# Natural Language

Larry Holder
School of EECS
Washington State University

# Natural Language: Why?

- Acquire information from written/spoken language
- Communicate with humans

# Language

- A language is a possibly infinite set of sentences
  - E.g., "The wumpus ate the agent." (S1)
- A grammar is a set of rules defining the syntax of acceptable sentences
  - Sentence → NounPhrase, Verb, NounPhrase
  - NounPhrase → Determiner, Noun
  - Noun → "wumpus" | "agent" | …
  - Verb → "ate" | …
  - Determiner → "a" | "an" | "the"
- Semantics expresses the meaning of sentences
  - eat(wumpus,agent,t1), ¬alive(agent,t2)

# Language Models

- Language as a set of sentences too brittle
  - S2: “Ate the agent, the wumpus did.” (Yodish)
- Language as a probability distribution over sentences
  - P(S1) > P(S2)
- Semantics also best expressed as a probability distribution over possible meanings P(M|S)
  - S: “The wumpus ate the agent with the gold.”
  - M1: ate(wumpus,agent) ^ ate(wumpus,gold)
  - M2: has(agent,gold) ^ ate(wumpus,agent)
- Natural languages are large and changing

# N-gram Word Models

- View language as a sequence of words
- $P(w_{1:N})$ = probability of sequence of N words $w_1$ to $w_N$
  - P("the wumpus ate the agent") = 0.00001
- N-gram is a sequence of N words
- N-gram model is the probability distribution over N-word sequences
- Special cases
  - 1-gram = "unigram"
  - 2-gram = "bigram"
  - 3-gram = "trigram"

# N-gram Word Models

- N-gram model is defined as a Markov chain of order N-1
  - I.e., next word depends only on previous N-1 words

$$P(w_{1:N}) = \prod_{i=1}^{N} P(w_i \mid w_{1:i-1})$$

  - E.g., bigram model

$$P(w_i \mid w_{1:i-1}) = P(w_i \mid w_{i-1})$$

$$P(w_{1:N}) = \prod_{i=1}^{N} P(w_i \mid w_{i-1})$$

# N-gram Word Models

- Probabilities estimated from training corpus
- Google books Ngram Viewer
  - http://books.google.com/ngrams
- Can encounter new words
  - P(new word) = ?
  - How often a previously unseen word first appears

# Bigram Word Model: Example

- Estimate probabilities from corpus
  - www.ngrams.info
- P("the wumpus ate the agent")

| Word 1 | Word 2 | Frequency |
|---|---|---|
| the | wumpus | 1,000 |
| wumpus | ate | 500 |
| ate | the | 10,000 |
| the | agent | 5,000 |

# Text Classification

- Classify text into one of a set of predefined categories
  - E.g., spam detection: text → {spam, ham}
- Approach
  - Given examples of spam and ham
    - E.g., www.kaggle.com/ishansoni/sms-spam-collection-dataset/notebook
  - Compute P(spam) and P(ham)
  - Compute n-gram language models
    - P(message|spam) and P(message|ham)
  - Classify new message using Bayes' rule:

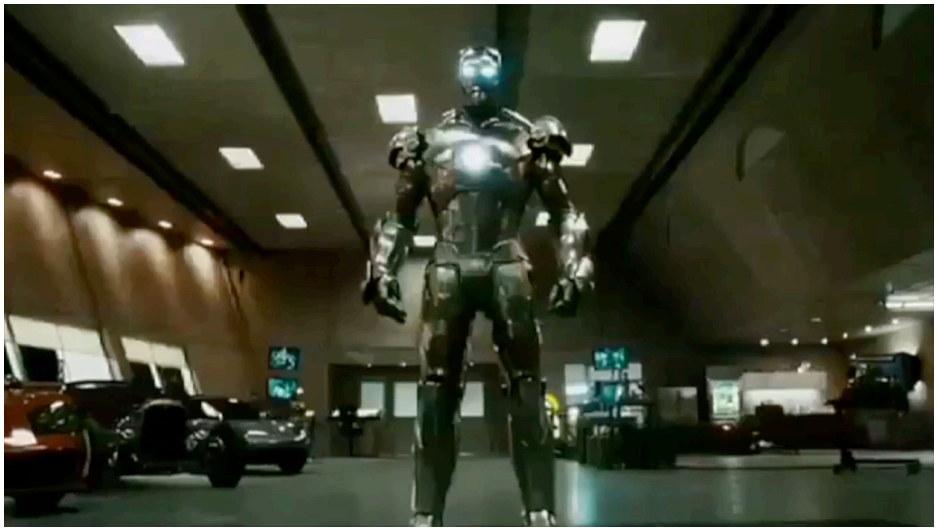
$$\arg\max_{c \in \{spam, ham\}} P(c \mid message) = \arg\max_{c \in \{spam, ham\}} P(message \mid c) P(c)$$

# Question Answering

- Want a short response to a well-formed question
- Approaches
  - Template-based
    - E.g., "What does a wumpus eat?" → [wumpus eats *]
  - Parser-based
    - Parse sentence and use POS tags to inform search
- Systems
  - Google?
  - START (start.csail.mit.edu)
  - WATSON

# Communication

- Natural language understanding and generation
- Speech recognition and synthesis



Iron Man (2008)

# Communication Challenges


- Stark: Jarvis, are you there?
- Jarvis: At your service, sir.
- Stark: Engage heads-up display.
- Jarvis: Check.
- Stark: Import all preferences from home interface.
- Jarvis: Will do, sir.
- Stark: All right, what do you say?
- Jarvis: I have indeed been uploaded, sir. We're online and ready.

# Communication Challenges

- Stark: Can we start the virtual walk-around?
- Jarvis: Importing preferences and calibrating virtual environment.
- Stark: Do a check on control surfaces.
- Jarvis: As you wish.
- Jarvis: Test complete. Preparing to power down and begin diagnostics.
- Stark: Uh, yeah. Tell you what. Do a weather and ATC check. Start listening in on ground control.
- Jarvis: Sir, there are still terabytes of calculations needed before an actual flight is...
- Stark: Jarvis! Sometimes you gotta run before you can walk.

Indirect speech act

Assumptions
Planning


Priorities
Utilities
Risk

???

# Communication Challenges

- Stark: All right, let's see what this thing can do.
- Stark: What's SR-71's record?
- Jarvis: The altitude record for fixed-wing flight is 85,000 feet, sir.
- Stark: Records are made to be broken!  Come on!
- Jarvis: Sir, there is a potentially fatal build-up of ice occurring.
- Stark: Keep going!  Higher!
- Stark: Aaaaah!
- Stark: Kill power.

Altitude or speed?

# Communication Challenges

- Part 2


Iron Man (2008)

# Communication Challenges

- Stark: Notes. Main transducer feels sluggish at plus 40 altitude. Hull pressurization is problematic. I'm thinking icing is the probable factor.
- Jarvis: A very astute observation, sir.
- Jarvis: Perhaps, if you intend to visit other planets, we should improve the exosystems.
- Stark: Connect to the sys. co. Have it reconfigure the shell metals. Use the gold titanium alloy from the seraphim tactical satellite. That should ensure a fuselage integrity while maintaining power-to-weight ratio.
- Stark: Got it?
- Jarvis: Yes. Shall I render using proposed specifications?
- Stark: Thrill me.

Facetious? Humor?

What is "it"?

Prompting Planning

→ "yes"?

# Communication Challenges

- Jarvis: The render is complete.
- Stark: A little ostentatious, don't you think?
- Jarvis: What was I thinking? You're usually so discreet.
- Stark: Tell you what. Throw a little hot-rod red in there.
- Jarvis: Yes, that should help you keep a low profile.
- Jarvis: The render is complete.
- Stark: Hey, I like it. Fabricate it. Paint it.
- Jarvis: Commencing automated assembly. Estimated completion time is five hours.


Humor? Sarcasm?

Another indirect speech act

More humor, sarcasm?

# Communication Challenges

- Contextual cues: "Got it?"
- Personality, attitude
- Understanding of priorities
  - "Sir, there are still terabytes of …"

- Much reasoning behind interaction, but let's focus on the natural language

# General Approach

- Define syntax using a grammar
- Parse sentences using grammar
- Add semantics to grammar
- Meaning composed during parsing
- Example
  - "The wumpus ate the agent." → ate(wumpus,agent)

# Grammars

- A grammar is a set of rules describing how syntactically correct sentences are formed
- A phrase structure grammar describes sentences in terms of phrases (e.g., noun phrase, verb phrase)
- Phrases are composed of lexical categories, or parts of speech (e.g., noun, verb)
- Lexicon is the set of words in the language grouped according to lexical categories

# Grammars

- Grammar rule consists of a left-hand side and a right-hand side
  - E.g., S → NP VP  (S can be replaced by NP VP)
  - E.g., Noun → "wumpus" | "agent" | …
- Terminal symbols are words or strings
- Non-terminal symbols are like variables, referring to a set of phrases (other rules)

# Grammars

- Context-free grammar (CFG) allows only one non-terminal on the left-hand side of rules
  - E.g., S → NP VP
  - E.g., Noun → "wumpus" | "agent" | …
- Context-sensitive grammar constrains right-hand side of rules to have at least as many symbols as the left-hand side
  - E.g., A X B → A Y B
- Probabilistic context-free grammar (PCFG) assigns a probability to every sentence in the language
  - E.g. NP → Article Noun (0.6) | Noun (0.4)

# Wumpus World Lexicon

Noun → stench | breeze | glitter | wumpus | pit | agent | gold | …
Verb → is | see | smells | shoot | feel | stinks | grab | eat | …
Adjective → right | left | smelly | breezy | dead | …
Adverb → here | there | nearby | ahead | …
Pronoun → me | you | I | it| …
RelativePronoun → that | which | who | whom | …
Name → John | Mary | Boston | …
Article → the | a | an | every | …
Preposition → to | in | on | of | near | …
Conjunction → and | or | but | yet | …
Digit → 0| 1| 2| 3| 4| 5| 6| 7| 8| 9

- Open classes change often
  - Nouns, names, verbs, adjectives, adverbs
- Closed classes rarely change
  - Pronoun, relative pronoun, article, preposition, conjunction

# Wumpus World Grammar

S → NP VP | S Conjunction S

NP → Pronoun | Name | Noun | Article Noun | Article Adjectives Noun
| Digit Digit | NP PP | NP RelativeClause

VP → Verb | VP NP | VP Adjective | VP PP | VP Adverb

Adjectives → Adjective | Adjective Adjectives

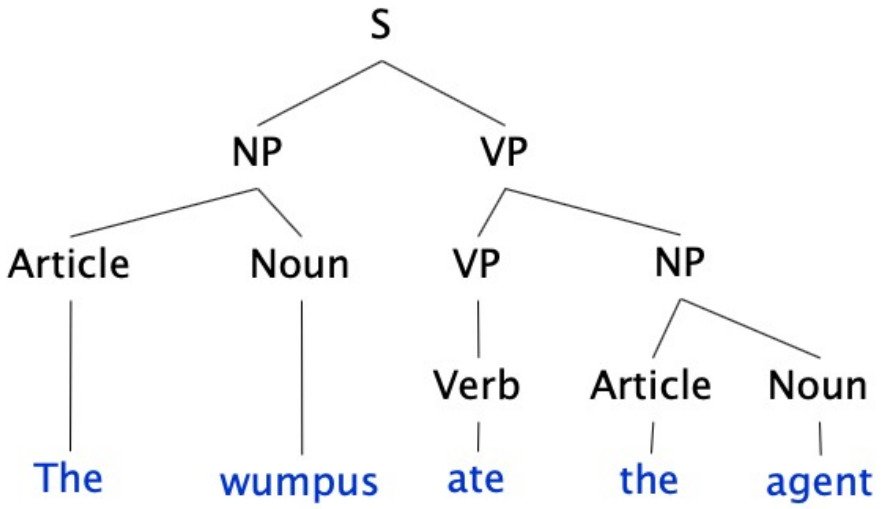
PP → Preposition NP

RelativeClause → RelativePronoun VP

- Overgenerates: accepts incorrect sentences
  - E.g., "I smells an wumpus."
- Undergenerates: rejects correct sentences
  - E.g., "The wumpus is not in 1 1."

# Parsing

- Parse tree is a proof that a sentence is consistent with (accepted by) the grammar
- Bottom-up parsing
  - Start with words of sentence
  - Use grammar rules in reverse to replace words with non-terminals until reach S



# Parsing

- E.g., “the agent smells the wumpus in 2 2”

the agent smells the wumpus in 2 2

# Parsing

- Stanford Parser
  - https://nlp.stanford.edu/software/
- Natural Language Toolkit (NLTK)
  - https://www.nltk.org/

- Parsing checks syntax, now how about semantics…

# Semantics

- Augmented grammar rules with variables and constraints on these variables
- Example
  - S(v(n1,n2)) → NP(n1) VP(v,n2) {agree(n1,v)}
  - "The wumpus eats the agent" → eats(wumpus,agent)
  - agree("wumpus","eats") = true
  - agree("wumpus","eat") = false

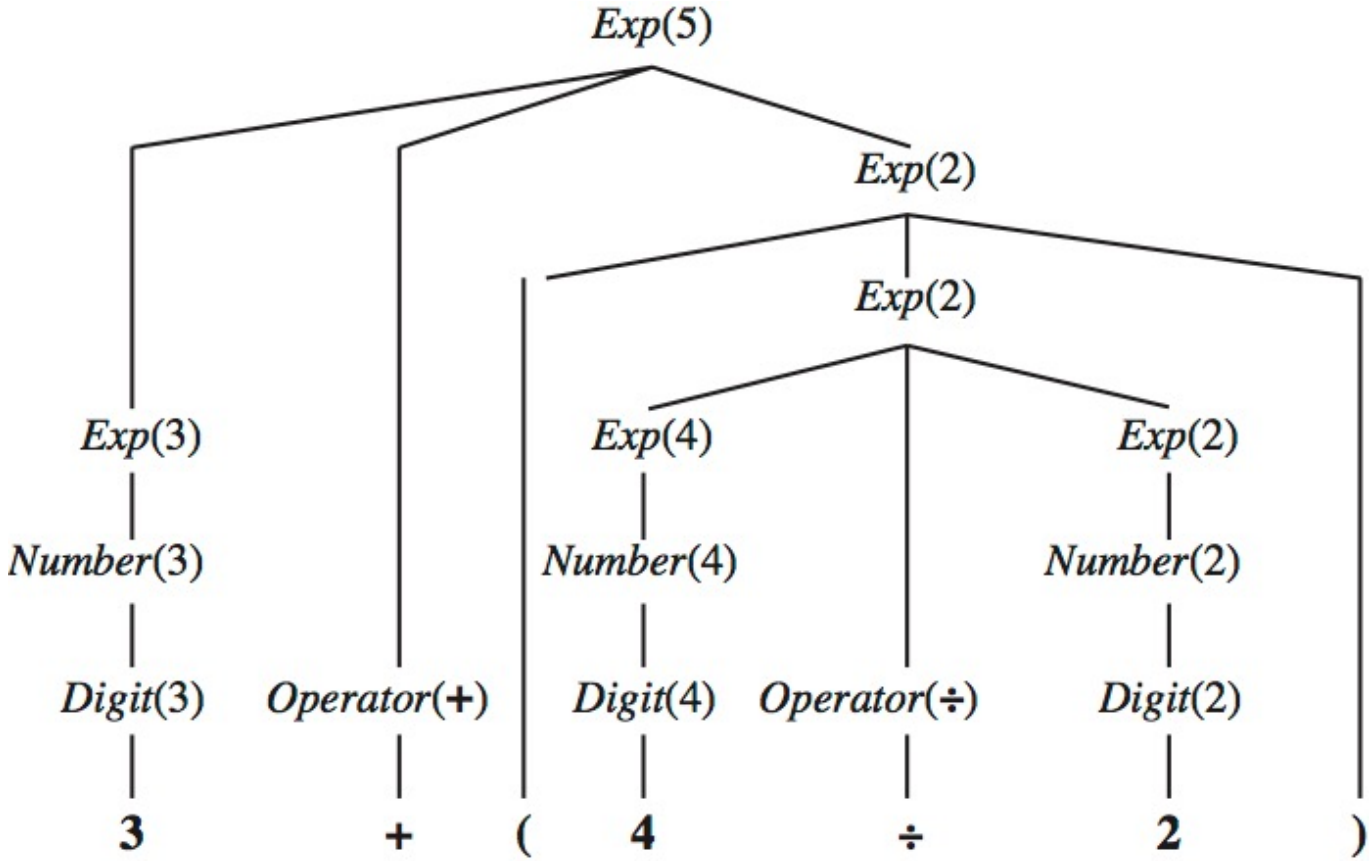
# Semantics (example)

- Augmented grammar for arithmetic expressions

| | |
|---|---|
| Exp(x) → Exp(x1) Operator(op) Exp(x2) | {x = Apply(op,x1,x2)} |
| Exp(x) → ( Exp(x) ) | |
| Exp(x) → Number(x) | |
| Number(x) → Digit(x) | |
| Number(x) → Number(x1) Digit(x2) | {x = 10 * x1 + x2} |
| Digit(x) → x | {0 ≤ x ≤ 9} |
| Operator(x) → x | {x ∈ {+,−,÷,×} } |

# Semantics (example)

- Parse tree for "3 + (4 ÷ 2)"



# Semantic Complications

- Tense (past, present, future)
  - "The wumpus ate the agent."
  - "The wumpus will eat the agent."
- Quantification
  - "There is a breeze next to every pit."
  - "There is a breeze next to the agent."
- Pragmatics
  - "I am now in location 1 2. It smells here."
  - "Shoot!"

# Semantic Complications

- Ambiguity
  - “The wumpus ate the agent with the gold.”
  - “The agent smells the wumpus in 2 2.”

- Disambiguation requires knowledge about the likelihood of different meanings
  - World model
  - Mental model
  - Language model
  - Acoustic model

# Natural Language Generation

- Internal representation? (e.g., logic)
- Template-based approach (e.g., Mitsuku)
- More sophisticated approach
  - Content determination: Facts
  - Document structuring: Order facts
  - Aggregation: Merge similar sentences
  - Lexical choice: Choose words
  - Referring expression generation: Use pronouns
  - Realization
    - Check against syntax and morphology
    - Output text
    - Simple NLG (github.com/simplenlg/simplenlg)

# Speech Recognition

- Identify a sequence of spoken words from an acoustic signal
- Successful systems available
- Built-in to MacOS, Windows
- Numerous applications
  - Menu systems
  - Navigation
  - Commands
  - Information retrieval
  - Dictation



Google Assistant

Siri

# Speech Recognition: Challenges

- Ambiguity: Different utterances sound the same
  - "recognize speech"
  - "wreck a nice beach"
- Segmentation: No pauses between words
- Coarticulation: Words spoken quickly form new sounds ("tomato")
- Homophones: Different words that sound the same
  - "to", "too", "two"

# Speech Recognition: Approach

- Given acoustic signal ($sound_{1:t}$)
  - Sequence of features extracted from signal
- Identify sequence of words ($word_{1:t}$)

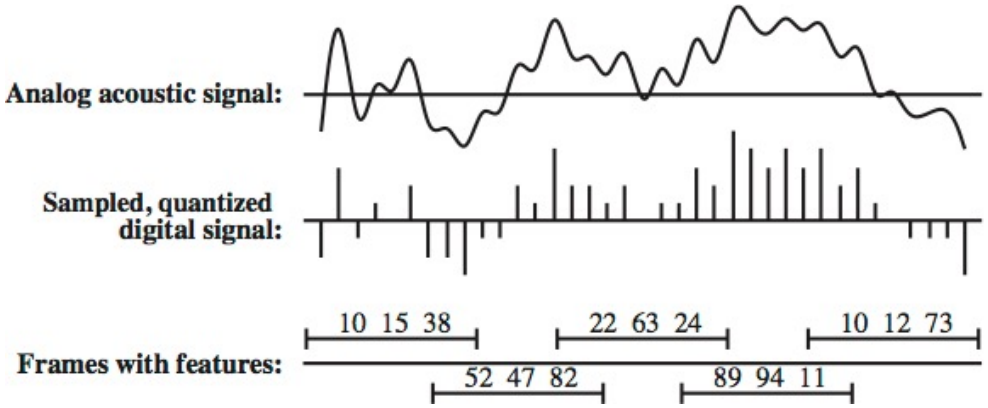
$$\arg\max_{word_{1:t}} P(word_{1:t} \mid sound_{1:t}) =$$

$$\arg\max_{word_{1:t}} P(sound_{1:t} \mid word_{1:t}) P(word_{1:t})$$

- Acoustic model: $P(sound_{1:t} \mid word_{1:t})$
- Language model: $P(word_{1:t})$
  - P("recognize speech") > P("wreck a nice beach")

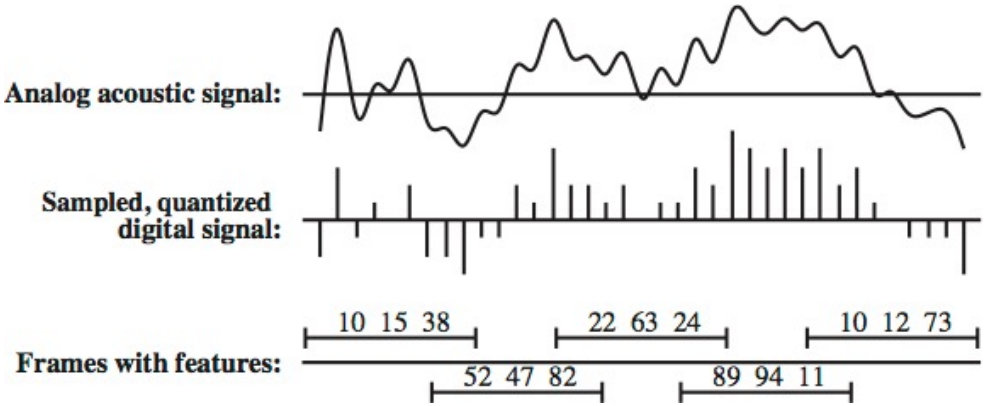
# Acoustic Model

- Sampling rate of acoustic signal
  - Typically 8 kHz
- Quantization factor (bits per sample)
  - Typically 8–12 bits
- Frames are overlapping time windows on the acoustic signal
  - Typically 10 milliseconds
- Feature vector summarizes each frame



# Acoustic Model: Features

- Dominate frequencies (Fourier transform)
- Difference from previous frame
- Difference between differences
- Discretize values



# Acoustic Model: Phones

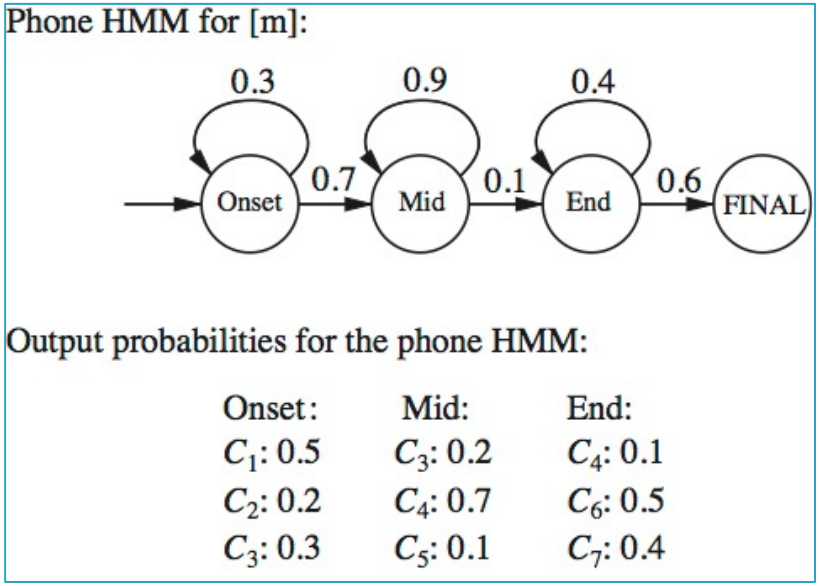
- Distinguish between about 100 sounds, or phonemes, used to compose words
- About 50 for all of English

| Vowels | | Consonants B–N | | Consonants P–Z | |
|---|---|---|---|---|---|
| Phone | Example | Phone | Example | Phone | Example |
| [iy] | b**ea**t | [b] | **b**et | [p] | **p**et |
| [ih] | b**i**t | [ch] | **Ch**et | [r] | **r**at |
| [eh] | b**e**t | [d] | **d**ebt | [s] | **s**et |
| [æ] | b**a**t | [f] | **f**at | [sh] | **sh**oe |
| [ah] | b**u**t | [g] | **g**et | [t] | **t**en |
| [ao] | b**ough**t | [hh] | **h**at | [th] | **th**ick |
| [ow] | b**oa**t | [hv] | **h**igh | [dh] | **th**at |
| [uh] | b**oo**k | [jh] | **j**et | [dx] | bu**tt**er |
| [ey] | b**ai**t | [k] | **k**ick | [v] | **v**et |
| [er] | B**er**t | [l] | **l**et | [w] | **w**et |
| [ay] | b**uy** | [el] | bott**le** | [wh] | **wh**ich |
| [oy] | b**oy** | [m] | **m**et | [y] | **y**et |
| [axr] | din**er** | [em] | bott**om** | [z] | **z**oo |
| [aw] | d**ow**n | [n] | **n**et | [zh] | mea**s**ure |
| [ax] | **a**bout | [en] | butt**on** | | |
| [ix] | ros**e**s | [ng] | si**ng** | | |
| [aa] | c**o**t | [eng] | washi**ng** | [-] | *silence* |

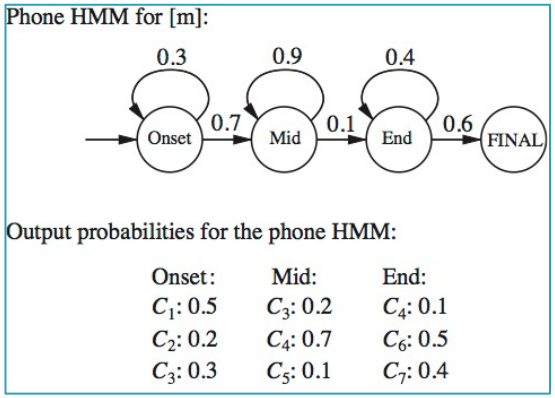
# Acoustic Model

- Phone model
  - Predict phone given sequence of frame features
- Hidden Markov Model (HMM)
  - Observations $\mathbf{e}_t$ are the frame features
  - Hidden states $\mathbf{x}_t$ are the components of the phone: Onset, Mid, End
  - $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-1})$
  - $\mathbf{P}(\mathbf{E}_t|\mathbf{X}_t)$
  - Learned

$C_i$ are feature values



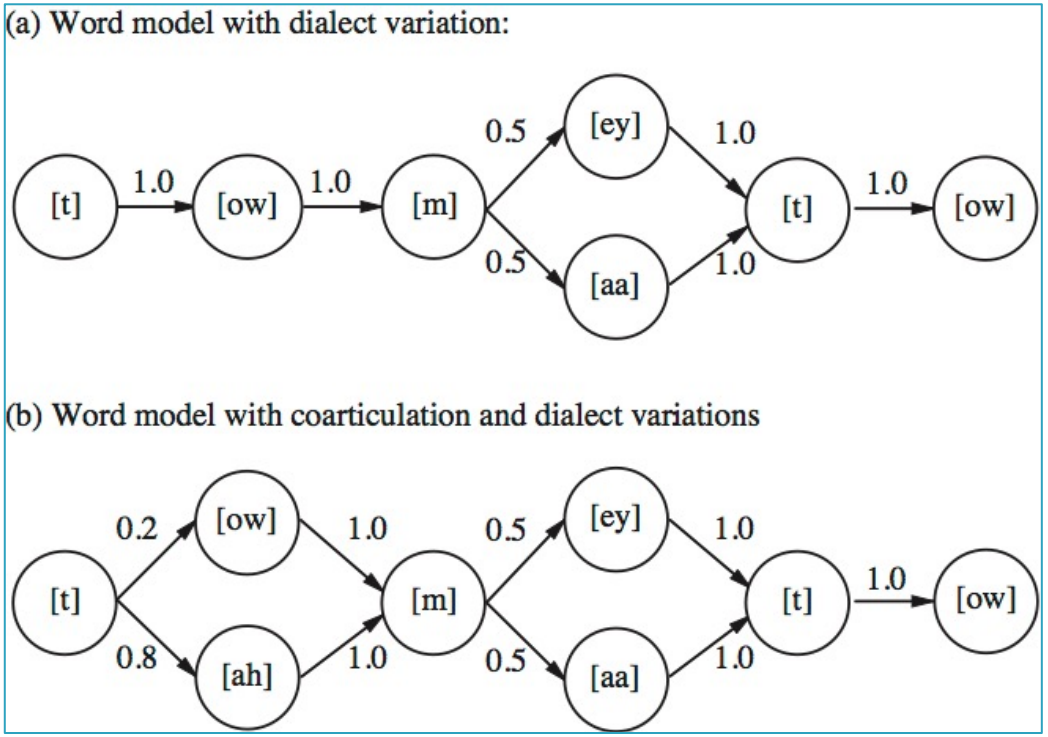
# Acoustic Model: Evaluation




- Given sequence of frame features
  - E.g., $C_1, C_2, C_3, C_4, C_6$
- What is the P(phone|sequence)?
  - Onset,Onset,Onset,Mid,End
    - (0.5)*[(0.3)(0.2)]*[(0.3)(0.3)]*[(0.7)(0.7)]*[(0.1)(0.5)]*(0.6)
    - $= 3.97 \times 10^{-5}$
  - Or: Onset,Onset,Mid,Mid,End
  - Or:
- Which is most likely explanation of features?
- Viterbi algorithm

# Acoustic Model

- Pronunciation (word) model
  - Predict word given sequence of phones
- Construct and train HMM for each word
- Use Viterbi to find most likely word



# Language Model

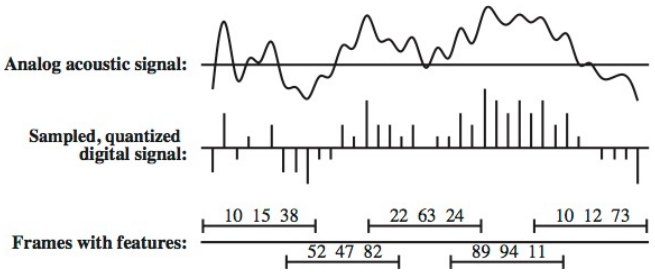
- Standard n-gram model from text corpus
- Better: Corpus of transcribed audio
- Even better: Task specific (e.g., flight information system)
  - Limits vocabulary (e.g., Amtrak unlikely)
  - Includes task-specific words (e.g., SEATAC)
- Current systems can achieve 99% accuracy

# Speech Recognition Tools

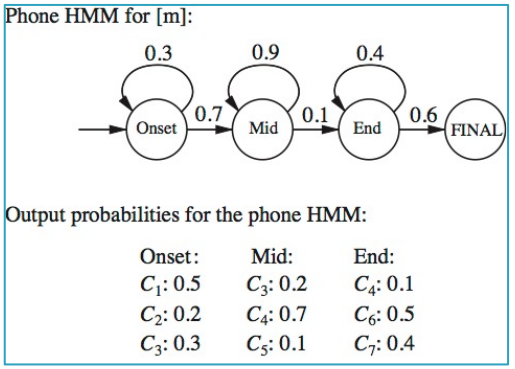
- Open source
  - Mozilla DeepSpeech (github.com/mozilla/DeepSpeech)
  - CMUSphinx (cmusphinx.github.io)
- Commercial
  - Dragon (www.nuance.com)
- Built-in to Windows, MacOS, and mobile platforms
- Corpora
  - Linguistic Data Consortium (www.ldc.upenn.edu)
  - Mozilla Common Voice (voice.mozilla.org)

# Deep Learning for Speech Recognition

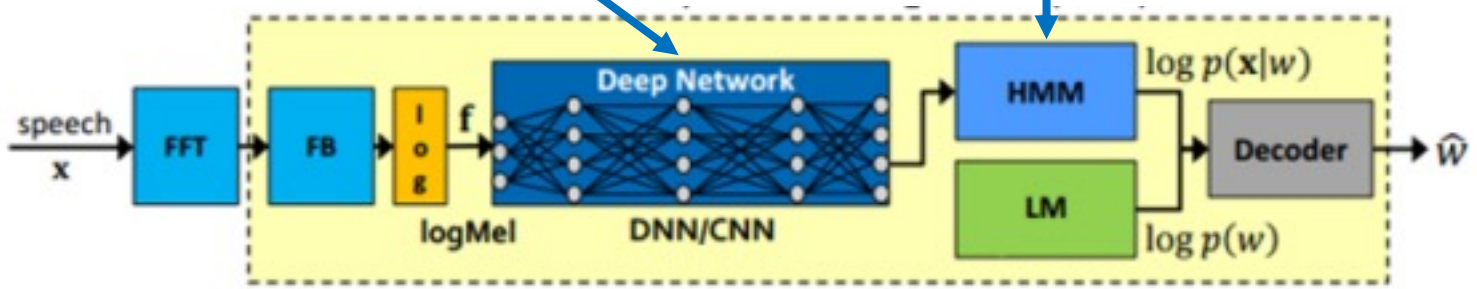
- Feature generation from acoustic signal




- Features → Phones

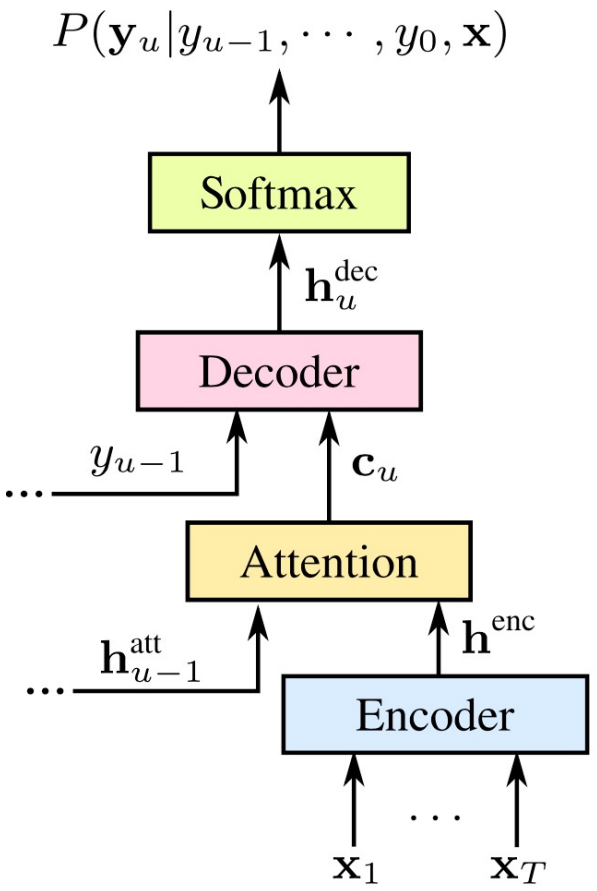



  - Recurrent NN
  - Long Short-Term Memory (LSTM) units



# Deep Learning for Speech Recognition

- Attention-based
- x = acoustic signal
- y = words
- h = features
- c = context
- Uses feedback
  - Previous words
  - Previous features
- No explicit acoustic or language models



# Speech Synthesis

- Approach 1: Invert acoustic model
  - Generate most likely sequence of phones for each word
  - Generate most likely sequence of frame features for each phone
  - Convert each frame to equivalent acoustic signal
- Approach 2: Invert word model
  - Generate most likely sequence of phones for each word
  - Concatenate and play pre-recorded phones
- Approach 3: Hybrid
  - Approach 2 with multi-word special cases

# Speech Synthesis

- Challenges
  - Intonation
  - Duration
  - Heteronyms (e.g., “project” as noun and verb)
  - Numbers
  - Abbreviations
- Most English language speech synthesizers are dictionary based

# Speech Synthesis Tools

- Open source
  - Festvox (www.festvox.org)
  - eSpeak (espeak.sourceforge.net)
- Built-in to Windows, MacOS, and mobile platforms

# Summary: Natural Language

- Natural language processing
  - Language models (n-gram)
  - Text classification
  - Question answering
  - Knowledge acquisition via reading
- Natural language communication
  - Grammars and parsing
  - Semantic interpretation
  - Language understanding and generation
  - Speech recognition and synthesis