REAL-TIME SPEECH RECOGNITION SYSTEM

Hy Murveit and Mitchel Weintraub

SRI International
Menlo Park, CA 94025

PROJECT GOALS

SRI and U.C.Berkeley are developing hardware for a real-time implementation of spoken language systems (SLS). Our goal is to develop fast speech recognition algorithms and supporting hardware capable of recognizing continuous speech from a bigram or trigram based 20,000 word vocabulary or a 1,000 to 5,000 word SLS system.

RECENT RESULTS

SRI and U.C. Berkeley's recent accomplishments on this project include:

- Eight special-purpose IC's were designed, fabricated and tested.
- Designed, fabricated, and tested the Output-Distribution printed-circuit board. The Output_Distribution board computes and loads HMM state output probabilities onto the HMM Word-Processor board.
- Designed, fabricated, and tested the HMM Word-Processor printed-circuit board. The HMM Word-Processor board updates HMM state probabilities.
- Developed software to support the hardware effort. This includes the following software modules: simulation, system initialization, control program coordinating different hardware components, and the grammar computation on the SKY Challenger dual-processor TMS32030.
- Ported current noise-robust software algorithms (from Symbolics Lisp Machine) to run on a Sun Sparcstation in C. Ported this C implementation to a Banshee TMS32030 to run in real-time.

PLANS FOR THE COMING YEAR

- Complete the construction of the current hardware design, and software tools to support this architecture.
- Design a multiple-processor TMS320C30 board with a high I/O bandwidth to interface with the special-purpose HMM-board, and an interface to the MTU A/D box to compute the front-end VQ values.
- Develop a large vocabulary recognizer to fully use the computational capabilities of this design.
- Implement multiple types of grammars using this hardware.
- Use the real-time hardware for collecting data about man-machine speech interactions.
- Integrate the real-time recognizer into our research trainer to shorten the development cycle for corrective-training systems.
- Evaluate the current architecture to determine the computational and algorithmic bottlenecks.
- Replicate the system and port to a DARPA and NASA site.