Design and Implementation of Speech Recognition Systems

Spring 2013
Bhiksha Raj, Rita Singh

Class 1: Introduction
23 Jan 2013

Administrivia

- Instructors:
  - Bhiksha Raj
    - GHC 6705
    - Office hours: Tuesday 2-3
    - bhiksha@cs.cmu.edu
  - Rita Singh
    - GHC 6703
    - Office hours: TBD
    - rsingh@cs.cmu.edu
- TA
  - TBD

Course webpage, Registration Etc.

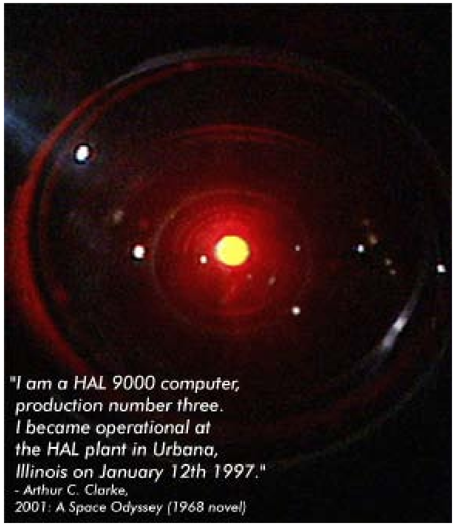
- Course webpage:
  - http://asr.cs.cmu.edu/spring2013/
  - Slides and handouts will be posted here
  - Notes to appear
- Google group: Will send invites to everyone, please sign up
  - Group email: 11756-18799D@googlegroups.com
  - Addr: http://groups.google.com/group/11756-18799D
- All notices will be posted on google group

Attendance

- Is compulsory
- Carries 10% of your points

- Grading: Relative, with bounds
  - Anyone not completing 50% of assignments automatically gets a C
    - This means everyone can get a “C”
  - Anyone successfully completing all projects gets an “A”
    - This means everyone can get an “A”
  - Between these bounds, we will use a relative scale
    - Based on histogram of scores

What is "Automatic" Speech Recognition



- Computer recognition of speech
  - Enabling a computer to "recognize" what was spoken
  - Usually understood as the ability to faithfully *transcribe* what was spoken
    - Something even humans cannot do often
  - More completely, the ability to *understand* what was spoken
    - Which humans do extremely well

Why Speech?

- Most natural form of human communication
- Highest bandwidth human communication as well
- With modern technology (telephones etc.) people can communicate over long distances
  - Voice-based IVR systems are virtually everywhere
  - Such automated systems can remain online 24/7
- Voice commands can free hands/eyes for other tasks
  - Especially in cars, where hands and eyes are busy

Some Milestones in Speech Recognition

- 1968? – Vintsyuk proposes dynamic time warping algorithm
- 1971 – DARPA starts speech recognition program
- 1975 – Statistical models for speech recognition
  - James Baker at CMU
- 1988 – Speaker-independent continuous speech recognition
  - 1000 word vocabulary; *not* real time!
- 1992 – Large vocabulary dictation from Dragon Systems
  - Speaker dependent, isolated word recognition
- 1993 – Large vocabulary, real-time continuous speech recognition
  - 20k word vocabulary, speaker-independent
- 1995 – Large vocabulary continuous speech recognition
  - 60k word vocabulary at various universities and labs
- 1997? – Continuous speech, real-time dictation
  - 60k word vocabulary, Dragon Systems *Naturally Speaking*, IBM *ViaVoice*
- 1999 – Speech-to-speech translation, multi-lingual systems
- 2004 – Medium/large vocabulary dictation on small devices
- 2011 – SIRI

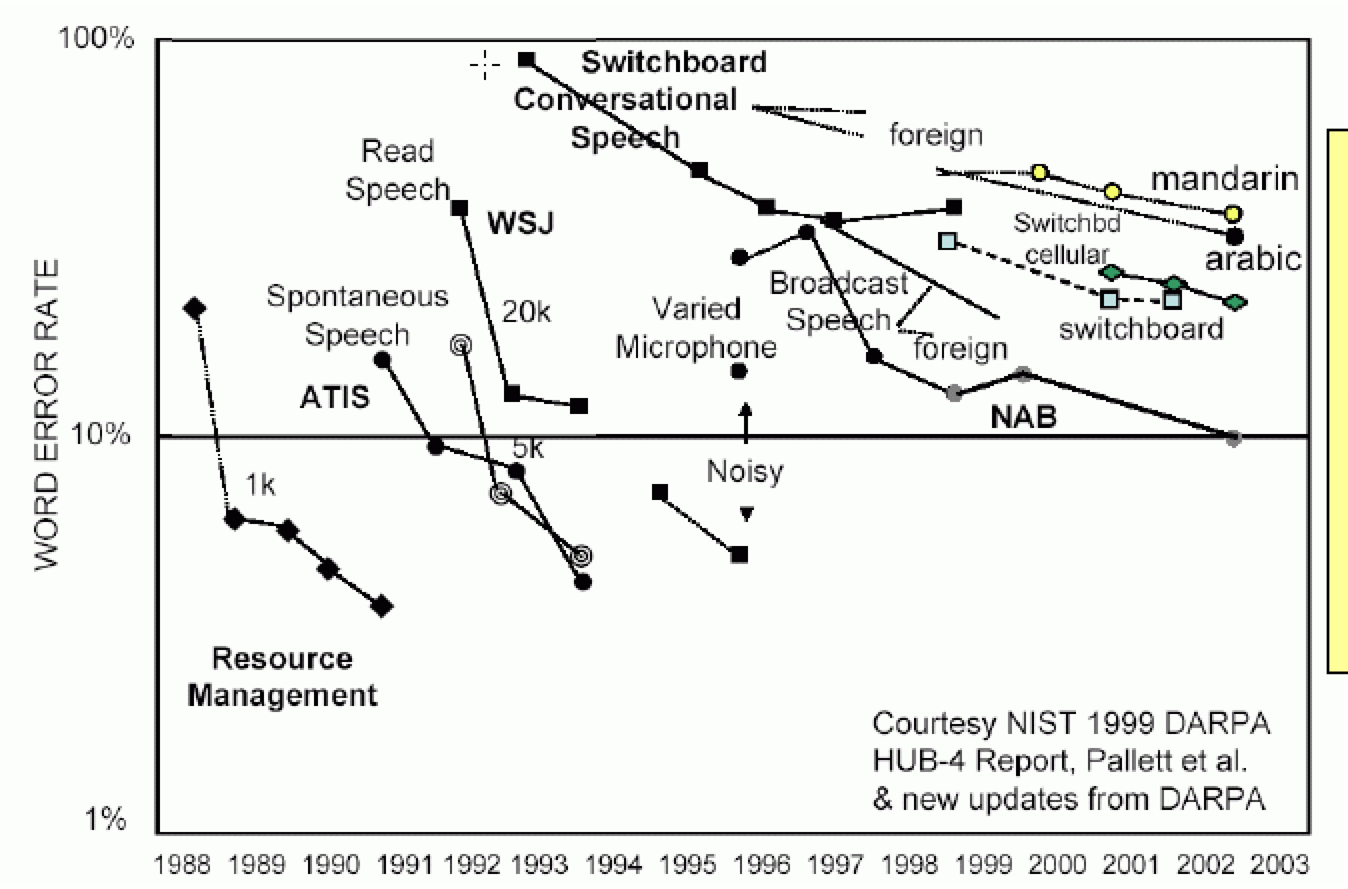
Some Reasons for the Rapid Advances

- Improvements in acoustic modeling
  - Hidden Markov models, context-dependent models
  - Speaker adaptation
  - Discriminative models
- Improvements in Language modeling
  - Bigram, trigram, quadgram and higher-order models
- Improvements in recognition algorithms

- Availability of more and more training data
  - Less than 10 hours to 100000 hours
  - Brute force
- Unprecedented growth in computation and memory
  - MHz to GHz CPUs, MBs to GBs memory
  - Brute force, again

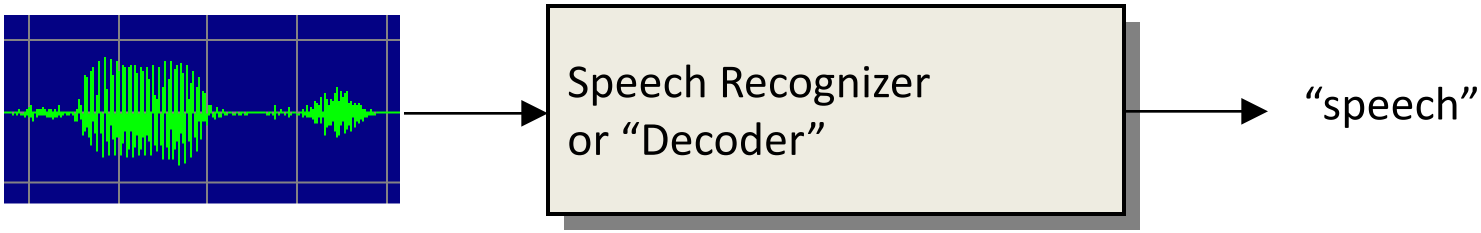
Speech Recognition Performance

- History of ASR performance in DARPA/NIST speech recognition evaluations



Every time ASR performance reached a respectable level, the focus shifted to a more difficult problem, broadening the research horizons

The Speech Recognition Problem



- Speech recognition is a type of pattern recognition problem
  - Input is a stream of sampled and digitized speech data
  - Desired output is the sequence of words that were spoken
- Incoming audio is “matched” against stored patterns that represent various sounds in the language
  - Sound units may be words, phonemes or other similar units
  - Patterns also represent *linguistic* constraints
    - “Fire And Ice” or “Firing Ice”?

Why is Speech Recognition Hard?

- Acoustic patterns vary from instance to instance
  - Natural variations: *Even the same person never speaks anything exactly the same way twice*
  - Systematic variations:
    - Human physiology: squeaky voice *vs*. deep voice
    - Speaking style: clear, spontaneous, slurred or sloppy
    - Speaking rate: fast or slow speech
      - Speaking rate can change within a single sentence
    - Emotional state: happy, sad, etc.
    - Emphasis: stressed speech *vs* unstressed speech
    - Accents, dialects, foreign words
  - Environmental or background noise

- Linguistic patterns are hard to characterize:
  - Large vocabulary and infinite language
  - Absence of word boundary markers in continuous speech
  - Inherent ambiguities: "I scream" or "Ice cream"?
    - Both are linguistically plausible; other context cues are needed

Technological Challenges

- Inherent variations in speech make pattern matching difficult
  - Solution must understand and represent what is *invariant*
    - This represents the message
- Pattern matching algorithms are by nature inexact
  - Compound an already hard problem
  - Solutions must account for imprecisions and assumptions of pattern matching algorithms

The Technological Challenges (contd.)

- As target vocabulary size increases, complexity increases
  - Computational resource requirements increase
    - Memory size to store patterns
    - Computational cost of matching
  - Most important, the degree of *confusability* between words increases
    - More and more words begin sounding alike
    - Requires finer and finer models (patterns)
    - Further aggravates the computational cost problem

Disciplines in Speech Technology

- Modern speech technology is combination of many disciplines
  - Physiology of speech production and hearing
  - Signal processing
  - Linear algebra
  - Probability theory
  - Statistical estimation and modeling
  - Information theory
  - Linguistics
  - Syntax and semantics
  - Computer science
    - Search algorithms
    - Machine learning
    - Computational complexity
    - Computer hardware

Typical ASR Applications

- Online:
  - Command and control
  - Dictation
  - Simple speech APIs

- Offline:
  - Transcription
  - Keyword spotting, Mining

What will the course be about

- This will be a hands-on course
  - Everyone is expected to code
- The stress will not be on theory
  - It will be on hands-on practice
- We will discuss algorithms and implementation details

Format of Course

- Lectures
- A series of projects/assignments of linearly increasing complexity
- Each project has a score
- Projects will be performed by teams
  - Size 2-3 members
- Projects will be presented in class periodically
  - Code description
  - Algorithmic and implementation details
  - Problems faced, solutions etc.
- Grading will be based mainly on completion of projects

Projects

- Project 1a: Capturing Audio

- Project 1b: Feature computation
  - Plug feature computation into audio capture
  - Modify feature computation for buffered audio
  - Visualize various partial results in feature computation
  - Modify various parameters and visualize output

- Project 2: A spellchecker
  - String matching

- Project 3: DTW-based recognition of isolated words
  - Generalize string matching to DTW
  - Record templates
  - Create feature-based templates
  - Pattern matching and recognition

Projects

- Project 4: HMM-based recognition of isolated words
  - Viterbi decoding with simple Gaussian densities
  - Viterbi decoding with mixture Gaussian densities

- Project 5: Training HMMs from isolated recordings (Viterbi method)
  - Recording data
  - Segmenting data
  - Training models

- Project 6: Training (and recognition) of isolated words from continuous recordings
  - Record data for a chosen vocabulary
  - Train models of different structures
  - Recognition

Projects

- Project 7: HMM-based recognition of continuous word strings
  - Continuous ASR of words
  - Continuous ASR of words with optional silences
  - Training a set of word models (carried over from previous exercise)
  - Evaluation

Projects

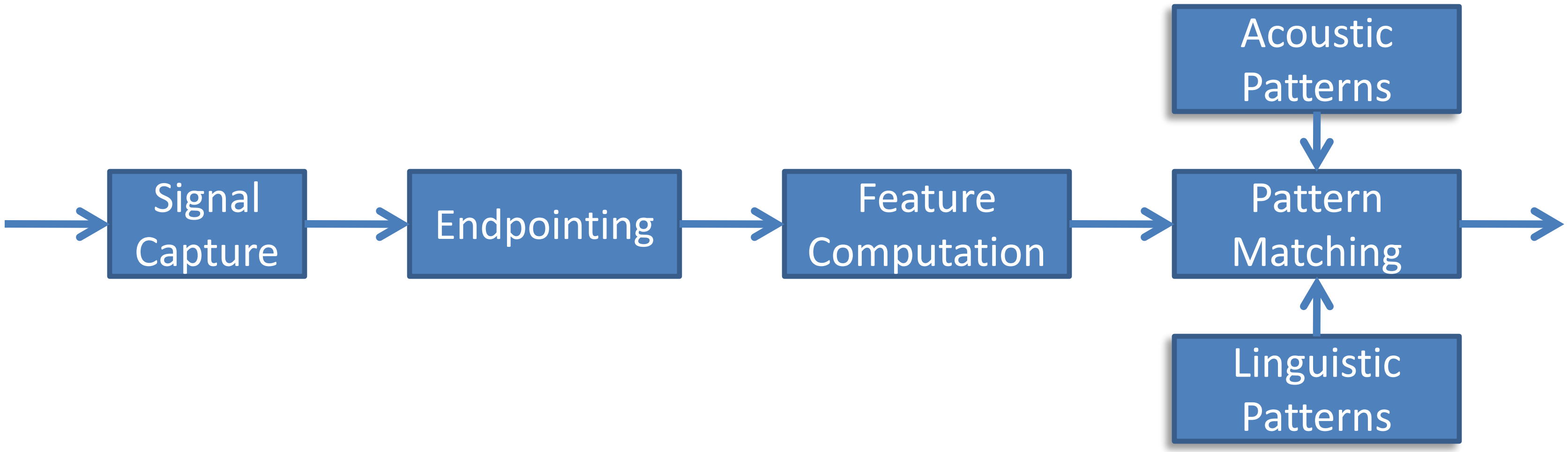
- Project 8: Grammar-based recognition of continuous words
  - Building graphs from grammars
  - Building HMM-networks from grammars
  - Recognition of continuous word strings from a grammar

Projects

- Project 9: Grammar-based recognition from Ngram models
  - Conversion of Ngrams to FSGs
  - Grammar-based recognition of continuous speech from Ngrams

- Project 10: Training and recognition with subword units

Typical ASR procedure



- Series of steps that translate spoken audio to text
  - Lets consider the steps
    - Several of which we will cover in this course

Preview of Topics in the Course

- Speech Signal capture
- Endpointing
- Feature extraction from speech signal (in brief)
- Template matching algorithm
- Hidden Markov modeling of speech
- Isolated word *vs* continuous speech recognition
- Small vocabulary *vs* large vocabulary considerations
- Pronunciation modeling
- Language modeling
- Obtaining multiple results from a recognizer
- Determining confidence in recognition results
- Accuracy and efficiency considerations
- Creating or training various models (in brief)

Preview of Topics in the Course

- Speech Signal capture
- Endpointing
- Feature extraction from speech signal (in brief)
- Template matching algorithm
- Hidden Markov modeling of speech
- Isolated word *vs* continuous speech recognition
- Small vocabulary *vs* large vocabulary considerations
- Pronunciation modeling
- Language modeling
- Obtaining multiple results from a recognizer
- Determining confidence in recognition results
- Accuracy and efficiency considerations
- Creating or training various models (in brief)

Speech Production: The Vocal Tract

- Speech is produced by the vocal tract
  - Much study about how the vocal tract produces speech
- The result is a series of pressure waves that fall on the listener's ear

3. Oral and nasal cavities trap resonances, shaping the sounds as the cavities change shape

2. Vocal cords vibrate (open and close rapidly), sending impulses of air pressure into the oral and nasal cavities

1. Air forced up from lungs

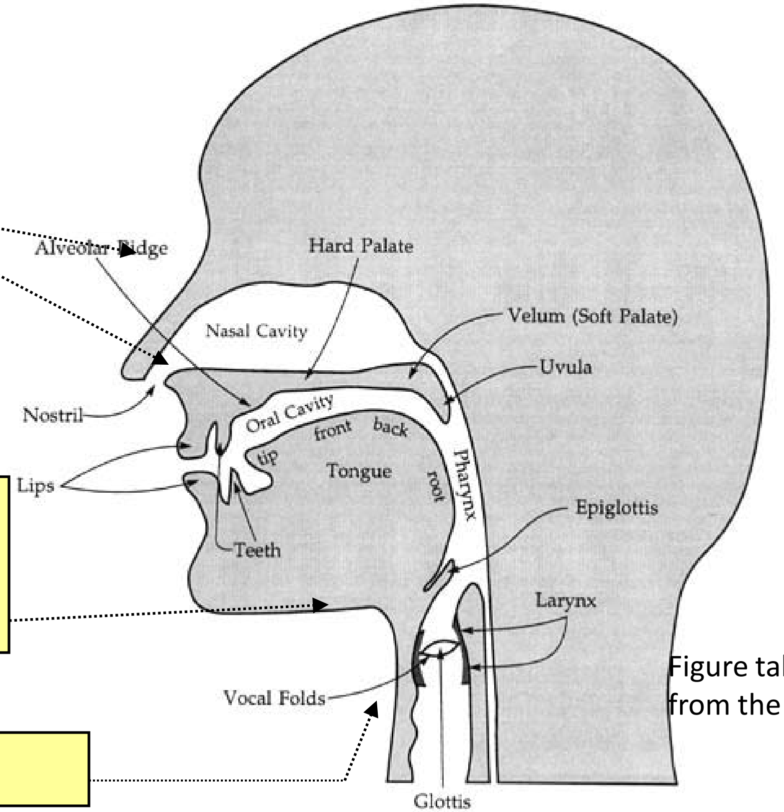
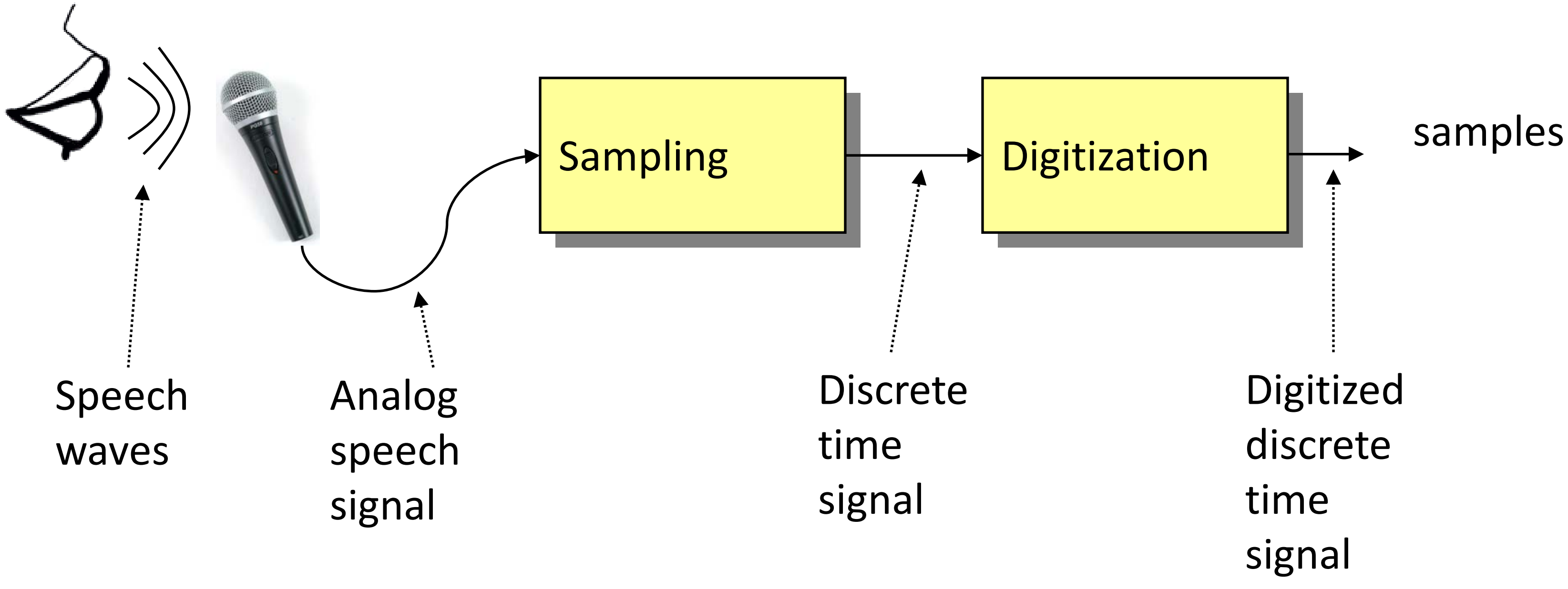


Figure taken from the web

Speech Signal Capture

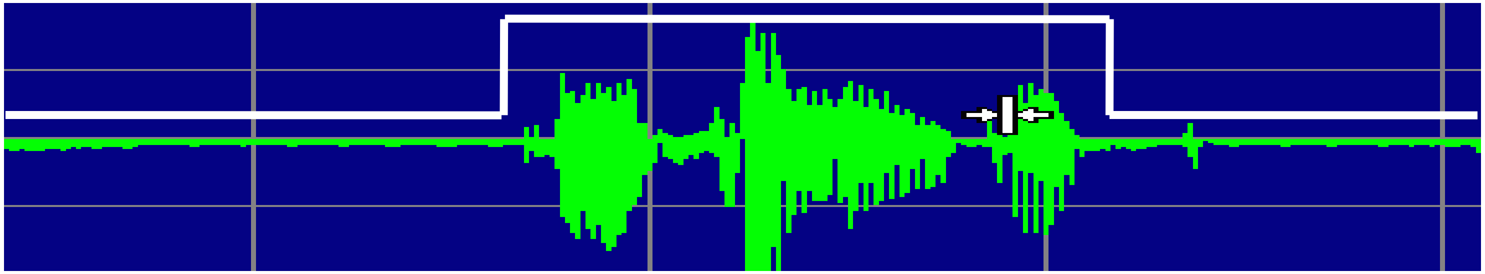
- The first step is capture of the pressure waves as a series of samples
  - We will consider this briefly later
- A simplified view:



Preview of Topics in the Course

- Speech Signal capture
- Endpointing
- Feature extraction from speech signal (in brief)
- Template matching algorithm
- Hidden Markov modeling of speech
- Isolated word *vs* continuous speech recognition
- Small vocabulary *vs* large vocabulary considerations
- Pronunciation modeling
- Language modeling
- Obtaining multiple results from a recognizer
- Determining confidence in recognition results
- Accuracy and efficiency considerations
- Creating or training various models (in brief)

Endpointing: Identifying presence of speech



- Necessary to identify where speech is present in a captured signal
- Avoid attempting to recognize speech in non-speech regions
  - Computational efficiency
  - Prevents hallucination of unspoken words

Endpointing: Identifying the presence of speech

- Speaking modes:
  - Push to talk
    - Press and speak
  - Hit to talk
    - Hit and speak
  - Continuous listening

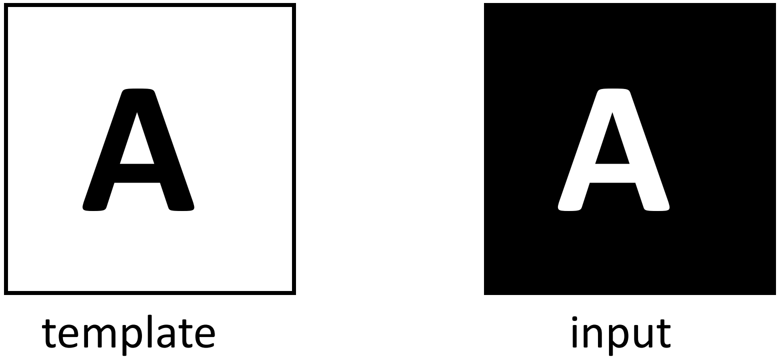
- Multi-pass endpointing

Preview of Topics in the Course

- Speech Signal capture
- Endpointing
- Feature extraction from speech signal (in brief)
- Template matching algorithm
- Hidden Markov modeling of speech
- Isolated word *vs* continuous speech recognition
- Small vocabulary *vs* large vocabulary considerations
- Pronunciation modeling
- Language modeling
- Obtaining multiple results from a recognizer
- Determining confidence in recognition results
- Accuracy and efficiency considerations
- Creating or training various models (in brief)

Feature Extraction

- Should pattern matching in speech be done directly on audio samples?
  - Raw sample streams are not well suited for matching
  - A visual analogy: recognizing a letter inside a box

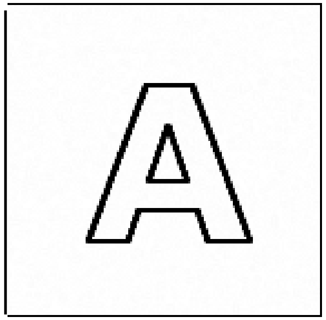


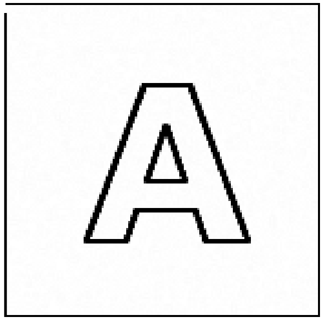
  - The input happens to be pixel-wise inverse of the template

- But blind, pixel-wise comparison (*i.e.* on the raw data) shows maximum *dis*-similarity

Feature Extraction (contd.)

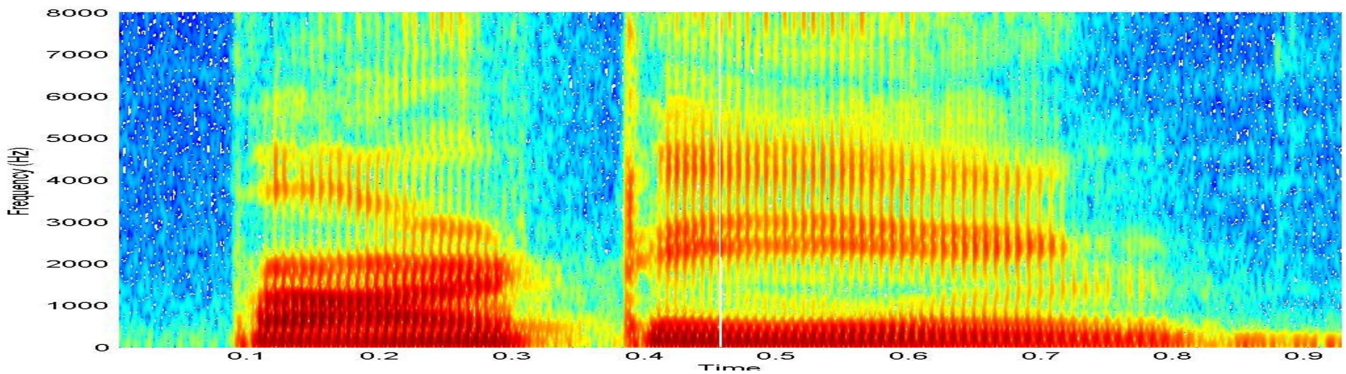
- Needed: identification of salient *features* in the images
  - E.g. edges, connected lines, shapes
    - These are commonly used features in image analysis
  - An *edge detection* algorithm generates the following for both images and now we get a perfect match





  - Our brain does this kind of image analysis automatically and we can instantly identify the input letter as being the same as the template

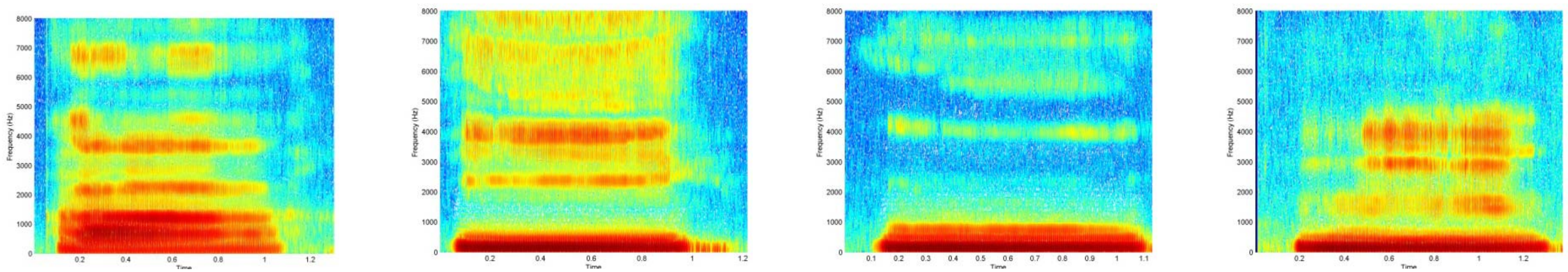
Feature Extraction: Speech



- The information in speech lies in the *frequency* distribution of the signal
  - The ear performs frequency analysis
- Visualization: Convert speech to a *time-frequency* representation
  - E.g. Spectrograms are 2-D time-frequency plots
  - The x-axis is time, the y-axis is frequency
  - The intensity at each location indicates the energy in the signal in that frequency band, during that time window
    - Red is highest value, blue is lowest

Spectrograms of Speech Signals

- The following are spectrograms for various phonemes or basic speech sounds, each about 1 sec long



AA IY UW M

- The phonemes have a distinct set of dominant frequency bands called *formants*
- Feature computation converts the signals into a representation such as the above
  - Where sound identity is clear
- We will learn to compute features

Preview of Topics in the Course

- Speech Signal capture
- Endpointing
- Feature extraction from speech signal (in brief)
- Template matching algorithm
- Hidden Markov modeling of speech
- Isolated word *vs* continuous speech recognition
- Small vocabulary *vs* large vocabulary considerations
- Pronunciation modeling
- Language modeling
- Obtaining multiple results from a recognizer
- Determining confidence in recognition results
- Accuracy and efficiency considerations
- Creating or training various models (in brief)

Simple Approach: Template Matching

- Consider a simple two word vocabulary
  - *YESTERDAY* and *TOMORROW*
- Pre-record one spoken example of each word
- Each pre-recorded example is a *template*
- When someone later speaks one of the two words:
  - "Somehow" compare this speech to each template
    - (After converting everything to feature vector streams, of course)
  - Select the word whose template more closely matches the input
- Any speech recognition system is, at its core, some version of this simple scheme

Template Matching: Dynamic Programming

- Template matching: how to compare templates to input speech
  - Input and template may be from different people
  - Input and template can be of different durations
  - How can two data streams differing in so many ways be compared?

- We will learn the *dynamic programming* (DP) algorithm to perform this matching efficiently, and *optimally*

- DP is the cornerstone of most widely used speech recognition systems

A little diversion…

- Generalization of templates …

General Concept of "Models"

- Templates, *e.g.* pre-recorded examples of spoken words, are an instance of *models*
  - Wikipedia definition of "model": "*An abstraction or conceptual object used in the creation of a predictive formula*"

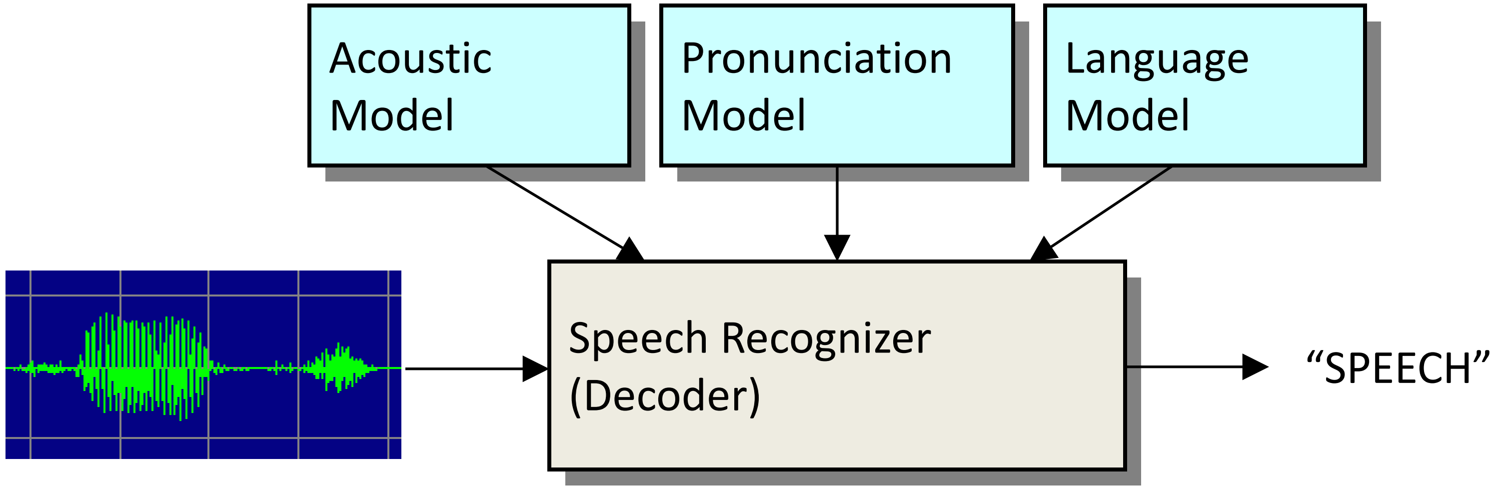
- Templates are models of what we expect future utterances of words to look like
  - Models are *trained* from *known examples* of what we expect to see in the future

- The quality of a model may be judged by:
  - How coarse or detailed it is
  - How robust or brittle it is
  - How accurately it classifies future inputs, etc.

Models for Speech Recognition

- In speech recognition, we have the following models:
  - *Acoustic* models, for modeling speech signals (e.g., templates)
  - *Pronunciation* models (*e.g.* as in dictionaries)
  - *Language* models, for modeling the structure of sentences
- Not all speech recognition systems need all three types of models
  - However, all do need acoustic models of some sort
- Many systems actually use extra models for other purposes as well:
  - *Dialog* models are used to structure a conversation between a user and a speech-based application
  - *Duration* models may be used to constrain word, phoneme or syllable durations

A Typical Speech Recognition System



- Acoustic, pronunciation and language models are inputs to the recognizer

- A complete speech recognition package must include:
  - The *recognizer* or *decoder*
    - Incorporates information from various models to recognize the speech
  - *Trainers* to train the various models

Our Focus

- For now, our focus will be more on the recognizer or decoder, than the trainers

- Other commonly used names for the recognizer: *decoder*, *recognition engine*, *decoding engine*, etc.

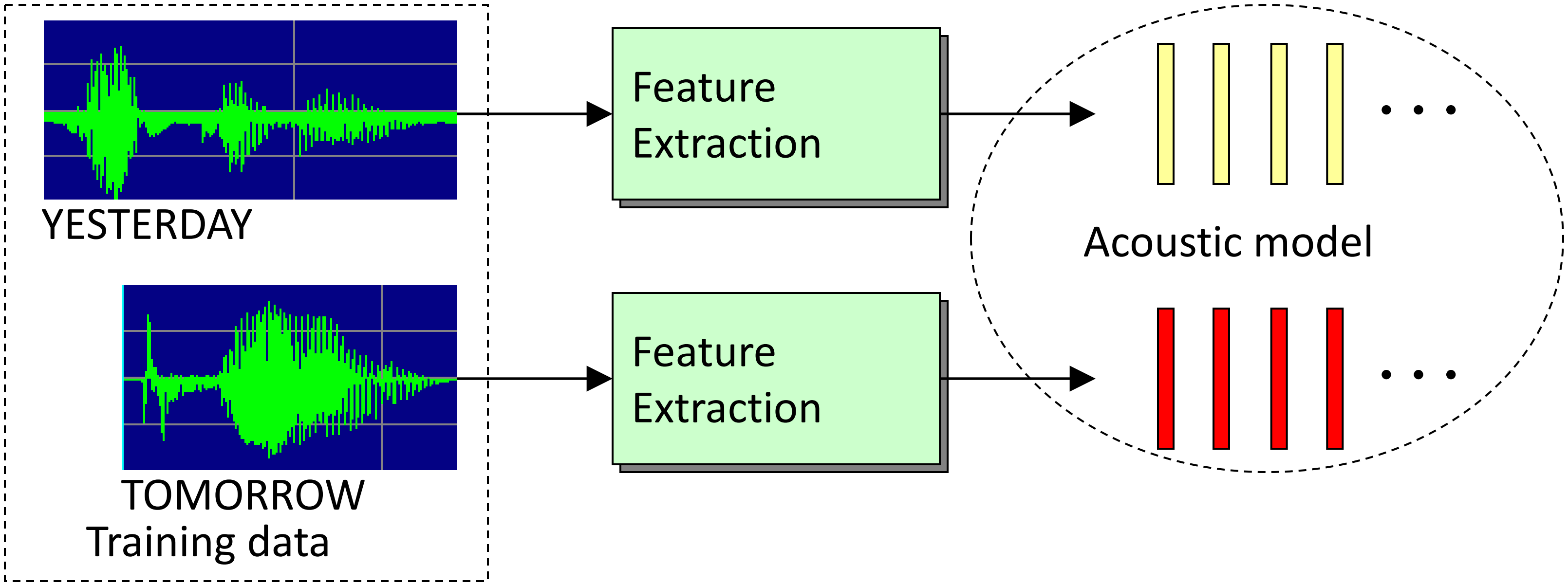
- The algorithm used by the recognizer is usually referred to as the *search algorithm*
  - Given some spoken input, it searches for the correct word sequence among all possibilities
  - It does this, in effect, by searching through all available templates

- End of diversion – Back to template matching

Preview of Topics in the Course

- Speech Signal capture
- Endpointing
- Feature extraction from speech signal (in brief)
- Template matching algorithm
- Hidden Markov modeling of speech
- Isolated word *vs* continuous speech recognition
- Small vocabulary *vs* large vocabulary considerations
- Pronunciation modeling
- Language modeling
- Obtaining multiple results from a recognizer
- Determining confidence in recognition results
- Accuracy and efficiency considerations
- Creating or training various models (in brief)

Templates as Acoustic Models



- Training data consist of the pre-recorded examples of the words
- "Training" the acoustic model is, in this case, trivial
  - The feature streams derived from the templates serve as the acoustic model

Hidden Markov Models for Acoustics

- Templates as acoustic models are quite brittle
  - Appropriate only for small vocabulary or "isolated-word recognition" situations
  - Also, inaccurate if speaker is different from the template

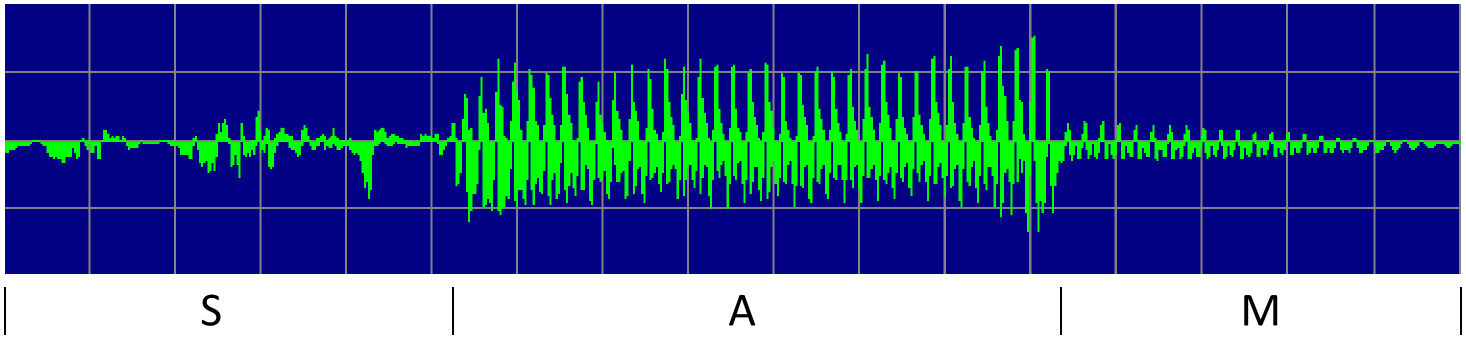
- *Hidden Markov models* (HMMs) are an elegant generalization that leads to more robust performance
  - We will learn about these

- HMMs are the most common framework for acoustic models in modern speech recognition systems

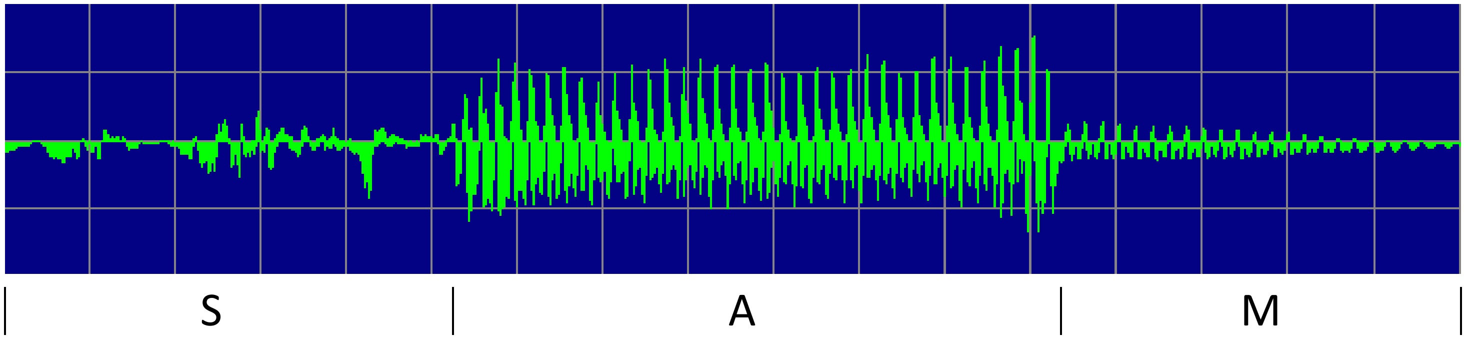
HMMs for Acoustics: Intuition

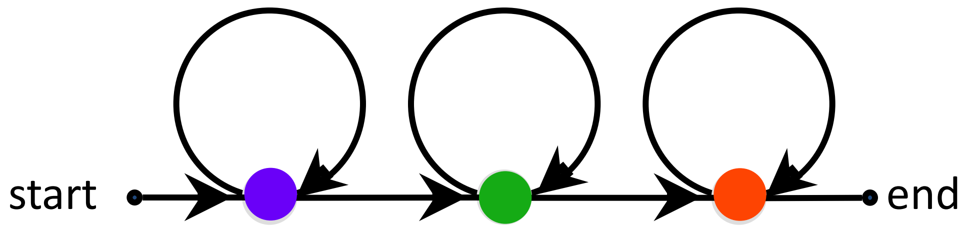
- HMMs model speech by breaking it into short "consistent" segments that are relatively uniform within themselves
  - E.g. the speech signal for "SAM" below can be broken down into three such segments
- Each segment is modeled by an HMM *state*
- Each state has an *underlying probability distribution* that describes the feature vectors for its segment
  - Principal source of robustness of HMMs
- The entire word can be described as a linear sequence of such states

Speech signal for "SAM"



HMMs for Acoustics: Intuition (contd.)





3-state HMM for the word SAM

- The HMM succinctly captures the essentials of the acoustics
- *Note:* this illustration is only for obtaining an intuitive idea of HMMs; details follow later in the course

HMM Based Speech Recognition

- The same dynamic programming algorithm used for template matching forms the basis for decoding based on HMMs

- Two important decoding algorithms will be presented
  - The *Forward* algorithm, used more in training acoustic models
  - *The Viterbi* algorithm, *the* most widely used decoding algorithm among speech recognizers

Preview of Topics in the Course

- Speech Signal capture
- Endpointing
- Feature extraction from speech signal (in brief)
- Template matching algorithm
- Hidden Markov modeling of speech
- Isolated word *vs* continuous speech recognition
- Small vocabulary *vs* large vocabulary considerations
- Pronunciation modeling
- Language modeling
- Obtaining multiple results from a recognizer
- Determining confidence in recognition results
- Accuracy and efficiency considerations
- Creating or training various models (in brief)

Isolated Word *vs* Continuous Speech

- Isolated word speech recognition relies on every word being spoken in isolation, surrounded by brief silence

- Early speech recognizers relied on *isolated word* speech for accuracy and speed
  - E.g. early dictation system from Dragon Systems (*Dragon Dictate*)

- In isolated word speech, the silence periods between words bracket word boundaries explicitly
- Requires a *speech/silence detection module* to separate word segments in input stream
  - Endpointing is critical

Isolated *vs* Continuous Speech (contd.)

- Continuous speech recognition systems can handle normal, continuous speech
  - Word boundaries are not explicitly demarcated

- Analogous to deciphering text without any spaces:

  ireturnedandsawunderthesunthattheraceisnottotheswiftnortheb attletothestrongneitheryetbreadtothewisenoryetrichestomenofu nderstandingnoryetfavourtomenofskillbuttimeandchancehappe nethtothemall

  - Even more accurately, analogous to where some corruption of the text has taken place
    - *E.g.* through deletion, substitution or insertion of letters

Preview of Topics in the Course

- Speech Signal capture
- Endpointing
- Feature extraction from speech signal (in brief)
- Template matching algorithm
- Hidden Markov modeling of speech
- Isolated word *vs* continuous speech recognition
- Small vocabulary *vs* large vocabulary considerations
- Pronunciation modeling
- Language modeling
- Obtaining multiple results from a recognizer
- Determining confidence in recognition results
- Accuracy and efficiency considerations
- Creating or training various models (in brief)

Small *vs* Large Vocabulary Systems

- Additional issues in large vocabulary systems:
- First: the need for a set of *sub-word* units
  - Cannot explicitly store templates/models for every word
    - Potentially unlimited number of words
    - Many will not have sufficient training data
  - Models for words must be *composed* from models for smaller sound units
    - Typically phonemes
  - Requires the maintenance of a *pronunciation model*, also commonly known as *dictionary* or *lexicon*
    - The lexicon defines each word in terms of the lower level units

Large Vocabulary Systems (contd.)

- Second: Larger vocabulary implies larger *confusability* between words
  - Many pairs of words may differ only in a single sound
    - *e.g.* BAR, PAR, TAR, CAR
- Acoustic models need to be significantly more sophisticated, and more discriminating
  - Need to distinguish between the same basic sound, occurring in different contexts
    - *E.g.* CAT *vs* BAT: must model the difference in the "A"s in the two words!
- This will be dealt with in *context-dependent acoustic modeling*

Large Vocabulary Systems (contd.)

- Three: Many phrases actually sound *exactly* alike
  - "Are Tea" or "Arty"?
- Especially true with continuous speech recognition where word boundaries are not known
  - Consider again the analogy of text with no spaces ireturnedandsawunderthesunthattheraceisnottotheswiftnorthebattletothestrongneitheryetbreadtothewisenoryetrichestomenofunderstandingnoryetfavourtomenofskillbuttimeandchancehappenethtothemall
  - How many different words can you identify in that running text?
  - Allowing for overlaps? If the text contained errors?
- Need higher level knowledge to resolve ambiguities
  - *Syntactic* knowledge or grammar
  - *Semantic* knowledge or meaning

Large Vocabulary Systems (contd.)

- Of the two (grammatical and semantic knowledge) the former is much easier to capture and represent
- Every large vocabulary speech recognition system uses such knowledge
  - Usually called *Language Models* (LM) or, sometimes, *Grammars*
- We will study two forms of LMs:
  - Structured grammars (*finite state* or *context free* grammars) used in small to medium vocabulary systems
  - *N-gram* LMs for medium and large vocabulary systems
    - Based on knowing probabilities of word sequences
- We will study how to naturally integrate such LMs into a decoder

Large Vocabulary Systems (contd.)

- Four: Large vocabulary systems have greater computational and memory size requirement
  - Acoustic models have to be much more detailed or fine grained
    - Must model all the details that distinguish between the various words
      - Which may differ only minimally
  - Language models can be enormous, especially N-grams
    - The number of linguistic structures that are possible with a very large vocabulary are very large
    - All must be modelled

- Thus, decoding algorithms for large vocabulary systems must pay close attention to *computational* and *memory efficiency*

Preview of Topics in the Course

- Speech Signal capture
- Endpointing
- Feature extraction from speech signal (in brief)
- Template matching algorithm
- Hidden Markov modeling of speech
- Isolated word *vs* continuous speech recognition
- Small vocabulary *vs* large vocabulary considerations
- Pronunciation modeling
- Language modeling
- Obtaining multiple results from a recognizer
- Determining confidence in recognition results
- Accuracy and efficiency considerations
- Creating or training various models (in brief)

Pronunciation Modelling

- Large (or flexible) vocabulary systems *compose* models for words from smaller sound units
- Pronunciation modelling specifies how words are composed from sub-word units
- Includes hand-crafted lexicons and automatically generated pronunciations
  - Will only be superficially covered
    - A simple pronunciation generator (if time permits)

Preview of Topics in the Course

- Speech Signal capture
- Endpointing
- Feature extraction from speech signal (in brief)
- Template matching algorithm
- Hidden Markov modeling of speech
- Isolated word *vs* continuous speech recognition
- Small vocabulary *vs* large vocabulary considerations
- Pronunciation modeling
- Language modeling
- Obtaining multiple results from a recognizer
- Determining confidence in recognition results
- Accuracy and efficiency considerations
- Creating or training various models (in brief)

Language Modelling

- Language models represent *valid* and *plausible* word sequences
  - Help the system choose between “Wreck a nice beach” or “recognize speech”

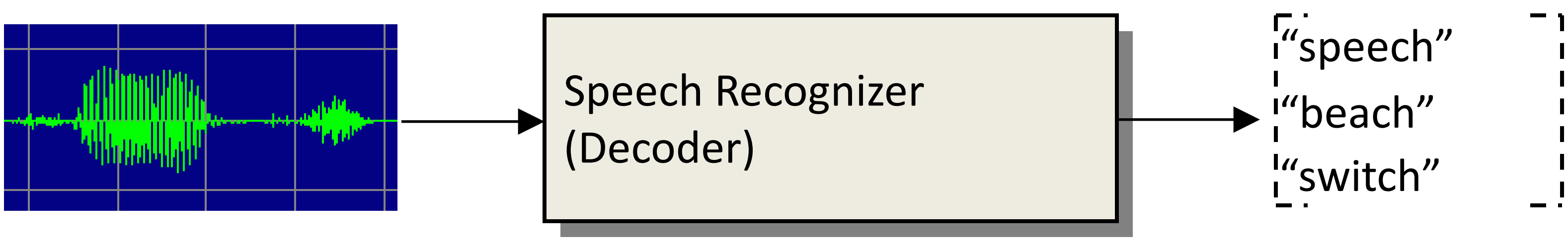
- Modelled in various ways
  - Rigid structure: Finite state and context free grammars
  - Statistical structure: N-gram language models
- We will cover both

Preview of Topics in the Course

- Speech Signal capture
- Endpointing
- Feature extraction from speech signal (in brief)
- Template matching algorithm
- Hidden Markov modeling of speech
- Isolated word *vs* continuous speech recognition
- Small vocabulary *vs* large vocabulary considerations
- Pronunciation modeling
- Language modeling
- Obtaining multiple results from a recognizer
- Determining confidence in recognition results
- Accuracy and efficiency considerations
- Creating or training various models (in brief)

Multiple Results from Recognizer

- Recognition systems usually produce a single recognition result, or *hypothesis*
  - The *best guess* for what was spoken
    - Which may be wrong
- Sometimes we desire more than just the single result:
  - The top N best hypotheses
  - Allows the application to consider other alternative explanations for the audio
  - Humans subconsciously generate N-best lists all the time
    - Especially when what they hear is ambiguous or unclear



Multiple Results from Recognizer (contd.)

- There are two commonly used forms of multiple results
  - *N-best lists*
  - *Lattices* or *DAGs* (directed acyclic graphs)

- An N-best list is a list of recognition results (or *hypotheses*)
  - Each hypothesis is a complete word sequence
  - The list is ranked, based on how well they match the input speech

- Examples of N-best lists, where N=2:
  - IT IS HARD TO RECOGNIZE SPEECH
  - IT IS HARD TO WRECK A NICE BEACH

  Or,
  - I'LL NEVER BE A BEAST OF BURDEN (Rolling Stones song lyrics)
  - I'LL NEVER BE A PIZZA BURNING

Multiple Results from Recognizer (contd.)

- A lattice is a *graph* of all possible words that might have reasonably matched some segment of the input speech
  - Each word includes starting and ending time information
  - Each word can have other information, such as a measure of its match to the input
  - The graph is formed by linking together words where one ends (in time) and another begins
  - A lattice can be much more compact than an N-best list

- Same old analogy of text without spaces:

  ireturnedandsawunderthesunthattheraceisnottotheswift…

i	turn	a	saw	
ire	urn	an		
	turned			…
return		and		
returned				

Lattice (edges of graph not shown)

Preview of Topics in the Course

- Speech Signal capture
- Endpointing
- Feature extraction from speech signal (in brief)
- Template matching algorithm
- Hidden Markov modeling of speech
- Isolated word *vs* continuous speech recognition
- Small vocabulary *vs* large vocabulary considerations
- Pronunciation modeling
- Language modeling
- Obtaining multiple results from a recognizer
- Determining confidence in recognition results
- Accuracy and efficiency considerations
- Creating or training various models (in brief)

Confidence Estimation

- *Problem*: How confident are we that the recognizer's hypothesis is correct?
- The question can be posed at the word or the utterance level
  - Assign a confidence value to each word in the hypothesis, or,
  - To the entire utterance as a single unit
- Confidence value is usually stated as a probability ($0<p<1$)
- It is one of the harder problems in speech recognition
- Variety of *ad hoc* solutions
  - Often from estimating competition from other possible hypotheses
    - Qualitatively, the more the competition, the less the confidence in any proposed hypothesis
  - Actual implementations of the theme can vary widely

Preview of Topics in the Course

- Speech Signal capture
- Endpointing
- Feature extraction from speech signal (in brief)
- Template matching algorithm
- Hidden Markov modeling of speech
- Isolated word *vs* continuous speech recognition
- Small vocabulary *vs* large vocabulary considerations
- Pronunciation modeling
- Language modeling
- Obtaining multiple results from a recognizer
- Determining confidence in recognition results
- Accuracy and efficiency considerations
- Creating or training various models (in brief)

Accuracy and Efficiency

- The complexity of speech recognition increases rapidly with vocabulary size

- We will study various methods for improving computational efficiency
  - *Pruning*: limiting the set of hypotheses under evaluation
    - Specific version called *beam search*, used extensively in speech recognition
  - Algorithmic improvements such as *sharing*
    - Words under consideration might have the same root, *e.g.* READ and READING
    - Share the match computation for the portion that is common
  - *Lookahead*, also called *fast match*: using very lightweight models to rapidly eliminate words unlikely to match a given segment of input speech
  - Other techniques specific to dealing with large acoustic or language models

Accuracy and Efficiency (contd.)

- Some efficiency techniques worsen recognition accuracy
  - *E.g.* Pruning and lookahead can both prematurely rule out candidate words
- Improving accuracy usually requires better (more detailed, and finer-grained) models
  - Implies greater search complexity

- Accuracy and efficiency are on a trade-off curve
- Search for accuracy *and* efficiency is one of the holy grails of speech recognition

Preview of Topics in the Course

- Speech Signal capture
- Endpointing
- Feature extraction from speech signal (in brief)
- Template matching algorithm
- Hidden Markov modeling of speech
- Isolated word *vs* continuous speech recognition
- Small vocabulary *vs* large vocabulary considerations
- Pronunciation modeling
- Language modeling
- Obtaining multiple results from a recognizer
- Determining confidence in recognition results
- Accuracy and efficiency considerations
- Creating or training various models (in brief)

Training

- The models used by a recognizer must be *trained*
- Acoustic and language models are typically trained based on statistics gathered from known *training data*
  - The templates used in template matching are one such, although trivial, example
- Pronunciation models are largely hand-crafted
- The statistical training algorithms are quite different from the search algorithms used in decoding
  - Deal with probability, statistics and estimation theory
- We will briefly cover the well-known Baum-Welch algorithm for training HMMs for acoustic models
- We will also cover the creation of N-gram language models for large vocabulary recognition

Resources

- “Spoken Language Processing”, by Huang, Acero and Hon
  - Extensive references to virtually all topics in speech
- “Fundamentals of Speech Recognition”, by Rabiner and Juang
  - An early book but outstanding for details
- Several speech technology toolkits:
  - CMU Sphinx
    - Sphinx2, Sphinx3, Sphinx4, SphinxTrain
  - Cambridge HTK
    - The HTK Book
  - Kaldi, Julius, other open-source speech recognizers
  - Microsoft
- For brief introductions to various topics as well as references, *wikipedia* is great