December 2, 2003

Chapter 4.1–4.4

Computational Phonology

Administrivia

- Presentations on Thursday (∼5 minutes each)
- Last day is Tuesday (review, course evaluations)
- Final exam Monday 12/15 4:30-6:20pm
- Final projects/papers due Monday 12/15 4:30pm
- Final projects must be documented

Overview

- Phonetics, IPA, ARPAbet
- Phonological rules
- FSTs for phonological rules
- Rule ordering and two-level phonology
- OT
- Machine learning of phonological rules
- Next time: text-to-speech (TTS)

Phonetics

- Phonetics: The study of the speech sounds of the world's languages.
- Speech sounds can be described by their place and manner of articulation, plus some other features (oral/nasal, length, released/unreleased).

Phonetics

- Alphabetic writing systems represent the speech sounds used to make up words, but imperfectly:
  - Predictable phonological processes not represented
  - Historical muddling of systems is common
- IPA: An evolving standard with the goal of transcribing the sounds of all human languages.
- ARPAbet: A phonetic alphabet designed for American English using only ASCII symbols.

Phonological rules

- Much of the distribution of actual speech sounds in any given language is predictable.
- Particular phones can be grouped into equivalence classes (allophones) that appear in phonologically describable environments.
- Phonological and morphophonological rules relate underlying representations to surface forms.

SPE/FST rules

- /t/ $\longrightarrow$ [flap] / $\acute{V}$ __ V
- FST implementing this is given in figure 4.10

Rule ordering

- Rules can *feed* or *bleed* each other, but creating or destroying the next rule's environment.
- A long standing issue in phonology is whether rule systems require extrinsic ordering, or whether all ordering is intrinsic.
- Example: faks+z ('faxes')
  - $\epsilon \rightarrow$ [barred i] / [+sibilant] ˆ __ z #
  - z $\rightarrow$ s / [– voice] ˆ __ #

More elaborate rule ordering: Yawelmani Yokuts

- Vowel harmony: suffix vowels agree in backness and roundness with the preceding stem vowel, if the vowels are of the same height.
- Lowering: Long high vowels become low.
- Shortening: Long vowels in closed syllables become short.
- Order: Harmony, Lowering, Shortening:

/?u:t'+it/	$\rightarrow$	[?o:t'ut]
/sudu:k+hin/	$\rightarrow$	[sudokhun]

Modeling rule ordering

- Cascaded or composed FSTs
- But: Most phonological rules are independent of each other.
- More efficient to run them in parallel.
- Koskenniemi's two-level rules finesse the issue of ordering by potentially referring to both underlying and surface forms.
- Example: Figs. 4.12–4.14

OT

- Grammar consists of GEN and EVAL
- GEN takes an underlying form and produces all possible surface forms.
- EVAL consists of a set of ranked constraints and an algorithm for choosing the best candidate.
- The best candidate is the one who's highest constraint violation is lower than any of the others. In the case of a tie, the next constraint violations are considered.

Implementing OT

- Explicit interpretation of constraints
- GEN: a regular relation (FST)
- EVAL: Cascade the constraints, but with ‘lenient composition’ (Karttunen 1998)
  - macro(priority_union(Q,R), $\{$Q, !domain(Q) $\circ$ R$\}$).
  - macro(lenient_composition(S,C), priority_union(S $\circ$ C,S)).
  - lenient_composition($\{$b x [b,b], a x [b,b]*$\}$, [b,b,b]*)

Learning Rankings

- Tesar & Smolensky (1993, 1998): Error-Driven Constraint Demotion, learns ordinal rankings.
- Boersma (1997, 1998, 2000): Gradual Learning Algorithm learns stochastic rankings, can handle optionality and variation, as well as noisy training data.

Learning Rules

- Machine learning systems automatically induce a model for some domain, given some data and potentially other information.
- Supervised algorithms are given correct answers for some of the data and use the answers to induce generalizations to apply to further data.
- Unsupervised algorithms works only from data, plus potentially some learning biases.

Learning Rules

- Ex: Gildea & Jurafsky (1996) specialize a learning algorithm for a subtype of FSTs to learn two-level phonological transducers from a corpus of input/output pairs.
- Learning biases: Faithfulness and Community

Overview

- Phonetics, IPA, ARPAbet
- Phonological rules
- FSTs for phonological rules
- Rule ordering and two-level phonology
- OT
- Machine learning of phonological rules
- Next time: text-to-speech (TTS)