Modelling exemplar-based phonologisation

Robert Kirchner
University of Alberta Linguistics Dept.
kirchner@ualberta.ca

Roger K. Moore
University of Sheffield Computer Science Dept.
r.k.moore@dcs.shef.ac.uk

U. Chicago Phonologization Symposium
26 April 2008

This research is supported in part by the Social Sciences and Humanities Research Council of Canada, Grant No. 410-2004-1356
What is phonologisation?

Gradient phonetic patterns come to be reanalysed as patterns over symbolic representations:

> -son → +cont / V__V
What is phonologisation?

Gradient phonetic patterns come to be reanalysed as patterns over symbolic representations:

> -son → +cont / V__V

Begs the question: are phonological patterns necessarily stated over symbolic representations?
Motivation for symbolic representations

- Phonologised patterns show discrete, categorical variation
  - e.g. [p,t,k] in some contexts or [ɸ,θ,x] in some others, but not a normally distributed range of intermediate outcome like 'almost stops'
- Such patterns are typically conditioned by coarse phonetic context (e.g. /V__V), rather than fine phonetic detail, e.g.
  - particular muscle activation levels, or
  - particular formant trajectories.
Objections

- The standard treatment merely stipulates these observations, in terms of distinct, formally independent phonetic and phonological levels of representation.
  - leads either to massive redundancy (e.g. Optimality Theory's markedness constraints, which recapitulate phonetic factors),
  - or to a theory of exceedingly narrow scope, which cedes most interesting questions to the phoneticians (e.g. Hale & Reiss's 'substance-free' phonology).

- No (built-in) explicit account of the mechanism that induces this jump from numeric to symbolic characterisation. Why no 'phoneticisation'?

- How do we reconcile this supposed abrupt jump with the observed incremental nature of sound change?

- Forces a categorical choice between phonological and phonetic treatments, precluding a natural treatment of partially phonologised patterns of variation.
Example of partial phonologisation

- Florentine Italian consonant lenition, Giannelli & Savoia (1979), Kirchner (1998 ch. 8, 2004).
  - Voiceless stops, /g/, and affricates /tʃ/ and /dʒ/ obligatorily lenite to continuants in 'weak position' (i.e. roughly intervocalic within an intonational phrase); but
  - the outcome of this lenition varies from close fricative to Ø, depending on place of articulation, speech rate and register;
  - additional consonants undergo various forms of lenition in weak position in faster/more casual speech; and
  - lenition expands beyond weak position in faster/more casual speech.

- The categorical bit of this pattern is just the tip of the iceberg.
- We suggest that once phonologists/phoneticians start looking for these partial phonologisations, they will prove to be ubiquitous.
Alternative view of phonologisation

- No abrupt diachronic jump from quantitative to symbolic characterisation of a pattern.
- A broad range of quantitative methods from machine learning could be applied to phonetic processing,
- which, with certain biases in the system, could induce stabilisation of the patterns, i.e. phonologisation as an emergent behaviour, rather than a representational stipulation.
- cf. Felman & Griffiths 2007 on a Bayesian account of perceptual magnet effect.
Exemplar-based processing

- One such quantitative approach.
- Attractions: could, in principle, elegantly handle episodic effects in speech processing (e.g. Goldinger & Azuma 2003), frequency effects, sociophonetic variation, as well as phonologisation.
- Seems to be emerging as the leading competitor to the standard view of phonetics-phonology interface, e.g. Kirchner 1999, Bybee 2001, Pierrehumbert 2001, Port 2007.
- But the exemplar-based production story requires some notion of 'blending' of exemplars to generate a concrete output.
Focus of this talk:

• How do you 'blend' a collection of unique variable-length signals?
  – Without an explicit computational model, claims of exemplar-based approaches cannot be tested.
  – Previous exemplar-based proposals (including Port's appeal to Hintzman's (1986) Minerva2 model) do not address the time problem: how to compare and generate a prototype output from multiple variable-length signals.
Not in this talk: patterns transcending word classes

• No results today modelling influence of tokens outside the target word class on the output,
  – although the claim is key to the exemplar-based phonology research programme,
  – and we've had such effects in mind in designing our model.
  • Actually, the problem is not getting such effects, it's getting erratic, phonologically implausible effects, probably due to tiny training corpora.
Introducing PEBLS

Phonological Exemplar-Based Learning System
Framing the problem:

- Randomly select one token of the target word as *input* (as e.g. spectral or cepstral frames)

- All other tokens in the corpus are the *field*.
  - NB: the input token is excluded from the field: no exact match to the input, just better and worse approximations.

- Generate an output as a *path* through the field that approximates the input but is biased toward patterns in the field.
The 'path' of PEBLS

- An output is computed as an optimal alignment of the input with a transition network (a data structure derived from the field), using dynamic programming – specifically, a variant of dynamic time warping (see generally Sankoff & Kruskal 1983).

- The 'optimality' of the alignment reflects not merely the lowest distance path through the field, but also a measure of confidence in the path, calculated by means of hierarchical clustering (cf. Manning & Schütze 1999).
Aligning two signals with dynamic time warping

- DTW: a well-understood DP technique for finding the optimal alignment between two variable-length signals, A and B.
- In effect, locally stretching or shrinking A to best fit B, or vice-versa.

From Dan Ellis,
DTW: How does it work?

- Given a matrix of distances between each frame of A and B.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2</td>
<td>0.9</td>
</tr>
<tr>
<td>3.2</td>
<td>8.5</td>
</tr>
<tr>
<td>3.4</td>
<td>3.7</td>
</tr>
<tr>
<td>2.7</td>
<td>2.7</td>
</tr>
<tr>
<td>3.6</td>
<td>5.8</td>
</tr>
<tr>
<td>1.3</td>
<td>3.4</td>
</tr>
<tr>
<td>4.2</td>
<td>4.9</td>
</tr>
<tr>
<td>7.4</td>
<td>4.8</td>
</tr>
<tr>
<td>1.3</td>
<td>1.9</td>
</tr>
<tr>
<td>4.7</td>
<td>5.5</td>
</tr>
<tr>
<td>7.3</td>
<td>7.4</td>
</tr>
<tr>
<td>1.3</td>
<td>1.8</td>
</tr>
<tr>
<td>2.5</td>
<td>3.3</td>
</tr>
<tr>
<td>3.5</td>
<td>5.6</td>
</tr>
<tr>
<td>7.7</td>
<td>8.8</td>
</tr>
<tr>
<td>1.4</td>
<td>1.3</td>
</tr>
</tbody>
</table>
DTW: How does it work?

- Given a matrix of distances between each frame of A and B.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th></th>
<th>B</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2</td>
<td>0.9</td>
<td>7.8</td>
<td>1.4</td>
<td></td>
</tr>
<tr>
<td>3.2</td>
<td>8.5</td>
<td>5.2</td>
<td>2.7</td>
<td></td>
</tr>
<tr>
<td>3.4</td>
<td>3.7</td>
<td>6.6</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>2.7</td>
<td>2.7</td>
<td>6.4</td>
<td>7.7</td>
<td></td>
</tr>
<tr>
<td>3.6</td>
<td>5.8</td>
<td>5.9</td>
<td>4.1</td>
<td></td>
</tr>
<tr>
<td>1.3</td>
<td>3.4</td>
<td>5.7</td>
<td>4.2</td>
<td></td>
</tr>
<tr>
<td>4.2</td>
<td>4.9</td>
<td>1.3</td>
<td>6.7</td>
<td></td>
</tr>
<tr>
<td>7.4</td>
<td>4.8</td>
<td>9.7</td>
<td>9.1</td>
<td></td>
</tr>
<tr>
<td>1.3</td>
<td>1.9</td>
<td>7.5</td>
<td>7.9</td>
<td></td>
</tr>
<tr>
<td>4.7</td>
<td>5.5</td>
<td>5.8</td>
<td>8.2</td>
<td></td>
</tr>
<tr>
<td>7.3</td>
<td>7.4</td>
<td>9.9</td>
<td>5.4</td>
<td></td>
</tr>
<tr>
<td>1.3</td>
<td>1.8</td>
<td>6.4</td>
<td>6.2</td>
<td></td>
</tr>
<tr>
<td>2.5</td>
<td>3.3</td>
<td>5.8</td>
<td>5.9</td>
<td></td>
</tr>
<tr>
<td>3.5</td>
<td>5.6</td>
<td>8.1</td>
<td>9.2</td>
<td></td>
</tr>
<tr>
<td>7.7</td>
<td>8.8</td>
<td>2.3</td>
<td>3.2</td>
<td></td>
</tr>
<tr>
<td>1.4</td>
<td>1.3</td>
<td>7.2</td>
<td>6.7</td>
<td></td>
</tr>
</tbody>
</table>
DTW: How does it work?

• Given a matrix of distances between each frame of A and B,

• DTW iteratively calculates the cost of getting to each cell, by moving
  - horizontally (= deletion, skipping a frame of B),
  - vertically (= insertion, skipping a frame of A), and
  - diagonally (= substitution, matching frames of A and B).
Dynamic Time Warping (cont'd)

- In this case, substitution is the lowest-cost move into this cell, which is recorded in a decision table.

- With each decision step, a cumulative distance matrix is updated.

```
     16.7 (5.8)  16.6 (5.9)
     17.5 (3.4)  22.0 (5.7)
     20.3 (4.9)  1.3
```
Dynamic Time Warping (cont'd)

- In this case, substitution is the lowest-cost move into this cell, which is recorded in a decision table.

- With each decision step, a cumulative distance matrix is updated.

- The cumulative distance for the highlighted cell is:
  - the cost of getting there (the cumulative distance previously computed for the blue cell, i.e. 3.4), plus
  - the cost of being there (the distance of the red cell itself, i.e. 1.3).
Dynamic Time Warping (cont'd)

- In this case, substitution is the lowest-cost move into this cell, which is recorded in a decision table.
- With each decision step, a cumulative distance matrix is updated.
- The cumulative distance for the highlighted cell is:
  - the cost of getting there (the cumulative distance previously computed for the blue cell, i.e. 3.4), plus
  - the cost of being there (the distance of the red cell itself, i.e. 1.3).
Dynamic Time Warping (cont'd)

- Once the cumulative distance matrix is complete (i.e. the bottom-right corner is reached),
- the globally optimal path through this matrix is found by traceback through the decision table.

```
ins  ins  del  ins  ins  sub
ins  ins  sub  sub  del  del
sub  sub  del  ins  ins  del
del  del  del  sub  ins  del
ins  sub  sub  sub  sub  del
del  del  sub  sub  del  sub
```
Dynamic Time Warping (cont'd)

- Once the cumulative distance matrix is complete (i.e. the bottom-right corner is reached),
- the globally optimal path through this matrix is found by traceback through the decision table.
- Starting with the bottom-right cell
  - recall the decision that got us there (i.e. substitution)

<table>
<thead>
<tr>
<th></th>
<th>ins</th>
<th>ins</th>
<th>del</th>
<th>ins</th>
<th>ins</th>
<th>sub</th>
</tr>
</thead>
<tbody>
<tr>
<td>ins</td>
<td>ins</td>
<td>sub</td>
<td>sub</td>
<td>del</td>
<td>del</td>
<td>sub</td>
</tr>
<tr>
<td>sub</td>
<td>sub</td>
<td>del</td>
<td>ins</td>
<td>ins</td>
<td>del</td>
<td>sub</td>
</tr>
<tr>
<td>del</td>
<td>del</td>
<td>del</td>
<td>del</td>
<td>sub</td>
<td>ins</td>
<td>del</td>
</tr>
<tr>
<td>ins</td>
<td>sub</td>
<td>sub</td>
<td>sub</td>
<td>sub</td>
<td>sub</td>
<td>del</td>
</tr>
<tr>
<td>del</td>
<td>del</td>
<td>sub</td>
<td>sub</td>
<td>sub</td>
<td>sub</td>
<td>del</td>
</tr>
</tbody>
</table>

sub
Dynamic Time Warping (cont'd)

- Once the cumulative distance matrix is complete (i.e. the bottom-right corner is reached),
- the globally optimal path through this matrix is found by **traceback** through the decision table.
- Starting with the bottom-right cell
  - recall the decision that got us there (i.e. substitution)
  - which leads us to the diagonally preceding cell

<table>
<thead>
<tr>
<th></th>
<th>ins</th>
<th>ins</th>
<th>del</th>
<th>ins</th>
<th>ins</th>
<th>sub</th>
</tr>
</thead>
<tbody>
<tr>
<td>ins</td>
<td>ins</td>
<td>sub</td>
<td>sub</td>
<td>del</td>
<td>del</td>
<td></td>
</tr>
<tr>
<td>sub</td>
<td>sub</td>
<td>del</td>
<td>ins</td>
<td>ins</td>
<td>del</td>
<td></td>
</tr>
<tr>
<td>del</td>
<td>del</td>
<td>del</td>
<td>sub</td>
<td>ins</td>
<td>del</td>
<td></td>
</tr>
<tr>
<td>ins</td>
<td>sub</td>
<td>sub</td>
<td>sub</td>
<td><strong>sub</strong></td>
<td>del</td>
<td></td>
</tr>
<tr>
<td>del</td>
<td>del</td>
<td>sub</td>
<td>sub</td>
<td>del</td>
<td><strong>sub</strong></td>
<td></td>
</tr>
</tbody>
</table>
Dynamic Time Warping (cont'd)

- Once the cumulative distance matrix is complete (i.e. the bottom-right corner is reached),

- the globally optimal path through this matrix is found by traceback through the decision table.

- Starting with the bottom-right cell
  - recall the decision that got us there (i.e. substitution)
  - which leads us to the diagonally preceding cell.

- Iterate, until we've traced back the decisions to the top-left corner.
Dynamic Time Warping

Raw distance matrix, and cumulative distance matrix, from spectrograms of two different tokens of “Cottage cheese with chives is delicious,” from TIMIT corpus. Red line shows path found by traceback.

From DTW to PEBLS: loosening up

- Classic DTW finds best alignment of whole signal to whole signal
  - a more or less diagonal path, no time reversals
- PEBLS, however, needs to be able to align portions of input to portions of field.
  - If you're storing whole exemplars of broad spans of speech, you need to be able to pool data from subparts,
  - e.g. using bits of _Cottage cheese with chives is delicious_ as basis for _Chives with cottage cheese is delicious._
- In PEBLS, the path can, in principle, jump from any frame to any frame.
From DTW to PEBLS: tightening up

- But some paths may be better than others.
- In PEBLS, the path is guided by a transition network.
  - Precomputed from a field self-similarity matrix, offset by one frame, encoding values for all intra-field pairwise frame-to-frame transitions.
  - For frames \{a,b,a',b\}, if a immediately precedes b within some token in the field, and \(a'\) is similar to a, then transition \(<a',b>\) has a high value.
  - Likewise for transition \(<a,b'>\) if \(b'\) is similar to b.
The transition network

- Encodes patterns within the field:
  - not just how the input aligns with the field, but also how the field aligns with itself,
  - getting emergent structure from self-similarity within the data.

- How is it used?
  - The score of “getting there” from each other cell = previous cumulative (input-to-field) similarity × (field-to-field) similarity from the transition network.
Frequency sensitivity

- Thus far, the model finds the highest similarity alignment of input to the field and transition network.

- But to reflect relative strength of phonological patterns in the field, the path decision should be sensitive to the **frequency** of patterns.
  - i.e. how typical within the field is (something like) a given path (or a subpart thereof)?
  - analogous to the statistical notion of **confidence** that a particular sample reflects the distribution of an underlying population.
Hierarchical clustering in PEBLS

- At each DP step, the “getting-there” scores are hierarchically clustered.

- The path decision is based not on the absolute highest “getting-there” score, but on the highest cluster score.
  - i.e. a function of the cluster's mean similarity, size, and variance.

- Distinguishes between high- and low-confidence paths, i.e. typical vs. aberrant.

Parameters

- **A (dis-)similarity constant, $c$**
  
  - scales the steepness of drop-off in similarity relative to Euclidean distance between frames, for input-field and transition network similarity scores:

    \[ s_{ab} = \exp(-c*d_{ab}) \]

- **A proportion constant, $p$**
  
  - scales the weight of (getting-there) similarity vs. cluster density (size over variance), for purposes of the cluster scores:

    \[ \text{cluster}=(1+\mu)^p \frac{N}{(1+\sigma)} \]
Further details

• Implementational details (including MATLAB code) available at
  http://www.ualberta.ca/~kirchner/PEBLS.html
Hypothesis

Given

- a field of tokens instantiating a predominant (but not absolute) pattern, e.g. allophonic $k \rightarrow x$ \textbackslash/V\_\_V/ ([exi], [ekt]) and
- an input that violates the pattern (e.g. [eki], or [ext])

PEBLS' output will conform to the predominant spirantisation pattern, rather than mere faithfulness to the input, notwithstanding the presence of a violating token in the field.
Data collection

• 10 tokens each of 1st author reading the following (mostly) nonsense words:

<table>
<thead>
<tr>
<th>ÆKS</th>
<th>ÆXS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ÆXÆ</td>
<td>ÆKÆ</td>
</tr>
<tr>
<td>IKS</td>
<td>IXS</td>
</tr>
<tr>
<td>IXI</td>
<td>IKI</td>
</tr>
<tr>
<td>SKÆ</td>
<td>SXÆ</td>
</tr>
<tr>
<td>SKI</td>
<td>SXI</td>
</tr>
<tr>
<td>ÅKT</td>
<td>ÅXT</td>
</tr>
<tr>
<td>ÅXE</td>
<td>ÅKE</td>
</tr>
<tr>
<td>ÅXI</td>
<td>ÅKI</td>
</tr>
<tr>
<td>EKT</td>
<td>EXT</td>
</tr>
<tr>
<td>EXE</td>
<td>EKE</td>
</tr>
<tr>
<td>EXI</td>
<td>EKI</td>
</tr>
<tr>
<td>IKT</td>
<td>IXT</td>
</tr>
<tr>
<td>IXÆ</td>
<td>IKAÆ</td>
</tr>
<tr>
<td>IXE</td>
<td>IKE</td>
</tr>
<tr>
<td>SKE</td>
<td>SXE</td>
</tr>
<tr>
<td>EKS</td>
<td>EXS</td>
</tr>
<tr>
<td>EKÆ</td>
<td>EXÆ</td>
</tr>
</tbody>
</table>

obey intervocalic allophonic spirantisation of /k/

violate spirantisation pattern
The fields

- 18 fields were constructed by assembling
  - all 10 tokens of each word which obey the allophonic spirantisation pattern,
  - but only 1 token each of words which violate the pattern.
- Each field thus displays a strong, albeit variable, pattern of allophonic intervocalic spirantisation of /k/.
Simulation and testing

- For each of the 18 violating words, for each 9 tokens not included in the corresponding fields, these were taken as inputs to PEBLS.
- Parameters $c$ set to 30, $p$ set to 1.
  - in addition, a similarity threshold of 0.1 was imposed, to speed up computation
- Output mfcc's were multiplied by the discrete cosine transform matrix to yield spectrograms.
- We measured mean energy during the medial consonant of the outputs.
Resynthesised examples

An input token of
[ækæ]

Resulting PEBLS output
Resynthesised examples

An input token of [ækæ]

Resulting PEBLS output
Resynthesised examples

An input token of \[ækæ\]

Resulting PEBLS output
Examples (cont'd)

An input token of [ext]

Resulting PEBLS output
Examples (cont'd)

An input token of [ext]

Resulting PEBLS output
Examples (cont'd)

An input token of [ext]

Resulting PEBLS output
Uninteresting results: pattern-conforming inputs

Mean energy of medial consonant in outputs

(each bar represents result for a different choice of input, 'a' is really [æ])
Interesting results: aberrant inputs

Mean energy of medial consonant in outputs
Discussion

- There is a strong effect of frequency of stop vs. fricative tokens within the field for each word:
  - fields with a preponderance of fricative tokens generally have high energy in the medial consonant of the outputs.
  - fields with a preponderance of stop tokens generally have low energy,
  - even when the input has a medial consonant of the opposite type.
  - Some words (particularly akt and ske) show more variation than others.
Discussion (cont'd)

- Although the outputs are not uniformly in accord with the intervocalic spirantisation pattern,
- Each case of fricative output for intervocalic position, and stop output for non-intervocalic position, represents a strengthening of the pattern.
  - in spite of pattern-violating inputs
  - *a fortiori*, the output uniformly conform to the pattern when the input itself does so.
  - randomly sampling violating and conforming inputs, there thus should be a progression towards stabilisation of the pattern in outputs.
Discussion (cont'd)

- Although the outputs are not uniformly in accord with the intervocalic spirantisation pattern,
- Each case of fricative output for intervocalic position, and stop output for non-intervocalic position, represents a *strengthening* of the pattern.
  - *in spite of* pattern-violating inputs
  - *a fortiori*, the output uniformly conform to the pattern when the input itself does so.
  - randomly sampling violating and conforming inputs, there thus should be a progression towards stabilisation of the pattern in outputs.

Problematic
Iterative outputs for /eKe/, with random selection of inputs
What's wrong?

- There's a design flaw in current version of PEBLS
  - Iterative addition of self-produced outputs introduces new tokens in the field with particular frames, or even long sequences of frames, which exactly match frames of the input.
  - PEBLS can't seem to resist the exact match: trumps confidence considerations.
    - At least under parameter settings we've explored as of yet.
Conclusion

- Conceptual point: phonologisation is a behaviour, which might emerge from a quantitative processing system, rather than by representational stipulation.

- Empirical point: we should be looking for partial phonologisations.

- PEBLS affords an explicit computational solution to the question: how to blend exemplars in production? (But there may be other, better solutions.)
  - Applied to very limited modelling tasks,
  - with tiny datasets.
  - Results are encouraging in some ways, discouraging in others.
References


