Speech-based Interaction:
Myths, Challenges, and Opportunities

Cosmin Munteanu
Institute of Communication, Culture, Information, and Technology
University of Toronto Mississauga
Cosmin.Munteanu@utoronto.ca

Gerald Penn
Dept. of Computer Science, University of Toronto
ICSI, UC Berkeley
gpenn@cs.toronto.edu

Speech-based Interaction:
Myths, Challenges, and Opportunities

Cosmin Munteanu
Institute of Communication, Culture, Information, and Technology
University of Toronto Mississauga
Cosmin.Munteanu@utoronto.ca

Gerald Penn
Dept. of Computer Science, University of Toronto
ICSI, UC Berkeley
gpenn@cs.toronto.edu

About the authors

- Cosmin Munteanu
  - Assistant Professor at the Institute for Communication, Culture, Information, and Technology (University of Toronto at Mississauga)
  - Associate Director of the Technologies for Ageing Gracefully lab, Computer Science Department
  - Research on speech and natural language interaction for mobile devices, mixed reality systems, and assistive technologies
  - Area of expertise: Automatic Speech Recognition and Human-Computer Interaction

http://cosmin.taglab.ca

- Gerald Penn
  - Professor of Computer Science at the University of Toronto and Research Scientist at ICSI, University of California, Berkeley
  - 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

About the tutorial

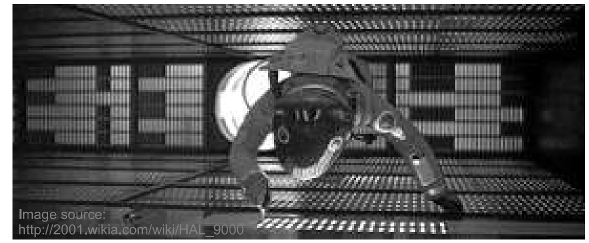
- What you'll learn today
  - How does Automatic Speech Recognition (ASR) work and why is it such a computationally-difficult problem?
  - What are the challenges in enabling speech as a modality for hands-free interaction?
  - What are the differences between the commercial ASR systems' accuracy claims and the needs of 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 Human-Computer Interaction (HCI) researchers in terms of enhancing systems' interactivity by enabling speech?

In the future ...

we were promised that we'll interact naturally with technology ...

The holy grail

True hands-free interaction



We (sort of) made it ...

But not quite

- We are still frustrated by the interaction with technology
  - Luckily some are going away (think voice-response customer service)
- We're still obsessing with using speech in the most unnatural ways, clinging to what was “space-age” a long time ago
- Often with disappointing outcomes ...

Often just saving face ...

Why speech?

- Simply, it's the most natural form of communication:
  - Transparent to users
  - No practice necessary
  - Comfortable
- Fast
- Modality-independent
  - Can be combined 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!”

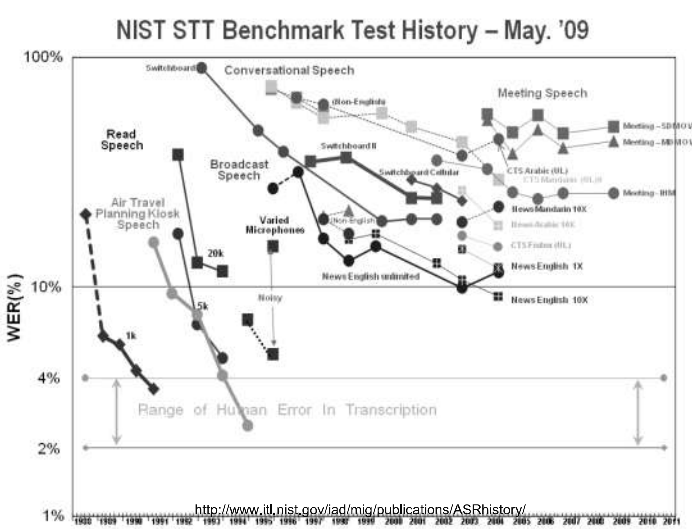
Images: IBM 2010, http://www-03.ibm.com/press/us/en/
Courtesy of International Business Machines Corporation.

- 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 = (# substitutions+deletions+insertions) / 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%
  - Youtube ~ 50% (still, as of 2014)

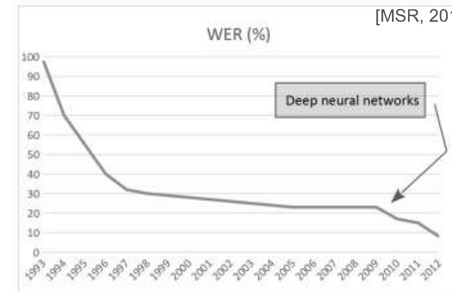
Shouldn't we have solved it by now?



We (sort of) did …

- But mostly for controlled tasks and domains
  - e.g., broadcast news read off a teleprompter by trained professionals in optimal acoustic conditions
- New methods based on Deep Neural Networks (Hinton, 2012) and using very large training data show promising results
  - Although still focused on improving word-level accuracies under controlled conditions ...

[MSR, 2016]



We (sort of) did …

- But mostly 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
- New methods based on Deep Neural Networks (Hinton, 2012) and using very large training data show promising results
  - Although still focused on improving word-level accuracies ...

Still, we're trailing users' demands

There's more to ASR than simply dictating to a desktop computer!

- How do we make critical interaction with technology more natural and more robust?
- How do we help users of mobile devices find info contained in the audio track of a large multimedia repository?

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]
- Speech-based personal organizers (e.g. Siri)
  - 10+ years of research in Artificial Intelligence at SRI International, initially under DARPA's program to develop a "Perceptive Assistant that Learns"

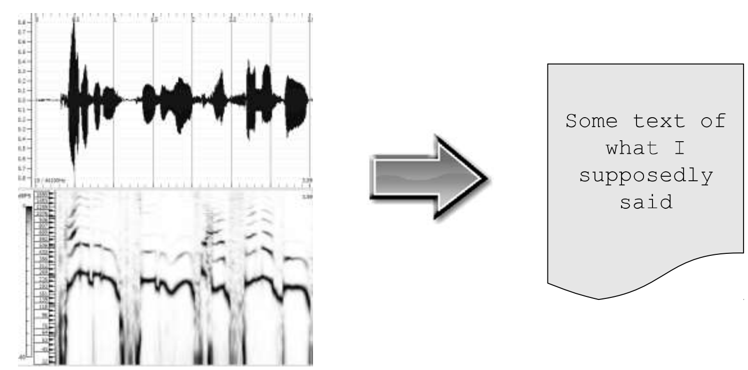
- All these employ not only ASR, but significantly more Natural Language Processing, and a good amount of Human-Computer Interaction – not all are dedicated to speech-based input!

Automatic Speech Recognition

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

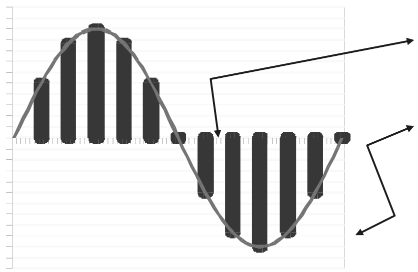
What is ASR?

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



How ASR works

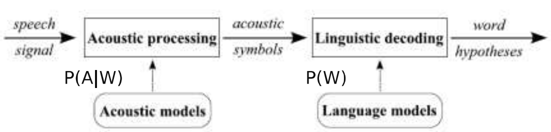
- Step 1: sample and digitize speech signal – convert the analog speech waveform into a digital representation



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)

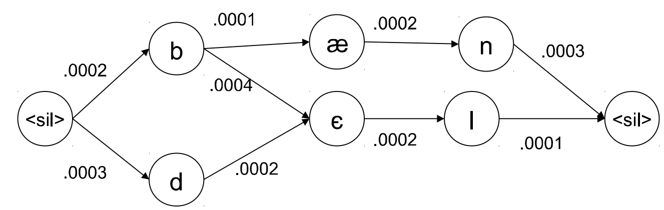
How ASR works



- Find the text (word sequence) most probable to have been spoken given the observed sequence of acoustic symbols that are derived from the speech signal $\widehat{W} = \underset{W}{argmax}\, P(W) \cdot P(A|W)$
- Acoustic model (AM) – state sequences / probability distributions (Hidden Markov) that model the way a word is pronounced
- Language model (LM) – model the way phrases are formed
  - Most ASR systems use N-gram models (N = 2, 3, or 4)
    e.g., P(cereal | crispy, rice) = 0.12
    P(serial | crispy, rice) = 0.01

How ASR works

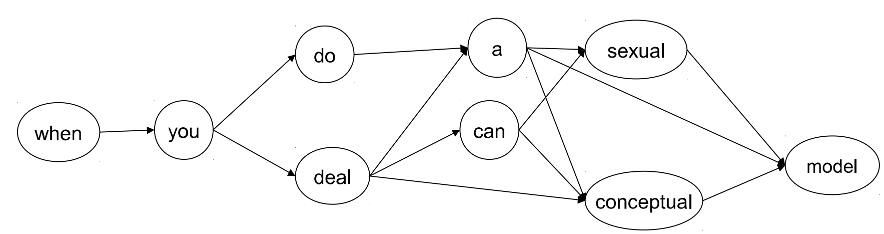
Decoding
- This is the “guessing” stage of the ASR process
- Question: given an observation sequence (of acoustic symbols), what is the most likely path of (hidden) states that produced the sequence?
- Viterbi – find the most likely path through the search space
  - Constructs a lattice (or trellis) of phones and/or words
  - The ASR output is the 1-best path in the lattice



ASR output

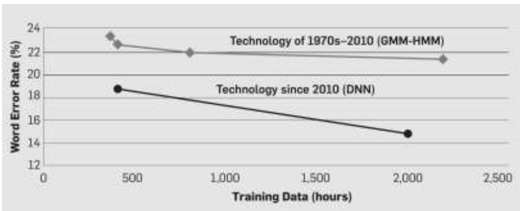
- This is a computationally-intensive optimization problem
- The best path is not always correct
- Having access to the (trimmed) lattice / n-best list before the output can be very useful!

-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
    - ~ 400 hours needed for Deep Neural Networks



[Huang, Baker, Reddy, 2014]

What's needed (to make it work)

- 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!
  - Often a source of errors and frustrations for the users!

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)

Factors affecting ASR quality

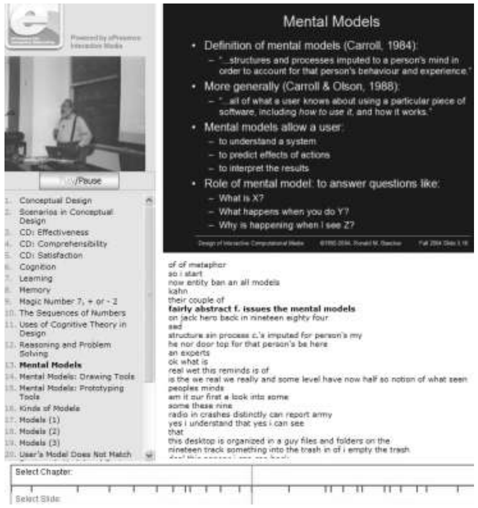
"today's speech recognition systems still degrade catastrophically even when the deviations are small in the sense the human listener exhibits little or no difficulty" [Huang, 2014]

The most critical issue
affecting the interaction!
(and the most ignored by UX designers)

How good does it have to be?

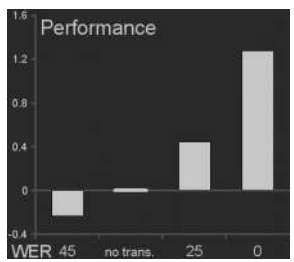
- User study: information-seeking tasks on archived lectures
- Typical webcast use – responding to a quiz about the content of a lecture
  - Factoid questions, some of which appear on slides, some of which are only spoken by instructor
  - Within-subject design: 48 participants (undergrad students, various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

[Munteanu et al., CHI '06]



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
  - **Most thought that the 0% WER condition was also machine-generated!**
- This is an ecologically valid use of transcripts - no one reads them verbatim, but uses them as navigational aids



Good enough doesn't always help

- When UX designers ignore that whole 1.5 factor and catastrophic degradation ...

Good enough doesn't always help

ASR in the wild

- EXERCISE 1, part 1

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 ...

There's more to speech than dictation

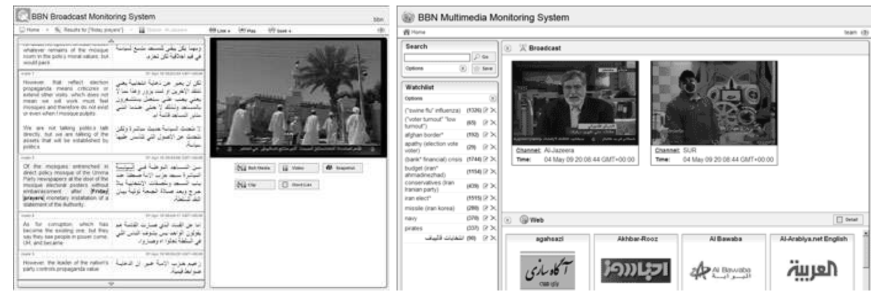
- Google News Indexer

There's more to speech than dictation

- OCADU / U of Toronto – CBC Newsworld Holodeck

There's more to speech than dictation

- BBN (Raytheon) Multilingual Audio Indexing



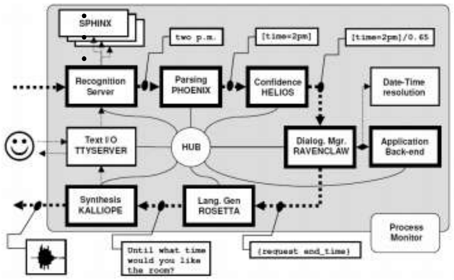
Speech-based interactive systems

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

Example – dialogue 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
  - Dialogue 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
  - User behaviour is modelled
  - 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

Define the domain & genre → Vocabulary, LM
Get to know the users' voices → AM
Define the interaction types → Dialog manager

↓↓ ↓↓

Design the interaction   Choose / Build the ASR

ASR choices

Source	Choice	Example	Gain	OOTB
Commercial	Off-the-shelf	Dragon, Microsoft SAPI		
Commercial	Enterprise grade	Vocon, Phonix, Lumenvox	-	+
Commercial / Research	Customizable system (enterprise / bundled)	Lumenvox, Sonic		
Research	Bundled (Recognizer + toolkit)	Sonic, Sphinx	+	-
Research	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, VoiceIn's Phonix, Lumenvox's SDK, Microsoft SAPI, Google android.speech
  - 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., CMU's Sphinx and PocketSphinx, Karlsruhe's Janus
  - Mostly toolkits for building an ASR, but come with prepackaged AM and LM good for some limited tasks (or easy-to-train AM/LMs)
  - 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

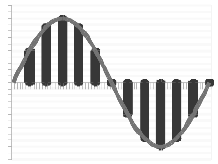
ASR toolkits choices

- ASR toolkits – "build-your-own"
  - E.g. Johns Hopkins' Kaldi, Cambridge's HTK
  - 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
  - Example: tablet-based controls for instructors

NRC's MINT simulator for public safety training

Critical factors

- Microphone choice significantly affects the ASR quality

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

Microphones (cont'd)

- Application-specific trade-off (human factors, interaction type, etc.)
- In general, the optimal choice is:
  - Hypercardiod (strongly directional)
  - Fixed position in relation to mouth
  - Wind insulated
  - Good sound-to-noise ratio

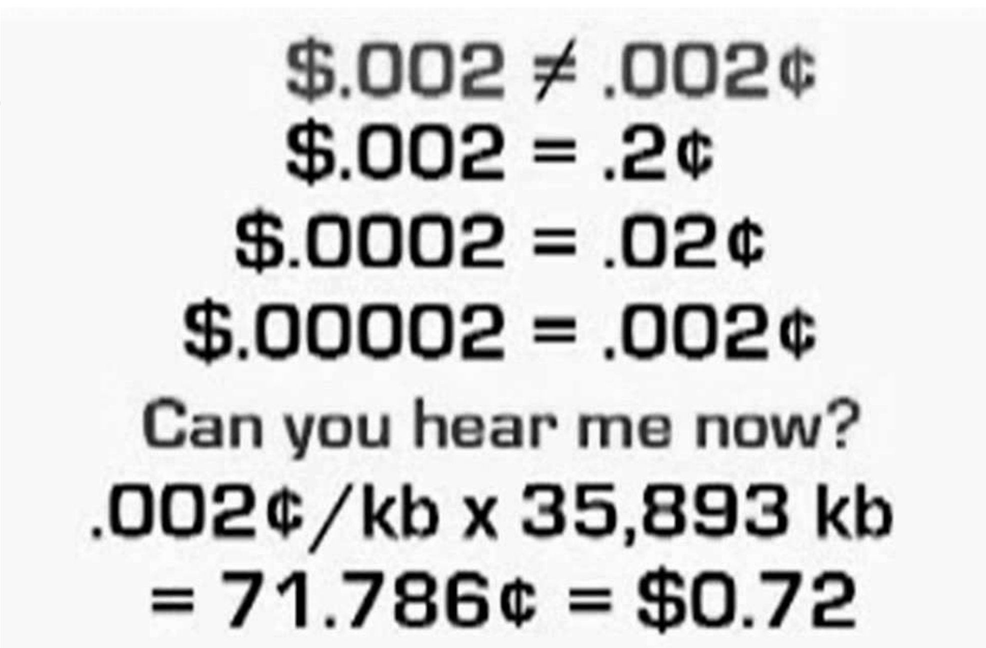
© 2007-2011 AKG ACOUSTICS GMBH

- 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)

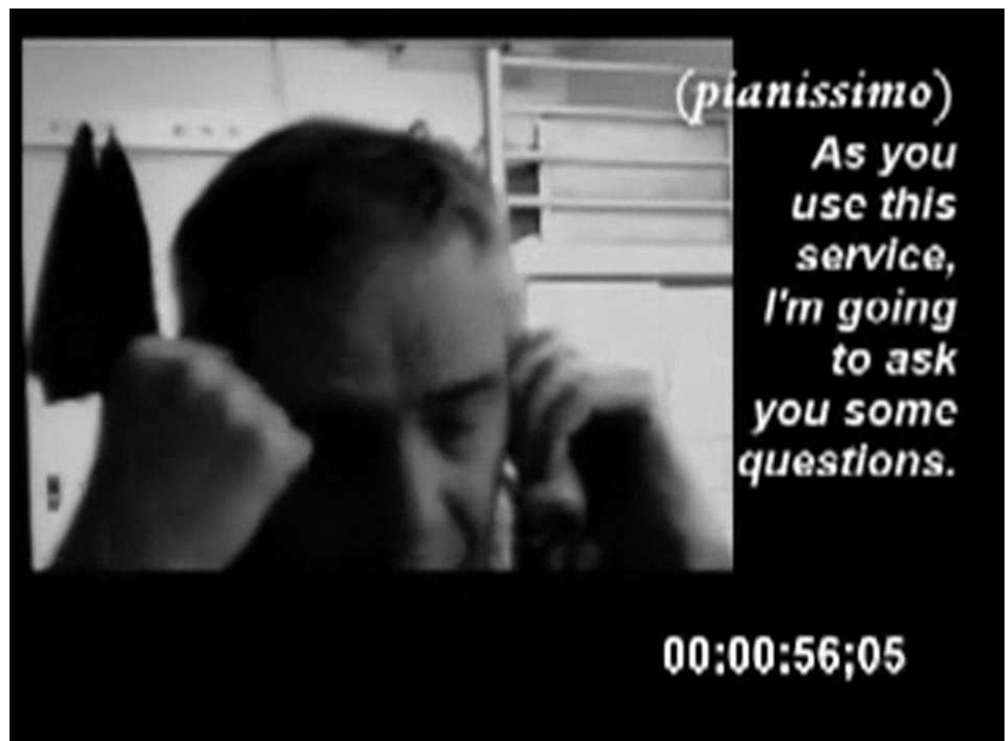
Most important: users

- 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 Interactive Voice Response (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, and recent successes (SIRI)
  - Goal orientation and keeping the user informed of their progress
  - Standardization and interoperability (VoiceXML)
  - Error correction (but needs to be used carefully – nobody wants to hear “I'm sorry, I didn't understand you” too many times!)

Although an apology is not always in order

- It seems not everyone got the memo about users and internal system errors ...

Although an apology is not always in order

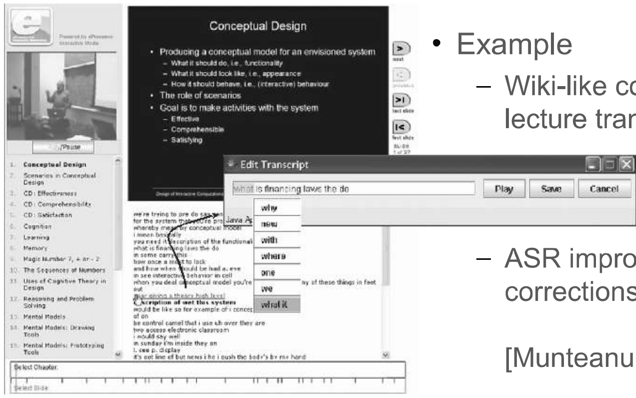
It's not a bug, it's a feature

- To Err is Human
- It may be impossible to completely eliminate ASR errors
- But they can be used to increase naturalness and realism of interaction

    - Samantha West – the Telemarketer (The Time, Dec. 10, 2013)

Human-Computer Interaction (HCI) 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]

Spoken interaction design

- Very little HCI research on user-centric design guidelines for speech
  - Need to leverage recent ASR progress to develop more natural, effective, or accessible user interfaces
    - We don't need to wait for 100% accuracy!
  - Workshop serires at CHI: Designing Speech and Language Interfaces
- Increased interest in and need for natural user interfaces (NUIs) by enabling speech interaction
  - As seen by many commercial applications, especially mobile
  - Although sometimes with very NSFW results!

Consumer speech (and multimodal) interfaces

Microsoft SYNC Speech Interface for Ford vehicles

Consumer speech (and multimodal) interfaces

Image: Adacel 2014.
http://www.adacel.com/MaxSimATC.html

Adacel Air Traffic Control Simulation & Training

Consumer speech (and multimodal) interfaces

Alelo Virtual Cultural Awareness Trainer and Operational Language and Culture Training

Images: Alelo 2014.
http://www.alelo.com/alelo_inc_us_dod_products.html

Consumer speech (and multimodal) interfaces

Microsoft Research Universal Speech-to-Speech Translator

Image: Microsoft Research 2012.
http://research.microsoft.com/en-us/research/stories/speech-to-speech.aspx

ASR in the wild

- EXERCISE 1, part 2

Speech Synthesis

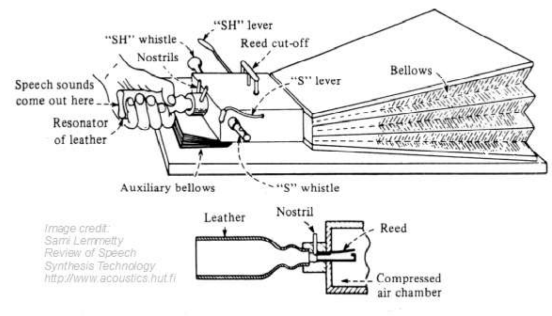
- *How does it work?*
- *How can you customize it?*
- *How good is it?*
- *How to tell that it's good enough?*

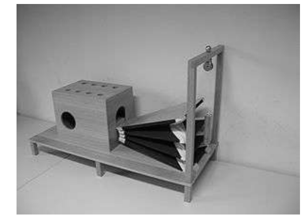
Synthesizing speech

- We've been trying this for centuries – before even thinking about automatic transcription
- History credits von Kempelen with inventing the first mechanical device able to reproduce human sounds
  - Incidentally – same guy who invented the Mechanical Turk

Kempelen's speaking machine

- Built in 1791
- Able to produce somewhat intelligible speech
- Mimics the human vocal tract





Wolfgang von Kempelen's

(1734-1804)

The VODER

- Things got better over time
- World Fair 1939 – the VODER machine (Bell Labs)
  - Same principles of emulating human speech production
  - Manually controlling the speech production parameters
  - Needed a highly trained operator
    - A total of 20 operators were trained
    - Quality of produced speech depended on the operator's skills

Nowadays ...

- Current Text-to-Speech engines

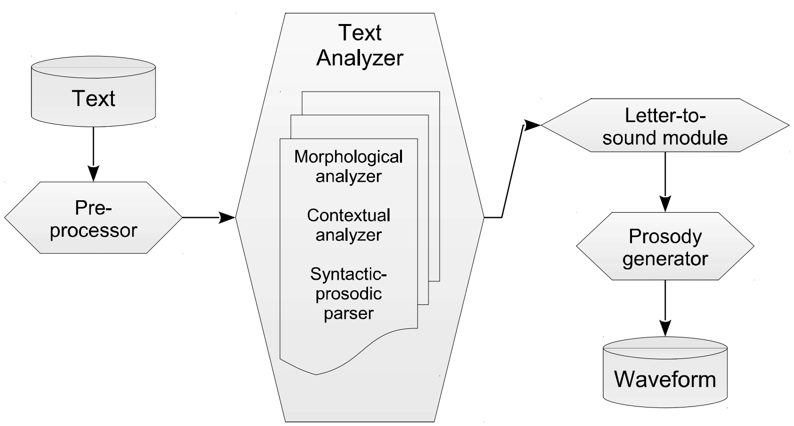
Nowadays ...

- Current Text-to-Speech engines

[Microsoft Anna]

Beyond just convenience ...

TTS Basics



Text analysis

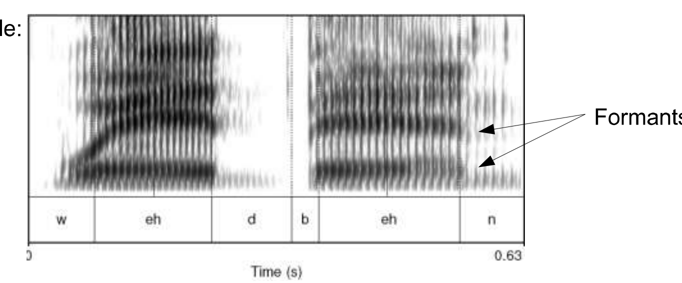
- Normalization
  - 100 → “one hundred”
  - 0.25 → “point two five”
  - Mr. → “Mister”
  - NASA vs. NHL
- Morphological analysis
  - Finding boundaries: words, syllables, sentences, ...
  - Determining: stress, accents, abbreviations, notations (e.g. email), origin of proper names, etc.
- Contextual analyzer & syntactic parser
  - Determine stress and intonation based on the sentences' grammatical structure and (some) semantics

Letter-to-sound mapping

- Map orthographic sequences of characters to sequences of diphones or triphones
  - (Uni-)phones are a terrible idea because of co-articulation effects and the lack of spectral stability at phone transitions
- Can use phonetic dictionaries, with or without stress markers:

```
interaction →  IH2 N T ER0 AE1 K SH AH0 N
               IH N T ER AE K SH AH N
```

Diphones example:



Prosody

- In charge of converting words+phones into boundaries, accent, F0 and duration information
- Prosodic phrasing
  - Need to break utterances into *intonation* phrases, e.g.,
    John can't throw, as far as I know
  - Punctuation is useful, but unreliable, and in any case insufficient
- Accents:
  - Which syllables should be accented
- Given accents/tones, generate intonation contour – depends on context:

```
she SAW me
she saw ME
SHE saw me
```


Duration

- Simplest: fixed size for all phones (100 ms)
- Next simplest: average duration for that phone (from training data), e.g.

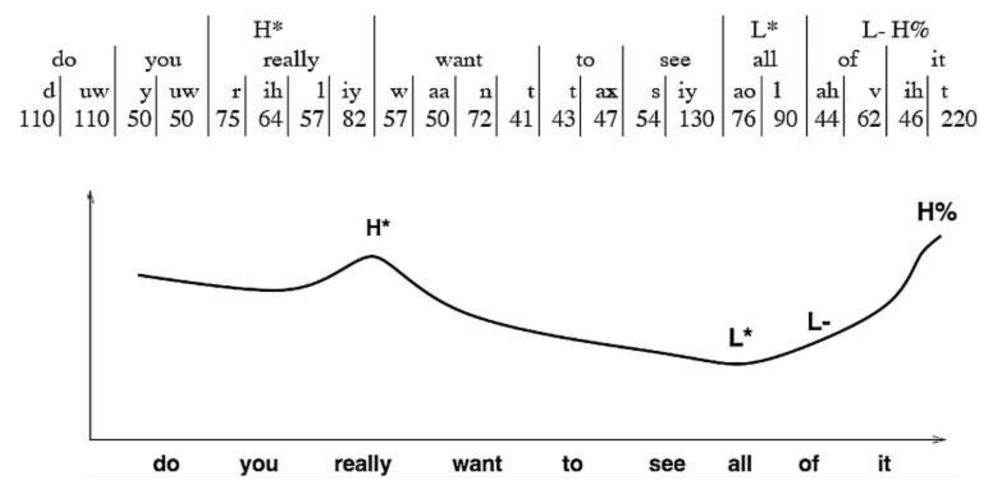
```
aa 118  / b 68  / ax 59  / d 68  / ay 138  / dh 44  / eh 87
```

- Next next simplest: add in phrase-final and initial lengthening plus stress
- Lots of fancy models of duration prediction, using:
  - Various clever normalizations
  - New features like word predictability
    - Words with higher bigram probability are shorter

Waveform synthesis

- Given:
  - String of phones
  - Prosody
    - Desired intonation contour for entire utterance
    - Duration for each phone
    - Stress value for each phone, possibly accent value
- Generate:
  - Waveforms

Internal waveform representation



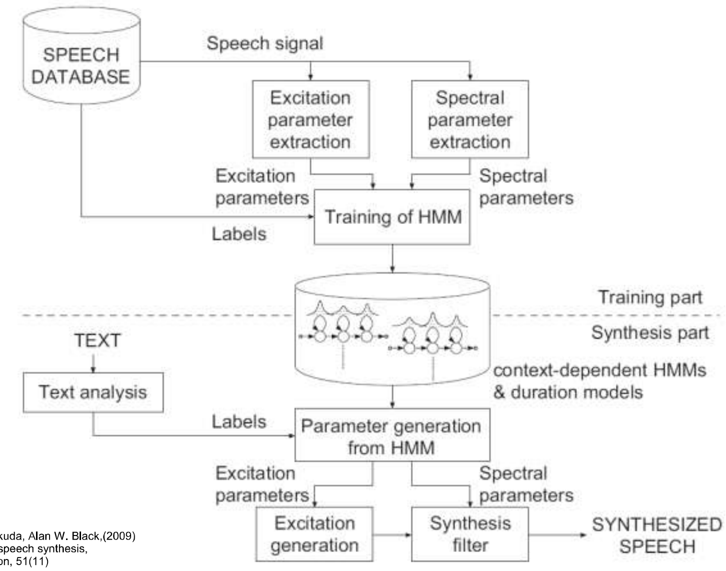
Waveform synthesis

- Articulatory Synthesis:
  - Model movements of articulators and acoustics of vocal tract
  - Common in 70's, but renewed interest as our articulatory models advance
- Formant Synthesis:
  - Start with acoustics, create rules/filters to create each formant
- Concatenative Synthesis:
  - Use databases of stored speech to assemble new utterances
  - Diphone or Unit Selection
  - Computationally very lightweight but requires a good database
- Statistical (HMM) Synthesis

Statistical (HMM) synthesis

- It is closer to ASR
- Hidden Markov Models (HMM) are trained from labelled data to learn how each phone is pronounced in each condition
  - It also learns its prosody
- Then, given a desired phoneme sequence and prosody pattern, it outputs the most probable audio sequence.

HMM-based speech synthesis system (HTS)



Heiga Zen, Keiichi Tokuda, Alan W. Black,(2009)
Statistical parametric speech synthesis,
Speech Communication, 51(11)

Using TTS

- Easier to set up than ASR
- Similar to ASR, there are some trade-offs
  - Commercial systems: good but not customizable
  - Research-grade systems: customizable but require skills to obtain good quality
- Some available systems:
  - Commercial: Acapela, AT&T
  - Commercial / SDK: Microsoft SAPI (built-in Windows)
  - Open source: eSpeak (http://espeak.sourceforge.net/)
  - Research:
    - CMU's Festvox, with extensive setup guide: http://festvox.org/
    - Edinburgh U's Festival: http://www.cstr.ed.ac.uk/projects/festival/
    - Nagoya Inst. of Technology's HTS: http://hts.sp.nitech.ac.jp/

TTS setup

- First – determine whether TTS is needed!
  - For simple IVR apps pre-recorded messages may be easier to set up
- Designing the text generation system, e.g.
  - For voice prompts – rules to generate the prompts
  - For read-aloud – rules to generate the prosody of the input text (this is not trivial and harder to do for some languages, e.g. Chinese)
  - Useful resource: ToBI (Tones and Breaks Indices) Framework for prosody transcription – used by many TTS systems http://www.ling.ohio-state.edu/~tobi/
- Pick a TTS system:
  - Research / toolkit – you will also need to set up a lexicon, text analysis module, selection of prosodic models, waveform synthesis, etc.
  - Commercial system – select “voice” and/or prosody

Evaluating TTS systems

- Significantly much harder to do than evaluating ASR!
- Two common metrics: intelligibility and quality

- Intelligibility – humans transcribing some TTS output
  - Rhyme tests – ability to transcribe acoustically confusable words, embedded in a carrier phrase

```
Now we will say bat again
Now we will say bad again
```

  - Transcribe Semantically Unpredictable Sentences with a fixed (and correct) syntactic pattern, e.g. DET ADJ NOUN VERB DET NOUN

```
The rainy desk applies the apple
```


Quality metrics

- Mean opinion score
  - Very subjective quality judgement
  - Human listeners ranking each utterance in a set with a 1 to 5 score
  - The mean for the set is that TTS system's quality score

- Sadly, no task-embedded evaluations or other ecologically-valid human subject experiments!

The Blizzard Challenge

- Yearly challenge aiming to evaluate state-of-the-art TTS systems on a common dataset
- Initiated in 2005 at CMU and Nagoya Institute of Technology http://www.festvox.org/blizzard/
- 10+ submissions in 2012
- Systems ranked according to intelligibility and subjective quality, judged by human listeners: speech experts, volunteers (random users), and English-speaking students (paid participants)
- The only significant, regular evaluation challenge for state-of-the-art research-grade TTS systems

TTS naturalness

- EXERCISE 2

Wrapping up ...

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
    - Use text-based search to locate something in a large collection of recorded video documentaries
    - Help mobile users with the pronunciation of unknown or difficult words
    - 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 Project on ASR for fishing boats [Lumsden, MobileHCI'08, '10]
  - Mixed-reality interaction for training simulators. E.g., “Multimodal Interactive Trainer” – MINT Project at NRC [Fournier, IITSEC'11]
  - Mobile language learning [Munteanu, CHI'10, '12, MobileHCI'10, '11]

Summary: final advice to system designers

- Moral of the story – what we've learnt:
  - 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!

Thank you!

Summary and discussion: 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:
   i. Choose the architecture (on-device vs. client-server, wearable computing vs. recording speech remotely)
   ii. Choose the ASR system – customization needs, environment
   iii. Define ASR restrictions – language, acoustic, dialogue
   iv. Design the ASR connection to the main application
4. Design the interactive interface – multimodality
5. Repeat steps as necessary