Research seminar week 3

Part 2

Tamás Biró
Humanities Computing
University of Groningen
t.s.biro@rug.nl

A few examples to work on

- Stress 1.
- Stress 2.
- Syllabification
- “String grammars”

General structure of examples

- Input $\longrightarrow$ set of candidates
- Constraints on the candidates
- OT: hierarchy of constraints;
  HG: weights to constraints.

Stress 1: motivation

- Cross-linguistic typology of stress:
  Type 1: stress on first syllable.
  Type 2: stress on last syllable.
  Type 3: stress on penultimate syllable.

Stress 1: candidates

- Input: n-syllable word (xxxx)
- Candidate set: stress on the first, second, etc. syllable.
- Example for input xxxx: {suuu, usuu, uusu, uuus}, where u = unstressed syllable, s = stressed syllable.

Stress 1: constraints

- AlignLeft: nr of syllables between left edge and stress.
- AlignRight: nr of syllables between stress and right edge.
- NoFinal: nr of stress on last syllable.

Stress 2: motivation

- Cross-linguistic typology of stress:
  Type 1: stress on first syllable.
  Type 2: stress on last syllable.
  Type 3: stress on penultimate syllable.
  And many other types!

Stress 2: candidates

(Metrical stress theory of Hayes)

- Input: n-syllable word (xxxx)
- Foot: group of one or two syllables, exactly one of which is stressed (either primary or secondary stress).

- A legitimate parse: contains exactly one foot with primary stress (1), and optionally further feet – with secondary stress (2) – and unfooted syllables.
- Candidate set: All possible parses of the input.
- Examples for input xxxx: u(1)uu, u(1u)u, (u2)u(1), (1u)(2u), etc.

Stress 2: constraints

Among many others:

- PARSE: nr. of unfooted syllables.
- BINARY: nr. of feet with a single syllable.
- MAINFOOTLEFT: nr. of syllables between left edge of the word and left

edge of the foot with primary stress.

- MainFootRight: nr. of syllables between right edge of the word and right edge of the foot with primary stress.
- Trochaic: nr. of iambic feet (us).
- Iambic: nr. of feet beginning with s.

Syllabification: motivations

- Cross-linguistic typology of possible syllables:
  Type 1: CV
  Type 2: CV, V
  Type 3: CV, CVC
  Type 4: CV, V, CVC, VC

- Dutch *melk* "melluk", etc.

Syllabification: candidates

- Input: word as a series of C's (consonants) and V's (vowels)
- Legitimate syllable: C^* V C^*
  Called: (onset) nucleus (coda).
- Insertion: add a C or a V not present in input (epenthesis, hiatus filling, etc.).

- Deletion: remove a C or a V.
- Candidate set: add any number of insertions (underlined), delete any number f original segments (crossed out), and then add syllable borders (dots) to obtain a sequence of legitimate syllables.
- Input: CVC to ∅V.C$\underline{V}$, C$\underline{CV}$.V∅, C$\underline{V}$.$\underline{C}$VC.
- NB: candidate set is infinite!

Syllabification: constraints

- ONSET: nr. vowels beginning a syllable.
- NOCODA: nr. consonants ending syllable
- NOCOMPLEXONSET: nr. of syllables beginning with more than one consonants.
- NOCOMPLEXCODA: nr. of syllables ending with more than one consonants.

- PARSE: number of segments deleted from input.
- FILL: number of segments inserted.
- FILLONSET: number of consonants inserted before the vowel of a syllable.
- FILLNUCLEUS: number of vowels inserted.

String grammar: motivation

- Easy to work with.
- Covers typical examples of (phonological) constraints.
- (A little bit too?) abstract

String grammar: candidates

- Input:
  approach 1: a number L;
  approach 2: a string of length L.

- *Candidates:* $\{0, 1, ..., P-1\}^L$
  E.g., $L = P = 4$: 0000, 0001, 0120, 0123,... 3333.

String grammar: constraints

Markedness constraints ($w = w_0 w_1 ... w_{L-1}$):

- No-n: $^*n(w) := \sum_{i=0}^{L-1}(w_i = n)$
- No-initial-n: $^*\textsc{Initial}n(w) := (w_0 = n)$
- No-final-n: $^*\textsc{Final}n(w) := (w_{L-1} = n)$
- Assimilation $\textsc{Assim}(w) := \sum_{i=0}^{L-2}(w_i \neq w_{i+1})$
- Dissimilation $\textsc{Dissim}(w) := \sum_{i=0}^{L-2}(w_i = w_{i+1})$

String grammar: constraints

- Faithfulness to input σ:

$$\textsc{Faith}_\sigma(w) = \sum_{i=0}^{L-1} d(\sigma_i, w_i)$$

What to measure

- Precision: prediction of competence model (exact implementation, all grammatical forms) vs. outputs of the performance model.
- Run time: number of iterations, or CPU time (Unix command `time`).

What to experiment on?

- Compare different parameters (cooling schedule, number of iterations; starting point of random walk; etc.).
- Compare different implementations (performance models) of the same grammar: e.g., gradient ascent vs. simulated annealing.

- Compare different languages: different grammars within the same architecture (different OT hierarchies, different HG weights).
- Compare different phenomena within the same architecture: syllabification vs. stress assignment.
- Comp different architectures: OT vs. HG.

And the most important:

Never forget to discuss your results!!!

- What did you expect *before* running the experiment? What was your motivation to run the experiment?
- Expectations confirmed? Surprising?
- If so, why?