SPEECH USER INTERFACE EVOLUTION

- *Authors:* John Karat, Jennifer Lai, Catalina Danis and Catherine Wolf
  - IBM T.J. Watson Research Center

Presented by: Leena Walavalkar

CSE 587 Introduction to Virtual Environments

Overview

- Introduction
- General characteristics of speech recognition (SR) systems considered in the paper
- Case studies
  - Personal Dictation Systems
  - MedSpeak
  - Conversation Machine
- Summary and Conclusions

Introduction

- Basic requirement of an automatic speech recognition (ASR) system – translate speech input into character strings or commands
- Relatively slow penetration of ASR into interfaces for computer systems – maybe speech is not a good modality!
  - Going from an acoustic signal to some computationally useful translation is technically challenging
  - Speech natural mode of communication (but for human-human, not human-computer)
- Takes time and practice to develop new form of interaction

Goals of this paper

- Help us understand how to incorporate speech into other systems and applications through practical examples
- Identify some common problems associated with speech interfaces in all systems and applications
- Design considerations and error handling
- Address human factors and related issues

PERSONAL DICTATION SYSTEM

- **User characteristics:**
  - General purpose system. Hence target audience- anybody
  - Assumption: users familiar with word processing on PC
- **Tasks:**
  - Speech enable existing word processing applications
  - Develop a new full function editor to address wide audience
  - Provided a speech window
- **Context of Use:**
  - Focus the design on production of medium to large quantities of text and view speech as an augmentation for other input devices

Capabilities and Constraints

- **Recognition Errors**
  - Delay in timing between speech event and feedback.
  - Make all error corrections in a single pass by providing information on what was said
  - Like keyboard, visually flag words that engine was less sure of. Also error correction mechanism modeled after spelling corrections dialogs (both did not work)
- **Lag in recognizing text**
  - IPDS algorithms based on language model. i.e probability of word match influenced by preceding and following word. Would not make a decision until then hence lag.
  - Also the best match could change with additional words

Capabilities and Constraints (contd)

- Mode Switching
  - Voice input directed to any applications running on a system.
  - Is flexible but need to know what application is expecting input
  - Proved easy for the system but difficult for users to keep track of
  - Command words that are also dictated words gives rise to mode split problem
  - Solution- inquire state of the desktop and create list of valid commands- but internal command names are not always obvious e.g copy and copy to clipboard
  - Not reached a satisfactory resolution to “what I can say”

Lessons Learned

- Tracked the performance of the system by creating email forums and gather usage reports
- Use of IPDS to compose email showed that it is fine for creating short pieces of informal text but not satisfactory for longer pieces
- SR technology provides some benefits for some tasks over keyboard but overall integration still lacking
    - ***Critique***
- *Makes the point of h-h communication against h-c communication*
- *Need more knowledge of text entry*

MedSpeak – Uses continuous recognition

- **User Characteristics:** user group-radiologists
- **Task:** Replace the existing method of radiologists using a tape or digital recording system to create a report
- **Context of Use:** Radiologists dictate in noisy rooms; can be more than one radiologist dictating in one room
- **Capabilities and Constraints**
  - Initial version of SR engine had a speaker independent model (i.e used merging and averaging of samples)
  - Use of more samples resulted in higher accuracy
- Difference in accuracy rates between read speech and spontaneous speech – increased error and user frustration
- Not robust to silence – recognizes mumble and pause

MedSpeak General Design Issues

- Narrowly defined set of users with known characteristics Less than 2 hours training as doctors are reluctant to use computers
- Should not be intimidating to first time users
- Default interface displays only basic function set – user decides when to move to advanced level of functionality
- Functions not organized as menus but accessible through large push buttons which can be activated by voice
- Also provided keyboard/mouse alternative to pushbuttons
- Even with buttons and functions users forget command to invoke the button-provide ‘what I can say’ command
- Gave users a sense of closure, increased satisfaction

MedSpeak SR Design Issues

- **Recognition failures**
  - Keep a history of recently recognized commands
- **Recognition errors**
  - Occurs less frequently than recognition failures because have longer commands
  - For most wrong recognized command undo possible, for destructive command user would confirm action.
- **Latency**
  - Not a major problem
- **Error correction**
  - Error due to word being out of vocabulary or due to mispronounciation – Solution dialogue box
  - Difference between error correction and mind change

MedSpeak SR Design Issues

- **Feedback of State**
  - User should know what mode he/she is in, dictation pause dictation or command. Achieved using color.
- **Eyes-Busy/Hands-Busy**
  - This constraint well supported by speech modality
  - Due to technology constraint, digit recognition, changing settings etc, had to be done using keyboard.
- **Enrollment**
  - Using system frequently gave accurate results especially for people with different accent

Lessons Learned

- Long way from building a prototype to usable version
- If accuracy is below a certain threshold, users are not interested in spending time
- Users that are disappointed the first time are difficult to motivate for a more sustained effort
- Oral composition a big factor (how clean and smooth the user speaks) – leads us to Natural Language Understanding

THE CONVERSATION MACHINE

- **User characteristics:**
  - Anyone who calls automated telephone banking service
- **Task:**
  - goal is to accomplish a transaction using speech
  - Has dialogue processing component, hence utterance need not be recognized word-for-word match.
  - Implications of what constitutes error and accuracy different.
  - Task involves transactions, should provide fail-safe method to recover from errors.
- **Context of Use:**
  - Acoustic context - background noise
  - Absence of visual display – feedback through auditory channel

Capabilities and Constraints

- Vocabulary limited for continuous recognition applications
- Limit on vocabulary depends on factors for e.g.
  - complexity of grammar
  - possibility of natural language/dialogue management component compensating for recognition errors
  - User, task, context of use
- Due to natural language processing error correction does not depend on recognition of unique utterances

Error Causes and recovery methods

Example	Cause of Error	Error Recovery
Mispronouncia-tion of a word	Speech misrecognition	Repeat or rephrase request
Surrounding noise	Background noise causes error	Repeat/rephrase in quiet environment
“what’s the damage going to be for my Visa bill?”	Use of phrase not in SR or natural language grammar	Rephrase request-”How much is my Visa bill?”
“How much was my electricity last month?”	System does not know information requested	Change goal

Summary and Conclusions

- Non technical factors (industry, negotiations etc) in making of a practical interface
- Targeted audience – makes a difference
- Human acceptance – less patience hence system should have less errors
- Whether an alternate approach exists- depends on application
- Personal factors i.e fluent speech, different accent etc (specific only to speech interface)
- Must include Natural Language Processing if want the interaction close to human-human