In the table we observe that, although the non-critical sentence error rate remains unchanged, the relative error rate for critical errors is reduced by 40%. A few correctly recognized sentences were incorrectly modified by our process, usually at the beginning of the dialogue where no information on previous turns is available for rule matching.

7. Conclusion and Future Work

In this paper, we verify that there are relationships among dialogue turns, which may apply across several dialogue turns. We use a string-based rule format to model the context relationships among several dialogue turns. A set of rules is used to model all the relationship patterns in an existing dialogue log. The rule set, trained using LCS, can be used to reprocess the N-best list, and thus to detect and correct ASR recognition errors.

There are several things that could be done to improve the performance. First is to increase the complexity of the training data, either by making the simulated user’s behavior more complex, or by collecting training data based on human-computer dialogues.

There could be benefit in defining more sophisticated context categories. For example, “I am going to play tennis tomorrow morning, would you like to join me?” is considered as a “Statement,” but actually there is a “Verify” clause as well.

Our algorithm could be improved by taking into account attribute values as well as speech acts. For example, if the previous dialogue turn was a verification question containing a content word such as “baseball,” then an “Affirm” or “Deny” response should also contain “baseball,” to be consistent. To be specific, if the system asks, “Do you like baseball?” the incorrect top candidate “Yes, I do like basketball” could conceivably be corrected, if an N-best item contains “baseball” instead.

We would also like to explore the possibility of incorporating additional context into the rule. We have identified three features within our domain that may be important: attribute values (e.g., hobbies), subject pronoun (I, you in the sentence), and time span (past, present, or future).

In future work, we plan to collect data from students playing the game and to assess the impact of our N-best selection algorithm on system performance.

8. Acknowledgements

This research is a partial result of the Project 5352BA5210 conducted by ITRI under sponsorship of the Ministry of Econimic Affairs, Taiwan, R.O.C.

9. References

[1] Gabsdil, M., and Lemon, O., “Combining acoustic and pragmatic features to predict recognition performance in spoken dialogue systems,” *Proc. Association for Computational Linguistics*, 2004.

[2] Holland, J. H. “A Mathematical Framework for Studying Learning in Classifier Systems.” Physica D, 22:307-317, 1986.

[3] Jonson, R., “Dialogue Context-based Re-Ranking of ASR Hypotheses,” *Proc. IEEE 2006 Workshop on Spoken Language Technology*, Aruba, 2006.

[4] Litman, D.J., Walker M.A., and Kearns M.S. “Automatic Detection of Poor Speech Recognition at the Dialogue Level,” *Proc. Association for Computational Linguistics*, pp. 309-316, 1999.

[5] Lemon, O., “Context-sensitive speech recognition in ISU dialogue systems: results for the grammar switching approach,” *Workshop on the Semantics and Pragmatics of Dialogue*, 2004.

[6] Seneff, S., “Interactive Computer Aids for Acquiring Proficiency in Mandarin,” Keynote Speech, pp. 1–11, *Proc. ISCSLP*, Singapore, 2006.