Words:
Computational Morphology and Phonology

CMSC 35100
Natural Language Processing
April 8, 2003
Roadmap

● Words: Surface variation and automata
  – FSTs and Morphological/Phonological Rules
    ● Morphology: Implementing spelling change
      – Fox example
      – Automatic acquisition
  ● Phonology:
    – Brief! Introduction to Phonetics and Phonology
      ● Phone classes
      – Implementing letter to sound rules (FST)
      ● Fox redux
Surface Variation: Morphology

- Searching for documents about
  - “Televised sports”

- Many possible surface forms:
  - Televised, televise, television, ..
  - Sports, sport, sporting

- Convert to some common base form
  - Match all variations
  - Compact representation of language
Surface Variation: Pronunciation

- Regular English plural: +s
- English plural pronunciation:
  - cat+s -> cats where s=s, but
  - dog+s -> dogs where s=z, and
  - base+s -> bases where s=iz
- Phonological rules govern morpheme combination
  - +s -> s, unless [voiced]^s -> z, or [sibilant]^s->iz
- Common lexical representation
  - Mechanism to convert appropriate surface form
Two-level Morphology

- Morphological parsing:
  - Two levels: (Koskenniemi 1983)
    - Lexical level: concatenation of morphemes in word
    - Surface level: spelling of word surface form
  - Build rules mapping between surface and lexical

- Mechanism: Finite-state transducer (FST)
  - Model: two tape automaton
  - Recognize/Generate pairs of strings
FSA -> FST

- Main change: Alphabet
  - Complex alphabet of pairs: input x output symbols
  - e.g. i:o
    - Where i is in input alphabet, o in output alphabet
- Entails change to state transition function
  - Delta(q, i:o): now reads from complex alphabet
- Closed under union, inversion, and composition
  - Inversion allows parser-as-generator
  - Composition allows series operation
Simple FST for Plural Nouns

- **reg-noun-stem**
  - +N:e
  - +SG: #

- **irreg-noun-sg-form**
  - +N:e
  - +PL:^s#

- **irreg-noun-pl-form**
  - +N:e
  - +PL: #
Rules and Spelling Change

- Example: E insertion in plurals
  - After x, z, s...: fox + -s -> foxes

- View as two-step process
  - Lexical -> Intermediate (create morphemes)
  - Intermediate -> Surface (fix spelling)

- Rules: (a la Chomsky & Halle 1968)
  - Epsilon -> e/\{x,z,s\}^__s#
    - Rewrite epsilon (empty) as e when it occurs between x, s, or z at end of one morpheme and next morpheme is -s
E-insertion FST

\[ q_0 \xrightarrow{^\Delta:e, \text{other}} q_1 \xrightarrow{\#} q_0 \]
\[ q_1 \xrightarrow{\#} q_2 \xrightarrow{\#} q_3 \]
\[ q_2 \xrightarrow{z,s,x} q_5 \]
\[ q_5 \xrightarrow{z,s,x} q_2 \]
\[ q_2 \xrightarrow{s} q_4 \]

\[ q_0 \xrightarrow{\#} q_1 \]
\[ q_1 \xrightarrow{\#} q_2 \]
\[ q_2 \xrightarrow{z,x} q_3 \]
\[ q_3 \xrightarrow{s} q_4 \]
Accepting Foxes

Lexical

<table>
<thead>
<tr>
<th>f</th>
<th>o</th>
<th>x</th>
<th>+N</th>
<th>+PL</th>
</tr>
</thead>
</table>

Intermediate

<table>
<thead>
<tr>
<th>f</th>
<th>o</th>
<th>x</th>
<th>^</th>
<th>s</th>
<th>#</th>
</tr>
</thead>
</table>

Surface

<table>
<thead>
<tr>
<th>f</th>
<th>o</th>
<th>x</th>
<th>e</th>
<th>s</th>
</tr>
</thead>
</table>
Implementing Parsing/Generation

- Two-layer cascade of transducers (series)
  - Lexical -> Intermediate; Intermediate -> Surface
    - I->S: all the different spelling rules in parallel
- Bidirectional, but
  - Parsing more complex
    - Ambiguous!
      - E.g. Is fox noun or verb?
Shallow Morphological Analysis

- Motivation: Information Retrieval
  - Just enable matching – without full analysis

- Stemming:
  - Affix removal
    - Often without lexicon
    - Just return stems – not structure
  - Classic example: Porter stemmer
    - Rule-based cascade of repeated suffix removal
      - Pattern-based
    - Produces: non-words, errors, ...
Automatic Acquisition of Morphology

- “Statistical Stemming” (Cabezas, Levow, Oard)
  - Identify high frequency short affix strings for removal
  - Fairly effective for Germanic, Romance languages

- Light Stemming (Arabic)
  - Frequency-based identification of affixes

- Minimum description length approach
  - Minimize cost of model + cost of lexicon | model
Computational Phonology & TTS

- Range of correspondences between sound and text
  - Writing systems from logographic to phonetic
- Question: How are words pronounced via phones?
  - Phones (basic speech units)
    - Crucial for TTS and ASR
  - Challenge: Variability!
    - Phones pronounced differently in different contexts (e.g. [t])
      Phonology models this variation
Phonetics & Transcription

- Word pronunciation model:
  - String of symbols representing phone

- Phone transcription:
  - International Phonetic Alphabet (IPA)
    - Goal: Transcription of all languages
      - Sounds and transcription rules
    - ARPABET: ASCII –based 1- or 2- character system
      - More English-focused, computational
    - NOT identical to alphabet in general
      - E.g. a -> aa or ax ar ae
ARPAbet Snippet

- iy: bee
- ih: hit
- ey: day
- eh: bet
- ae: cat
- aa: father
- ao: dog
- ow: show
- uw: sue....

- p: put
- t: top
- th: thin
- dh: this
- jh: jay
- zh: ambrosia
- dx: butter
- nx: winter
- el: little....
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)
  - Alveolar: roof of mouth behind teeth (t, d)
  - Palatal: palate: (y); Palato-alveolar: (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
  - Fricative (s, sh,) turbulence: Affricate: stop+fricative (jh, ch)
  - Approximant (w, l, r)
  - Tap/Flap: quick touch to alveolar 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: $e: [ix] \leftrightarrow [+sibilant]:^z$

Diagram:

```
q0  q1  q2  q3  q4
^:e, other
^:e, other
^:e
^:e, other
^:e, other
^:e, other
^:e
^:e
^:e
^:e
^:e
^:e
^:e
^:e
^:e
^:e
^:e
^:e
```

States and transitions:

- $q0$: Initial state.
- $q1$: Transition on $+$sib.
- $q2$: Transition on $+$sib.
- $q3$: Transition on $e:ix$.
- $q4$: Final state.

Symbols:

- $+$sib: Sibilant.
- $^z$: Zerovowel.
- $e$: Vowel.
- $#$: Other.
- $S, sh$: Special symbols.

Notes:

- The diagram represents a finite state transducer (FST) for the foxes redux rules.
- The arrow directions indicate the transitions and the rules for each state.
- The states $q0$ to $q4$ are connected with transitions based on the rules for insertion and deletion of [ix] and [+sibilant] marks.
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
Modeling Lexicon

- Enrich lexicon:
  - Orthographic + Phonological
    - E.g. cat = c|k a|ae t|t; goose = g|g oo|uw s|s e|e