Journal of Phonetics 38 (2010) 540–547

Contents lists available at ScienceDirect

Journal of Phonetics

journal homepage: www.elsevier.com/locate/phonetics

Computing phonological generalization over real speech exemplars

Robert Kirchner[a,*], Roger K. Moore[b], Tsung-Ying Chen[a]

[a] *Linguistics Department, University of Alberta, Canada*
[b] *Department of Computer Science, University of Sheffield, UK*

ARTICLE INFO

Article history:
Received 22 January 2009
Received in revised form
26 July 2010
Accepted 27 July 2010

ABSTRACT

Though it has attracted growing attention from phonologists and phoneticians, Exemplar Theory (e.g. Bybee, 2001) has hitherto lacked an explicit production model that can apply to speech signals. An adequate model must be able to generalize; but this presents the problem of how to generate an output that generalizes over a collection of unique, variable-length signals. Rather than resorting to a priori phonological units such as phones, we adopt a dynamic programming approach, using an optimization criterion that is sensitive to the frequency of similar subsequences within other exemplars: the Phonological Exemplar-Based Learning System. We show that PEBLS displays pattern-entrenchment behaviour, central to Exemplar Theory's account of phonologization.

1. Introduction

1.1. The need for an explicit exemplar theoretic model

Since Goldinger's (1996, 2000) experiments suggesting memory for speaker voices as part of lexical representation, and Johnson's (1997) seminal application of this idea to speech perception, Exemplar Theory has attracted steadily increasing interest among phonologists and phoneticians (e.g. Bybee, 2001; Gahl & Yu, 2006; Kirchner, 1999; Pierrehumbert, 2001, 2002, and articles contained therein, Gahl, 2008; Port, 2007). Exemplar Theory potentially affords elegant accounts of frequency effects, sociophonetic variation, gradient sound change; and more generally, provides a seamless phonetics–phonology interface. Exemplar-based approaches have also attracted recent interest in the automatic speech recognition (ASR) field, for their ability to exploit fine phonetic detail in recognition (e.g. Moore & Maier, 2007).

Exemplar Theory's development, however, has been hindered by the lack of an explicit computational speech processing model, capable of applying to real speech data, without which its claims cannot be rigourously tested. The recognition side of the model is not the central problem. A number of exemplar-based recognition models have been put forward, from Johnson's (1997) original X-Mod to the large-vocabulary continuous ASR system of DeWachter (2007). All a recognition model need do is assign a category label (or a sequence thereof) to an input signal based on its similarity to the variously labelled speech exemplars in memory.[1] (For concreteness' sake we assume these categories to be *words*, though they might extend to phrases or whole utterances as well; they do not, for our purposes, include phonological units: segments, syllables, and the like.) Most of the interesting phonological phenomena attributed to Exemplar Theory, however, pertain to the production side of the model, or at least crucially involve production as part of the story.

1.2. The production problem

Production involves a harder problem: generation of a concrete signal (in principle, a motor plan[2]) from a target word category (or a sequence thereof).

Naively, one might suppose that an exemplar-based production system could work simply by selecting some exemplar of the target word and reproducing it verbatim (i.e. playback). The playback method, however, lacks any mechanism for *generalizing*

* Corresponding author.
E-mail addresses: kirchner@ualberta.ca (R. Kirchner), r.k.moore@dcs.ac.uk (R.K. Moore), tsungyin@ualberta.ca (T.-Y. Chen).

0095-4470/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.wocn.2010.07.005

[1] This is an oversimiplification. Moore (2007) argues for a recognition system that includes an analysis-by-synthesis component (and likewise, for a synthesis-by-analysis component in production). In exemplar-based terms, while the recognition system decides, on auditory grounds, what category to assign to the input, it also emulates the production of the input, and uses the resulting articulatory similarity to influence the recognition decision. By giving special weight to self-produced exemplars, this partial analysis-by-synthesis thus induces some speaker normalization of the input signal for recognition purposes.

[2] The modelling results presented below, alas, do not include motor plans, only acoustic data. If we had articulatory data, we could simply add them as further dimensions to the exemplars, and PEBLS, with only minimal modification, could incorporate this information in its computation. Hofe and Moore's (2008) development of an animatronic model of the vocal tract promises to make articulatory data easier to acquire in future.

over a set of exemplars, and so its productions are limited to its previous experiences. It thus fails to model many key properties of human speech processing (and many desirable properties of an automatic speech processing system). For example, humans have the capacity to produce words which they have never uttered before, e.g. repeating a word just learned from another speaker. At the point of hearing this new word (and recognizing it as such), the relevant speaker acquires an exemplar encoding her auditory experience of the word, but no corresponding articulatory experience. Without articulatory information for this word, no motor plan can be ‘‘played back’’ as output to the speaker’s vocal tract. This deficiency can only be overcome by generalizing: in Exemplar Theory terms, forming a motor plan based on subsequences of exemplars of other words with similar auditory cues. In fact, the generalization issue is pervasive in speech production. Consider production of a word in some previously unencountered syntactic or pragmatic context, e.g. where it is subject to some phrasal phonological process; where it receives contrastive stress; or where a whispered or shouted production of the word is felicitous. Again, an adequate production model needs to generate a *composite* output – one that blends together some exemplars of the target word with contextually appropriate subsequences grabbed, perhaps, from exemplars of other word categories. More generally, Pierrehumbert (2001) shows that, in an exemplar-based production model without generalization, categories (word, phone, or any other level) increase their variances with each iteration of the production-perception loop, leading to massive collapse of the categories.[3]

Pierrehumbert therefore proposes generation of an output by averaging over a group of exemplars, namely some randomly selected exemplar of the target category, and its neighbours within a certain distance radius. However, Pierrehumbert applies this model – the most explicit exemplar-based production model to date – only to low-dimensional static data. Pierrehumbert’s model can readily be extended to higher-dimensional data. But it is not clear how it might be extended to real speech, which, in addition to being multi-dimensional, is variable-length time-series data. To recap, the production system needs to be able to generalize, but how can it generalize over a collection of unique, variable-length speech signals?

One response to this problem, adopted (but not computationally fleshed out) in Pierrehumbert (2002), is to appeal to less time-variable units, such as phones (=segments in the phonology literature). Phones can be characterized, albeit crudely, in terms of relatively static phonetic targets. Thus, if our exemplar system parses signals into phone as well as word categories, we can pool together all exemplars of, e.g. /s/, reduce these to fixed-dimensional vectors representing the phone “target” (perhaps with contextual target measurements as well), abstracting away from temporal variation within the exemplars. We can now generate an output based on an average of these fixed-dimensional vector values. However, this segmentation into a priori phonological units seems contrary to the spirit of Exemplar Theory. Categories, to the extent that they play a role in speech processing, should emerge bottom-up from comparison over the exemplars. This approach also fails to do justice to the rich dynamic structure of speech.

1.3. A way forward

Rather than segmenting the dynamic signal into quasi-static chunks, one might adopt a dynamic model ab initio. In Section 2 below, we present such a dynamic exemplar-based production model: the Phonological Exemplar-Based Learning System (PEBLS). In Section 3 we report results of an experiment testing PEBLS’ pattern generalization capacity with real speech. We further show, in a second experiment, that PEBLS propensity for generalization increases with iteration, thus capturing *pattern entrenchment*, one of the core properties attributed to Exemplar Theory in the literature, but never before demonstrated with real speech data. Finally we discuss parallels between this conception of Exemplar Theory and Optimality Theory.

The long-term goal of this research program is a comprehensive model of human speech production and perception, with particular attention to the learning of phonological patterns directly from exemplars of speech signals. We do not attempt to address the neuro-biological plausibility of this program here, other than to allude to the abundant neuro-biological motivation for the general framework of DIVA (Directions Into Velocities of Articulators model, Guenther, Ghosh, & Tourville, 2006). Specifically, PEBLS (or an extension thereof) can be seen as an exemplar-based variant of DIVA’s neural net method of learning correlations between perceptual cues and articulatory gestures – or more generally, a system of phonological patterns – for purposes of speech production (incidentally overcoming DIVA’s arbitrary restriction to syllable-sized units). The present study, however, is merely intended as a small step towards that goal: a proof-of-concept that it is possible to compute an output that generalizes over a collection of unique, variable-length signals.

2. PEBLS

2.1. Framing the problem

To generate an output for a given word, PEBLS begins, as in Pierrehumbert’s model, by randomly selecting an exemplar from this word class for use as the *input*.[4] Following Pierrehumbert’s terminology, the remainder of the exemplars are the *cloud*. (In the results presented below, we arbitrarily restrict clouds to other exemplars of the same word category.[5]) The clouds thus contain collections of exemplars which are more-or-less similar, but never identical, to the input.

The production problem can now be cast as finding an optimal *alignment* between the input and the cloud.

That is, the output is constructed from subsequences of the cloud exemplars which more-or-less correspond to subsequences of the input, and which more-or-less reflect typical subsequences (i.e. generalizations) within the cloud, as schematically represented in Fig. 1. The challenge lies in specifying an alignment criterion that can find these subsequences.[6]

2.2. Dynamic time warping

Dynamic time warping (DTW) provides a computational technique for optimally aligning two variable-length signals A

[3] This is under the assumption that outputs are subject to some non-deterministic variation from their inputs, as a consequence of their implementation by a physical system, namely the vocal tract (or ’slips ’twixt brain and lips’).

[4] This method, generating an output based on a particular input exemplar, was chosen to highlight PEBLS’ similarities and differences with Pierrehumbert’s (2001) model. It is not, however, crucial to PEBLS; we have also developed a version of the model in which the input is simply a vector indicating which word class is to be activated.

[5] As we are ultimately interested in capturing phonological generalizations that transcend individual lexical items, this is a restriction that we are eager to get away from in future research.

[6] As an anonymous referee notes, our framing of the problem – selection of an optimal set of subsequences from a rich database of exemplars – is substantially identical to that used in concatenative speech synthesis, see generally Hunt and Black (1996).

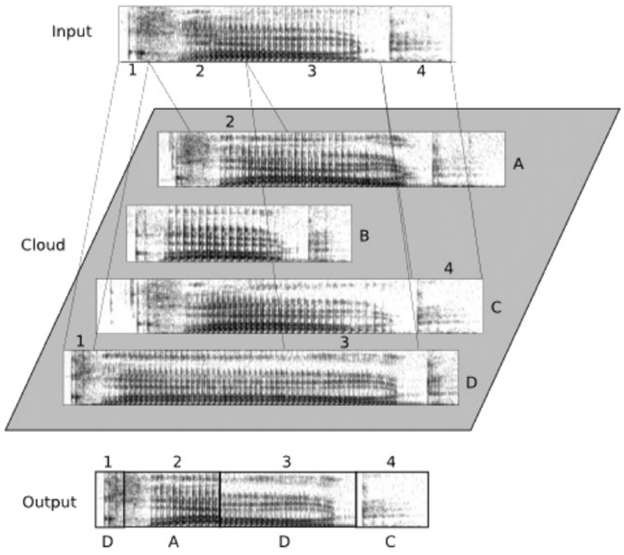

Fig. 1. Output as alignment of input with cloud. Numbers indicate corresponding subsequences within the input and cloud, and the concatenation of these subsequences which form the output. Letters show the particular exemplar from which each output subsequence was taken.

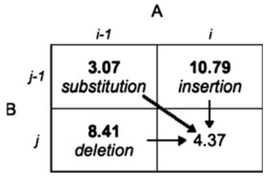

Fig. 2. Fragment of a hypothetical similarity matrix, illustrating choices for the originating cell for (i, j). Similarity values in boldface are cumulative.

and B, locally stretching or shrinking subsequences within A to best fit B, or vice-versa (see generally Sankoff & Kruskal, 1983). Since PEBLS builds upon this technique, it bears some examination.[7] Firstly, DTW presupposes some meaningful measure of similarity between timepoints of each of the signals to be aligned. For concreteness' sake, assume we are aligning two speech spectrograms, A and B. Each spectrogram is a series of spectral frames, and we can take the Euclidean distance between each frame of A and each frame of B to construct a distance matrix. Distance d can be transformed into similarity s, by

$$s = \exp(-cd) \qquad (1)$$

where c is a parameter that scales the steepness of drop-off (following Johnson, 1997).

DTW (like all dynamic programming) works by recursively breaking a complex problem down into alternative subsolutions, and finding the optimal *sub*-subsolution from which this alternative could have been reached. In classic DTW, each subsolution corresponds to a cell in the similarity matrix, which can be reached from at most three other cells: by deletion, insertion, and substitution of frames.

Cell (i, j) of Fig. 2, for example, can be reached from $(i, j-1)$ (i.e. insertion of a frame of B, relative to A), from $(i-1, j)$ (deletion of a frame of B, relative to A), or from $(i-1,\ j-1)$ (substitution: advancing a frame in both A and B). The *cumulative* similarity of (i, j) is computed as

$$S_{i,j} = \max(S_{i,j-1}, S_{i-1,j}, S_{i-1,j-1}) + s_{i,j} \qquad (2)$$

where S denotes cumulative similarity, and s denotes raw similarity. In this case, substitution has the highest cumulative similarity (10.79) of the possible originating cells, so we add this “benefit of getting there” to the raw similarity (4.37), the “benefit of being there”, to obtain the cumulative similarity, 15.16 for the current cell. We also record the *decision*: which cell has the maximum cumulative similarity (i.e. argmax $(S_{i,\ j-1},\ S_{i-1,\ j},\ S_{i-1,\ j-1})$). Because the algorithm proceeds iteratively from upper left to lower right in the matrix, the cumulative similarities of the three possible originating cells are always recorded before they are needed for computing the cumulative similarity of the current cell. Once all the cumulative similarities have been computed, we can trace the decision from which the bottom-right corner cell was reached, then the decision from which *that* cell was reached, iteratively, until we reach the upper-left corner of the matrix. This traceback procedure gives us the alignment, provably the globally maximum similarity path through the matrix.

2.3. The intra-cloud transition network

DTW aligns a whole signal with another whole signal: because the choice at every step is limited to insertion, deletion and substitution, the path is monotonic, moving more or less diagonally from upper left to lower right. DTW cannot align, for example, both corresponding parts in tokens of *housework* and *workhouse*. In PEBLS, however – particularly as we wish to avoid a priori segmentation of exemplars into phonological units – we must crucially be able to find alignments of subsequences of one exemplar with subsequences of another exemplar, as suggested in Fig. 1. That is, we must be able to pool data on a less-than-whole-exemplar basis.[8] In principle, then, we allow alignment of any frame of the input with any frame of any exemplar within the cloud, transitioning forward or backward in time within any given exemplar, or from part of one exemplar to another. Intuition suggests, though, that some transitions are better than others, namely transitions similar to those instantiated within the cloud. More precisely, if the input contains the frame sequence $\langle p, q\rangle$ while the cloud contains frames r and s (in any location), then the alignment of $\langle p, q\rangle$ to $\langle r, s\rangle$ is permissible to the extent that

- p is similar r,
- q is similar to s, and
- there is a sequence $\langle r, s'\rangle$ or $\langle r', s\rangle$ within an exemplar in the cloud s.t.
  - ○ r' is similar to r, or
  - ○ s' is similar to s.

To compute this permissibility, we construct an intra-cloud transition network: a similarity matrix of the entire cloud to itself, offset by one frame. Cell (i, j) of this matrix thus encodes not the similarity of frame i to j, but the similarity of i to the frame that immediately precedes j (or, equivalently, the similarity of j to the

[7] Indeed, DTW has formed the basis of almost all automatic speech recognition devices since the 1970s. Currently used hidden Markov models are an extension of the DTW technique, optimizing probability encoded as similarity/distance (hence providing “maximum likelihood classification”), see generally Holmes and Holmes (2001). DTW and HMMs are both instantiations of dynamic programming. DP is an established general technique for performing an optimal search through a graph. It is used in a number of different fields, e.g. “critical path anlaysis” (for searching plans), satellite navigation systems (for finding the shortest or fastest route). It also underlies the notion of minimum edit distance.

[8] Indeed, with this ability, PEBLS can handle whole-utterance exemplars, using its alignment method to find words within longer stretches of speech. This feature becomes necessary if we want to model patterns of phrasal phonology, or the sorts of lexicalization of high-frequency phrases discussed by Bybee (2001). Cf. Tucker and Tremblay (2008) showing onset latencies in a reading task continuously correlating (inversely) with word n-gram probability within several large speech corpora.

frame that immediately follows i).[9] By means of this transition network, PEBLS takes into account not only how the input aligns with each exemplar in the cloud, but how the cloud aligns with itself – getting emergent structure from self-similarity within the data.

The algorithm proceeds, as in DTW, by computing a $U \times V$ cumulative similarity matrix for the alignment of the input (V frames long) with the intra-cloud transition network t (size $U \times U$, with U frames in the whole cloud).[10] The cumulative similarity S of the vth frame of the input to the uth frame of the cloud is given by

$$S_{u,v} = \max_{i=1}^{U}(S_{i,v-1}t_{i,u}) + S_{u,v} \tag{3}$$

Within the max function of the first term, the "getting-there" score is the cumulative similarity, previously computed, for the $(v-1)$th frame of the input to the ith frame of the cloud, times the transition network score for moving from frame i to frame u: that is, a good originating point is one with a high cumulative similarity score thus far, and whose transition value into frame u is also high. The second term corresponds to the "being-there" score, the raw similarity of frame u to v.[11] Finally, the decision is given by

$$\operatorname*{argmax}_{i=1}^{U}(S_{i,v-1}t_{i,u}) \tag{4}$$

2.4. Confidence sensitivity

The model presented thus far finds the maximum similarity alignment between input and intra-cloud transition network. It thus solves the technical problem of how to generate a concrete speech output from a collection of variable-length speech exemplars. What we want, though, is an alignment that *generalizes* over the cloud (see Section 1.2 above), reflecting frame sequences which are in some sense prototypical of the cloud. To highlight this difference, consider a cloud of exemplars, predominantly, but not uniformly, reflecting some phonological pattern, e.g. intervocalic spirantization. If we select as input a token containing a non-spirantized intervocalic sequence, the presence of even a single pattern-violating exemplar in the cloud licenses transitions from vowel to plosive to vowel, notwithstanding the aberrancy of this subsequence relative to the rest of the cloud; and since the non-spirantized subsequence best matches the input, this is the alignment which will be chosen by the maximum similarity criterion. To capture the generalization effect, we need a different criterion: the "getting-there" score should include some measure of the frequency of similar subsequences within the cloud. This problem is analogous to the statistical notion of *confidence* that a particular sample reflects the distribution of an underlying population.

We calculate this confidence-sensitive measure by *hierarchically clustering*[12] the whole (U-point) vector of "getting-there" scores from the previous frame (still calculated as the product of the cumulative and transition scores, as in Eq. (3)) at each dynamic programming step (under the assumption that similar subsequences will have similar "getting-there" scores). We identify the optimal cluster w according to the following criterion:

$$w = \operatorname*{argmax}_{i=1}^{2U-1}\left(\frac{\mu_i N_i}{\sigma_i^2+1}\right) \tag{5}$$

where μ_i is the mean "getting-there" score, N_i the size, and σ_i^2 the variance, of cluster i. The optimal "getting-there" score is then μ_w (the mean of the optimal cluster), and the decision is

$$\operatorname*{argmin}_{i=1}^{U}(|u_i - \mu_w|) \tag{6}$$

i.e. the originating cell whose "getting-there" score is closest to the optimal cluster's mean. The confidence-sensitive criterion thus involves a potential trade-off between similarity (which figures into the "getting-there" score) and density (i.e. size over variance): a high-similarity but atypical alignment may lose to a somewhat lower-similarity alignment if drawn from a higher-density cluster.

Table 1
Word list, in IPA transcription.

Pattern-conforming		Pattern-violating	
Intervocalic [x]	Non-intervocalic [k]	Intervocalic [k]	Non-intervocalic [x]
æxæ	æks	ækæ	æxs
æxe	ækt	æke	æxt
æxi	eks	æki	exs
exæ	ekt	ekæ	ext
exe	iks	eke	ixs
exi	ikt	eki	ixt
ixi	skæ	iki	sxæ
ixæ	ske	ikæ	sxe
ixe	ski	ike	sxi

The complexity of the PEBLS algorithm can be estimated as O(N^2M), where N is the total number of frames in the cloud, and M is the number of frames in the input.

3. Experiment I: output generation for multiple clouds

3.1. Hypotheses

We were interested whether (a) as a threshold matter, PEBLS generated appropriate outputs for given target words, which could be resynthesized into reasonably natural-sounding speech; and (b) PEBLS' outputs showed *generalization*, focussing on a pattern of allophonic intervocalic /k/ spirantization.

3.2. Method

We recorded ten tokens each of the first author saying (in randomized order) a set of (mostly) nonsense words, shown in Table 1.

These words consisted of voiceless velar obstruents [k,x] flanked by vowels [i,e,æ] or consonants [s,t], yielding nine word types each of [k] and [x] in intervocalic position, and nine word types each of [k] and [x] in non-intervocalic position, or eighteen types each that conform to, or violate, a pattern of allophonic

[9] This network is thus analogous to the transition matrix of an ergodic hidden Markov model, albeit with a unique state for every observation.

[10] Unlike DTW, PEBLS does not allow deletion or insertion. It finds some frame in the cloud as a substitution for every frame in the input.

[11] Though we are not concerned, for present purposes, with recency effects, à la Pierrehumbert (2001), this factor could be incorporated into the model, simply by multiplying the word recency value by the "being-there" score in Eq. (3).

[12] Hierarchical clustering is an algorithm for efficiently finding all possible similarity-based groupings of a set of data points. Specifically, we use agglomerative average-linkage clustering. See http://www.resample.com/xlminer/help/HClst/HClst_intro.htm. The average-linkage method has the advantage of clustering based on the same statistics (mean, variance, and cluster size) that we use in the

(*footnote continued*)
confidence-sensitivity score. For a vector of U data points, the number of possible clusters is $2U-1$.

spirantization of /k/ in intervocalic position (i.e. /k/ → [x]/V__V). Eighteen clouds were then constructed, consisting of

- all ten tokens of each of the pattern-conforming words, plus
- one token each of the pattern-violating words.

Each of the clouds thus reflects a strong, albeit variable pattern of [x] in intervocalic position and [k] in non-intervocalic position. We operationalize the notion of generalization as follows: if an input is selected which violates the spirantization pattern, and it is fed through PEBLS, with the corresponding cloud constructed as above, and the resulting output nevertheless conforms to the pattern (i.e. [x] between vowels, [k] elsewhere), then PEBLS has generalized the pattern. If, however, the output violates the pattern, remaining faithful to the input, then PEBLS has not generalized the pattern.

The recordings were made with an Andrea NC7100 head-mounted USB microphone in a quiet office environment, directly to a computer hard-drive, at 41.5 kHz. The audio signals were preprocessed into frames of thirteen mel-frequency cepstral coefficients (MFCCs) using Slaney's (1998) Auditory Toolbox in Matlab.[13] Formant synthesis parameters were also computed from the audio signals, using Holmes' (1988) formant analysis software, on the same timescale as the MFCCs. We could thus match each MFCC frame in the cloud with its formant synthesis parameters, and used the latter to resynthesize audio signals from PEBLS' outputs. The similarity drop-off parameter c (see Eq. (1)) was set to 30. In addition, a similarity threshold of 0.1 was imposed on the transition network, to speed up computation.

For each of the eighteen clouds, each of the nine pattern-violating tokens not included in the cloud was successively selected as input, for which PEBLS generated an output. For purposes of comparison, outputs were also generated for each of the ten pattern-conforming tokens, using a leave-one-out procedure in constructing the clouds. The resulting MFCCs were transformed into 40-point filter bank (quasi-spectrographic) representations by multiplying by the discrete cosine transform matrix (see Slaney, 1998).

We measured mean energy during the medial consonant[14] of the outputs. The global minimum was rescaled to zero. High values reflect a spirantized output, whereas low values reflect stop closure.

3.3. *Results and discussion*

A few illustrative spectrograms (Fig. 3b and d) show that PEBLS' outputs meet a threshold level of adequacy: they are appropriate outputs for the given target words.

Resynthesized audio signals of these inputs and outputs are available with the on-line version of the paper. The outputs are natural-sounding speech. They further show generalization of the intervocalic allophonic spirantization pattern. It must be acknowledged, however, that with c set at 30, the optimal alignment turned out to be a straight line through a particular exemplar in the cloud, because transitions to immediate successor frames had much higher values than other transitions. (At lower settings of c, the alignment became highly erratic. Subsequent analysis suggests that this behaviour was due to the scaling of variance relative to cluster size in Eq. (3), a problem to be explored in future research.) Not so trivially, though, PEBLS' confidence-sensitive criterion ensured that the particular exemplar chosen was a representative one.

More general results are shown in Fig. 4. For every intervocalic cloud (the top row of boxes in Fig. 4), the majority of outputs show fricative allophones of the medial velar consonant; whereas in non-intervocalic clouds, the outputs are predominantly stops. Broadly speaking, then, the results show generalization of the allophonic spirantization pattern instantiated in each cloud. In some words (/æ_æ/, /i_i/, /æ_s/, /e_t/, /s_æ/), the outputs uniformly adhere to the pattern; whereas in others, the outputs vary in their pattern conformity. In the less interesting case of selection of pattern-conforming inputs, the outputs (not shown here) uniformly conform to the pattern.

4. Experiment II: Iterative production

4.1. *Hypothesis*

Putting together the results of the Experiment I,

- When a pattern-conforming input is selected, the output uniformly conforms to the pattern.
- When a pattern-violating input is selected, the output conforms to the pattern in a majority of cases.

It should thus be the case that, as the system generates outputs iteratively, adding each new output to the cloud and then randomly selecting another exemplar from within the cloud as the new input, the word type should show a progression toward uniform adherence to the pattern, i.e. pattern entrenchment.

4.2. *Method*

Iteration with PEBLS' current input selection method, however, is problematic. Addition of self-produced outputs introduces new tokens in the cloud with particular frames, or even long sequences of frames, which may *exactly* match frames of the input. In PEBLS, these exact matches seem to trump confidence sensitivity. This technical problem can be overcome, though, by adding a modicum of normally distributed, variance-scaled random noise to the MFCCs of each output as it is appended to the cloud (indeed, variable deformation of outputs is a crucial part of Pierrehumbert's model, see fn. 3, though we acknowledge that our method is a crude way to implement this idea).

With this modification, we tested PEBLS' productions for /e_e/ (one of the still-variable clouds in Experiment 1), starting with the original cloud plus the results of Experiment 1 (with both pattern-conforming and violating inputs), and then iterating with random selection of inputs. Results were measured as in Experiment 1.

4.3. *Results and discussion*

The hypothesis that PEBLS would model pattern entrenchment was confirmed.

The results in Fig. 5 show intermittent stop outputs which begin to taper off after about 100 iterations, ceasing altogether after the 411th iteration, and continuing with only fricative outputs for 200 iterations thereafter. We infer that, for this word, after these iterations, the spirantized allophone has become obligatory.

[13] Our choice of MFCCs is not crucial to the model. PEBLS can handle any sort of signal, provided it gives reasonable similarity measures. MFCCs are standard in ASR, and have generally been found to yield more useful similarity results than e.g. spectrograms, due to the independence of the coefficients.

[14] The consonant boundaries were visually identified based on onset and offset of aperiodic energy in the case of fricatives (or abrupt shifts in the energy's frequency, if flanked by another fricative), and onset and offset of closure in the case of stops. Release bursts were not included in the stop measurements.

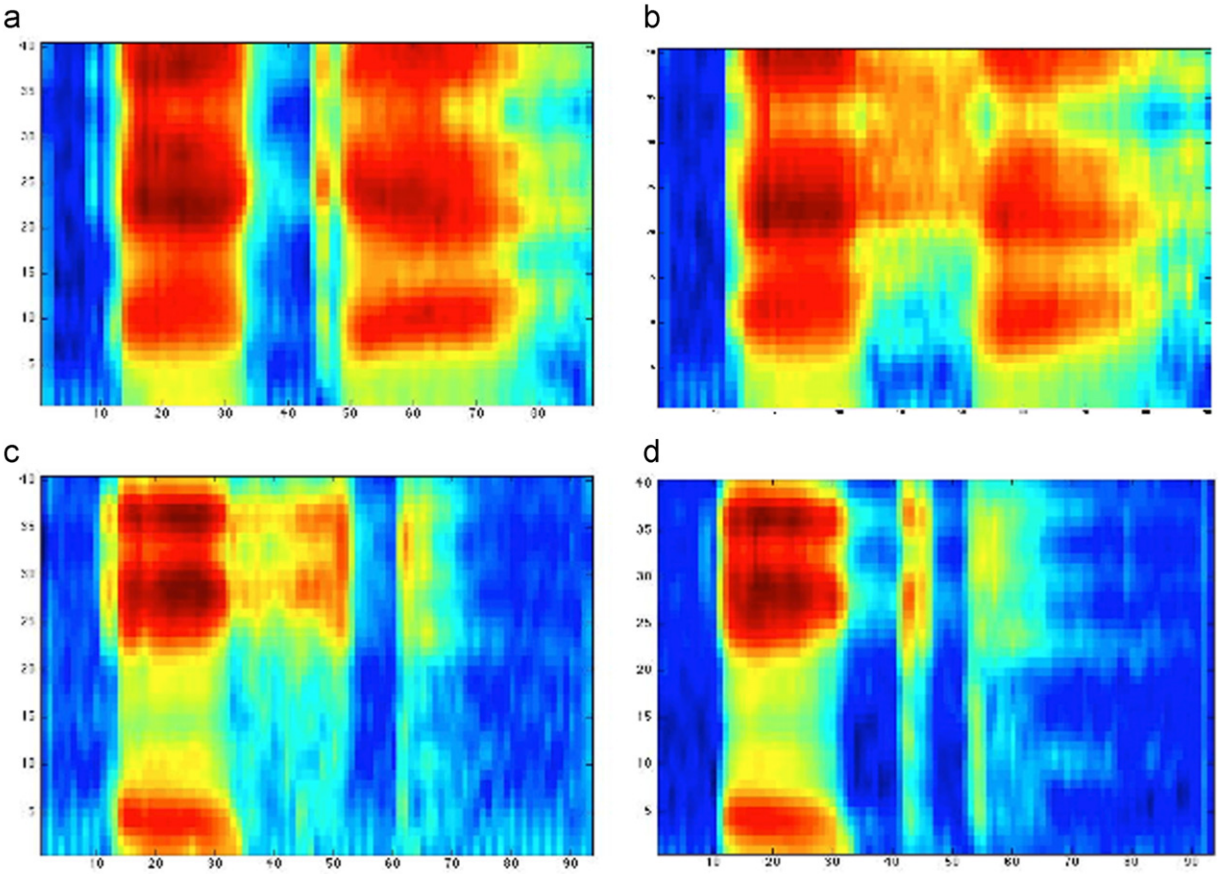

Fig. 3. Filter bank spectrograms of input tokens of [ækæ] (a) and [ext] (c), and resulting PEBLS outputs (b and d, respectively). The outputs both show generalization of the patterns in their clouds: (b) by substituting a fricative interval for the input stop in intervocalic position, and (d) by substituting a stop closure interval in place of the input fricative in non-intervocalic position.

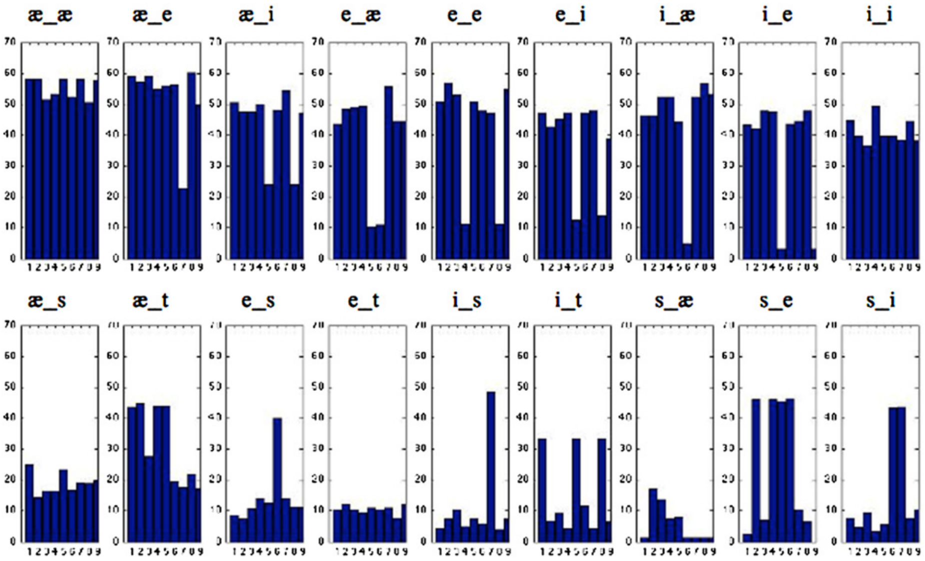

Fig. 4. Mean energy of medial consonant in PEBLS' outputs. Each box shows results for a given word cloud. Within each box, the bars show results for each of 9 pattern-violating inputs. High values (> 30) reflect fricatives, low values, stops.

To test pattern entrenchment in the opposite direction, occlusivization in non-intervocalic position, the iterative simulation was repeated, but with the cloud for /i_t/. For reasons of time, only 100 iterations were run.

Because there are much fewer iterations shown in Fig. 6 than in Fig. 5, we cannot rule out, with any confidence, the possibility of further spirantized realizations after the 100th iteration. Nevertheless, these results suggest the same movement towards

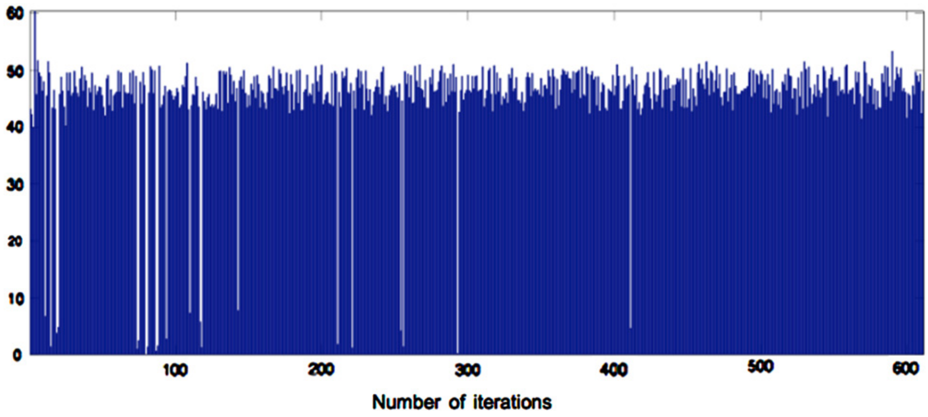

Fig. 5. Mean energy of medial consonant in iterative productions of /e_e/.

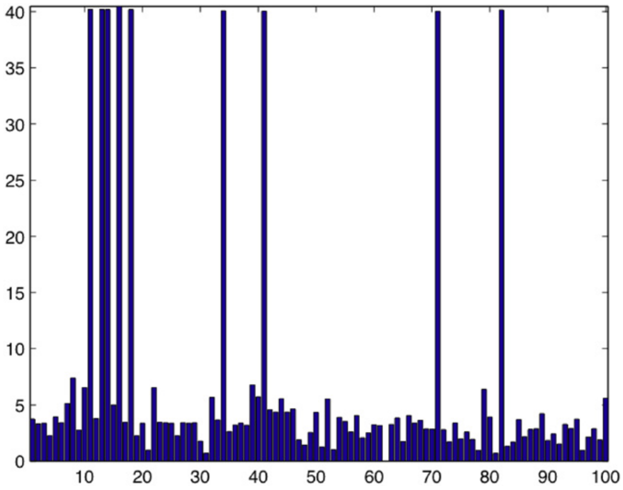

Fig. 6. Mean energy of medial consonant in iterative productions of /i_t/.

pattern entrenchment, with spirantized realizations of the medial consonant becoming increasingly rare after the 18th iteration.

5. Conclusions

PEBLS provides a solution (though perhaps better solutions remain to be discovered) to the modelling problem which has hindered the development of Exemplar Theory, namely how to generate a composite output from a set of unique, variable-length signals.

The notion of pattern entrenchment in exemplar dynamics has been a central claim of the Exemplar Theory literature. It is the sum and substance of the Exemplar Theory story on where phonology comes from – how categorical, stable (i.e. quasi-symbolic) behaviour arises from numerical signals. PEBLS provides the first explicit model of this emergent effect with real speech signals.

The next step in this research programme is to show generalization outside the word class. That is, expanding the cloud to include all other exemplars in the corpus, we hope to show that a pattern of, e.g., intervocalic spirantization, strongly instantiated in most of the word types, can be extended by PEBLS even to outputs for word types with intervocalic contexts which initially contain only non-spirantized exemplars, i.e. lexical diffusion of the spirantization pattern. It should further be the case in PEBLS that this lexical diffusion occurs more readily to word types of low token frequency.

Finally, we note that, inasmuch as PEBLS computes a global optimization for the output, there exist deep parallels to Optimality Theory. The alignment described in Section 2 is analogous to OT enforcement of correspondence constraints. Specifically, the input token in PEBLS plays much the same role as an input representation in OT: the cumulative similarity-based criterion enforces faithfulness (i.e. similarity of frame sequences) between the input and output. However, the PEBLS output is also constrained to be faithful to the tokens in the cloud, through the factoring of the transition network values into the cumulative similarity-based criterion. This dynamic is reminiscent of, if not precisely equivalent to, the effect of output-output correspondence constraints in OT (e.g. Benua, 1997), insofar as the output emerges from faithfulness to the input vs. potentially conflicting faithfulness to a set of related forms. In OT, the related forms relevant to OO correspondence constraints are *morphologically* related words, whereas in the current version of PEBLS they are merely other tokens of the same word (to this extent, the transition network is just another manifestion of IO faithfulness). However, with the expansion of the cloud beyond the word class, described above, the PEBLS transition network could also encode the influence of tokens of morphologically related words on the output for the target word. A more elaborated version of PEBLS would also include soft constraints reflecting phonetic pressures as part of the optimization criterion, e.g. an energy minimization imperative, analogous to Pierrehumbert's lenition bias, but also analogous to OT markedness constraints "grounded" in ease of articulation, cf. Kirchner (1998). In PEBLS then, as in OT, phonological patterns would arise from conflict between constraints favouring current patterns (including patterns within the word class, as with IO faithfulness, and patterns within the paradigm, as with OO faithfulness), as well as constraints favouring phonetic naturalness. Unlike OT's assumption of strict domination, but very much in the spirit of Harmonic Grammar (e.g. Pater, 2009), PEBLS's optimization criterion folds together the effect of the various faithfulness and phonetic naturalness constraints into a single numerical value. Unlike both OT and Harmonic Grammar, cross-linguistic variation in phonological patterns is attributed not to extrinsic ranking or weighting of constraints, but directly to the tokens of speech to which the learner is exposed.[15] Moreover, PEBLS computes over numeric signals rather than symbolic representations, thus providing a seamless phonetics–phonology interface.

[15] Of course, this is also true, albeit indirectly, of any version of OT (or Harmonic Grammar) that incorporates some learning algorithm: the exposure to primary linguistic data affects the ranking (or weighting) of constraints, which in turn determines the output patterns.

Appendix A. Supplementary materials

Supplementary data associated with this article can be found in the online version at doi:10.1016/j.wocn.2010.07.005.

References

Benua, L. (1997). *Transderivational identity: Phonological relations between words*. Amherst, Massachusetts: Doctoral dissertation, University of Massachusetts.

Bybee, J. (2001). *Phonology and language use*. Cambridge University Press.

DeWachter, M. (2007). *Example based continuous speech recognition*. Doctoral dissertation, Katholieke Universiteit Leuven.

Gahl, S. (2008). *Time and Thyme* are not homophones: The effect of lemma frequency on word durations in spontaneous speech. *Language*, *84*(3), 474–496.

Gahl, S., & Yu, A. (2006). Introduction to the special issue on exemplar-based models in linguistics. *The Linguistic Review*, *23*(3), 213.

Goldinger, S. D. (1996). Words and voices: Episodic traces in spoken word identification and recognition memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *22*, 1166–1183.

Goldinger, S. D. (2000). The role of perceptual episodes in lexical processing. In A. Cutler, J. M. McQueen, & R. Zondervan (Eds.), *Proceedings of SWAP (Spoken Word Access Processes)* (pp. 155–159). Nijmegen: Max Planck Institute for Psycholinguistics.

Guenther, F., Ghosh, S., & Tourville, A. (2006). Neural modeling and imaging of the cortical interactions underlying syllable production. *Brain and Language*, *96*, 280–301.

Hofe, R., & Moore, R. K. (2008). Towards an investigation of speech energetics using 'AnTon': an animatronic model of a human tongue and vocal tract. *Connection Science*, *20*(4), 319–336.

Holmes, J. (1988). *Speech synthesis and processing*. Van Nostrand Reinhold.

Holmes, J., & Holmes, W. (2001). *Speech synthesis and recognition* (2nd ed.). Taylor and Francis.

Hunt, A., & Black, A. (1996). Unit selection in a concatenative speech synthesis system using a large speech database. In: *Proceedings of ICASSP 1996* (pp. 373–376). Atlanta, Georgia.

Johnson, K. (1997). Speech perception without speaker normalization. In K. Johnson, & J. Mullennix (Eds.), *Talker variability in speech processing*. San Diego: Academic Press.

Kirchner, R. (1998). *An effort-based approach to consonant lenition*. Doctoral dissertation, UCLA. (Published by Routledge, 2001).

Kirchner, R. (1999). Preliminary thoughts on phonologization within an exemplar-based speech-processing system. In: M. Gordon (Ed.), *UCLA working papers in linguistics* (Papers in Phonology 2, vol. 1, pp. 205–231).

Moore, R. K. (2007). Spoken language processing: piecing together the puzzle. *Journal of Speech Communication (Special Issue on Bridging the Gap Between Human and Automatic Speech Processing)*, *49*, 418–435.

Moore, R. K., & Maier, V. (2007). Preserving fine phonetic detail using episodic memory: Automatic speech recognition with MINERVA2, *Proceedings of ICPhS*, Saarbruchen.

Pater, J. (2009). Weighted constraints in generative linguistics. *Cognitive Science*, *33*, 999–1035.

Pierrehumbert, J. (2001). Exemplar dynamics: Word frequency, lenition, and contrast. In J. Bybee, & P. Hopper (Eds.), *Frequency effects and the emergence of linguistic structure* (pp. 137–157). Amsterdam: John Benjamins.

Pierrehumbert, J. (2002). Word-specific phonetics. In Carlos Gussenhoven, & Natasha Warner (Eds.), *Papers in laboratory phonology VII* (pp. 101–140). Berlin: Mouton de Gruyter.

Port, R. (2007). How are words stored in memory? Beyond phones and phonemes. *New Ideas in Psychology*, *25*, 143–170.

Sankoff, D., & Kruskal, J. (1983). *Time warps, string edits and macromolecules*. CSLI Publications.

Slaney, M. (1998). Auditory Toolbox version 2, Technical Report #1998-010. Interval Research Corporation, 〈http://cobweb.ecn.purdue.edu/∼malcolm/interval/1998-010〉.

Tucker, B. & Tremblay, A. (2008). Effects of transitional probability and grammatical structure on the production of four-word sequences. Poster presented at Mental Lexicon 6 Conference, Banff, October.