November 16, 2004

Chapter 4.1–4.5

Computational Phonology

Overview

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

Evaluation (General)

- Two questions to consider:
  - Is the code you wrote doing what you meant it do?
  - Is what you meant it to do a good (or decent, or better, or above-baseline) solution to the problem you're targeting?
- The latter (only) is usually reported in papers on computational linguistics. When it's not, people get grumpy.
- Why might matters of evaluation be particularly hard in computational linguistics?
- How could you evaluate a parser?

Evaluating parsers

- Create a “gold standard”
- C = # of correct constituents in candidate parse
- N = total # of constituents in candidate parse
- N_s = total # of constituents in gold standard parse
- Precision: C/N
- Recall: C/N_s
- Cross-brackets: number of occurrences of ((A B) C) for (A (B C))

More on Precision and Recall

- Precision and recall tend to conflict: maximizing one can be done at the cost of sacrificing the other.
- Example: Find all the aces in a deck of cards:
  - 100% precision: Turn over one card, it's an ace, stop.
  - 100% recall: Turn over all the cards; you've found all the aces.
- F-Score: balance of precision and recall:

$$F = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

$\beta > 1$, precision is favored, $\beta < 1$, recall is favored.

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). [articulatory phonetics]
- Also: acoustic phonetics and perceptual phonetics

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] / V́ __ V
- FST implementing this is given in figure 4.10
- "accepts any string in which flaps occur in the correct places, and rejects strings in which flapping doesn't occur, or in which flapping occurs in the wrong environment"
- What forms should we test to validate these claims?
- What paths does the transducer take for those forms?

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')
  - $\epsilon \longrightarrow$ [barred i] / [+sibilant] ˆ __ z #
  - z $\longrightarrow$ 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/	$\longrightarrow$	[?o:t’ut]
/sudu:k+hin/	$\longrightarrow$	[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.
- (Likewise for XFST).
- Example: Figs. 4.12–4.14

- Computational phonology: What kinds of rules are required to model NL phonological systems, and how can they be implemented (with finite-state technology or otherwise)?

More on two level rules

- Two level rules can refer to upper or lower tape (or both) for both left and right context.
- Different types of two level rules differentiated by when they apply: a is realized as b whenever it appears in the context c __ d, only in that context, always and only, or never.
- What about always and only outside that context, and always outside and sometimes inside that context?
- XFST has a slightly different set of rules, with different notation (sorry!)

Another approach: Optimality Theory (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) ∘ R}).
  - macro(lenient_composition(S,C), priority_union(S ∘ 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)