Pronunciation Variation: TTS & Probabilistic Models

CMSC 35100

Natural Language Processing

April 10, 2003

Fast Phonology

Consonants: Closure/Obstruction in vocal tract

- Place of articulation (where restriction occurs)
  - Labial: lips (p, b), Labiodental: lips & teeth (f,v)
  - Dental: teeth: (th,dh)
  - Alvoelar:roof of mouth behind teeth (t.d)
  - Palatal: palate: (y); Palato-alvoelar: (sh, jh, zh)…
  - Velar: soft palate (back): k,g ; Glottal
- Manner of articulation (how restrict)
  - Stop (t): closure + release; plosive (w/ burst of air)
  - Nasal (n): nasal cavity
  - Frictative (s,sh,) turbulence: Affricate: stop+fricative (jh, ch)
  - Approximant (w,l,r)
  - Tap/Flap: quick touch to alvoelar ridge

Fast Phonology

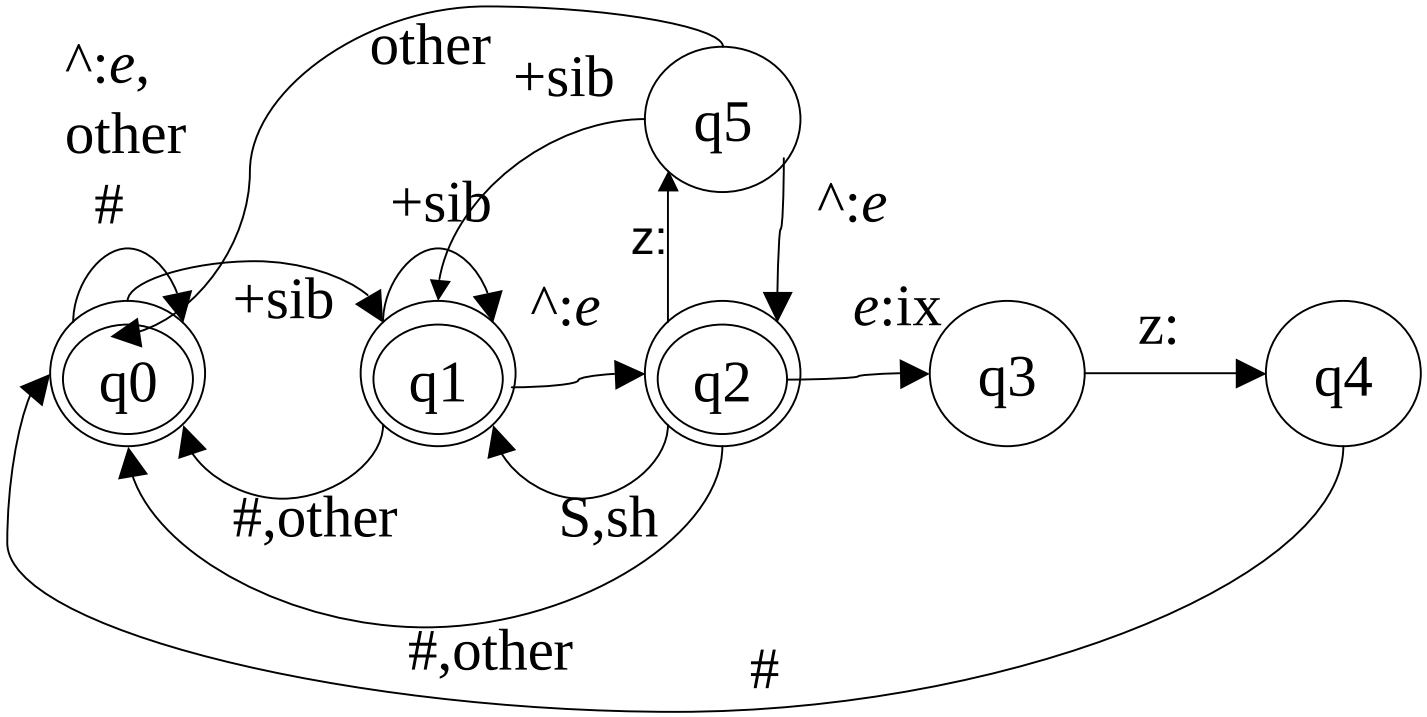
- Vowels: Open vocal tract: Articulator position
  - Vowel height: position of highest point of tongue
    - Front (iy) vs Back (uw)
    - High: (ih) vs Low (eh)
    - Diphthong: tongue moves: (ey)
  - Lip shape
    - Rounded: (uw)

Phonological Variation

- Consider t in context:
  - -talk: t – unvoiced, aspirated
  - -stalk: d – often unvoiced
  - -butter: dx – just flap, etc
- Can model with phonological rule
  - Flap rule: {t,d} -> [dx]/V'__V
    - T,d becomes flap when between stressed & unstressed vowel

Phonological Rules & FSTs

- Foxes redux:
  - [ix] insertion: *e*:[ix] <-> [+sibilant]:^_z



Harmony

- Vowel harmony:
  - Vowel changes sound be more similar to other
    - E.g. assimilate to roundness and backness of preceding
    - Yokuts examples:
      - dub+hin -> dubhun
      - xil+hin -> xilhin
      - Bok'+al -> bok'ol
      - Xat+al -> xatal
- Can also be handled by FST

Text-to-Speech

- Key components:
  - Pronouncing dictionary
  - Rules
- Dictionary: E.g. CELEX, PRONLEX, CMUDict
  - List of pronunciations
    - Different pronunciations, dialects
    - Sometimes: part of speech, lexical stress
  - Problem: Lexical Gaps
    - E.g. Names!

TTS: Resolving Lexical Gaps

- Rules applied to fill lexical gaps
  - Now and then
- Gaps & Productivity:
  - Infinitely many; can't just list
    - Morphology
    - Numbers
      - Different styles, contexts: e.g. phone number, date,..
    - Names
      - Other language influences

FST-based TTS

- Components:
  - FST for pronunciation of words & morphemes in lex
  - FSA for legal morpheme sequences
  - FSTs for individual pronunciation rules
  - Rules/transducers for e.g. names & acronyms
  - Default rules for unknown words

FST TTS

- Enrich lexicon:
  - Orthographic + Phonological
    - E.g. cat = c|k a|ae t|t; goose = g|g oo|uw s|s e|*e*
- Build FST for lexicon to intermediate
  - Use rich lexicon
- Build FSTs for pronunciation rules
- Names & Acronyms:
    - Liberman&Church: 50000 wd list
    - Generalization rules
      - Affixes: s, ville, son..; Compounds
      - Rhyming rules

Probabilistic Pronunciation

- Sources of variation:
  - Lexical variation: Represent in lexicon
    - Differences in what segments form a word
      - E.g. vase, brownsville
      - Sociolinguistic variation: e.g. dialect, register, style
  - Allophonic variation:
    - Differences in segment values in context
      - Surface form: phonetic & articulatory effects
        - » E.g. t: about
      - Coarticulation: Dis/Assimilation, Deletion, Flapping, Vowel reduction, epenthesis

The ASR Pronunciation Problem

Given a series of phones, what is the most probable word?

Simplification: Assume phone sequence known, word boundaries known

Approach: Noisy channel model

Surface form is an instance of lexical form that has passed through a noisy communication path

Model channel to remove noise, find original

Bayesian Model

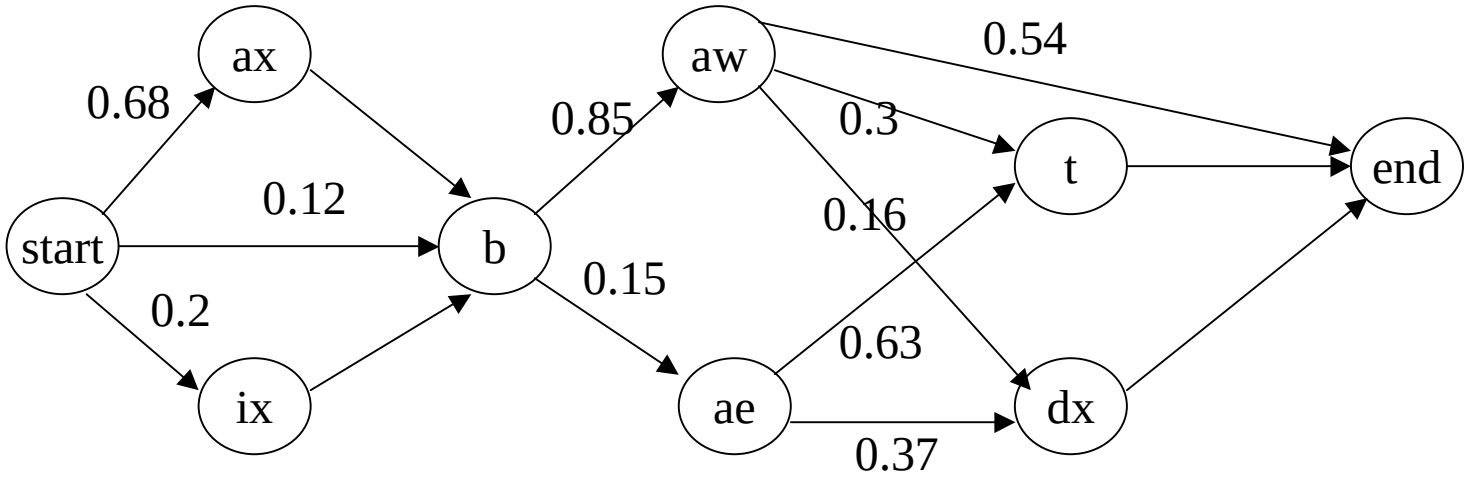
- Pr(w|O) = Pr(O|w)Pr(w)/P(O)
- Goal: Most probable word
  - Observations held constant
  - Find w to maximize Pr(O|w)*Pr(w)
- Where do we get the likelihoods? – Pr(O|w)
  - Probabilistic rules (Labov)
    - Add probabilities to pronunciation variation rules
      - Count over large corpus of surface forms wrt lexicon
- Where do we get Pr(w)?
  - Similarly – count over words in a large corpus

Automatic Rule Induction

- Decision trees
  - Supervised machine learning technique
    - Input: lexical phone, context features
    - Output: surface phone
  - Approach:
    - Identify features that produce subsets with least entropy
    - Repeatedly split inputs on features until some threshold
    - Classification:
      - Traverse tree based on features of new inputs
      - Assign majority classification at leaf

Weighted Automata

- Associate a weight (probability) with each arc
  - Determine weights by decision tree compilation or counting from a large corpus



Computed from Switchboard corpus

Forward Computation

- For a weighted automaton and a phoneme sequence, what is its likelihood?
  - Automaton: Tuple
    - Set of states Q: q0,…qn
    - Set of transition probabilities between states aij,
      - Where aij is the probability of transitioning from state i to j
    - Special start & end states
  - Inputs: Observation sequence: O = o1,o2,…,ok
  - Computed as:
    - forward[t,j] = $P(o1,o2\ldots ot,qt=j|\lambda)p(w)=\Sigma i$ forward[t-1,i]*aij*bjt
      - Sums over all paths to qt=j

Viterbi Decoding

- Given an observation sequence o and a weighted automaton, what is the mostly likely state sequence?
  - Use to identify words by merging multiple word pronunciation automata in parallel
  - Comparable to forward
    - Replace sum with max
- Dynamic programming approach
  - Store max through a given state/time pair

Viterbi Algorithm

Function Viterbi(observations length T, state-graph) returns best-path
Num-states<-num-of-states(state-graph)
Create path prob matrix viterbi[num-states+2,T+2]
Viterbi[0,0]<- 1.0
For each time step t from 0 to T do
  for each state s from 0 to num-states do
    for each transition s' from s in state-graph
      new-score<-viterbi[s,t]*at[s,s']*bs'(ot)
      if ((viterbi[s',t+1]=0) || (viterbi[s',t+1]<new-score))
        then
          viterbi[s',t+1] <- new-score
          back-pointer[s',t+1]<-s
Backtrace from highest prob state in final column of viterbi[] & return