Using Task-Oriented Spoken Dialogue Systems for Language Learning: Potential, Practical Applications and Challenges

Antoine Raux **Maxine Eskenazi**

Language Technologies Institute
Carnegie Mellon University
5000 Forbes Avenue
15213 Pittsburgh, USA

{antoine, max+}@cs.cmu.edu

Abstract

The technology developed for task-based spoken dialogue systems (SDS) has a significant potential for Computer-Assisted Language Learning. Based on the CMU Let's Go SDS, we describe two areas in which we investigated adaptations of the technology to non-native speakers: speech recognition and correction prompt generation. Although difficulties remain, particularly towards robust understanding, results prove that this technology can be used to provide realistic, involved environment for language learning.

1 Introduction

Outside of classic learning settings, students in an immersion situation learn from each experience where their speech in the target language is modified according to the reaction to what they have said. Learning experiences are plentiful. But such an experience poses a challenge. In a real situation, getting a result and doing so quickly are more important than learning something new. The new element must be presented rapidly, with no negative consequence if the person does not modify his or her speech, and it must take a different form from classical language learning feedback.

State-of-the-art Spoken Dialogue Systems (SDS) provide a framework for users to have simple conversations with a machine to do a specific task like giving bus schedule information or reserving a flight. These systems have clear potential for Computer-Assisted-Language Learning systems that place the student in a realistic situation where a specific task has to be accomplished in the target language. However, traditional SDS assume that the user has perfect mastery of the interaction language, and so any disruption of the interaction, and the corresponding repair process, is due to the machine imperfect understanding of spoken language caused by speech recognition errors, misinterpretations of the user's intentions, etc. Thus a number of challenges need to be addressed. After a brief description of Let's Go, a telephone-based bus schedule information system built for this research, we present two challenges and describe our solutions to them: understanding non-native speech and responding appropriately to ungrammatical utterances to help the user/student acquire grammar structures in the target language.

2 The Let's Go Spoken Dialogue System

2.1 Overview

Current spoken dialogue systems are composed of a set of modules each handling a different part of the process and exchanging information with one another using a central or distributed architecture. The three main sub-tasks handled by these modules are:

- spoken language understanding
- dialogue management
- spoken language generation

We give a broad description of the system here. For more details, see (Raux et al. 2003).

2.2 Spoken Language Understanding

Understanding is usually accomplished in two separate passes. First the user's speech is transcribed into a string of words by a speech recognizer. Then a natural language understanding module parses the transcription hypothesis into some internal semantic representation on which the system can perform reasoning.

Speech recognition in Let's Go is performed by the CMU Sphinx 2 recognizer (Huang et a.l. 1992), a speaker independent real-time speech recognizer; natural language understanding is performed by Phoenix, a robust parser particularly adapted to speech.

2.3 Dialogue Management

For meaningful, natural interaction with the user, dialogue systems follow a model of human task-oriented dialogue. In our case, we use RavenClaw

(Bohus and Rudnicky 2003), a general purpose dialogue management framework developed at CMU. RavenClaw has two main components:

- a description of the task to be performed, indicating the information that the system must gather from the user and in what order, what to do with this information, and which information to give to the user in return.
- a set of strategies modeling the behavior of a human speaker (e.g. asking for repetition or confirmation).

2.4 Spoken Language Generation

The last element needed for a conversation is conveying or requesting information in natural spoken language. The understanding part is usually in two steps. First, natural language sentences are generated from the system's internal data representation. This has different levels of complexity, from simple template-based approaches to complex planning systems that dynamically decide what information to actually give, in which order and with which words. Due to our fairly limited target domain, we used the simpler approach and hand-wrote the prompt templates required for our task.

The second step is speech synthesis, i.e. transforming the generated sentence into a spoken utterance understandable to the user. Here again, decades of research in the area have created a wide range of solutions, from flexible but unnatural general purpose to task-specific high quality synthesizers. In Let's Go, we opted for the latter solution, although, as we will see later, we backed up to a more flexible system for certain experiments.

3 Understanding Non-Native Speakers

3.1 Discrepancy in Performance between Native and Non-Native Speakers

The first major issue in enabling non-native speakers to use spoken dialogue systems - the one that has received most attention, is to be able to understand non-native speech as well as human listeners. Indeed, while speech recognition has recently reached a satisfying level of accuracy for native speakers in limited domains, it remains brittle when confronted with non-native speech.

This is mostly due to the mismatch between the accent of non-native speakers and of the native speakers used to train the recognizer's acoustic models. For example in (Raux and Eskenazi 2004), word error rate of the original Let's Go system on non-native speakers was 52%, more than 2.5 times that of native speakers (20.4%). By using gender-specific acoustic models, we reduced WER to 43.1% for non-native speakers (resp. 17% for native speakers), but the ratio of the two remained unchanged at 2.5.

Improving the accuracy of the recognition of non-native speech is a pre-requisite to any use of spoken dialogue systems for language learning.

3.2 Past Work on Non-Native Speech Recognition

Since data is costly for a fully trained acoustic model for a specific accent, let alone for non-native speakers in general, researchers have mostly resorted to speaker adaptation. This amounts to using a small amount of transcribed non-native accented speech to "adapt", or modify, acoustic models that were originally trained on a large amount of native speech. Standard adaptation techniques, such as Maximum Likelihood Linear Regression, have been used with some success, most with English as the target language (L2) and a variety of L1s (e.g. Spanish (Byrne et al 1998), Japanese (Mayfield Tomokiyo and Waibel 2001)).

However, these all assume that the user's native language is known and that some data and/or expert knowledge about her accent is available. In practice, there are many cases where the user 's L1 cannot be known in advance nor can acoustic models for all possible accents be trained.

3.3 Acoustic and Lexical Adaptation on Multiple Accents

In order to improve speech recognition for as broad a range of non-native speakers as possible in the Let's Go system, we first performed acoustic adaptation on 169 minutes of speech from non-native calls to the system, covering a variety of accents including speakers from Japan, India, Germany and China. As shown in (Raux 2004), performing this accent insensitive adaptation on gender-specific models reduced the WER by 17%, down to 35.8%. In order to further improve performance, we automatically captured the pronunciation habits of speakers in the training set by estimating the phonetic lexicon matching each speaker and clustering speakers with similar lexicons. Using two clusters per gender, we got a WER of 29.8%.

We achieved these results by automatically selecting the cluster for each new utterance at runtime, based solely on the recognizer score. Moreover, using an oracle to always pick the actual best recognition hypothesis yields a WER of 23.5%, suggesting performance could be improved by selecting hypotheses more accurately at recognition time, for instance using a richer set of features.

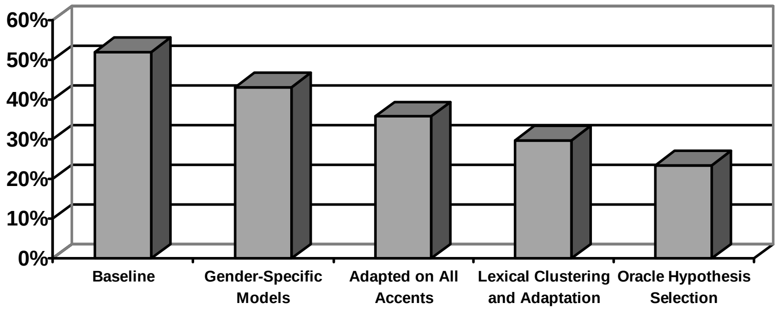

Figure 1. Word Error Rate on Non-Native Speech

3.4 Language Model Adaptation

While acoustic mismatch is the leading cause of misrecognition, we showed (Raux and Eskenazi 2004) that there is also a linguistic mismatch because the linguistic patterns (choice of words, syntax…) of non-native speakers don't always match those in the recognizer's language model and in the parser, both usually built on native utterances.

In Let's Go, including non-native data in the language model significantly improved both perplexity and out-of-vocabulary (OOV) rate. More importantly, it reduced the gap in these metrics between the performance of the model on non-native data and native data. At the recognition level, while a small WER reduction was observed using a language model including non-native data, this reduction was similar on non-native and native speech, indicating that more data is necessary to properly train the model, even for native speakers. However, on more open tasks, it is likely that the difference in language patterns between non-native speakers and native speakers would be greater and have a larger impact on recognition accuracy. For instance, this would probably be the case in a "free conversation" CALL system, where the student is asked to conduct a dialogue with a computer agent.

4 Dialogue Strategies to Enhance Language Learning

4.1 Lexical Entrainment for Language Learning

In immersion, where a non-native has to perform everyday tasks in the target language, native listeners use a wide range of strategies when faced with ungrammatical, expressions, from ignoring them to explicitly correcting them.

Similarly, while it would be enough for a traditional SDS to ignore ungrammaticalities and complete its task, a task-oriented language learning system has a dual goal: completing the task and improving the conversational ability of the user. In order to achieve the second goal, we can rely on a standard dialogue manager, for which strategies specifically dedicated to help the user learn the target language can be designed.

Indeed, this allows "corrective" conversational strategies to be put in competition with more standard confirmation or repetition strategies and to be selected appropriately by the dialogue manager according to some predefined heuristic-based policy (e.g. "send a correction prompt when the user utterance diverges from a grammatical sentence by at most two words, otherwise ask her to repeat") or other, more complex, algorithm.

4.2 Corrective Prompts in Let's Go

One issue when providing corrections embedded within a task-directed conversation is that they have to be as concise as possible while still providing understandable feedback to the user. In particular, such corrections should not contain too many words or grammatical structures that are unknown to the user. Thus, we designed an algorithm to generate corrections that are as close as possible to the user utterance and provide feedback by putting emphasis on the erroneous words. This method, represented in Figure 2, assumes that a list of target sentences is available beforehand.

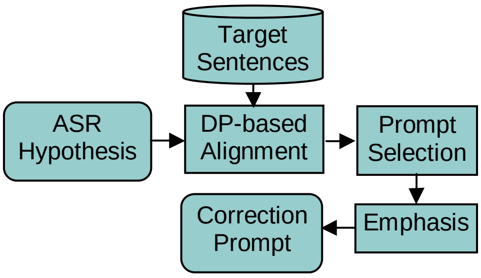

Figure 2 Corrective prompt generation algorithm

These are designed to cover as wide a range of possible user utterances to the system inside the task, while still being both grammatical and inside the language model and grammar of the system. The idea is that whenever a non-native speaker utters an ungrammatical utterance, her goal was actually to utter one of the target sentences but that she made a mistake like inserting, deleting or substituting a word. Based on the recognition result for the user utterance, we compute its distance to each target sentence using dynamic programming and select the closest target. If the two match exactly, no correction is produced and the dialogue continues normally. If words were deleted, inserted or substituted by the non-native, we generate a prompt that is both confirmation and correction, as exemplified in Figure 3 (second system prompt).

```
S: What can I do for you?
U: I want to go the airport.
S: Sorry, I didn’t get that.
Did you mean:
I want to go TO the airport?
U: Yes
S: To the airport.
Where are you leaving from?
U: ...
```

Figure 3 Dialogue with a corrective prompt

In order to focus the user’s attention on the corrected words, we generate natural F0 contours for emphasis based on a database of recorded emphasized prompts (see Raux and Black 2003). This allows the system to provide implicit corrective feedback while minimally interfering with the main task of the dialogue.

4.3 Experimental Results

To evaluate the proposed method, we conducted a small experiment on 24 calls to a modified version of Let’s Go containing correction prompts. The data shows that 40.7% of the correction prompts were “false positives” (i.e. triggered although the user utterance was grammatical and the dialogue could have proceeded correctly).

In addition, 64.4% of the prompts were triggered by recognition errors due to the system rather than ungrammatical user utterances (conducted on an early version of the system, which did not include most of the improvements described in 3 above). Unfortunately, this left too little clean data to draw conclusions on the effectiveness of the approach for language learning. To address both issues, we are working on improving speech recognition as described in section 3, and on improving the triggering mechanism by more fully integrating the corrective prompts into the dialogue manager and taking into account advanced features to estimate the recognizer’s confidence.

5 Conclusion

We described work on the Let’s Go project to adapt spoken dialogue systems to non-native users. Significant improvement of recognition accuracy was achieved through acoustic and lexical adaptation. To improve spoken dialogue system behavior on with ungrammatical utterances, a method to automatically generate corrective prompts was proposed. These improvements lay the base for using spoken dialogue system technology for computer-assisted language learning systems targeting conversational skills.

6 Acknowledgements

This work is supported by the US National Science Foundation under grant number 0208835, "LET'S GO: improved speech interfaces for the general public". Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

D. Bohus and A. Rudnicky. 2003. RavenClaw: Dialog management using hierarchical task decomposition and an expectation agenda. Eurospeech '03. Geneva, Switzerland.

W. Byrne, E. Knodt, S. Khudanpur and J. Bernstein. 1998. Is Automatic Speech Recognition Ready for Non-Native Speech? A Data Collection Effort and Initial Experiments in Modeling Conversational Hispanic English. STiLL 98, Marholmen, Sweden.

X. Huang, F. Alleva, H.-W. Hon, and K.-F. Hwang, K.-F., M.-Y. Lee and R. Rosenfeld. 1992. The SPHINX-II speech recognition system: an overview. *Computer, Speech and Language,* 7(2):137-148.

L. Mayfield Tomokiyo and A. Waibel. 2001. Adaptation Methods for Non-Native Speech. Multilinguality in Spoken Language Processing. Aalborg, Denmark.

A. Raux. 2004. Automated Lexical Adaptation and Speaker Clustering based on Pronunciation Habits for Non-Native Speech Recognition. Submitted to ICSLP 2004.

A. Raux and A. Black. 2003. A Unit Selection Approach to F0 Modeling and its Application to Emphasis. ASRU 2003. St Thomas, US Virgin Islands.

A. Raux and M. Eskenazi. 2004. Non-Native Users in the Let's Go!! Spoken Dialogue System: Dealing with Linguistic Mismatch. HLT/NAACL 2004. Boston, MA.

A. Raux, B. Langner, M. Eskenazi and A. Black. 2003. LET'S GO: Improving Spoken Dialog Systems for the Elderly and Non-natives. Eurospeech '03. Geneva, Switzerland.