Maxent phonotactics

A. Maxent grammar learning fully implemented

(1) gj.R implements all of the maxent grammar tool in R.

(2) Sample tableau format in gjsample.txt.

B. Welsh onsets

(3) Data from an on-line dictionary: http://www.gutenberg.org/ebooks/19704

(4) Command-line tools (all on Mac, various versions available for Windows):
   a. grep to extract the headwords and their onsets.
   b. Unix tools sort and uniq to make the counts.
   c. Scriptable text editor vim to massage data into something the software would recognize.

(5) Three data files:
   a. WelshFeatures.txt
   b. WelshLearningData.txt
   c. WelshTestingData.txt

C. Comparing models of phonotactics

(6) Daland et al. (2011) ask: is simple statistical experience (word learning) sufficient to account for phonotactic data or is some knowledge of phonological universals necessary?

(7) The sonority effect: English speakers prefer [bnIk] to [nbIk].

(8) Their claim: the sonority effect follows from feature-based statistical experience plus something like syllable structure.

(9) Daland et al. (2011) do an on-line (Mechanical Turk) judgment task for occurring, marginal, and non-occurring clusters: experimental results correlate with experience.
(10) Models considered:
   a. bigram
   b. featural bigram
   c. syllabic parser
   d. maxent
   e. phonotactic probability calculator
   f. generalized neighborhood model

(11) Conditional probability

\[ p(a|b) = \frac{c(ba)}{c(b)} \]

(12) Classical bigram model

\[
\begin{align*}
p(kæt) &= p(#kæt#) = p(\#) \cdot p(k|\#) \cdot p(æ|\#k) \cdot p(t|\#kæ) \cdot p(\#|\#kæt) \\
&\approx p(\#) \cdot p(k|\#) \cdot p(æ|k) \cdot p(t|æ) \cdot p(\#|t)
\end{align*}
\]

(13) Syllabic parser:

\[ \sigma_1 \quad \sigma_2 \\
R_1 \quad R_2 \\
O_1 \quad O_2 \\
| \quad | \\
k \quad \text{à} \quad m \quad p \quad l \quad \text{é} \quad \text{k} \quad s \]

(14) \[ p(\text{kàmplékks}) = p(O_1) \cdot p(N_1) \cdot p(C_1) \cdot p(O_2) \cdot p(N_2) \cdot p(C_2) \]

(15) Featural bigram

\[ I(xy) = \max_{A,B} p(AB) \times p(x|A) \times p(y|B) \]
where
A and B represent natural classes to which x and y respectively belong
p(AB) is the type frequency of natural class bigram AB in the training lexicon
p(x|A) = 1/|A| and p(y|B) = 1/|B|

(16) “phonotactic probability calculator”: positional n-gram model?
Generalized neighborhood model

\[ \text{score}_i = \sum_j (A \log f_j^2 + B \log f_j + C) \cdot \exp(-D \cdot d_{ij}) \]

where

\[ \log f_j = \log(\text{token frequency of } w_j + 2) \]

\( d_{ij} \) is the string-edit distance between \( w_i \) and \( w_j \)

A, B, C, D are constants

Results

a. sonority effect is possible from the lexicon alone
b. syllabification and features are needed to get the sonority effect

Statistical learning (machine learning; ML) successful with features and structure only:

a. But ML can learn anything.
b. What if a language exhibited reversed sonority?
c. ML techniques could learn that without problem.
d. Where do the features come from? In ML, typically built in. What does that imply?

Evolutionary phonology doesn’t answer the question either.

D. References


