

# PatPho: A phonological pattern generator for neural networks

PING LI

*University of Richmond, Richmond, Virginia*

and

BRIAN MACWHINNEY

*Carnegie Mellon University, Pittsburgh, Pennsylvania*

Much of the power of neural network modeling for language use and acquisition derives from a reliance on statistical regularities implicit in the phonological properties of words. Researchers have devised several methods for representing the phonology of words, but these methods are often either unable to represent realistically sized lexicons or inadequate in the ways they represent individual words. In this paper, we present a new phonological pattern generator (PatPho) that allows connectionist modelers to derive accurate phonological representations of the English lexicon. PatPho not only generates phonological patterns that can scale up to realistically sized lexicons, but also accurately and parsimoniously captures the similarity structures of the phonology of monosyllabic and multisyllabic words.

Rumelhart and McClelland's (1986) connectionist model of the acquisition of the English past tense had a profound positive impact on the fields of artificial neural networks, language acquisition, and cognitive psychology. However, that model was also heavily criticized for the way it represented phonological patterns of the verbal input. The fundamental structure of the past tense learning model was a nonstandard phonological structure called the *Wickelfeature*. Critics (Lachter & Bever, 1988; Pinker & Prince, 1988) argued that these distributed feature structures were unable to faithfully represent the phonological structures of words and the differences between words. As a result of these problems, connectionist researchers subsequently abandoned the use of Wickelfeatures as a way to represent phonological input, using instead a variety of alternative systems for phonological representations. These methods fall roughly into three categories.

The first class of methods (e.g., Plunkett & Marchman, 1991, 1993) treats the word as a simple string of phonemes. For example, Plunkett and Marchman used 6 binary units to code each of the three positions in a set of English consonant-vowel-consonant (CVC), VCC, and CCV word-like strings. Features included voicing, sonority, and place and manner of articulation. Because each of the three seg-

ments used six features, 18 units were needed to code a three-phoneme word. A representation of this type provides only an approximation to the phonology of words, owing to its use of arbitrarily determined binary values for phonological features. In addition, the representation accommodates only a limited number of monosyllables. Because of these problems, it is not a good choice for simulations that attempt to model the learning of a realistic lexicon.

Miikkulainen (1997) used a variant of this scheme with five units on a continuous scale to represent the phonological features of each English phoneme. In his scheme, a word is a simple concatenation of its component phonemes. This extended representation scheme can accommodate words beyond monosyllables. It also provides a more accurate representation of the phonological features, because of its use of continuous units instead of binary units. However, it has problems capturing the similarity between words of different phonemic lengths. For example, *spot* and *pot* in this coding will end up sharing very little similarity, because phonemic concatenation leads to dislocated positioning of similar phonemes: For *spot*, Units 1-5 represent /s/ and Units 6-10 represent /p/, whereas for *pot*, Units 1-5 represent /p/ and Units 6-10 represent /t/, and so on. Thus, the same phoneme activates completely different units in the representation (see Plaut, McClelland, Seidenberg, & Patterson, 1996, for a discussion of a similar problem in orthographic representations).

A second method for representing phonological patterns encodes no more than a single segment at a time. For example, the NetTalk system (Sejnowski & Rosenberg, 1988) uses a read-head approach to processing, which ac-

---

This research was supported by Grants BCS-9975249 and BCS-998009 from the National Science Foundation. We thank Xiaoming Zhao and Lihua Chen, who assisted in the development of the source code, and Igor Farkas for helping with the binary coding and conducting the PCA analyses. Please address correspondence to P. Li, Department of Psychology, University of Richmond, Richmond, VA 23173 (e-mail: pli@richmond.edu).

cepts single English orthographic letters one by one and then outputs the corresponding English sounds. To do this, the system maintains a local memory of the context. This form of representation is unable to capture larger phonological patterns and cannot deal with word-based irregularities or nonlocal phonological patterns.

A third method for representing phonological patterns relies on the slot-based representation introduced by MacWhinney and Leinbach (1991) and applied in a variety of later models (Joanisse & Seidenberg, 1999; Plaut et al., 1996; Plunkett & Juola, 1999). MacWhinney and Leinbach showed how the switch from Wickelfeatures to slot-based representations solved many of the problems with Rumelhart and McClelland's (1986) model of past tense learning. By using slot-based representations, the phonology of a word is encoded in terms of a template with a fixed set of slots, rather than as a string with either a fixed or a variable length or as a series of isolated segments. This method has its basis in autosegmental phonological theory, according to which phonemes are bundles of features in metric syllabic grids (Goldsmith, 1976; Levelt, 1989). Each segment in a word is assigned to a different slot, depending on which syllable it belongs to and whether it appears in the syllable's onset, nucleus, or coda. For a monosyllabic word, it is relatively simple to assign phonemes to their appropriate positions. For example, Joanisse and Seidenberg used the CCVVCCC template to represent English monosyllables, in which a consonant initial would occur in the first C position and consonant clusters would occupy the first two CC positions; single vowels occur at the first V position, but diphthongs occupy both VVs, and so on. Plunkett and Juola used a CC-CVVCCC template, which could additionally accommodate consonant clusters such as /str/ at the word-initial position. Thus, in this type of coding, *spot* could occur as *spCoViCC* in the template, whereas *pot* could occur as *CpCoViCC*, thus preserving their phonological similarities.

The representations used by Joanisse and Seidenberg (1999) and by Plunkett and Juola (1999) are restricted to monosyllables. MacWhinney and Leinbach (1991) also used slots to represent multisyllabic English verbs. For example, a full trisyllabic template in MacWhinney and Leinbach's representation had a CCCVVCCCVVCC-CVVCCC form. Recently, Bullinaria (1997) presented a model that combined the slot-based representation with aspects of the single-segment processing used in NetTalk. However, it appears that the successes of that model are based largely on its use of slot-based templates. The slot-based representation captures the phonological similarity structure better, thereby promoting better generalization in connectionist models. Note that this virtue comes with a price: The representation becomes longer and computationally more expensive. For example, with 18 binary units and seven slots, it takes 126 units to code a monosyllable (Joanisse & Seidenberg, 1999). Even in the condensed representations used by MacWhinney and Leinbach, 214 units are needed to encode a full trisyllabic structure.

In this paper, we will introduce a phonological pattern generator (PatPho) that builds on the idea of syllabic templates, while at the same time capturing phonological similarities more accurately and reducing the dimension of computationally costly representations. Our primary goal is to provide an accurate representation system for the phonology of English words and a computational tool (PatPho) that facilitates the generation of phonological patterns based on this system. Note that PatPho is not a complete phonological learning model per se, but, rather, a representational tool that allows connectionist researchers to compose input word representations that can capture the similarity structure of lexical phonology. Thus, use of PatPho can significantly increase the accuracy and ease of the task of connectionist language modeling. Although PatPho is designed for researchers who wish to derive phonological patterns of English words of variable length (monosyllabic, disyllabic, and multisyllabic), this method can also be used for languages other than English (see MacWhinney, Leinbach, Taraban, & McDonald, 1989).<sup>1</sup>

## METHOD

### Phonological Features

Connectionist researchers have used various phonological features to characterize phonemes in a given language, all of which are based on some type of phonological/acoustic analyses (although this is often not explicitly discussed). In selecting phonological features for English phonemes, we decided to use as our basis the articulatory phonetic features as outlined in Ladefoged's (1982) well-known textbook, *A Course in Phonetics*. Ladefoged illustrated the different levels of dimensions that can be used to distinguish consonants (cf. Figure 2.1; p. 33) and vowels (cf. Figure 2.2; p. 34). (1) For consonants, there are two major dimensions, *manner of articulation* and *place of articulation*. Within manner of articulation, Ladefoged distinguished five levels: nasal, stop, fricative, approximant, and lateral. Within place of articulation, he distinguished eight levels: bilabial, labio-dental, dental, alveolar, palato-alveolar, palatal, velar, and glottal. (2) For vowels, there are two major dimensions, *height* and *tongue position*. Within height, Ladefoged distinguished five levels: high, mid-high, mid, mid-low, and low. Within position, he distinguished three levels: front, central, and back.<sup>2</sup>

Ladefoged's (1982) classification gives rise to 40 (5 manner  $\times$  8 place) manner/place types for consonant, and 15 (5 height  $\times$  3 position) height/position types for vowels. The 40 manner/place types can easily distinguish the 24 English consonants (including the two affricates, /tʃ/ and /dʒ/), and the 15 height/position types can distinguish the 14 vowels (including /e/ and /a/, which occur only in diphthongs—e.g., /eɪ/ and /aɪ/). Our proposed PatPho representation includes each of these two dimensions. In addition, we introduce a third dimension, phoneme status, to determine whether a given segment is a vowel or a consonant. We will also use this third dimension to characterize whether a consonant is voiced or voiceless. Thus, we need a minimum of three units to represent the three di-

mensions of articulatory features for each English phoneme, as in Table 1.

In Table 1, the first column represents the ASCII symbols of the English phonemes (as used in the CELEX database; Baayan, Piepenbrock, & van Rijn, 1995), the second the IPA symbols, and the third the examples. The fourth through sixth columns are the three proposed dimensions (D1–D3). Note that D2 and D3 lump together two kinds of features (D2, position for vowels and manner of articulation for consonants; D3, sonority for vowels and place of articulation for consonants). This juxtaposition provides a way to condense the representation but makes it difficult to name the dimension uniformly. One could alternatively use separate dimensions to represent each of the features for vowels and for consonants, in which case five (instead of three) dimensions would be required, as is done in Mäikkulainen (1997). In that case, the five dimensions would be status, manner, place, height, and position.

Table 1 follows the phonological features of Ladefoged (1982) as closely as possible, with four exceptions.

1. /h/ (voiceless, glottal, approximant) is added in the table; Ladefoged (1982) did not include this phoneme in his scheme because he considered it simply as a voiceless counterpart of its subsequent vowel.
2. /ɜ:/ (“3,” vowel, central, mid) is added in the table; this vowel is assigned the value of mid to be distinguishable from /ʌ/ (“V”), even though both should be mid-low in articulation.
3. For the same reason as in (2), /ɪ/ and /e/ are assigned mid-high and mid, respectively, even though both should be mid-high in articulation.
4. Similarly, /ɔ:/ (“O”), /ɒ/ (“Q”), and /ɑ:/ (“A”) are assigned mid, mid-low, and low for differentiation, respectively, even though they should be mid-low (/ɔ:/) and low (/ɒ/ and /ɑ:/).

**Table 1**  
The Representation of English Phonemes  
by Three Phonological Dimensions (D1–D3)

Phonemes	IPA	Examples	D1	D2	D3
i	i:	tea	vowel	front	high
I	ɪ	pit	vowel	front	mid-high
e	e	bay	vowel	front	mid
E	e	pet	vowel	front	mid-low
&	æ	pat	vowel	front	low
@	ə	above	vowel	central	mid-high
3	ɜ:	burn	vowel	central	mid
V	ʌ	but	vowel	central	mid-low
a	a	buy	vowel	central	low
u	u:	put	vowel	back	high
U	ʊ	foot	vowel	back	mid-high
O	ɔ:	born	vowel	back	mid
Q	ɒ	pot	vowel	back	mid-low
A	ɑ:	barn	vowel	back	low
p	p	pit	voiceless	bilabial	stop
t	t	tip	voiceless	alveolar	stop
k	k	kick	voiceless	velar	stop
b	b	bit	voiced	bilabial	stop
d	d	dip	voiced	alveolar	stop
g	g	game	voiced	velar	stop
m	m	mad	voiced	bilabial	nasal
n	n	net	voiced	alveolar	nasal
N	ŋ	ping	voiced	velar	nasal
l	l	lip	voiced	alveolar	lateral
r	r	rip	voiced	alveolar	approximant
f	f	few	voiceless	labio-dental	fricative
v	v	view	voiced	labio-dental	fricative
s	s	sad	voiceless	alveolar	fricative
z	z	zoo	voiced	alveolar	fricative
S	ʃ	shoe	voiceless	palato-alveolar	fricative
Z	ʒ	measure	voiced	palato-alveolar	fricative
j	j	yip	voiced	palatal	approximant
h	h	hop	voiceless	glottal	approximant
w	w	witch	voiced	velar	approximant
T	θ	thin	voiceless	dental	fricative
D	ð	then	voiced	dental	fricative
C	tʃ	cheap	voiceless	palatal	fricative
J	dʒ	jeep	voiced	palatal	fricative

Note—IPA, International Phonetic Alphabet.

**Table 2**  
**Conversion of Phonological Dimensions (D1–D3)**  
**to Numerical Representations**

	D1		D2		D3
Vowel	0.100	front	0.100	high	0.100
		central	0.175	mid-high	0.185
		back	0.250	mid	0.270
				mid-low	0.355
				low	0.444
Voiced	0.750	bilabial	0.450	nasal	0.644
Voiceless	1.000	labio-dental	0.528	stop	0.733
		dental	0.606	fricative	0.822
		alveolar	0.684	approximant	0.911
		palato-alveolar	0.762	lateral	1.000
		palatal	0.841		
		velar	0.921		
		glottal	1.000		

It is important to note that these four additional assignment decisions are specific to English. When applying this three-dimensional system to other languages, other assignment customizations may be necessary in order to maximize the utility of the coding.

**Feature Conversion**

To convert the above articulatory features to numerical representations for each phoneme, we replaced the features with numerical values, scaled between the range of 0 and 1. Thus, the closer the numerical values are, the more similar the articulatory features should be, as in Table 2 (upper panel for vowel features, lower panel for consonant features). For D1, the values for voiced (.750) and voiceless (1.0) are closer to each other than to that for vowel (.1) so that we can maximally distinguish the vowels and consonants. For D2, the position features of vowels (front, central, back) are closer to each other than to the place features of the consonants. This provides additional contrast between the vowels and the consonants. For D3, the height features of vowels are closer to each other than to the manner features of consonants, further separating the two types of sounds. Within each subgroup of features, the numerical distances are spaced evenly. With this conversion table, we can then represent all the English phonemes (14 vowels and 24 consonants) with only three dimensions, as in Table 3.

A hierarchical cluster analysis (Manly, 1994) was applied to the three values of all phonemes. Figure 1 presents the result of this analysis in a dendrogram. It shows that phonetically similar phonemes are grouped together (i.e., close in distance space): The higher up the phonemes are connected in the cluster tree, the more dissimilar they are. One can also see three general clusters (vowels, voiceless consonants, and voiced consonants) in the dendrogram. This analysis indicates that the above representation scheme accurately captures the similarity structure of English phonemes, as well as the overall distances between vowels and consonants and between voiced and voiceless consonants (note that /b/ and /m/ have been treated as outliers to the consonants in this analysis, perhaps owing to their unique combination of voiced and bilabial features).

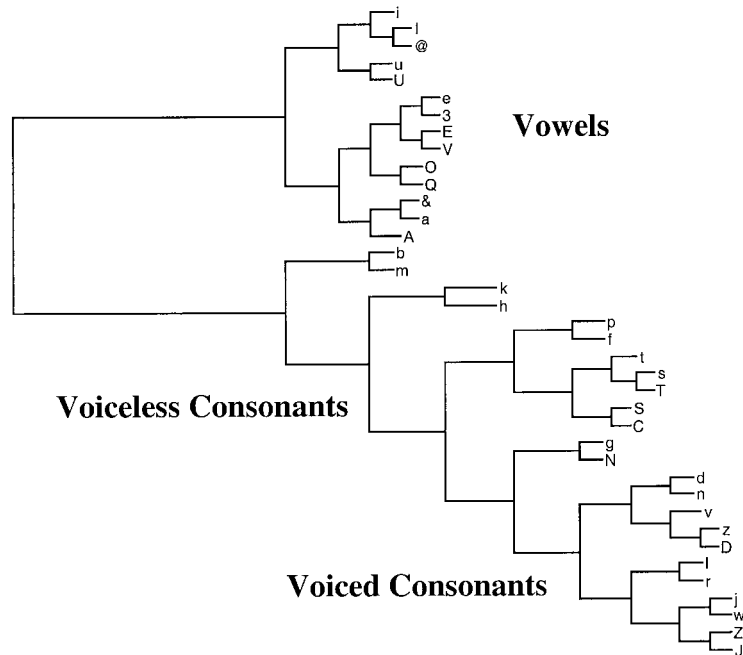
An alternative to the above conversion is to use binary codes to represent the features. This alternative approach provides a clearer representation with fewer additional assumptions. We have implemented this type of binary representations in PatPho. Table 4 shows the corresponding look-up table (cf. Table 2) that we used to convert the phonological features with binary coding. In Table 4, the vowels and the consonants are coded differently with different number of binary units, given that there are different numbers of vowel features versus consonant features to consider. For the vowels, only five units are needed to code the three tongue positions (two units) and the five height levels (three units). For consonants, however, we need seven units to code the voicing (one unit), the place of articulation (three units), and the manner of articulation (three units).

**Vector Representations**

In a string-based representation (Miikkulainen, 1997), we could simply concatenate the values of the three dimensions in Table 3 for all the phonemes of a word to form the phonological representation. Thus, *cat* /kæt/ and

**Table 3**  
**Three-Dimensional (D1–D3) Representation**  
**of English Phonemes**

Phonemes	D1	D2	D3
i	0.100	0.100	0.100
I	0.100	0.100	0.185
e	0.100	0.100	0.270
E	0.100	0.100	0.355
&	0.100	0.100	0.444
@	0.100	0.175	0.185
3	0.100	0.175	0.270
V	0.100	0.175	0.355
a	0.100	0.175	0.444
u	0.100	0.250	0.100
U	0.100	0.250	0.185
O	0.100	0.250	0.270
Q	0.100	0.250	0.355
A	0.100	0.250	0.444
p	1.000	0.450	0.733
t	1.000	0.684	0.733
k	1.000	0.921	0.733
b	0.750	0.450	0.733
d	0.750	0.684	0.733
g	0.750	0.921	0.733
m	0.750	0.450	0.644
n	0.750	0.684	0.644
N	0.750	0.921	0.644
l	0.750	0.684	1.000
r	0.750	0.684	0.911
f	1.000	0.528	0.822
v	0.750	0.528	0.822
s	1.000	0.684	0.822
z	0.750	0.684	0.822
S	1.000	0.762	0.822
Z	0.750	0.762	0.822
j	0.750	0.841	0.911
h	1.000	1.000	0.911
w	0.750	0.921	0.911
T	1.000	0.606	0.822
D	0.750	0.606	0.822
C	1.000	0.841	0.822
J	0.750	0.841	0.822



**Figure 1. Dendrogram of the phonemic distances by hierarchical cluster analysis. Clear clusters can be discerned on the dendrogram's upper and lower branches (vowels vs. consonants) and within the lower branch (voiceless vs. voiced consonants). Scale of distance (Euclidean distances between phoneme representations) ranges from 0.07 (most similar; e.g., between /a/ and /&/) to 1.05 (most dissimilar; e.g., between a vowel and a consonant).**

*pat* /pæt/, which have the CVC structure, would be represented as the following vectors, numerically (each phoneme in the CVC structure is represented by three feature units):

*cat* 1.000 0.921 0.733 0.100 0.100 0.444 1.000 0.684 0.733

*pat* 1.000 0.450 0.733 0.100 0.100 0.444 1.000 0.684 0.733.

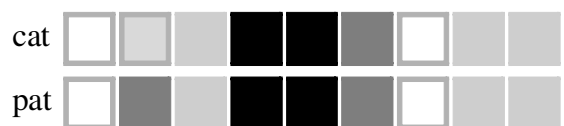
We can see that these two vectors differ only by the second unit, and a corresponding gray-scale representation shows this more clearly, as in Figure 2 (white indicates 1 and black 0).

However, as was discussed earlier, this type of representation becomes problematic when words of varying lengths are considered. Thus, we adopted the syllabic template scheme for our vector representations, as implemented in PatPho. In PatPho representations, each word consists of phonemes that are made up of phonological features in a syllabic template: A full trisyllabic template would be CCCVCCCVCCCVCC, with each CCC-VV representing one syllable and the last CCC represents consonant finals. This trisyllabic template has 18 consonant (C) and vowel (V) units. Using three articulatory features to represent each C and V, then, any word up to three syllables will be represented as a vector of 54 units. For example, *cat* (monosyllabic), *pat* (monosyllabic), *patio* (disyllabic), and *catalyst* (trisyllabic) differ in terms of numbers of phonemes and syllables. Their phonological

representations in our scheme would look like Figure 3 (in gray-scale).

As can be seen from this example, the longer the word is, the greater the number of active units is in the vector representation. Units that are solid black are inactive. Monosyllables have active phonological features only in the first 18 units (CCCVVC), whereas trisyllabic words have active features in all units of the template. Even trisyllables will have some inactive units when they have a single vowel segment in the nucleus or a single consonant rather than a cluster in the onset or coda.

This representational scheme allows us to easily represent multisyllabic words with a reasonable number of dimensions. To represent all English words up to three syllables, we need a template of only 54 units, and to represent all English words up to six syllables, we need a template of only 108 units. The system thus provides a significant reduction in the size of representation, as compared with the 162 units required only for monosyllables in models such as that of Joanisse and Seidenberg (1999).



**Figure 2. Gray-scale representation of phonemic features for *cat* and *pat*.**

**Table 4**  
**Conversion of Phonological Dimensions (D1–D3)**  
**with Binary Codes**

D1		D2		D3	
Vowel		front	01	high	011
		central	11	mid-high	001
		back	10	mid	101
				mid-low	110
				low	100
Voiced	0	bilabial	000	nasal	001
Voiceless	1	labio-dental	001	stop	010
		dental	010	fricative	100
		alveolar	011	approximant	011
		palato-alveolar	100	lateral	110
		palatal	101		
		velar	110		
		glottal	111		

In Table 4, we provided an alternative to code the features with binary values. That implies that we need more dimensions for the vectors. Thus, for a full trisyllabic word (i.e., with the template of CCCVVCCCCVVCCCCV-CCC), the vector length will be 114 units ( $7 \times 12$  C slots plus  $5 \times 6$  V slots). This is twice as long as the vectors with real values, although still significantly more compact than the other aforementioned schemes. Clearly, there is a tradeoff between the number of dimensions and the density of data storage: For the binary vectors, each unit requires only one bit of storage (hence, the longer dimensions), whereas for the continuous value vectors, each unit stores many bits of information (hence, the shorter dimensions). In our own simulations, we have found that use of binary representations can add a small degree of coding precision that improves the final learning outcome. However, this effect is typically small, involving an increase in accuracy of only two to three percentage points, at the most. Moreover, for larger problems, the use of binary vectors can significantly slow down the running of simulations. Thus, PatPho representations generally take longer to compute for the same program if binary instead of real value vectors are used (owing to the greater number of feature units in the representation). But if computing time is not a main concern (or if algorithmic shortcuts can be found), researchers may favor the use of the binary representations.

**Justification of Phonemes in the Template**

In the example shown in Figure 3, all the phonemes within a given syllable are encoded in the first available C or V slots in the template. This is called the left-justified representation. For example, *cat* would be encoded as *kCæVtCCVVCCCCVVCCC* and *catalyst* as *kCCæVtCCa*

*VICCVstC* in a left-justified representation. Alternatively, we can use right-justified representations, in which we start from the rightmost Cs of the template. For example, *cat* would be encoded as *CCCVVCCCCVVCCkæCCt* and *catalyst* as *CCkæCCtVaCCVICst* in a right-justified representation. Left-justified representations place emphasis on phonological similarities at the beginnings of words in linguistic processing (see Marslen-Wilson, 1987; Treiman & Zukowski, 1996), and this type of representations would be particularly appropriate, for example, for models that simulate the acquisition of prefixes (e.g., Li & MacWhinney, 1996), in which the initial contrasts of phonemes are at stake. Right-justified representations, on the other hand, are more appropriate for models that emphasize the phonological similarities at the ends of words in language acquisition (see Slobin, 1985), such as models of the acquisition of tense-aspect suffixes (Joanisse & Seidenberg, 1999; Li, 2000; Li & Shirai, 2000; MacWhinney, 1993; MacWhinney & Leinbach, 1991).

In a right-justified representation, the examples in Figure 3 would thus look like those in Figure 4. Comparing this figure with Figure 3, we can clearly see the effects of left- versus right-justification on how phonological similarities are represented and captured (differently) in these vectors.

A potential problem with the right-justification method is that the representations of the inflected forms and their stems will become very different, since the suffix (e.g., *-ing* of *seeing*) may occupy the rightmost syllable slot, whereas the root (e.g., *see* of *seeing*) occupies the next-to-rightmost slot. In fact, the same problem occurs with left-justification for prefixes. One way to solve this problem is to reserve an extra syllable slot for the affixes (i.e., the prefix or the suffix slots), so that all root forms will be shifted to the second syllable slot (for a prefix) or the second last syllable slot (for a suffix). Manipulations of this type, of course, would require careful considerations of the characteristics and phonotactics of the target language; PatPho is, so far, limited in handling the many typological variations of morphology and phonotactics that one finds in natural languages (e.g., infixation, as opposed to prefixation and suffixation).

**PATPHO: AN IMPLEMENTATION**

To implement the above scheme of phonological representation, we built the PatPho program that can automatically generate phonological vectors for English words of variable length. To derive representations for a large-scale lexicon, we extracted the phonological codings for all English words from the CELEX database, and after some



**Figure 3. PatPho representations for words of variable syllabic length: *cat*, *catalyst*, *pat*, and *patio*.**

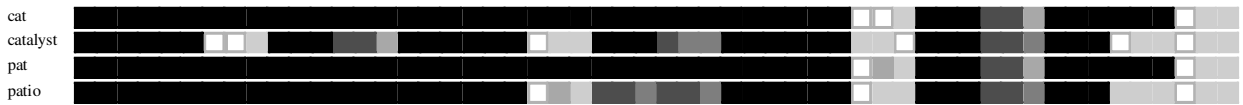


Figure 4. Right-justified PatPho representations for *cat*, *catalyst*, *pat*, and *patio*.

preprocessing (e.g., removing the stress markers and syllabic hyphens), we submitted the CELEX codings to PatPho. The CELEX database is organized either by lemmas (excluding all inflections, a total of 39,780 items) or by word forms (including all inflectional forms of a word, a total of 73,211 items). Thus, PatPho can be used to generate representations either for lemmas or for word forms, depending on whether the researcher is interested in modeling inflectional variations of a word. Although our phonological output representations are based on the CELEX database, PatPho can, in principle, take any phonological codings as input and generate corresponding phonological representations. Currently, PatPho can generate phonological representations for all English words up to three syllables (a total of 31,114 lemmas and 58,265 word forms), which cover about 80% of the English lexicon as recorded in the CELEX database. The program can easily be extended to encode still longer syllable strings.

To evaluate whether PatPho generates accurate phonological representations, we examined the euclidean distances between words in a randomly selected sample of 180 vectors/words. This analysis shows that the smaller the Euclidean distances are, the more phonologically similar the two words are. For example, the closest distances in this sample were between word pairs like *look* and *lock*, *work* and *walk*, and *curl* and *call*. In addition, to verify that the representation scheme is adequate for modeling purposes in actual simulations, we trained a self-organizing feature map (Kohonen, 1989) to learn the PatPho representations of 1,000 English words. The network achieved a 96% recognition rate—that is, it can successfully differentiate almost all words, except some homophonic words (e.g., *too* and *two*, *see* and *sea*).

The program source code, along with installation instructions and sample representation outputs, is available at <http://cogsci.richmond.edu/patpho/patpho.sit>. A set of 12 text files serves as the user's documentation on PatPho. The "README" file contains information about all files and how each file should be interpreted and used. "Patpho.c" contains the source code in C. Use of "makefile" allows the user to compile the source code using the GCC or C++ compiler running under various platforms of Unix and Linux. After compilation and running of the program, PatPho produces both left-justified and right-justified phonological representations for given phonological feature codes (see below). It also produces syllabic templates for every input word, against which the researcher can verify the phonological output representations. A sample input file with 75 words based on CELEX phonological codings ("phones.cx") is included with the program for testing. For users who want to use

PatPho under the Windows environment, we have also included the DevLex simulator, of which PatPho is a special tool. Clicking on the DevLex icon and then PatPho under Tools shall evoke the command window for the relevant parameters.

The PatPho program provides a flexible code that allows individual researchers to tailor the representations to their own needs. It can output two types of representations, the real value vectors (54 units) and the binary value vectors (114 units), to suit individual researchers' needs. The researcher needs to specify whether to use continuous phonemic features (as in the file "phonemes.real.codes"; see Table 2) or to use binary codes (as in "phonemes.binary.codes"; see Table 4). Researchers can further modify the program in two ways. (1) They can modify the feature-code file ("phonemes.codes") so that *any* number of phonological feature dimensions can be used in the representation: for example, one can replace the dimensions in Table 2 or Table 4 with codings that contain richer information about vowel quality or consonant properties, or use the researcher's own binary feature codes of each phoneme (e.g., the 18 binary features as in Joanisse & Seidenberg, 1999, or the 16 binary features as in Plunkett & Juola, 1999). (2) They can run PCA analyses on the binary representations to derive more efficient and condensed representations with *any* number of vector dimensions. A PCA subroutine can be requested from the authors for this purpose. The program is also adjustable to take any ASCII phonemic symbols as input—for example, MRPA, CPA, DISC, and UNIBET, in addition to CELEX (see Baayan et al., 1995, and MacWhinney, 2000, for discussions of these symbol systems). Or they can make use of the *makemod* function in the CLAN program (MacWhinney, 2000; downloadable from <http://childes.psy.cmu.edu>) to generate SAMPA phonetic symbols from transcribed speech, which can then be read as input by PatPho. However, researchers need to be aware of the subtle differences between these various symbol systems and their correspondence to the International Phonetic Alphabet symbols. Whenever in doubt, they should contact the authors of this article to request more specific information about PatPho's compatibility with those systems.

## SUMMARY

Learning of linguistic structure in neural network models depends heavily on accurate encoding of the statistical regularities implicit in the phonological properties of words. These models are important in such areas as child language acquisition, adult language processing, second language learning, and language disorders. Researchers

have devised several methods for representing the phonology of words, but these methods are often either unable to represent realistically sized lexicons or inadequate in the ways they represent individual words. In this paper, we presented a new phonological pattern generator (PatPho) that allows connectionist researchers to derive accurate phonological representations of the English lexicon. PatPho not only generates phonological patterns that can scale up to realistically sized lexicons, but also captures the similarity structures of the phonology of words accurately and parsimoniously.

PatPho builds on the idea of syllabic templates for coding phonemes of words, while at the same time attempting to reduce the potentially large dimensions associated with such a method. It avoids the huge dimensions by condensing the phonological features of phonemes. To capture the phonological properties of words accurately, the system uses articulatory features as outlined in Ladefoged (1982). PatPho is designed for neural network researchers who wish to derive phonological representations of English lexical items of variable length. Although PatPho has been designed for use in English, the method and the idea on which it is built can be used for other languages as well. The program itself can also be modified to generate phonological representations for other languages.

## REFERENCES

- BAAYAN, H., PIEPENBROCK, R., & VAN RIJN, H. (1995). *The CELEX lexical database* (CD-ROM Release 2). Philadelphia: University of Pennsylvania, Linguistic Data Consortium.
- BULLINARIA, J. A. (1997). Modelling reading, spelling and past tense learning with artificial neural networks. *Brain & Language*, **59**, 236-266.
- GOLDSMITH, J. (1976). An overview of autosegmental phonology. *Linguistic Analysis*, **2**, 23-68.
- JOANISSE, M., & SEIDENBERG, M. (1999). Impairments in verb morphology after brain injury: A connectionist model. *Proceedings of the National Academy of Sciences*, **96**, 7592-7597.
- KOHONEN, T. (1989). *Self-organization and associative memory*. Heidelberg: Springer-Verlag.
- LACHTER, J., & BEVER, T. (1988). The relation between linguistic structure and associative theories of language learning: A constructive critique of some connectionist learning models. *Cognition*, **28**, 195-247.
- LADefOGED, P. (1982). *A course in phonetics* (2nd ed.). San Diego: Harcourt Brace.
- LEVELT, W. (1989). *Speaking: From intention to articulation*. Cambridge, MA: MIT Press.
- LI, P. (2000). The acquisition of lexical and grammatical aspect in a self-organizing feature-map model. In L. Gleitman & A. K. Joshi (Eds.), *Proceedings of the 22nd Annual Conference of the Cognitive Science Society* (pp. 304-309). Mahwah, NJ: Erlbaum.
- LI, P., & MACWHINNEY, B. (1996). Cryptotype, overgeneralization, and competition: A connectionist model of the learning of English reverse prefixes. *Connection Science*, **8**, 1-28.
- LI, P., & SHIRAI, Y. (2000). *The acquisition of lexical and grammatical aspect*. Berlin: Mouton de Gruyter.
- MACWHINNEY, B. (1993). Connections and symbols: Closing the gap. *Cognition*, **49**, 291-296.
- MACWHINNEY, B. (2000). *The CHILDES project: Tools for analyzing talk* (3rd ed.). Hillsdale, NJ: Erlbaum.
- MACWHINNEY, B., & LEINBACH, J. (1991). Implementations are not conceptualizations: Revising the verb learning model. *Cognition*, **40**, 121-157.
- MACWHINNEY, B., LEINBACH, J., TARABAN, R., & McDONALD, J. (1989). Language learning: Cues or rules? *Journal of Memory & Language*, **28**, 255-277.
- MANLY, B. (1994). *Multivariate statistical methods: A primer* (2nd ed.). New York: Chapman & Hall.
- MARSLEN-WILSON, W. (1987). Functional parallelism in spoken word-recognition. *Cognition*, **25**, 71-102.
- MIKKULAINEN, R. (1997). Dyslexic and category-specific aphasic impairments in a self-organizing feature map model of the lexicon. *Brain & Language*, **59**, 334-366.
- PINKER, S., & PRINCE, A. (1988). On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition*, **28**, 73-193.
- PLAUT, D., McCLELLAND, J., SEIDENBERG, M., & PATTERSON, K. (1996). Understanding normal and impaired word reading: Computational principles in quasi-regular domains. *Psychological Review*, **103**, 56-115.
- PLUNKETT, K., & JUOLA, P. (1999). A connectionist model of English past tense and plural morphology. *Cognitive Science*, **23**, 463-490.
- PLUNKETT, K., & MARCHMAN, V. (1991). U-shaped learning and frequency effects in a multi-layered perceptron: Implications for child language acquisition. *Cognition*, **38**, 43-102.
- PLUNKETT, K., & MARCHMAN, V. (1993). From rote learning to system building: Acquiring verb morphology in children and connectionist nets. *Cognition*, **48**, 21-69.
- RUMELHART, D., & McCLELLAND, J. (1986). On learning the past tenses of English verbs. In J. McClelland, D. Rumelhart, & the PDP Research Group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition* (Vol. 2, pp. 216-271). Cambridge, MA: MIT Press.
- SEJNOWSKI, T. J., & ROSENBERG, C. R. (1988). NETalk: A parallel network that learns to read aloud. In J. A. Anderson & E. Rosenfeld (Eds.), *Neurocomputing: Foundations of research* (pp. 661-672). Cambridge, MA: MIT Press.
- SLOBIN, D. (1985). Crosslinguistic evidence for the language-making capacity. In D. Slobin (Ed.), *The crosslinguistic study of language acquisition* (Vol. 2, pp. 1157-1249). Hillsdale, NJ: Erlbaum.
- TREIMAN, R., & ZUKOWSKI, A. (1996). Children's sensitivity to syllables, onsets, rimes, and phonemes. *Journal of Experimental Child Psychology*, **61**, 193-215.
- XING, H., SHU, H., & LI, P. (2002). A self-organizing connectionist model of character acquisition in Chinese. In W. Gray & C. Schunn (Eds.), *Proceedings of the 24th Annual Conference of the Cognitive Science Society* (pp. 950-955). Mahwah, NJ: Erlbaum.

## NOTES

1. We have already used it successfully to generate phonological patterns for all monosyllables in Chinese. See Xing, Shu, and Li (2002).
2. Note that from a phonetician's perspective, this is still a rough characterization of vowels. For example, the two dimensions say nothing about the variation in the degree of lip rounding in different vowels.

(Manuscript received January 9, 2001;  
revision accepted for publication May 5, 2002.)