Spoken Language Interaction

Introduction
NLP Basics
Speech Recognition
Prosody

Using *speech* to interact with systems

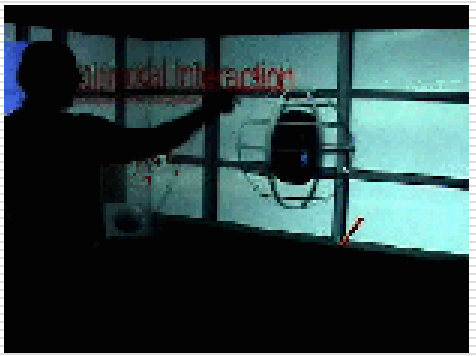
A vision?

Using *speech* to interact with systems

- Intuitive form of communication, no need for training
- Relates to way of thinking; *but* images, maps, …
- Paradigm: Computer adapts fully to the human

Speech interaction nowadays

- desktop systems
- telephone applications,
  e.g. ticket booking systems
- mobile devices
- automotive interaction
- Virtual Reality
- conversational agents
- mobile robot companions
- …



Spoken Dialogue Systems

- A system that allows a user to *speak* his queries in natural language and receive useful spoken *responses* from it
- Provides an interface between the user and a computer-based application that permits *spoken interaction* with the application in a "relatively natural manner"

Levels of sophistication

- Touch-tone replacement:

  System Prompt: "For checking information, press or say one."
  Caller Response: "One."

- Directed dialogue:

  System Prompt: "Would you like checking account information or rate information?"
  Caller Response: "Checking", or "checking account," or "rates."

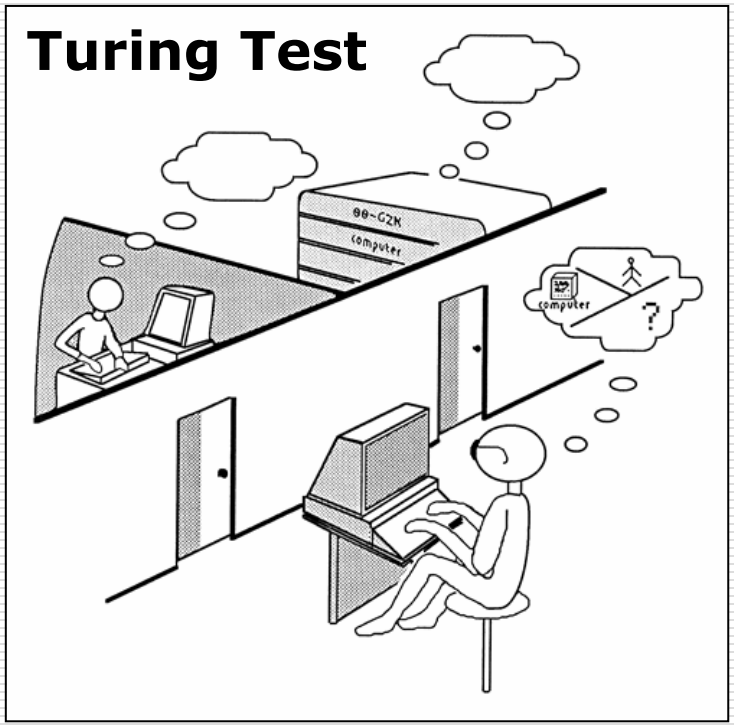
- Natural language:

  System Prompt: "What transaction would you like to perform?"
  Caller Response: "Transfer 500 dollars from checking to savings."

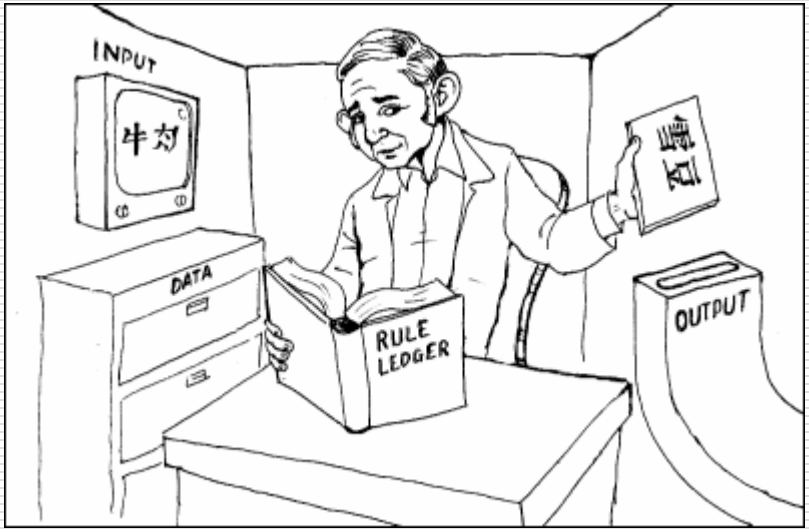
Levels of sophistication

Controlled language	limited task vocabulary, simple grammar (e.g. command language)
Natural language	huge vocabulary, complex grammar, grammatical variation, ambiguities, unclear sentence boundaries, omissions, word fragments
Natural dialog	turn-taking, initiative switch, discourse grounding, restarts, interruptions, interjections, speech repairs

Perfect natural dialog - „Holy Grail“ of AI



I propose to consider the question "Can machines think?" This should begin with definitions of the meaning of the terms "machine" and "think.“
[Turing, 1950]



Critics: Understanding not really needed (no intelligence?)

- “Chinese Room” (Searl, 1980)
- ELIZA (Weizenbaum, 1966)

Central aspects

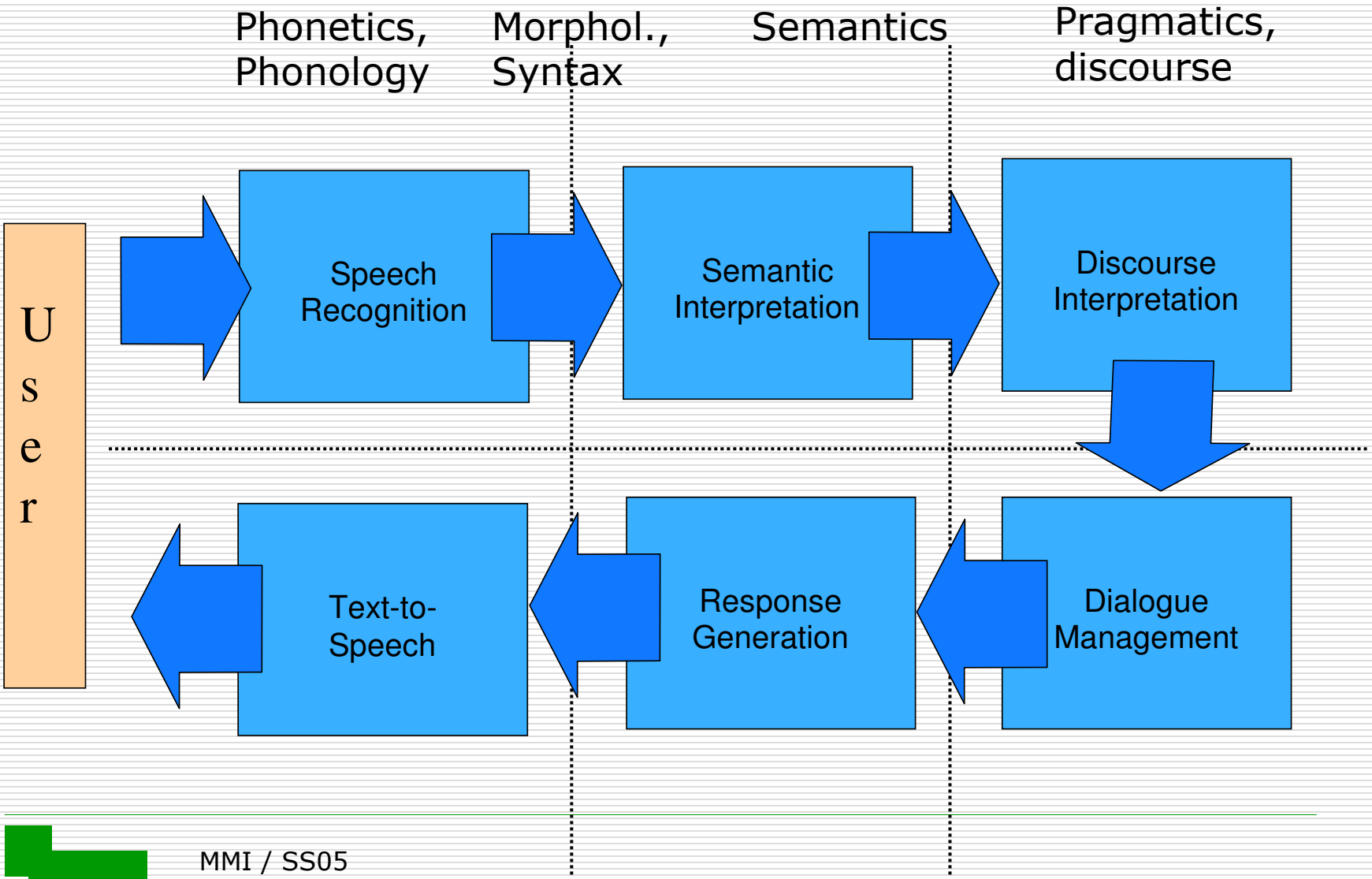
- *Natural language understanding*, NLU
  (Verarbeitung natürlich-sprachlicher Eingabe)
- *Natural language generation*, NLG
  (Generierung natürlich-sprachlicher Ausgaben)
- *Dialog management*, knowledge representation and reasoning
  (Führen eines konsistenten, zielgerichteten Dialogs)

Natural Language Processing, NLP = NLU + NLG

Natural language – levels to look at

- *Phonology and Phonetics*
  study of speech sounds and their usage
- *Morphology*
  study of meaningful components of words
- *Syntax*
  study of structural relationship between words
- *Semantics*
  study of meaning, of words (lexical semantics) and of word combinations (compositional semantics)
- *Pragmatics*
  study of how language is used to accomplish goals (said: „I'm cold" → meant: „shut the window")
- *Discourse*
  study of linguistic units larger than single utterances

Spoken Dialogue System - overview



Spoken Dialogue System - overview

- Speech Recognition:
  - Decode the sequence of feature vectors into a sequence of words.
- Syntactic Analysis and Semantic Interpretation:
  - Determine the meaning of the words.
- Discourse Interpretation:
  - Understand what the user intends by interpreting the utterances in context.
- Dialogue Management:
  - Determine system goals in response to user utterances based on user intention.
- Response Generation:
  - Express the system goals in natural utterances
- Text-to-speech:
  - Generate synthetic speech audio for the words

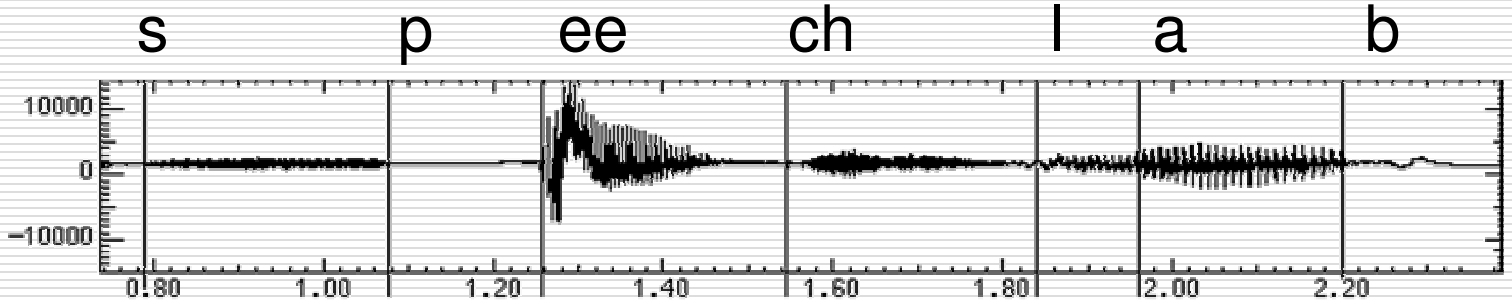
phonetic & phonological processing

Speech recognition

Text-to-speech

Starting and end point: acoustic waves

- Human speech generates a wave
- A wave for the words “speech lab”:



Basics

- *Phonetics*: study of speech sounds
  - *Phone* (*segment*) = speech sound, represented in some phonetic alphabet (e.g. IPA, „[t]“)
  - Phones = *vowels*, *consonants*
  - *Diphone*, *triphone*, … = combination of phones
  - *Syllables* = made up of vowels and consonants, not always clearly definable („syllabification problem“)
  - *Prominence* = *Accented* syllables that stand out
    - Louder, longer, pitch movement, or combination
  - *Lexical stress* = accented syllable if word is accented
    - „PARsley“
    - „CONtent“ (noun) vs „conTENT“ (adjective)

Basics (II)

- Phones have variation, get differently pronounced in different contexts
  - [t] in „tunafish“ → aspirated, voicelessness after it
  - [t] in „starfish“ → unaspirated

- *Phonology*: describes the systematic ways that sounds are differently realized
  - *Phoneme* = abstract, meaningful sound unit („/t/“), generalizes over different phonetic realizations (*allophones*)
  - *Phonological rules* = relation between phoneme and its allophones

Example: allophones of /t/

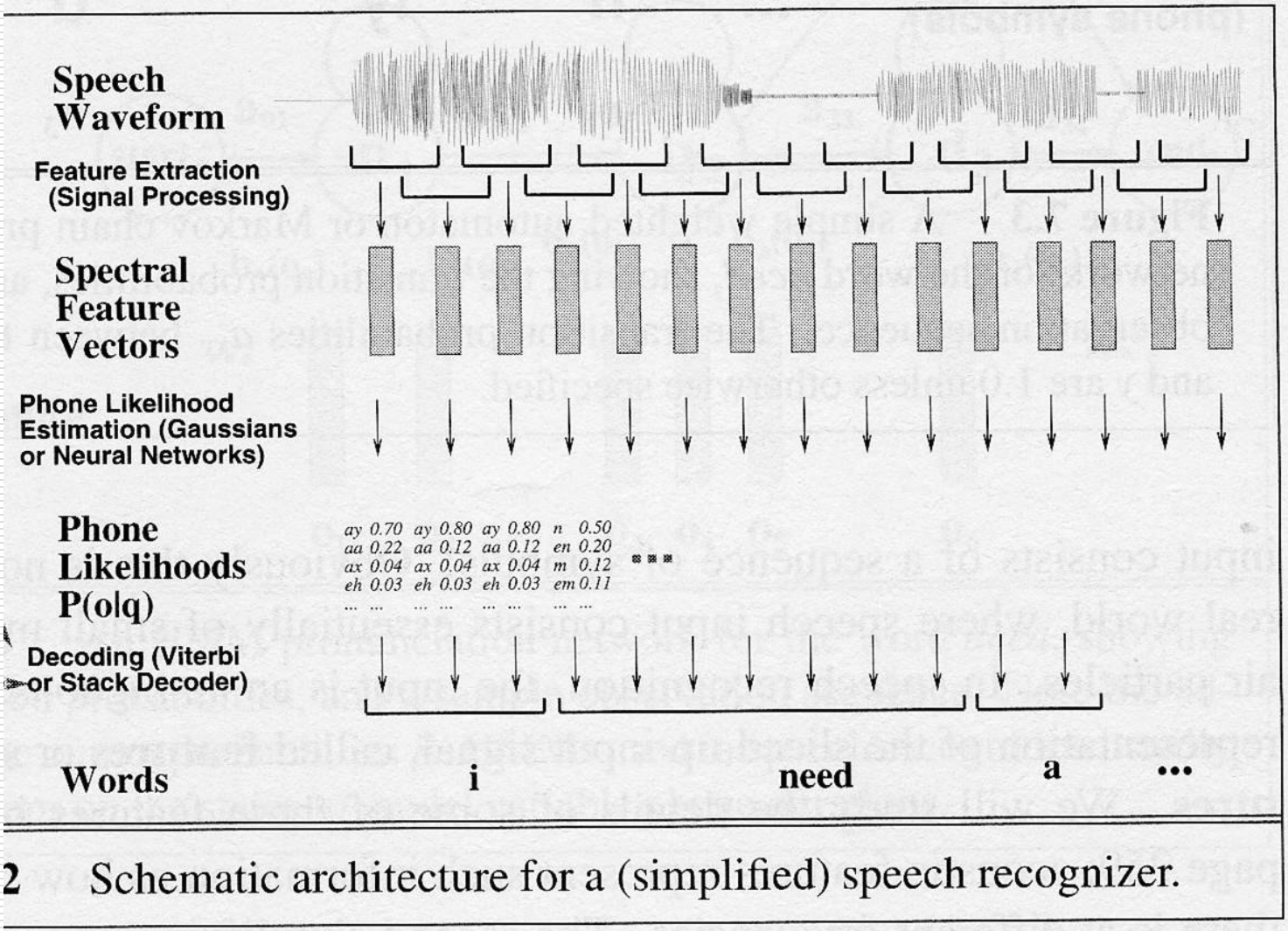
Phone	Environment	Example	IPA
[tʰ]	in initial position	*toucan*	[tʰukʰæn]
[t]	after [s] or in reduced syllables	*starfish*	[stɑrfɪʃ]
[ʔ]	word-finally or after vowel before [n]	*kitten*	[kʰɪʔn]
[ʔt]	sometimes word-finally	*cat*	[kʰæʔt]
[ɾ]	between vowels	*buttercup*	[bʌɾɚkʰʌp]
[t̚]	before consonants or word-finally	*fruitcake*	[frut̚kʰeɪk]
[t̪]	before dental consonants ([θ])	*eighth*	[eɪt̪θ]
[]	sometimes word-finally	*past*	[pæs]

Figure 4.8 Some allophones of /t/ in General American English.

(Jurafsky & Martin, 2000)

Speech recognition

Signal processing
Acoustic modeling
Language modeling
Decoding

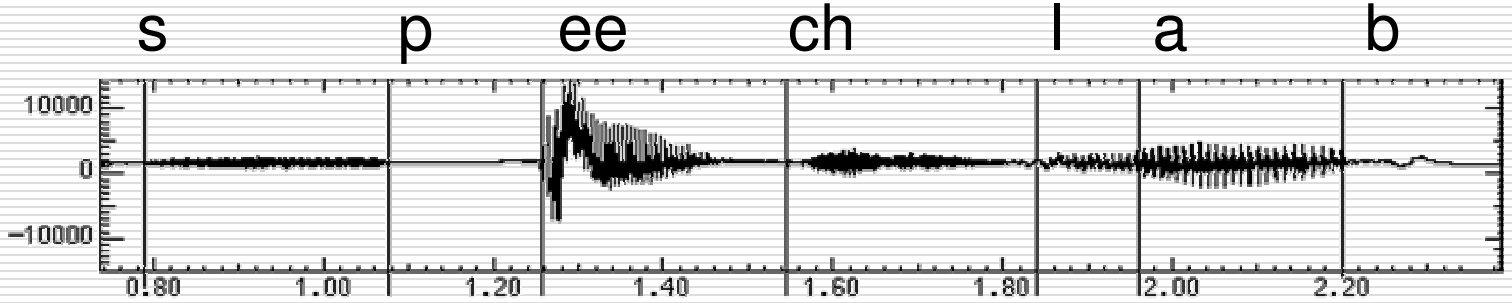


2 Schematic architecture for a (simplified) speech recognizer.

(Jurafsky & Martin, 2000)

Acoustic Waves

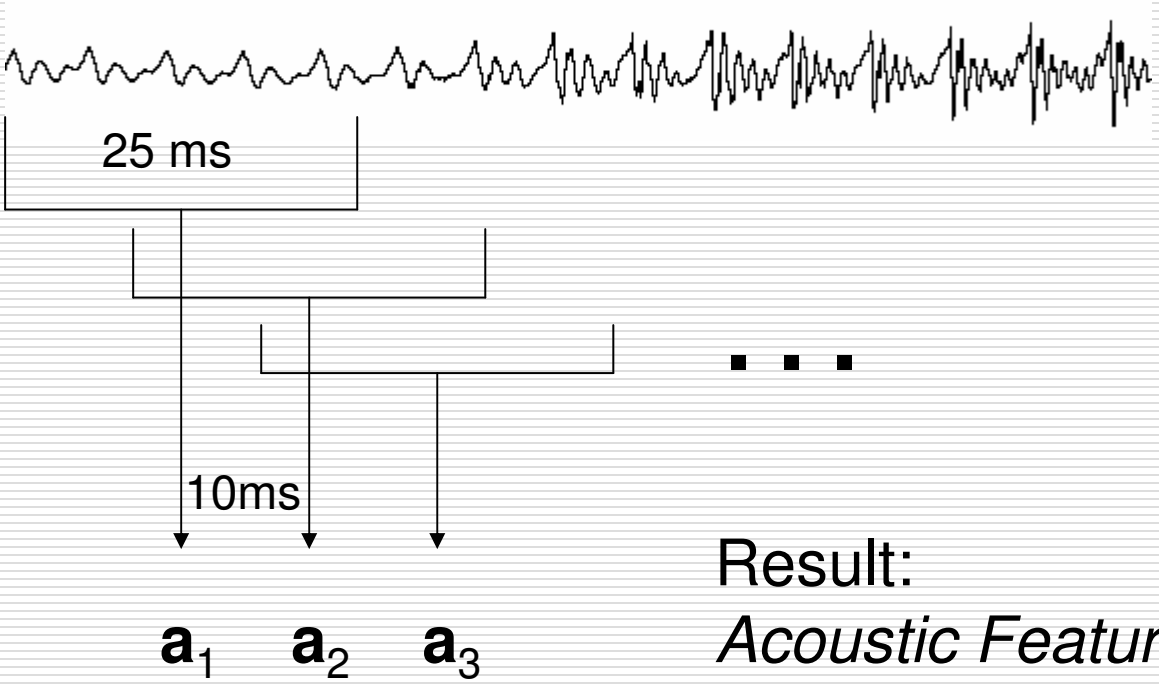
- A wave for the words “speech lab” looks like:



“l” to “a” transition:

Acoustic Sampling

- 10 ms frame (= 1/100 second)
- ~25 ms window around frame to smooth signal processing



Result:
Acoustic Feature Vectors

Signal Processing

Involves…

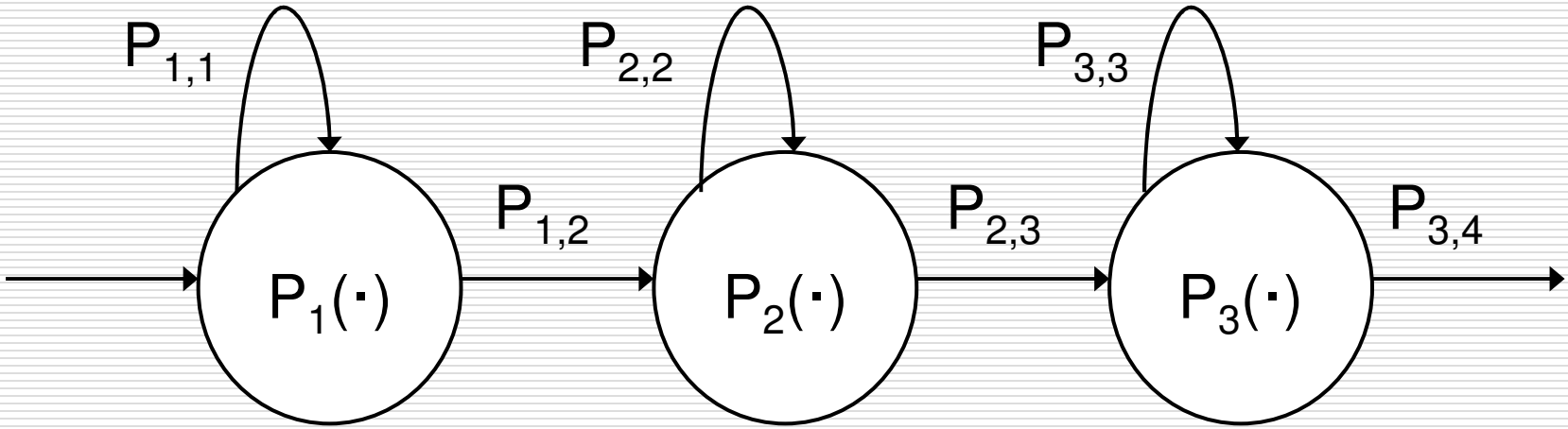
- *Pre-emphasis* to boost high level energy and minimize signal-to-noise ration
- *Spectral Analysis* break down into different frequencies using Fourier Transformation
- *Acoustic Features* to model non-linear human audio perception and create indep. features
- *Channel Adaptation* to deal with line and microphone characteristics
- *Echo Cancellation* to remove background noise
- Adding a *Total (log) Energy* feature (+/- normalization)
- *End-pointing* to detect signal start and stop

The Speech Recognition Problem

- *Bayes' law*
  - $P(a,b) = P(a|b)\,P(b) = P(b|a)\,P(a)$
  - Joint probability of *a* and *b* = probability of *b* times the probability of *a* given *b*
- the **recognition problem**
  - Find most likely sequence **w** of "words" given the sequence of acoustic observation vectors **a**
  - Use Bayes' law to create a *generative model*
  - $\text{ArgMax}_{\mathbf{w}}\ P(\mathbf{w}|\mathbf{a}) = \text{ArgMax}_{\mathbf{w}}\ P(\mathbf{a}|\mathbf{w})\,P(\mathbf{w}) / P(\mathbf{a})$
    $= \text{ArgMax}_{\mathbf{w}}\ P(\mathbf{a}|\mathbf{w})\,P(\mathbf{w})$
- *acoustic model*: $P(\mathbf{a}|\mathbf{w})$
- *language model*: $P(\mathbf{w})$

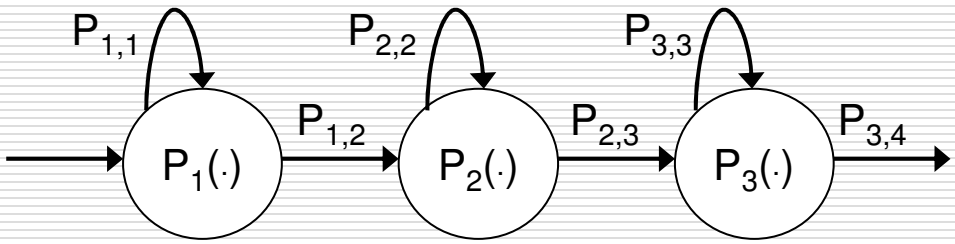
Hidden Markov Models (HMMs)

- HMMs provide *acoustic models* P(**a**|**w**)
- probabilistic, non-deterministic FSA
  - state *n* generates feature vectors with density P_n
  - transitions from state *j* to *n* are probabilistic $P_{j,n}$



Acoustic modeling with HMMs

- Train HMMs to represent *subword* units
- Units typically segmental; may vary in granularity
  - phonological (~40 phonemes for English)
  - phonetic (~60 phones for English)
  - *context-dependent triphones* (~14,000 for English): models temporal+spectral transitions between phones
  - *silence* and *noise* are usually additional symbols
- standard architecture is three successive states per phone:

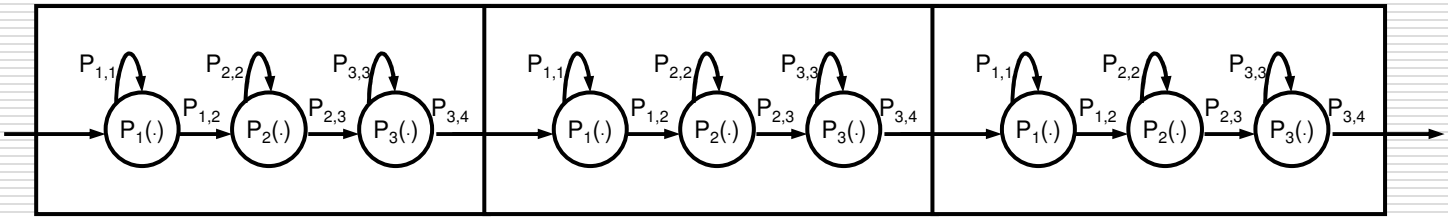


Lexical HMMs

- Create *compound* HMM for each lexical entry by concatenating the phones for the pronunciation
  - example of HMM for 'lab' (following 'speech' for crossword triphone)

phone: l a b

triphone: ch-**l**+a l-**a**+b a-**b**+#



- Multiple pronunciations can be weighted by likelihood into compound HMM for a word
- (Tri)phone models are independent parts of word models

Probabilistic language modeling

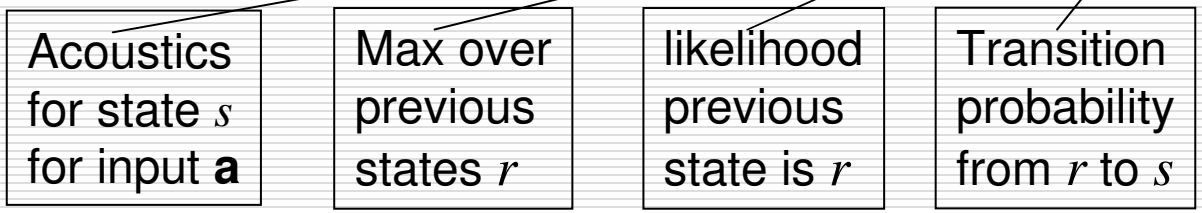
- Assign probability P(**w**) to word sequence
  w = $w_1, w_2, \ldots, w_k$
- Bayes' law provides a *history-based* model:
  $P(w_1, w_2, \ldots, w_k) = P(w_1)\, P(w_2|w_1)\, P(w_3|w_1,w_2) \cdots P(w_k|w_1,\ldots,w_{k-1})$
- *Cluster* histories to reduce number of parameters
- ***n*-gram** assumption clusters based on last *n*-1 words
  - $P(w_j|w_1,\ldots,w_{j-1}) \sim P(w_j|w_{j-n-1},\ldots,w_{j-2}, w_{j-1})$
  - unigrams ~ $P(w_j)$
  - bigrams ~ $P(w_j|w_{j-1})$
  - trigrams ~ $P(w_j|w_{j-2}, w_{j-1})$
- Possible extensions of language model
  - adapt over time
  - include indication of semantic topic
  - estimate categories (syntactic and/or semantic)

HMM Decoding

- Given a sequence of acoustic features (a_i), for which we don't know where the word boundaries are
- **Decoding problem** is finding best word sequence:
  - **ArgMax** $_{w1,\ldots,wm}$ $P(w_1,\ldots,w_m \mid \mathbf{a}_1,\ldots,\mathbf{a}_n)$
- Words $w_1 \ldots w_m$ are fully determined by sequences of states (=subword units), but many state sequences produce the same words (different pronounciations,…)
- **Viterbi** assumption**:**
  - the word sequence derived from the most likely path will be the most likely word sequence, as would be computed over all paths
  - implies a dynamic programming invariant:
    if the best paths goes through S_i, it always includes the best path from the start up to S_i and including it
  - not always valid, hence other decodes used sometimes (A*, stack, …)

Viterbi algorithm

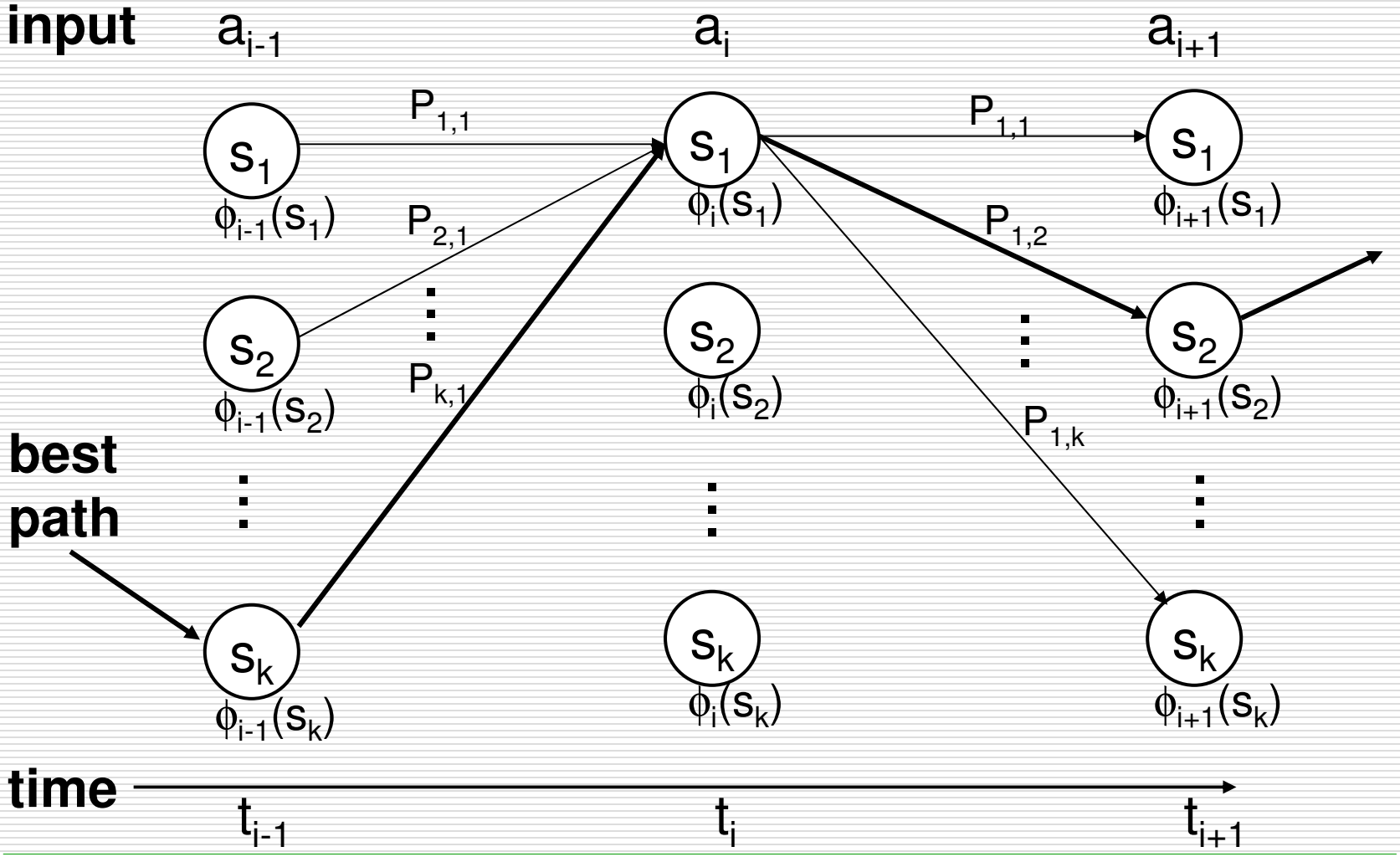
$$\phi_i(s) = \text{Max} \;\; \text{P}(s_1,\ldots,s_i \mid \mathbf{a}_1,\ldots,\mathbf{a}_i) = \text{P}_s(\mathbf{a}_i)\,\text{Max}_r\,\phi_{i-1}(r)\,\text{P}_{r,s}$$



- also known as *dynamic programming alignment*, *one-pass encoding*, *dynamic time warping*
- computes prob. matrix, with each cell (i,s_j) containing the probability of the *best* path, which accounts for the first i observations and ends in state s_j of the HMM
- dynamic programming: the best path at time t ending in state j is the best extension (with max prob.) of every possible previous path from time t-1 to t

Viterbi Decoding

$$\phi_i(s) = \text{Max}\ \text{P}(s_1,\ldots,s_i \mid \mathbf{a}_1,\ldots,\mathbf{a}_i) = \text{P}_s(\mathbf{a}_i)\,\text{Max}_r\,\phi_{i-1}(r)\,\text{P}_{r,s}$$



Example

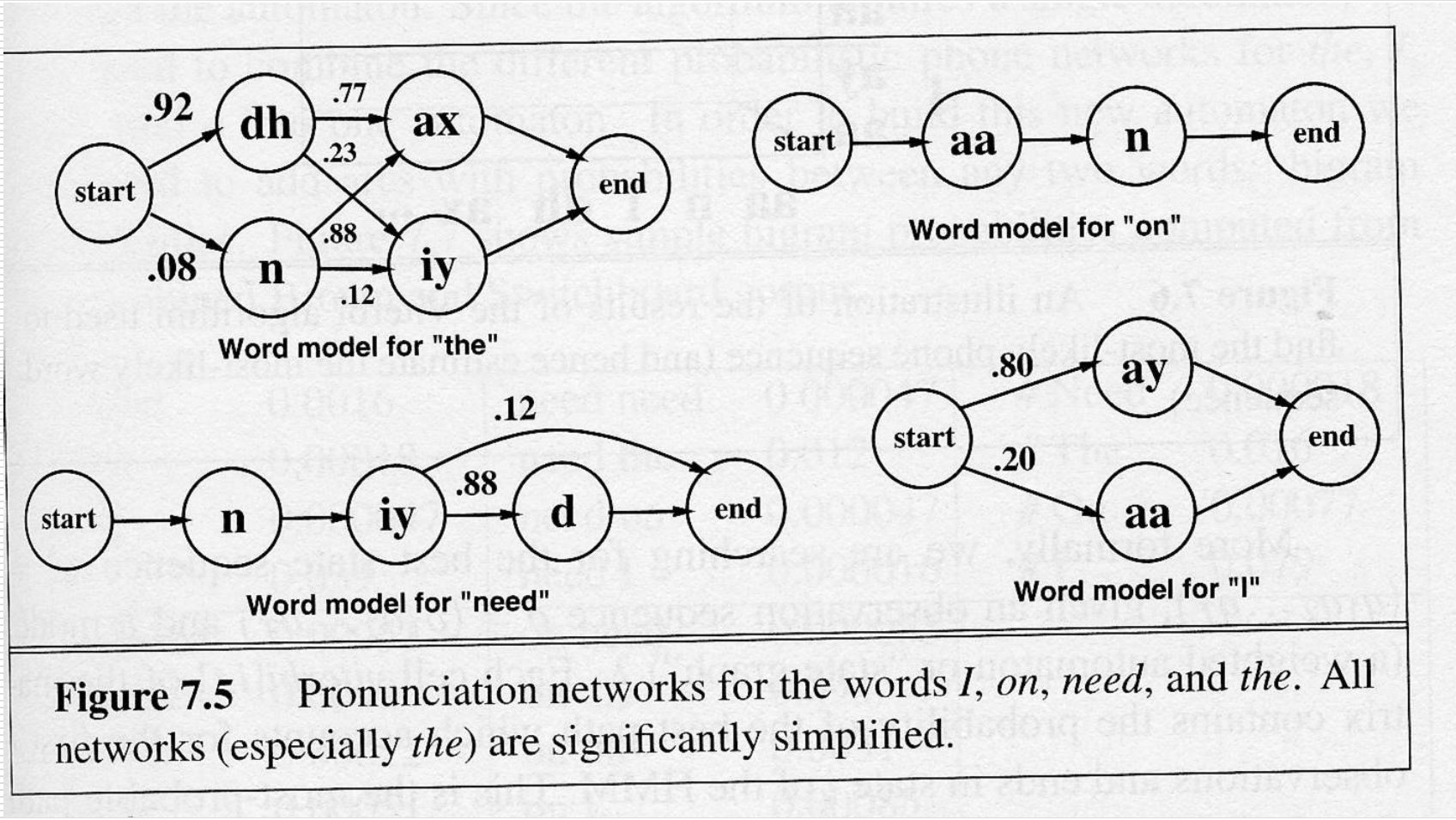


Figure 7.5 Pronunciation networks for the words *I*, *on*, *need*, and *the*. All networks (especially *the*) are significantly simplified.

(Jurafsky & Martin, 2000)

Example

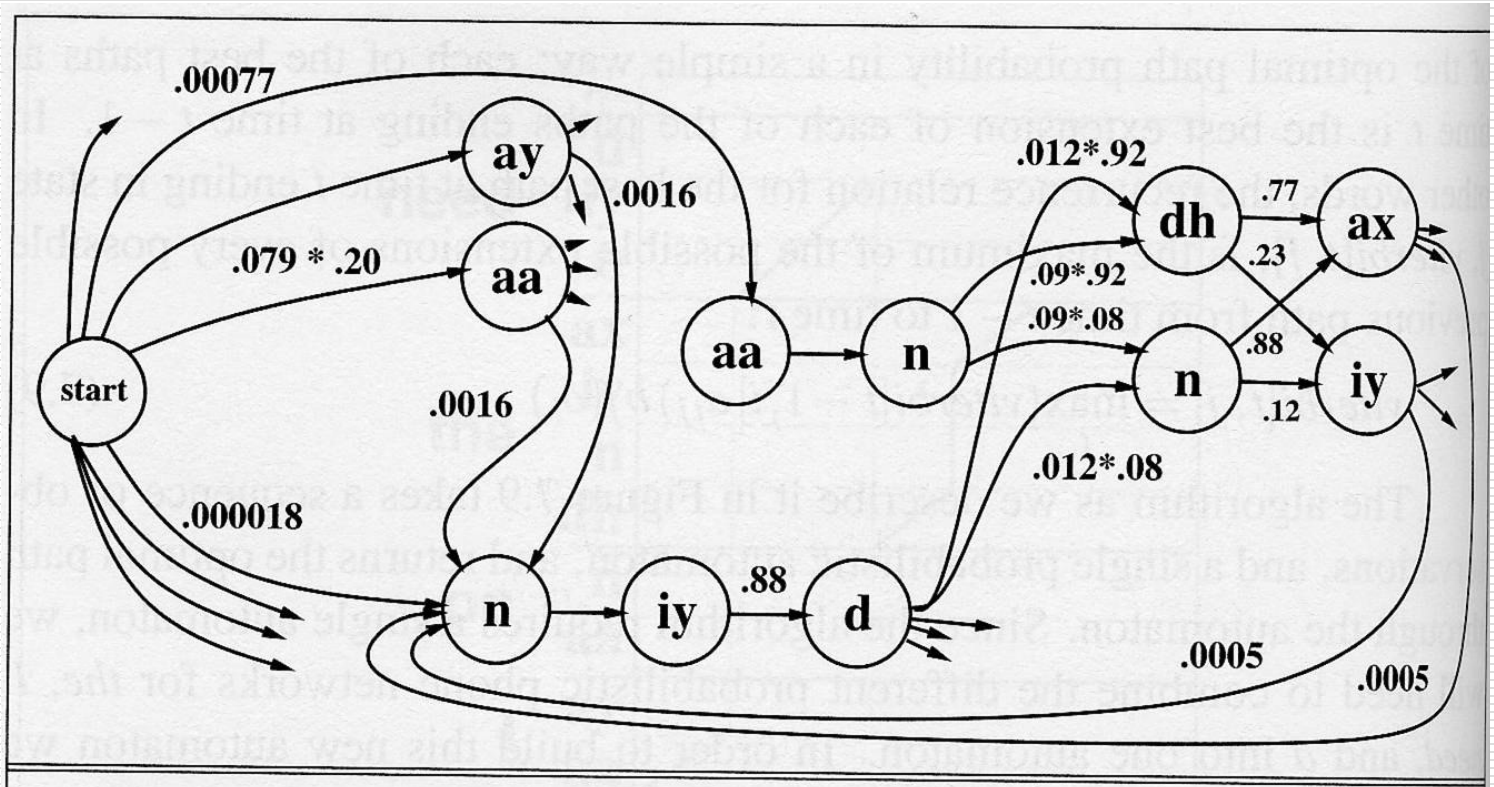


Figure 7.8 Single automaton made from the words *I*, *need*, *on*, and *the*. The arcs between words have probabilities computed from Figure 7.7. For lack of space the figure only shows a few of the between-word arcs.

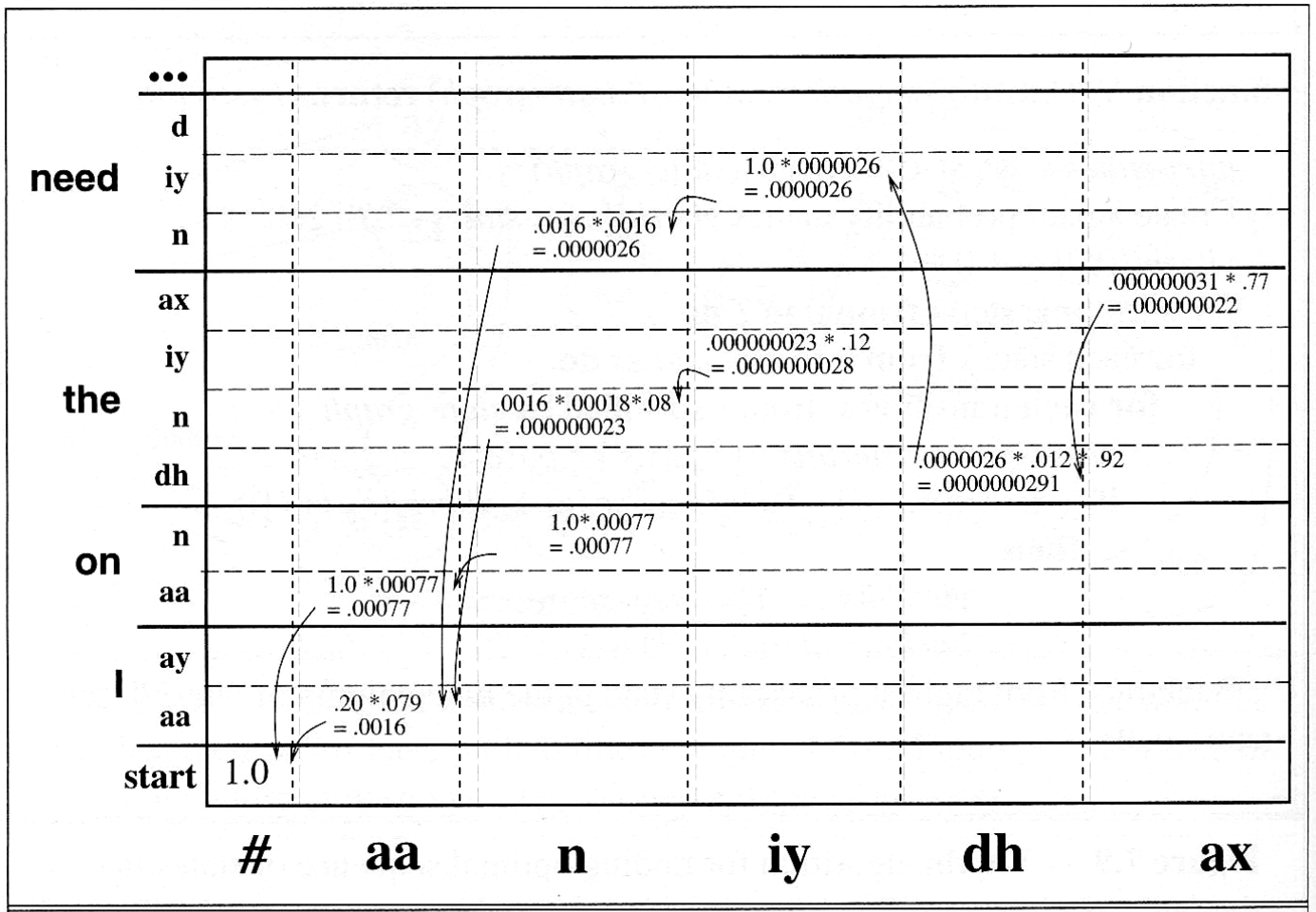


Figure 7.10 The entries in the individual state columns for the Viterbi algorithm. Each cell keeps the probability of the best path so far and a pointer to the previous cell along that path. Backtracing from the successful last word (*the*), we can reconstruct the word sequence *I need the*.

Properties of Speech Recognizers

- Speaker:
  - independent vs. dependent
  - adapt to speaker vs. non-adaptive
- Speech:
  - recognition vs. verification
  - continuous vs. discrete (single words)
  - spontaneous vs. read speech
  - large vocabulary (2K-200K) vs. limited (2-200)
- Acoustics
  - noisy environment vs. quiet environment
  - high-res microphone vs. phone vs. cellular
- Performance
  - real time? low vs. high Latency
  - anytime results vs. final results

Text-to-speech

Basics
Phonological and phonetic aspects of prosody
Duration
Intonation
Tone sequence models (ToBI)

Text-to-speech

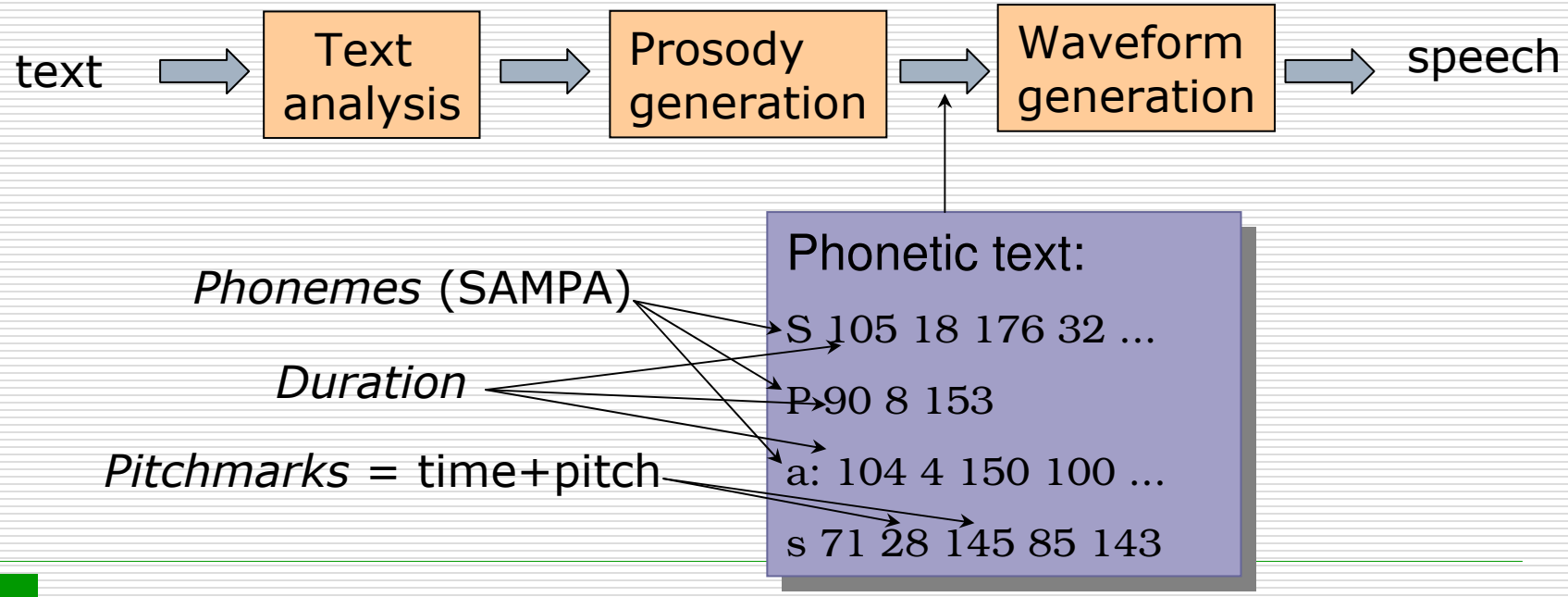
- Mapping text to phones

- The simplest (and most common) solution is to record prompts spoken by a (trained) human
- Produces human quality voice
- Limited by number of prompts that can be recorded
- Can be extended by limited cut-and-paste or template filling

Text-to-speech

Central tasks:

1. Analyse text and select the right sounds
2. Determine prosody
3. Turn into acoustic waveform (*speech synthesis*)



Prosody – phonological aspects

1. Prominence:
   - *Stress* or *accent* on syllables
   - Accent prediction complex: **app**le cake vs apple **pie**
2. Structure:
   - *Phrasing*: words that naturally group together
   - *Intonation phrase* (|), *intermediate phrase* (‘)
   - „I wanted ‘ to go ‘ to London, | but ‘ could only ‘ get tickets ‘ for France.“
3. Tune:
   - *Intonation*, mostly influenced by pitch
   - *Pitch accent*: characteristic pattern of pitch on stressed syllable

Prosody – functional aspects

Functional information can be encoded via combination of the above three features:

- *Lexical stress, sentence stress* determined by placement of words
- *Emphatic stress* determined by given/new information (information status)
  ("And then I saw a **church**.")
- *Contrastive* stress
  („I like **blue** tiles better than **green** tiles.")
- *Phrase boundaries* to signal clause structure
- Signal intention behind speech (statement/question/command)

Prosody - phonetic aspects

Phonological aspects are *perceived* sound qualities
→ caused by phonetic, psycho-acoustic features

- Prominent syllables are generally louder and longer
- Intonation determined by fundamental frequency (F0)
- Loudness created by duration, F0, amplitude
- Pitch accents manifested as F0 contours
- Phrase boundaries indicated by pauses, lengthened syllable just before, lowered pitch

Prosody in TTS

- Task: form a linguistic representations of prosody and generate acoustic patterns from them
- Output: {phoneme+pitch+duration}
  → input to waveform synthesis
- But, determining prosody is difficult!
  - Real-world knowledge needed
  - Semantic information needed
- Systems often shoot for „neutral declarative“ prosody → they sound wooden

„Accent is predictable (if you‘re a mind reader).“ (Bolinger, 1972)

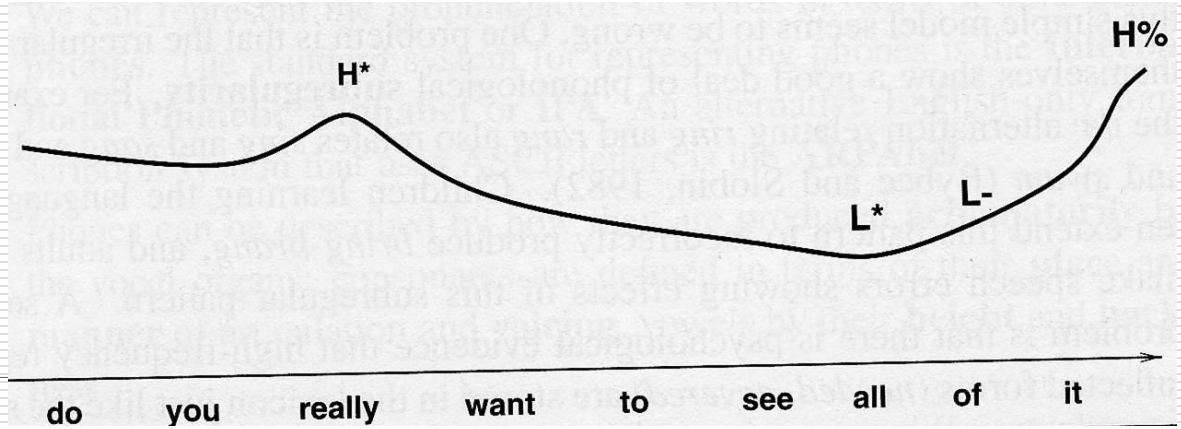
Duration modeling

Generate segments with appropriate duration. Influenced by

- Segmental identity
  - /ai/ in ‘like’ twice as long as /I/ in ‘lick’
- Surrounding segments
  - vowels longer following voiced fricatives than voiceless stops
- Syllable stress
  - stressed syllables longer than unstressed
- Word “importance”
  - word accent with major pitch movement lengthens
- Location of syllable in word
  - word ending longer than starting longer than word internal
- Location of the syllable in the phrase
  - phrase final syllables longer than in other positions

Intonation - tone sequence models

- Fundamental frequency (F0) contours generated from phonologically distinctive tones (High or Low) which are *locally independent* (Pierrehumbert, 1980)
- This gives a sequence of tonal *pitch targets* to fit with signal processing
- All intonation movements are *relative*
  - to each other (higher/lower instead of high/low)
  - to an overall *reference level,* which slightly declines over the intonation phrase



ToBI - Tone and Break Indices

- *Pitch Accent* (*): H*, L*, H*+L, H+L*, L*+H, L+H*
- *Phrase Accent* (-): H-, L-
- *Boundary Tone* (%): H%, L%
- *Intonation Phrase:* <Pitch A.>$^{+}$ <Phrase A.> <Bound.T.>

- Pitch accents cover important functional accents
  - 6 accents in German (GToBI; Reyelt et al.,1996):

H*		Gipfelakzent Emphase
L+H*		Steiler Anstieg auf Akzentsilbe Kontrastakzent
L*		Tonhöhe fällt auf Akzentsilbe ab oder bleibt dort

L*+H		Tieftonige Akzentsilbe gefolgt von Gipfel
H+L*		Abfall aus Höhenlage auf tieftonigeAkzentsilbe
H+!H*		Abfall auf abgesenkten Hochton *(downstep)*

		H*										L*		L- H%			
do		you		really				want				to		see		all		of		it	
d	uw	y	uw	r	ih	l	iy	w	aa	n	t	t	ax	s	iy	ao	l	ah	v	ih	t
110	110	50	50	75	64	57	82	57	50	72	41	43	47	54	130	76	90	44	62	46	220

Figure 4.25 Output of the FESTIVAL (Black et al., 1999) generator for the sentence *Do you really want to see all of it?* The exact intonation contour is shown in Figure 4.26. Thanks to Paul Taylor for this figure.

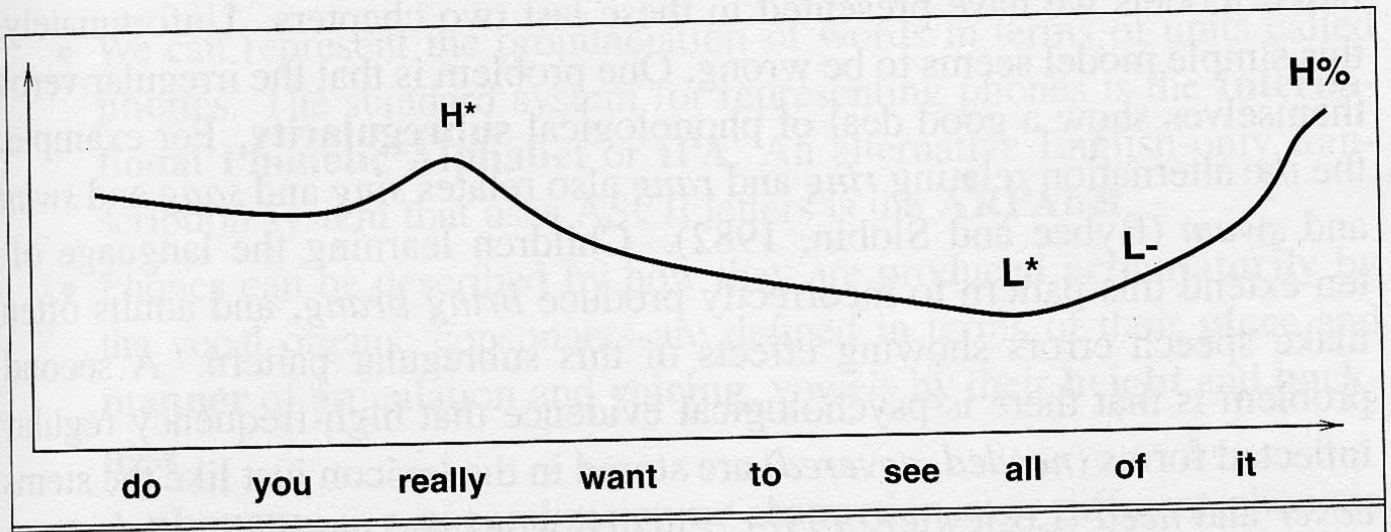


Figure 4.26 The F0 contour for the sample sentence generated by the FESTIVAL synthesis system in Figure 4.25, thanks to Paul Taylor.

Festival Synthesizer:
http://www.cstr.ed.ac.uk/projects/festival.html

Text Markup for TTS

- Bell Labs TTS Markup
  - r(0.9) L*+H(0.8) ***Humpty*** L*+H(0.8) ***Dumpty*** r(0.85) L*(0.5) ***sat on a*** H*(1.2) ***wall***.
  - Tones: Tone(Prominence)
  - Speaking Rate: r(Rate) and pauses
  - Top Line (highest pitch); Reference Line (reference pitch); Base Line (lowest pitch)
- SABLE: emerging XML standard
  - http://www.cstr.ed.ac.uk/projects/sable.html
  - marks: emphasis(#), break(#), pitch(base/mid/range,#), rate(#), volume(#), semanticMode(date/time/email/URL/...), speaker(age,sex)