A Voice Enabled Procedure Browser for the International Space Station

Manny Rayner, Beth Ann Hockey, Nikos Chatzichrisafis, Kim Farrell
ICSI/UCSC/RIACS/NASA Ames Research Center
Moffett Field, CA 94035–1000
`mrayner@riacs.edu, bahockey@email.arc.nasa.gov`
`Nikos.Chatzichrisafis@web.de, kfarrell@email.arc.nasa.gov`

Jean-Michel Renders
Xerox Research Center Europe
6 chemin de Maupertuis, Meylan, 38240, France
`Jean-Michel.Renders@xrce.xerox.com`

Abstract

Clarissa, an experimental voice enabled procedure browser that has recently been deployed on the International Space Station (ISS), is to the best of our knowledge the first spoken dialog system in space. This paper gives background on the system and the ISS procedures, then discusses the research developed to address three key problems: grammar-based speech recognition using the Regulus toolkit; SVM based methods for open microphone speech recognition; and robust side-effect free dialogue management for handling undos, corrections and confirmations.

1 Overview

Astronauts on the International Space Station (ISS) spend a great deal of their time performing complex procedures. Crew members usually have to divide their attention between the task and a paper or PDF display of the procedure. In addition, since objects float away in microgravity if not fastened down, it would be an advantage to be able to keep both eyes and hands on the task. Clarissa, an experimental speech enabled procedure navigator (Clarissa, 2005), is designed to address these problems. The system was deployed on the ISS on January 14, 2005 and is scheduled for testing later this year; the initial version is equipped with five XML-encoded procedures, three for testing water quality

and two for space suit maintenance. To the best of our knowledge, Clarissa is the first spoken dialogue application in space.

The system includes commands for navigation: forward, back, and to arbitrary steps. Other commands include setting alarms and timers, recording, playing and deleting voice notes, opening and closing procedures, querying system status, and inputting numerical values. There is an optional mode that aggressively requests confirmation on completion of each step. Open microphone speech recognition is crucial for providing hands free use. To support this, the system has to discriminate between speech that is directed to it and speech that is not. Since speech recognition is not perfect, and additional potential for error is added by the open microphone task, it is also important to support commands for undoing or correcting bad system responses.

The main components of the Clarissa system are a speech recognition module, a classifier for executing the open microphone accept/reject decision, a semantic analyser, and a dialogue manager. The rest of this paper will briefly give background on the structure of the procedures and the XML representation, then describe the main research content of the system.

2 Voice-navigable procedures

ISS procedures are formal documents that typically represent many hundreds of person hours of preparation, and undergo a strict approval process. One requirement in the Clarissa project was that the procedures should be displayed visually exactly as they
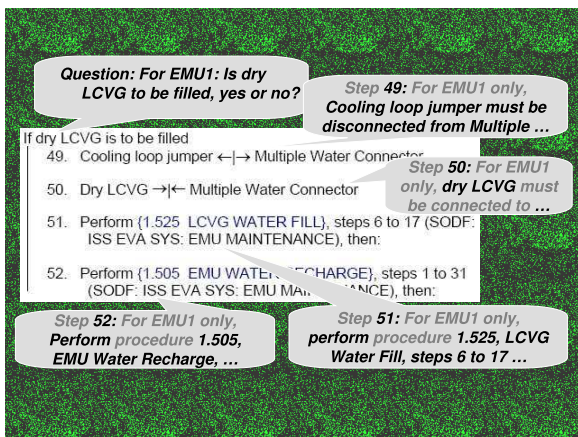
Figure 1: Adding voice annotations to a group of steps

Rec	Patterns	Errors		
		Reject	Bad	Total
Text	LF	3.1%	0.5%	3.6%
Text	Surface	2.2%	0.8%	3.0%
Text	Surface+LF	0.8%	0.8%	1.6%
SLM	Surface	2.8%	7.4%	10.2%
GLM	LF	1.4%	4.9%	6.3%
GLM	Surface	2.9%	4.8%	7.7%
GLM	**Surface+LF**	**1.0%**	**5.0%**	**6.0%**

Table 1: Speech understanding performance on six different configurations of the system.

appear in the original PDF form. However, reading these procedures verbatim would not be very useful. The challenge is thus to let the spoken version diverge significantly from the written one, yet still be similar enough in meaning that the people who control the procedures can be convinced that the two versions are in practice equivalent.

Figure 1 illustrates several types of divergences between the written and spoken versions, with "speech bubbles" showing how procedure text is actually read out. In this procedure for space suit maintenance, one to three suits can be processed. The group of steps shown cover filling of a "dry LCVG". The system first inserts a question to ask which suits require this operation, and then reads the passage once for each suit, specifying each time which suit is being referred to; if no suits need to be processed, it jumps directly to the next section. Step 51 points the user to a subprocedure. The spoken version asks if the user wants to execute the steps of the subprocedure; if so, it opens the LCVG Water Fill procedure and goes directly to step 6. If the user subsequently goes past step 17 of the subprocedure, the system warns that the user has gone past the required steps, and suggests that they close the procedure.

Other important types of divergences concern entry of data in tables, where the system reads out an appropriate question for each table cell, confirms the value supplied by the user, and if necessary warns about out-of-range values.

3 Grammar-based speech understanding

Clarissa uses a grammar-based recognition architecture. At the start of the project, we had two main reasons for choosing this approach over the more popular statistical one. First, we had no available training data. Second, the system was to be designed for experts who would have time to learn its coverage, and who moreover, as former military pilots, were comfortable with the idea of using controlled language. Although there is not much to be found in the literature, an earlier study in which we had been involved (Knight et al., 2001) suggested that grammar-based systems outperformed statistical ones for this kind of user. Given that neither of the above arguments is very strong, we wanted to implement a framework which would allow us to compare grammar-based methods with statistical ones, and retain the option of switching from a grammar-based framework to a statistical one if that later appeared justified. The Regulus and Alterf platforms, which we have developed under Clarissa and other earlier projects, are designed to meet these requirements.

The basic idea behind Regulus (Regulus, 2005; Rayner et al., 2003) is to extract grammar-based language models from a single large unification grammar, using example-based methods driven by small corpora. Since grammar construction is now a corpus-driven process, the same corpora can be used to build statistical language models, facilitating a direct comparison between the two methodologies.

On its own, however, Regulus only permits comparison at the level of recognition strings. Alterf (Rayner and Hockey, 2003) extends the paradigm to

ID	Rec	Features	Classifier	Error rates				
				Classification				Task
				In domain		Out	Av	
				Good	Bad			
1	SLM	Confidence	Threshold	5.5%	59.1%	16.5%	11.8%	10.1%
2	GLM	Confidence	Threshold	7.1%	48.7%	8.9%	9.4%	7.0%
3	SLM	Confidence + Lexical	Linear SVM	2.8%	37.1%	9.0%	6.6%	7.4%
4	GLM	Confidence + Lexical	Linear SVM	2.8%	48.5%	8.7%	6.3%	6.2%
5	SLM	Confidence + Lexical	Quadratic SVM	2.6%	23.6%	8.5%	5.5%	6.9%
6	**GLM**	**Confidence + Lexical**	**Quadratic SVM**	**4.3%**	**28.1%**	**4.7%**	**5.5%**	**5.4%**

Table 2: Performance on accept/reject classification and the top-level task, on six different configurations.

the semantic level, by providing a trainable semantic interpretation framework. Interpretation uses a set of user-specified patterns, which can match either the surface strings produced by both the statistical and grammar-based architectures, or the logical forms produced by the grammar-based architecture.

Table 1 presents the result of an evaluation, carried out on a set of 8158 recorded speech utterances, where we compared the performance of a statistical/robust architecture (SLM) and a grammar-based architecture (GLM). Both versions were trained off the same corpus of 3297 utterances. We also show results for text input simulating perfect recognition. For the SLM version, semantic representations are constructed using only surface Alterf patterns; for the GLM and text versions, we can use either surface patterns, logical form (LF) patterns, or both. The "Error" columns show the proportion of utterances which produce no semantic interpretation ("Reject"), the proportion with an incorrect semantic interpretation ("Bad"), and the total.

Although the WER for the GLM recogniser is only slightly better than that for the SLM recogniser (6.27% versus 7.42%, 15% relative), the difference at the level of semantic interpretation is considerable (6.3% versus 10.2%, 39% relative). This is most likely accounted for by the fact that the GLM version is able to use logical-form based patterns, which are not accessible to the SLM version. Logical-form based patterns do not appear to be intrinsically more accurate than surface (contrast the first two "Text" rows), but the fact that they allow tighter integration between semantic understanding and language modelling is intuitively advantageous.

4   Open microphone speech processing

The previous section described speech understanding performance in terms of correct semantic interpretation of in-domain input. However, open microphone speech processing implies that some of the input will not be in-domain. The intended behaviour for the system is to reject this input. We would also like it, when possible, to reject in-domain input which has not been correctly recognised.

Surface output from the Nuance speech recogniser is a list of words, each tagged with a confidence score; the usual way to make the accept/reject decision is by using a simple threshold on the average confidence score. Intuitively, however, we should be able to improve the decision quality by also taking account of the information in the recognised words.

By thinking of the confidence scores as weights, we can model the problem as one of classifying documents using a weighted bag of words model. It is well known (Joachims, 1998) that Support Vector Machine methods are very suitable for this task. We have implemented a version of the method described by Joachims, which significantly improves on the naive confidence score threshold method.

Performance on the accept/reject task can be evaluated directly in terms of the classification error. We can also define a metric for the overall speech understanding task which includes the accept/reject decision, as a weighted loss function over the different types of error. We assign weights of 1 to a false reject of a correct interpretation, 2 to a false accept of an incorrectly interpreted in-domain utterance, and 3 to a false accept of an out-of-domain utterance. This

captures the intuition that correcting false accepts is considerably harder than correcting false rejects, and that false accepts of utterances not directed at the system are worse than false accepts of incorrectly interpreted utterances.

Table 2 summarises the results of experiments comparing performance of different recognisers and accept/reject classifiers on a set of 10409 recorded utterances. "GLM" and "SLM" refer respectively to the best GLM and SLM recogniser configurations from Table 1. "Av" refers to the average classifier error, and "Task" to a normalised version of the weighted task metric. The best SVM-based method (line 6) outperforms the best naive threshold method (line 2) by 5.4% to 7.0% on the task metric, a relative improvement of 23%. The best GLM-based method (line 6) and the best SLM-based method (line 5) are equally good in terms of accept/reject classification accuracy, but the GLM's better speech understanding performance means that it scores 22% better on the task metric. The best quadratic kernel (line 6) outscores the best linear kernel (line 4) by 13%. All these differences are significant at the 5% level according to the Wilcoxon matched-pairs test.

5 Side-effect free dialogue management

In an open microphone spoken dialogue application like Clarissa, it is particularly important to be able to undo or correct a bad system response. This suggests the idea of representing discourse states as objects: if the complete dialogue state is an object, a move can be undone straightforwardly by restoring the old object. We have realised this idea within a version of the standard "update semantics" approach to dialogue management (Larsson and Traum, 2000); the whole dialogue management functionality is represented as a declarative "update function" relating the old dialogue state, the input dialogue move, the new dialogue state and the output dialogue actions.

In contrast to earlier work, however, we include task information as well as discourse information in the dialogue state. Each state also contains a back-pointer to the previous state. As explained in detail in (Rayner and Hockey, 2004), our approach permits a very clean and robust treatment of undos, corrections and confirmations, and also makes it much

simpler to carry out systematic regression testing of the dialogue manager component.

References

Clarissa, 2005. http://www.ic.arc.nasa.gov/projects/clarissa/. As of 26 April 2005.

T. Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*, Chemnitz, Germany.

S. Knight, G. Gorrell, M. Rayner, D. Milward, R. Koeling, and I. Lewin. 2001. Comparing grammar-based and robust approaches to speech understanding: a case study. In *Proceedings of Eurospeech 2001*, pages 1779–1782, Aalborg, Denmark.

S. Larsson and D. Traum. 2000. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural Language Engineering, Special Issue on Best Practice in Spoken Language Dialogue Systems Engineering*, pages 323–340.

M. Rayner and B.A. Hockey. 2003. Transparent combination of rule-based and data-driven approaches in a speech understanding architecture. In *Proceedings of the 10th EACL (demo track)*, Budapest, Hungary.

M. Rayner and B.A. Hockey. 2004. Side effect free dialogue management in a voice enabled procedure browser. In *Proceedings of INTERSPEECH 2004*, Jeju Island, Korea.

M. Rayner, B.A. Hockey, and J. Dowding. 2003. An open source environment for compiling typed unification grammars into speech recognisers. In *Proceedings of the 10th EACL*, Budapest, Hungary.

Regulus, 2005. http://sourceforge.net/projects/regulus/. As of 26 April 2005.