



NRC Publications Archive Archives des publications du CNRC

Speech-based interaction

Munteanu, Cosmin; Penn, Gerald

This publication could be one of several versions: author's original, accepted manuscript or the publisher's version. /
La version de cette publication peut être l'une des suivantes : la version prépublication de l'auteur, la version acceptée du manuscrit ou la version de l'éditeur.

Publisher's version / Version de l'éditeur:

Conference Proceedings - I/ITSEC Conference 2011, 2011

NRC Publications Record / Notice d'Archives des publications de CNRC:

<https://nrc-publications.canada.ca/eng/view/object/?id=0fc4042a-5865-474c-8ee4-5c2bbe1afa71>
<https://publications-cnrc.canada.ca/fra/voir/objet/?id=0fc4042a-5865-474c-8ee4-5c2bbe1afa71>

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at

<https://nrc-publications.canada.ca/eng/copyright>

READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

L'accès à ce site Web et l'utilisation de son contenu sont assujettis aux conditions présentées dans le site

<https://publications-cnrc.canada.ca/fra/droits>

LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D'UTILISER CE SITE WEB.

Questions? Contact the NRC Publications Archive team at

PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n'arrivez pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.





NRC-CNRC

*Institute for
Information Technology
e-Business*



Speech-based Interaction

Cosmin Munteanu

National Research Council Canada

`Cosmin.Munteanu@nrc-cnrc.gc.ca`

Gerald Penn

University of Toronto

`gpenn@cs.toronto.edu`



National Research
Council Canada

Conseil national
de recherches Canada

Canada

About the authors

- **Cosmin Munteanu**
 - Research Officer with the National Research Council Canada – Institute for Information Technology
 - Leads several research projects exploring speech and natural language interaction for mobile devices and mixed reality systems
 - Area of expertise: Automatic Speech Recognition and Human-Computer Interaction
- **Gerald Penn**
 - Associate Professor of Computer Science at the University of Toronto
 - Actively conducting research and publishing in Speech and Natural Language Processing
 - Area of expertise: Computational Linguistics, Speech Summarization, Parsing in Freer-Word-Order Languages

<http://www.cs.toronto.edu/~gpenn>

<http://nrc-ca.academia.edu/CosminMunteanu>

<http://iit-iti.nrc-cnrc.gc.ca/>

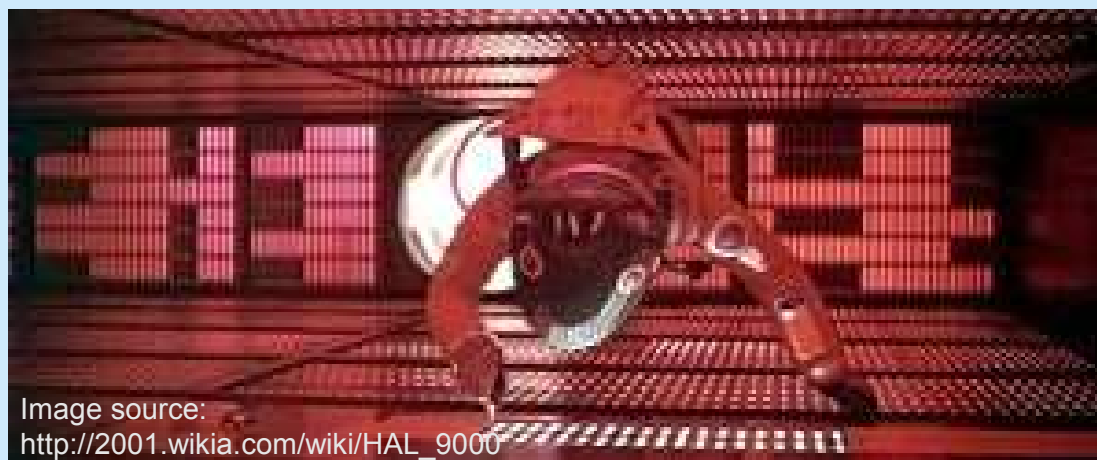
About the tutorial

- What you'll learn today
 - How does Automatic Speech Recognition (ASR) work and why is it such a computationally-difficult problem
 - What are the challenges in enabling speech as a modality for hands-free interaction
 - What are the differences between the commercial ASR systems' accuracy claims and the needs of mobile interactive applications
 - What do you need to enable speech in an interactive application
 - What are some usability issues surrounding speech-based interaction systems
 - What opportunities exist for researchers and developers in terms of enhancing systems' interactivity by enabling speech
 - What opportunities exist for HCI researchers in terms of enhancing systems' interactivity by enabling speech

In the future, we expected ...

The holy grail

True hands-free interaction



Instead, ...

We are still frustrated by the interaction with technology






Because,

the cold reality is slightly different ...

Why speech?

- Simply, it's the most natural form of communication:
 - Transparent to users
 - No practice necessary
 - Comfortable
- Fast
- Modality-independent
 - combines with other modalities

Why speech?

Mode	CPM	Reliability	Devices	Practice	Other tasks
Handwriting 	200-500	recognition errors	tabloid, scanner BIG	No (requires literacy)	hands and eyes busy
Typing 	200-1000	~ 100% (typos)	keyboard BIG	yes, if high bdwidth	hands and eyes busy
Speech 	1000-4000	recognition errors	micro SMALL	no	hands and eyes free

Still ... why is it difficult?

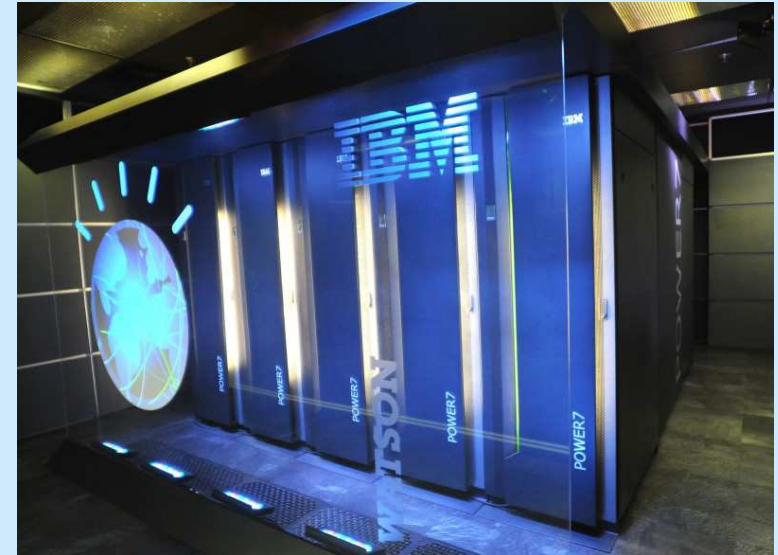
- **COMPLEXITY**
 - lots of data compared to text: typically 32000 bytes per second
 - tough classification problem: 50 phonemes, 5000 sounds, 100000 words
- **SEGMENTATION**
 - ... of phones, syllables, words, sentences
 - actually: no boundary markers, continuous flow of samples, e.g. “I scream” vs. “ice cream,” “I owe Iowa oil.”
- **VARIABILITY**
 - acoustic channel: different mic, different room, background noise
 - between speakers
 - within-speaker (e.g., respiratory illness)
- **AMBIGUITY**
 - homophones: “two” vs. “too”
 - semantics: “crispy rice cereal” vs. “crispy rice serial”

Is that a big deal?

- Don't we have super-powerful computers to deal with that complexity?
- We have – even competing on “Jeopardy!”



Images: IBM 2010, <http://www-03.ibm.com/press/us/en/>
Courtesy of International Business Machines Corporation.



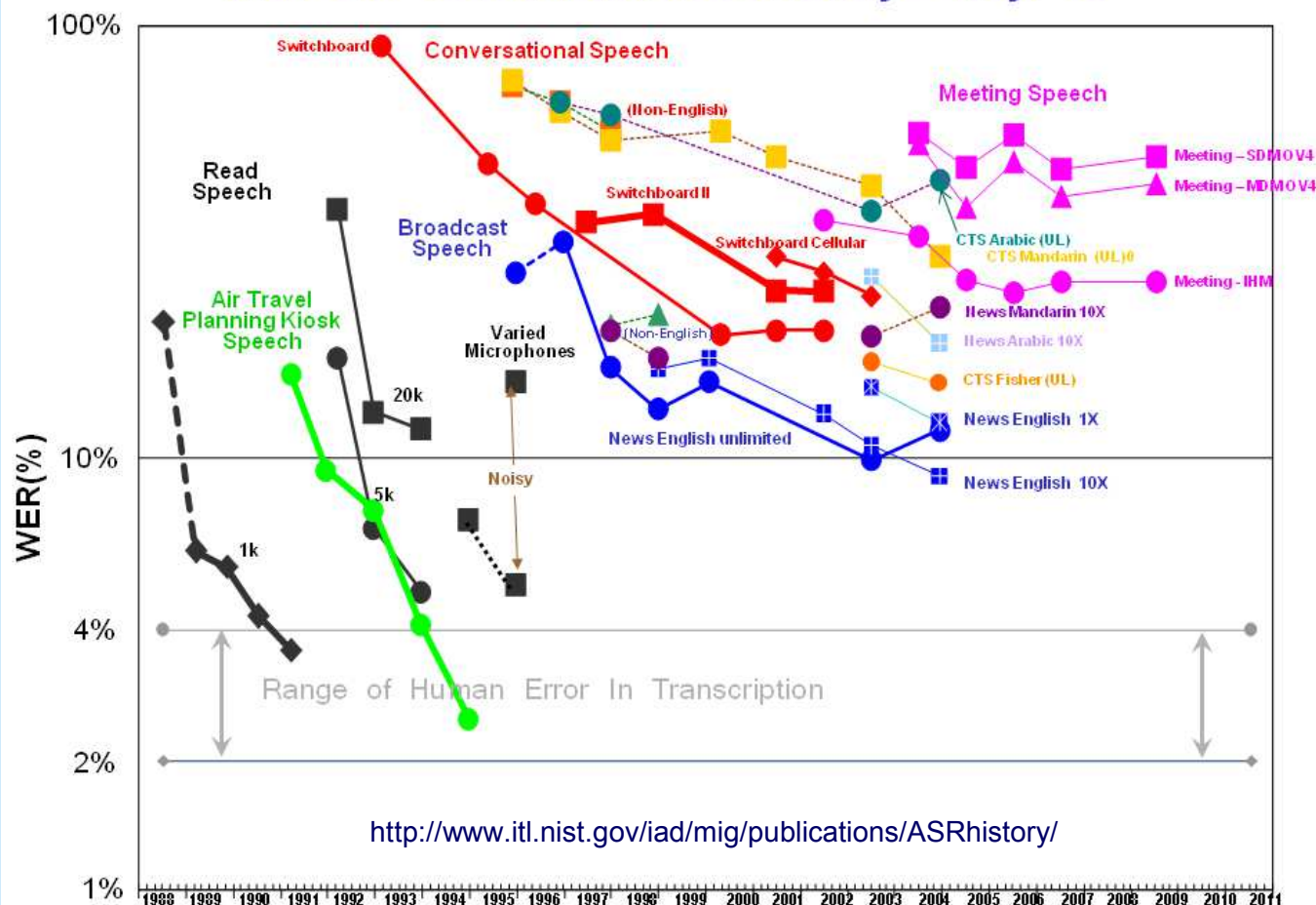
- But sadly, with no speech recognition.
 - Despite IBM having one of the world's leading ASR research programs

How accurate is it?

- For speech-to-text (automated transcription / dictation), the most common measure is WER (Word Error Rate)
 - The edit distance in words between ASR output and correct text
 - $WER = (\# \text{ substitutions} + \text{deletions} + \text{insertions}) / \text{sentence length}$
 - It is task-independent, based on 1-best output, and does not differentiate between types of words (e.g. keywords)
- Examples of WERs:
 - Isolated words (commands) < 1%
 - Read speech, small vocab. ~ 1-3%
 - Read speech, large vocab. (news) ~ 5-15%
 - Phone conversations (goal-oriented) ~ 15-20%
 - Lecture speech ~ 20-40%

Shouldn't we have solved it by now?

NIST STT Benchmark Test History – May. '09



We (sort of) did ...

- But only for controlled tasks and domains
 - e.g. broadcast news read off a teleprompter by trained professionals in optimal acoustic conditions
- For everything else, we need to work around, e.g.
 - “Shadow speakers” - professional speaker repeats parliamentary debates into expensive microphone in a sound booth as he listens
 - “Re-lecturing” - speech recognizer is evaluated on me giving this same lecture again next year
 - “Re-training” - speech recognizer is trained on me through a month-long iterative enrollment process

Still, we're trailing users' demands

There's more to ASR than simply dictating to a desktop computer!
e.g. how do we make critical interaction with technology more natural and more robust?



But we're on the right track ...

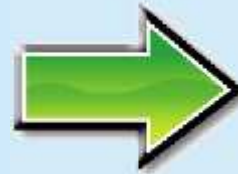
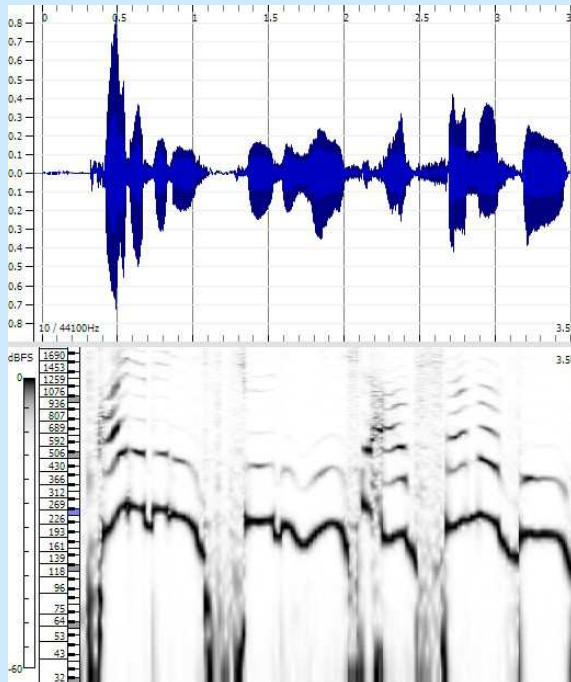
- Enhanced dialog systems
 - Face recognition, gesture interpretation (Microsoft / [Bohus '09])
- Speech-to-speech machine translation
 - Real-time lecture translation (CMU)
- Speech summarization
 - Audio or textual summaries of spoken documents [Zhu '07, '09]
- Speech indexing
 - Improved textual search in spoken documents [Kazemian '09]
- Speaker verification
 - Secure authentication through speech (field overview in [Bimbot '04])
- All these employ not only ASR, but significantly more Natural Language Processing, and a good amount of Human-Computer Interaction – not all are dedicated to speech-based input!

Automatic Speech Recognition

- *What is it?*
- *How does it work?*
- *When does it work?*
- *How good is it?*
- *How good is good enough?*

What is ASR?

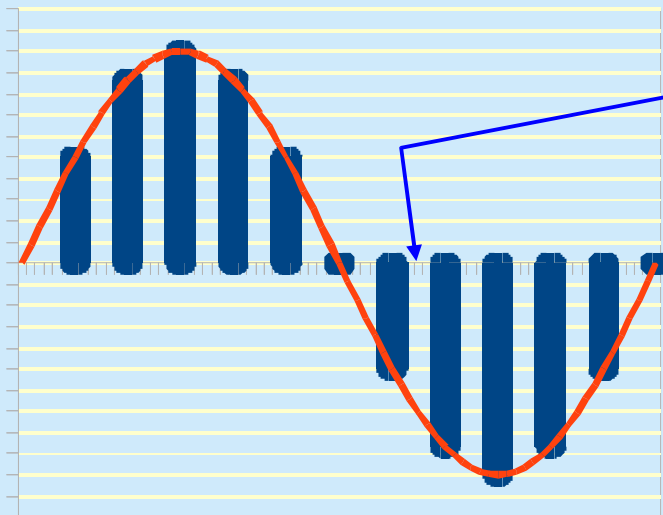
- Textbook definition: a speech recognizer is a device that automatically transcribes speech into text [Jelinek, 1997]



Some text
of what I
supposedly
said

How ASR works

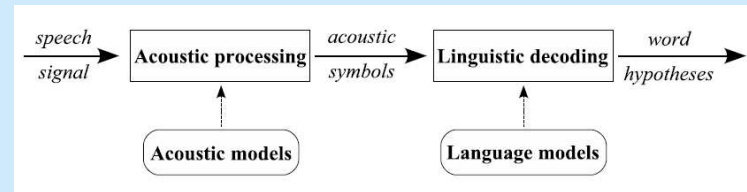
- Step 1: sample and digitize speech signal – convert the analog speech waveform into a digital representation



Sample rate: how often we “take” a sample (measure) from the analog signal

Sample size: on how many bits we can represent the analog value of the sample (how many “digital levels” we have for approximating the analog values)

How ASR works



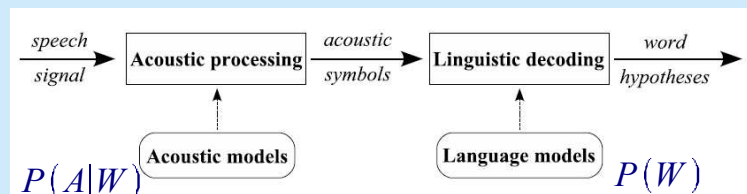
- Step 2: perform spectral analysis, estimate acoustic features from spectral analysis, convert into acoustic symbols

$$A = a_1, a_2, \dots, a_m$$

- Step 3: find the text (word sequence) most probable to have been spoken given the observed sequence of symbols A

$$\hat{W} = \underset{W}{\operatorname{argmax}} P(W|A)$$

How ASR works



Bayes:

$$P(W|A) = \frac{P(W)P(A|W)}{P(A)}$$

A is fixed, so:

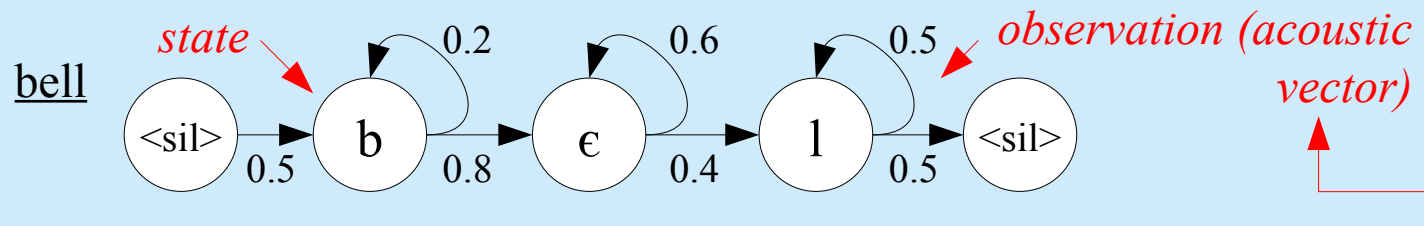
$$\hat{W} = \underset{W}{\operatorname{argmax}} P(W)P(A|W)$$

For large-vocabulary, continuous speech, speaker-independent speech: $P(A|W), P(W)$ – very large probability spaces!

How ASR works

Acoustic model (AM) – $P(A|W)$

- Model the way a word or phone sequence is pronounced
- Trained on transcribed speech
- Simplistically, each word has a Hidden Markov Model (HMM)



- HMMs are usually even more fine-grained – built for phone (basic sound unit) sequences

How ASR works

Language model (LM) – $P(W)$

- Model the way phrases are formed
 - Can be as simple as the probabilities of each word in the dictionary
 - Usually model the way words are sequenced one after another
- Trained on large collections of texts
- Most ASR systems use N-gram models ($N = 2, 3, \text{ or } 4$)
 - e.g. $P(\text{cereal} \mid \text{crispy, rice}) = 0.12$
 - $P(\text{serial} \mid \text{crispy, rice}) = 0.01$

How ASR works

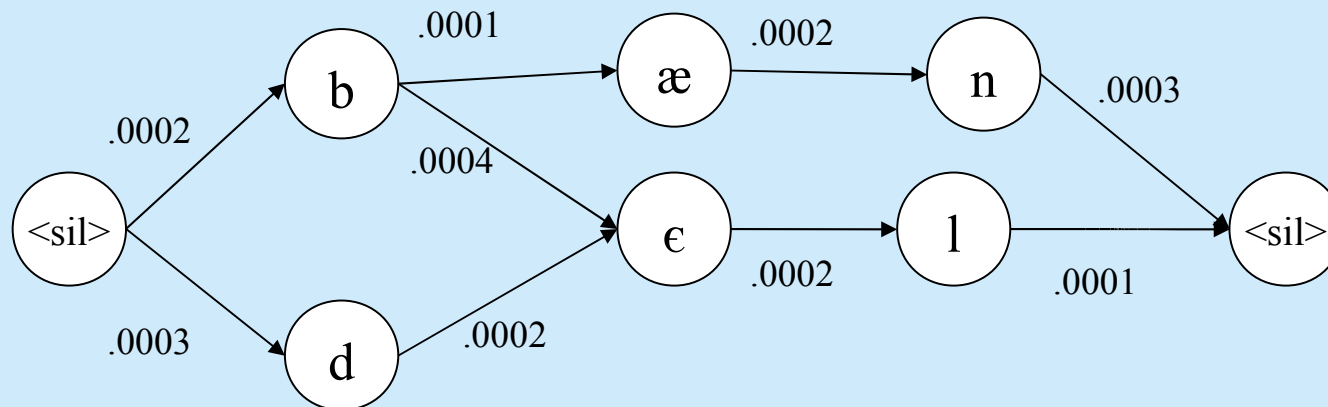
Decoding

- This is the “guessing” stage of the ASR process
- Question: given an observation sequence (of acoustic symbols), what is the most likely path of (hidden) states that produced the sequence?
 - The path represents linguistic units (e.g. phonemes, words)
 - The likelihood is computed given the probabilities captured by both the AM and the LM
 - Like guessing the recipe while tasting food, blind-folded!
- Various algorithms: Dynamic Time Warping, Viterbi search, etc.
- This is a computationally-intensive optimization problem

How ASR works

Decoding – Viterbi search

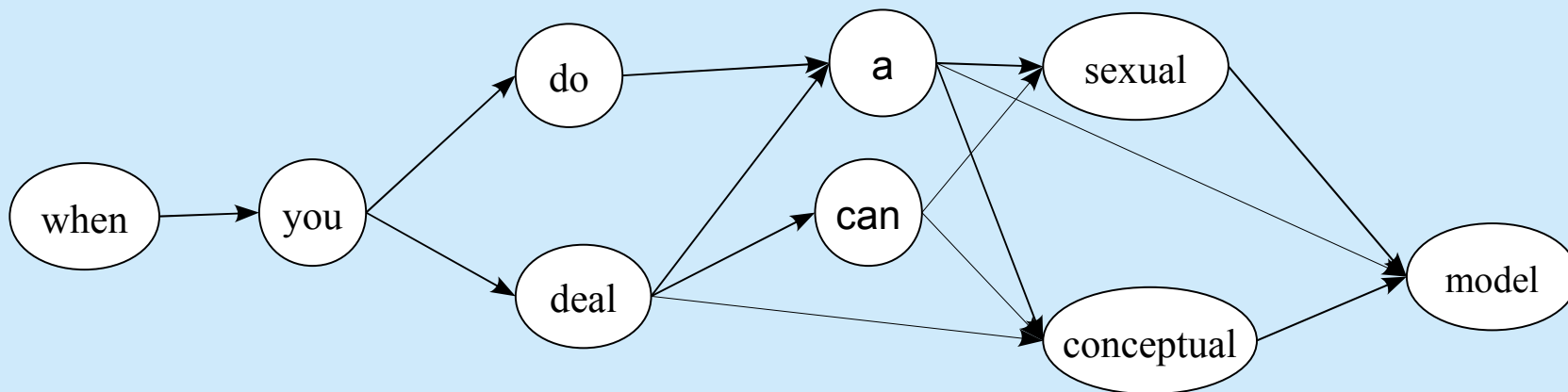
- Dynamic programming to find the most likely path through the search space
- Constructs a lattice (or trellis) of phones and/or words
- The ASR output is the 1-best path in the lattice



ASR output

- The best path is not always correct
- Having access to the (trimmed) lattice / n-best list before the output can be **very** useful!

-2156.45	when you deal can sexual model
-2178.31	when you do a sexual model
-2356.23	when you deal conceptual model
-2389.41	when you do a conceptual model
-2902.92	when you deal a model



What's needed (to make it work)

- Data, data, and more data – the LM and AM need to be trained!
- Requirements (and source of problems):
 - AM: need ~ 100 hours of diverse speakers recorded in acoustic conditions similar to the domain of the application
 - Speaker: dependent vs. independent, read vs. unconstrained
 - Acoustic: quiet vs. noisy, microphone type
 - LM: need large collection of texts that are similar to the domain of the application: vocabulary, speaking style, word patterns, ...
 - Vocabulary: large vs. small, topic-specific vs. general
 - Speaking style and word patterns: variations across genres and across speakers
- Under controlled acoustic conditions, the LM needs to be “just right” (no overfitting, no overgeneralization) – hard to achieve for unconstrained tasks!!

Factors affecting ASR quality

- **Word Error Rate (WER)** increases by a **factor of 1.5** for each unfavourable condition
 - Accented speaker (if ASR is speaker-independent)
 - Temporary medical conditions (if ASR is speaker-dependent)
 - Noise, esp. if different than that of the training data
 - Variations in the vocabulary, genre, and style of the target domain
 - And a variety of others at acoustic level (e.g. microphone change, physical stress) or language level (e.g. psychological stress, such as giving a lecture, training in a simulator, banking over the cellphone on the street)

How good does it have to be?

- User study: information-seeking tasks on archived lectures
- Typical webcast use – responding to a quiz about the content of a lecture
 - Factoid questions, some of which appear on slides, some of which are only spoken by instructor
 - Within-subject design: 48 participants (undergrad students, various disciplines, 26/22 females/males)

[Munteanu et al., CHI '06]

The screenshot shows a webcast player. At the top left is a logo with a green 'e' and the text 'ePresence Interactive Media'. Below it is a video window showing a lecturer. To the right of the video is a 'Play/Pause' button. Below the video is a table of contents with 20 items. Item 15, 'Mental Models: Prototyping Tools', is highlighted. At the bottom is a progress bar with a 'Select Chapter:' dropdown and a 'Select Slide:' dropdown.

1.	Conceptual Design
2.	Scenarios in Conceptual Design
3.	CD: Effectiveness
4.	CD: Comprehensibility
5.	CD: Satisfaction
6.	Cognition
7.	Learning
8.	Memory
9.	Magic Number 7, + or - 2
10.	The Sequences of Numbers
11.	Uses of Cognitive Theory in Design
12.	Reasoning and Problem Solving
13.	Mental Models
14.	Mental Models: Drawing Tools
15.	Mental Models: Prototyping Tools
16.	Kinds of Models
17.	Models (1)
18.	Models (2)
19.	Models (3)
20.	User's Model Does Not Match

Mental Models

- Definition of mental models (Carroll, 1984):
 - "...structures and processes imputed to a person's mind in order to account for that person's behaviour and experience."
- More generally (Carroll & Olson, 1988):
 - "...all of what a user knows about using a particular piece of software, including *how to use it*, and how it works."
- Mental models allow a user:
 - to understand a system
 - to predict effects of actions
 - to interpret the results
- Role of mental model: to answer questions like:
 - What is X?
 - What happens when you do Y?
 - Why is happening when I see Z?

Design of Interactive Computational Media ©1992-2004, Ronald M. Baecker Fall 2004 Slide 5.16

of of metaphor
so i start
now entity ban an all models
kahn
their couple of
fairly abstract f. issues the mental models
on jack hero back in nineteen eighty four
sad
structure sin process c.'s imputed for person's my
he nor door top for that person's be here
an experts
ok what is
real wet this reminds is of
is the we real we really and some level have now half so notion of what seen
peoples minds
am it our first a look into some
some these nine
radio in crashes distinctly can report army
yes i understand that yes i can see
that
this desktop is organized in a guy files and folders on the
nineteen track something into the trash in of i empty the trash

How good does it have to be?

- Measures:
 - Task performance data
 - Indicators of user perception data
- Results:
 - In general, transcripts are useful if WER is approx. 25% or less (compared to having no transcripts at all)
 - For some tasks (e.g. questions that are not on the slides), there is even a (slight) improvement for WER of 45%
 - Users would rather have transcripts with errors than no transcripts
- This is an example of an ecologically valid use of transcripts among users with no accessibility challenges - no one reads them verbatim, but they either scan them or search through them with a web browser to perform some other task



Speech-based interaction

- *What applications use ASR?*
- *What do you need to enable speech?*
- *What should you pay attention to?*
- *How do users crash it?*
- *What can you do with speech beside transcribe it?*

Speech-based interfaces

- Examples of typical commercial ASR applications
 - Interactive Voice Response (IVR) systems
 - Call routing (customer service, directory assistance)
 - Simple phone-based tasks (customer support, traffic info, reservations, weather, etc.)
 - Desktop-based dictation
 - Home/office use
 - Transcription in specific domains: legal, medical
 - Assistive technology
 - Automated captions
 - Interacting with the desktop / operating system
 - Language tutoring
 - Gaming
- Ideally – ASR is enhancing, not replacing, existing interactions

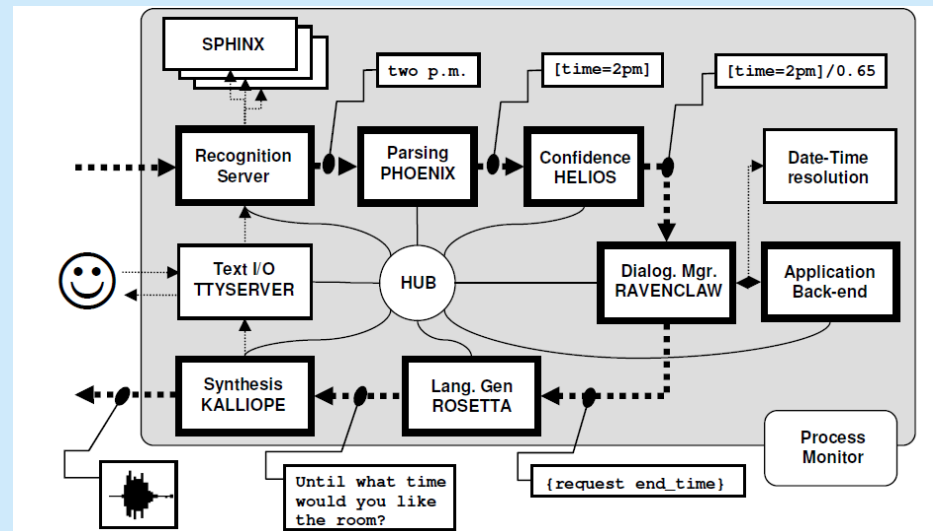
Speech-based interactive systems

The ASR system
can contribute to /
control various
aspects of
human interaction
with technology
and/or information



Example – dialog systems

- A common example of a speech-based interactive system
- Goal oriented: users interact with a system by voice to achieve a specific outcome (typically: info request, reservation, etc.)
- Usual modules:
 - ASR
 - Keyword / named entity extraction
 - Dialog manager
 - Application back-end
 - Nat. language generation
 - Text-to-speech



CMU's Olympus Dialog Manager [Bohus '07, HLT]

Example – dialog systems

- To ensure successful completion of task:
 - LM is limited to the domain (e.g. typical words used to reserve hotel rooms)
 - AM is specific to the channel (e.g. phone)
 - AM can be adapted to the speaker if recurrent calls (e.g. telebanking)
 - System has lots of error-correction strategies
 - Modelling of user behaviour
 - The interaction is (often) controlled to reduce vocabulary and language complexity
 - System initiative (prompts)
 - User initiative (no prompts)
 - Mixed (system leads, but user can interrupt)

A handyman's guide to building speech interfaces

- (ASR-related) steps to building a speech interface

1. Define the domain & genre → Vocabulary, LM

2. Get to know the users' voices → AM


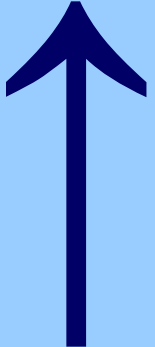
3. Define the interaction types → Dialog manager



1. Design the interaction

Choose / Build the ASR

ASR choices

Source	Choice	Example	Gain	OOTB
Commercial { Research	Off-the-shelf	Dragon, Microsoft SAPI		
	Enterprise grade	Vocon, Phonix, Lumenvox		
	Customizable system (enterprise / bundled)	Lumenvox, Sonic		
	Bundled (Recognizer + toolkit)	Sonic, Sphinx		
	Toolkit – build from scratch	HTK		

Gain : ASR performance as function of engineering effort
 OOTB: Out-of-the-box performance

Commercial ASR choices

- Off-the-shelf ASR
 - E.g. Dragon
 - Adequate out-of-the-box ASR
 - Easy development
 - No control/customization of the ASR
- Enterprise-grade
 - E.g. Nuance's Vocon, Voiceln's Phonix, Lumenvox's SDK (Microsoft SAPI is somewhere in between)
 - Good for large-scale projects: good SDK, integration with apps
 - Good WER for most tasks that are well constrained
 - Some control over the ASR (mostly vocabulary, maybe grammar to manually specify phrase patterns)

Research ASR choices

- Research-grade ASR system
 - E.g. U of Colorado's Sonic, CMU's Sphinx
 - Mostly toolkits for building an ASR, but come with prepackaged AM and LM good for some limited tasks
 - Good to get started; more control than commercial ASR
 - Out-of-the-box accuracy may be lower than commercial systems', but can be improved
 - AM suitable for most tasks, can be adapted if some transcripts for the speaker and/or application's domain exist
 - LM usually needs adaptation or completely built from scratch using toolkits (e.g. SRI, CMU) – not that hard! [Munteanu '07, Interspeech]
 - Access to word and/or phone lattices on the output side

ASR toolkits choices

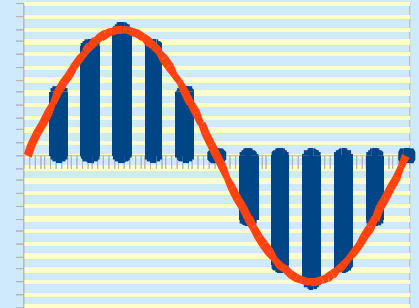
- ASR toolkits – “build-your-own”
 - Best control over the ASR
 - Can be custom built for a domain and/or types of speakers (topic, genre, speaker)
 - Doesn't work “out-of-the-box”, needs dedicated ASR engineering:
 - Everything needs to be built almost “from scratch”
 - Most difficult: building the AM (~ 100 hrs of transcribed speech)
 - Likely requires programming (C/C++/Java/...) for integration with other components of the interactive system

Critical factors

- ASR can be seriously affected by external factors
 - Acoustics (e.g. noise on the street)
 - CPU power (client-server vs. on-device ASR)
- When designing a spoken interactive system:
 - Know what is against you (environment, channel, etc.)
 - Know the domain (can improve accuracy by limiting the vocabulary and phrases)
 - Know the users!
 - Speakers: single vs. few vs. many
 - Speech: continuous vs. prompted vs. mixed
 - Level of stress: physical (walking), psychological (driving)
 - Can you “model” them? (constraints → task, goal, discourse, ...)

Critical factors

- Digitization constraints also affect ASR:
 - Sampling (analog-to-digital conversion)
 - Ideally – use a good sample rate / size
(20 KHz / 16 bit)
 - Do not change sample rates / sizes between recording and AM!
 - Codecs (lossy formats, compression, non-linear representation)
 - Use lossless compression (e.g. flac codec or zip) if low bandwidth
 - Ideally use only uncompressed formats (wav or raw)!
 - If using mp3, have AMs for mp3!
 - Do not switch between formats (never mp3 with AMs built for wav)
 - Transmission over networks (packet loss, etc.)










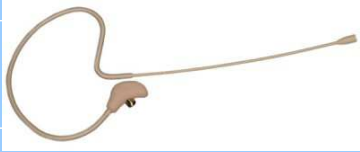



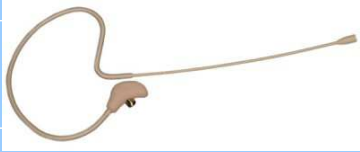

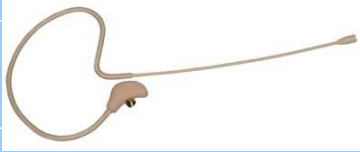

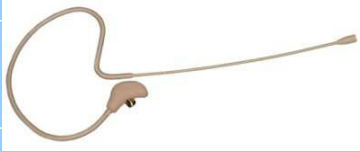

Critical factors

- Lack of complementary modalities
 - Gestures can help disambiguate ASR errors [Oviatt '03]), even if gesture recognition is in itself error-prone
 - Other actions by users can be further used to disambiguate, compensate for, or override ASR errors (e.g. tablet-based controls for instructors – NRC's IRET project)



Critical factors

- Microphone choice significantly affects the ASR quality

Source	Choice	Example		ASR
Consumer	Handheld (*)			
	Desktop (e.g. webcam)			
	Bluetooth			
	Headset (e.g. USB)			
Professional	Lectern / gooseneck			
	Lapel			
	Headworn - omnidirectional			
	Headworn - hypercardioid			

Microphones (cont'd)

- Application-specific trade-off (human factors, interaction type, etc.)
- In general, the optimal choice is:
 - Hypercardioid (strongly directional)
 - Fixed position in relation to mouth
 - Wind insulated
 - Good sound-to-noise ratio
- Other features to be considered:
 - Personal vs. area microphones (e.g. for meetings)
 - Availability of power supplies (dynamic vs. condenser)
 - Digitization (e.g. quality of sound mixer)



© 2007-2011 AKG ACOUSTICS GMBH

Most important: users

- Pushing the ASR boundaries is good, but we should never forget the users
 - ASR on its own will not solve all problems!
 - ASR errors and/or bad interactions can frustrate users and can lead to tasks not being completed!
- Example: significant commercial development for IVR systems is driven by the desire (and well-justified need!) to replace this type of human-human interaction ...

To avoid such errors in customer service, human operators are often replaced with automated systems (e.g. IVR), since machines are “smarter”, and of course, never wrong ...

Automated agents: an apology

- Telephone-based speech systems (IVR, phone reservations, automated enquiries, etc.) were all the rage 25 years ago
 - The envisioned end-appliance was the telephone
 - It was the only bi-directional personal communication device widely available
 - Privacy was not a (major) issue
- We've learned a lot - systems such as AT&T's successfully handled millions of calls
 - Significant ASR and usability improvements – see all research on dialogue systems and user modelling
 - Error correction (but needs to be used carefully – nobody wants to hear “I'm sorry, I didn't understand you” too many times!)
 - Goal orientation (system-initiated prompts or mixed initiative)
 - Informing the user of their progress toward achieving the goal

Human-Computer Interaction and ASR

- HCI needs to be aware of ASR's capabilities and limitations (and the other way around)
- One successful approach – human-in-the-loop
- Example

- Wiki-like corrections of webcasts lecture transcripts
- ASR improves based on user corrections

[Munteanu et al.,
CHI '08, ACL '09]

The screenshot shows a presentation slide titled "Conceptual Design" with a blue background. The slide content includes:

- Producing a conceptual model for an envisioned system
 - What it should do, i.e., functionality
 - What it should look like, i.e., appearance
 - How it should behave, i.e., (interactive) behaviour
- The role of scenarios
- Goal is to make activities with the system
 - Effective
 - Comprehensible
 - Satisfying

Below the slide, there is a video player showing a person at a computer. To the left of the video is a table of contents with 15 items, including "Conceptual Design", "Scenarios in Conceptual Design", and "Mental Models".

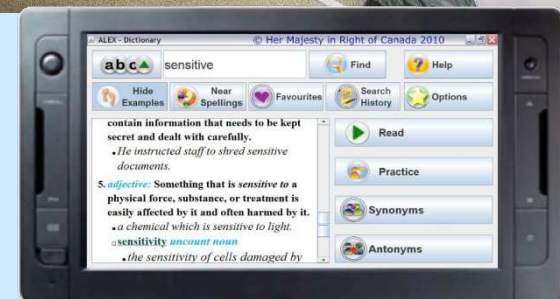
An "Edit Transcript" window is open in the foreground, showing a text input field with the text "what is financing laws the do". Below the input field is a list of suggested words: "why", "new", "with", "where", "one", "we", "what it". The window also has "Play", "Save", and "Cancel" buttons.

Focus: users

- Do not use speech just because it is possible
 - There should be a good reason why you need speech
 - Speech is not the answer to everything, sometimes it is not beneficial even if we think it's natural
- Integrated/holistic system design: human factors + ASR
- Not everything is desktop-based dictation or spoken commands
 - ASR is needed in many other areas
 - Display on a mobile device a text summary of a recorded lecture when listening to the entire lecture is not possible, or
 - Use text-based search to locate something in a large collection of recorded video documentaries
 - Interact with a training simulator (aviation, military, etc.) that replicates real-life scenarios

Use speech where needed ...

- Examples of applied research:
 - Speech-based input when hands are busy (e.g. NRC-IIT Project on ASR for fishing boats)
 - Mixed-reality interaction for training simulators (e.g. “Immersive Reflexive Engagement Trainer” – IRET Project at NRC-IIT)
 - Mobile language learning [Munteanu '10, CHI, MobileHCI]



Summary: final advice to system designers

- Moral of the story – what we've learned:
 - ASR is difficult, but we can still benefit from it
 - We don't always need 100% accuracy
 - We need to look beyond 1-best output (lattices)
 - For a good ASR-powered interactive system we need:
 - Ability to control/customize (at least the LM, ideally the AM) – various choices, each with advantages/disadvantages
 - Knowledge of what's against us – can't always go around it, but at least we can try to not make it worse ourselves
 - Knowledge of the domain / application / topic / genre / speakers
 - To never forget the user!

Summary: suggested design / decision steps

1. Define the users and the domain – Interaction modes, ASR resources
2. Choose the audio hardware – microphone choices and usage
3. Evaluate needs, environment, and users:
 - 1) Choose the architecture (on-device vs. client-server, wearable computing vs. recording speech remotely)
 - 2) Choose the ASR system – customization needs, environment
 - 3) Define ASR restrictions – language, acoustic, dialog
 - 4) Design the ASR connection to the main application
4. Design the interactive interface – multimodality
5. Repeat steps as necessary

Future ...

- Potential for future research and applications
 - Designing meaningful evaluation protocols (in both "camps", ASR and HCI)
 - Finding relevant metrics that capture users' needs together with the ASR & NLP performance
 - Interaction design with ASR low-performance in mind
 - We're moving away from desktop-based systems
 - Access to usable information should not rely on powerful systems
 - Interactive access to naturally-generated information
 - We still haven't solved the problem of searching through audio/video archives – and YouTube is growing fast!
 - Plus many more, including how to do all these on mobile devices!



NRC CNRC

*Institute for
Information Technology
e-Business*

Thank you!

Science
— at work for —
Canada



National Research
Council Canada

Conseil national
de recherches Canada

Canada