Learning Orthographic and Phonological Representations in Models of Monosyllabic and Bisyllabic Naming

Daragh E. Sibley

Department of Psychology
University of Wisconsin, Madison

Christopher T. Kello

Cognitive Science Program
University of California, Merced

Mark S. Seidenberg

Department of Psychology
University of Wisconsin, Madison

Short Title: Learning Representations

Corresponding Author:

Daragh E. Sibley
Department of Psychology
University of Wisconsin, Madison
1202 West Johnson St.
Madison, WI 53705
dsibley@wisc.edu
Voice: 608-262-7346, Fax: 608-262-4029

Abstract

Most current models of word naming are restricted to processing monosyllabic words and pseudowords. This limitation stems from difficulties in representing the orthographic and phonological codes for words varying substantially in length. Sibley, Kello, Plaut, & Elman (2008) described an extension of the simple recurrent network architecture, called the sequence encoder, that learned orthographic and phonological representations of variable-length words. The present research explored the use of sequence encoders in models of monosyllabic and bisyllabic word naming. Performance in these models is comparable to other models in terms of word and pseudoword naming accuracy, as well as accounting for naming latency phenomena. Although the models do not address all naming phenomena, the results suggest that sequence encoders can learn orthographic and phonological representations, making it easier to create models that scale up to larger vocabularies, while at the same time accounting for behavioral data.

For over 20 years, theoretical advances in the study of word reading have been marked by the development and refinement of computational models (e.g., McClelland & Rumelhart, 1981; Seidenberg & McClelland, 1989; Plaut, McClelland, Seidenberg, & Patterson, 1996; Harm & Seidenberg, 1999; Coltheart, Rastle, Perry, Langdon, & Ziegler, 2001; Perry, Ziegler, & Zorzi, 2007). These models implement cognitive mechanisms thought to underlie reading and other tasks. The models are typically evaluated by their ability to simulate data from word naming, lexical decision, semantic decision, and other reading tasks. Computational models have informed debates about the nature of the mapping between the written and spoken forms of a word (Kello, 2003; Kello & Plaut, 2003; Kello, 2006), the use of letter-to-sound correspondence rules (Coltheart et al., 2001), the use of lexical nodes (McClelland & Rumelhart, 1981; Kello, 2006), the representation of semantics (Grainger & Jacobs, 1996; Harm & Seidenberg, 2004), and the learning of word-specific information (Sibley, Kello, Plaut, & Elman, 2008), among other issues.

At its core, word naming involves mapping sequences of letters onto sequences of sounds (typically construed as phonemes). Mechanisms that map spelling to sound are greatly influenced by properties of their input and output representations, i.e., orthography and phonology, respectively. For instance, the principal flaw in the Seidenberg and McClelland (1989) model was a limited capacity to generate correct pronunciations for pseudowords, which resulted from imprecision in its representations of orthography and phonology. Plaut et al. (1996) addressed this limitation, which yielded models that produced more accurate nonword performance (see also Harm & Seidenberg, 1999).

Representations of orthography or phonology must code sequences of letters or phonemes that vary substantially in length. Difficulties inherent to coding variable length sequences have led to several systems for representing written and spoken words. Such representational schemes are specified by the modeler prior to the learning process, in order to focus on other phenomena, such as properties of the mappings between codes. This methodology does not address how the representations themselves are learned. Most models have also been limited to monosyllabic words (but see Ans, Carbonnel, & Valdois, 1998). In contrast, the current work explores a technique for learning lexical representations in the context of reading acquisition, while at the same time extending the scope of the model to bisyllabic words.

We present connectionist models of single word naming that demonstrate how simple recurrent networks (SRNs; Elman, 1990) can learn representations that overcome difficulties in representing words of variable length. This work is essentially an extension of the parallel distributed processing (PDP) models developed by Seidenberg and McClelland (1989), Plaut et al. (1996), and others. Utilizing SRNs enables our model to map between sequences of letters and phonemes for both monosyllabic (Simulation 1) and bisyllabic (Simulation 2) words. These models are benchmarked against naming latency data from the English Lexicon Project (ELP; Balota, Cortese, Hutchison, Neely, Nelson, Simpson, & Treiman, 2002). With respect to item variance in naming latencies, we show that model performance is comparable to the connectionist dual-process plus (CDP+) model reported by Perry et al. (2007). We conclude with a discussion of how these models may be improved and extended to handle full-scale English corpora.

Representing Sequences for Word Naming

English is difficult to read due to the lack of a one-to-one relationship between letters and phonemes. Multiple letters may correspond to one phoneme (e.g. PH corresponds to /f/ in SPHINX), and one letter may correspond to multiple phonemes (e.g. X corresponds to /ks/ in SPHINX). This many-to-many relationship can be partially addressed by grouping letters into graphemes like PH that tend to correspond to individual phonemes (graphemes may also include non-adjacent letters like A_E to handle the silent E in words like MALE). This solution is limited, however, because letters should not always be parsed into graphemes, as with the PH in UPHILL and A_E in MALEVOLENT.

More generally, graphemes correspond to different phonemes in different contexts (e.g. TOUGH versus THROUGH). These contextual dependencies cannot be handled by a process that operates serially over individual letters or graphemes. Morphological units may span multiple letters and graphemes, and lexical information (e.g., semantics) spans entire words. So the reading process seems to have access to a word's full orthography and phonology. To date, most models of word naming utilize *slot-based* representations to create representations that bind letters and their positions (Seidenberg & McClelland, 1989; Plaut el al., 1996; Harm & Seidenberg, 1999; Coltheart et al., 2001; Perry et al., 2007). Slot-based representations addressed contextual dependencies by assigning letters or letter clusters (or phonemes or phoneme clusters) to particular positions, such as A in the first position of a word, or NG at the end of a word.

Slot-based representations have been used with notable success in simulations of monosyllabic reading, but these representations do not easily scale to multisyllabic words. Learning about letters or letter units will not readily generalize across positions when units are position-specific; this is referred to as the dispersion problem. Consider a connectionist model in which the letter A is represented separately for each possible position in a word. Learning on connection weights associated with "A" in one position will be independent of those associated with "A" in other positions unless additional mechanisms are proposed. To alleviate the dispersion problem, Plaut et al. (1996) used slots for onsets, vowel clusters, and codas, instead of individual letter positions. Thus the letter R, for instance, is not represented by separate units for the onsets R, TR, and SPR. The problem still exists, however, for consonants in onsets versus codas (e.g., separate units represent the R in RAP versus PAR). This has posed a major problem for any scheme that attempts to integrate the learning of orthographic and phonological representations with reading acquisition. Moreover it is exacerbated as syllables are added to form multisyllabic words, and so has restricted models to simulating monosyllabic reading.

Including monosyllabic and multisyllabic words in a single model also engenders an alignment problem: Letters and phonemes in words of different lengths may not align depending on how positions are represented. The consequence is that the dispersion problem manifests differently depending on the representation of position. If the letters in a pair of different length words are left-aligned, then the ends may share no similarity (e.g., SING and PLEASING) and so learning about suffixes and other word-final regularities will be impaired. If

right-aligned, then learning about prefixes and word-initial regularities will be dispersed (e.g., CAT and CATAPULT).

A number of alternatives to slot-based representations have been proposed, including wickelfeatures (Wickelgren, 1969), open bigram codes (Grainger & Whitney, 2004), spatial codes (Davis, 1999), holographic codes (Levy, 2007; Plate, 1994), and recursive auto-associative memories (Pollack, 1990). Of these, only wickelfeatures have been used in a model of word naming (Seidenberg & McClelland, 1989), but these have representational and empirical shortcomings (Plaut et al., 1996). Schemes have not been designed for learning open bigram codes, spatial codes, and holographic codes are hence they do not yet explain a sensitivity to statistical dependencies among letters or phonemes. Without proposing additional mechanisms, these codes are insensitive to orthotactic and phonotactic variables that affect word naming performance (Bailey & Hahn, 2001). Recursive auto-associative memories are ill-suited to representing words because they impose a prescribed hierarchical structure onto sequences, whereas orthographic and phonological structures are variable and learned.

One method for learning sequential information is the SRN (Elman, 1990). SRNs are connectionist models that process sequences one element at a time. Unlike slot-based representations, SRNs learn to integrate sequential information through time. However, as originally designed, SRNs do not actually learn representations of sequences, i.e., SRNs do not learn representations that encode all the elements of a sequence and their positions. Instead, SRNs predict subsequent elements (Elman, 1990), activate target representations associated with sequences (Dell, Juliano, & Govindjee, 1993), or generate target sequences associated with input representations (Plaut & Kello, 1999).

Motivated by the need for models of word naming that process mono- and multisyllabic words, we recently extended the SRN architecture to learn orthographic and phonological representations (Kello, Sibley, & Colombi, 2004; Sibley et al., 2008). The basic function of our *sequence encoder* architecture is to encode an input sequence into a "plan" for generating an output sequence. In connectionist terms, an *encoder SRN* integrates an input sequence into a distributed representation that is learned in the service of generating a target sequence via a *decoder SRN* (see Figure 1). Even when input and output sequences vary in length, the sequence encoder learns normalized (fixed-width) representations.

To learn representations the sequence encoder was trained to store and reproduce input letter or phoneme sequences as output sequences. Two separate sequence encoder models, one for orthography and the other phonology, learned representations for over 70,000 English words ranging from 1 to 18 letters or phonemes in length. Learning generalized well to untrained letter strings (pseudowords) that were well-formed, i.e., conformed to graphotactic and phonotactic regularities. Learning did *not* generalize well to ill-formed pseudowords (e.g., SBTAMVLI, OEWPN), which demonstrates that representations were shaped by statistical dependencies among sequence elements.

Large-Scale Modeling of Word Reading

The previously mentioned sequence encoders simulated wordform learning, rather than word naming, because orthographic representations were not mapped onto phonological representations. Sequence encoders have been integrated into a large-scale model of lexical processing designed to simulate both word naming and lexical decision tasks (Kello, 2006;

Sibley, 2008). Unlike previous PDP models of word naming, the mapping from spelling to sound in these models were not accomplished via learned, distributed representations. Instead, an orthographic sequence encoder was connected to a phonemic sequence encoder via lexical nodes, where each node represented an individual word. These intermediate lexical nodes facilitated simulation of word recognition behaviors and helped us explore whether graded activation across 60,000 lexical nodes could support the processing of novel inputs. In essence, this lexical layer was designed to achieve an analogy-based or similarity-based process of generalization akin to the proposition of Glushko (1979).

These large scale models accounted for notable amounts of variance in word naming and lexical decision data compiled in the ELP database (Balota et al., 2002). For over 28,000 mono- and multisyllabic English words, the model accounted for 33.9% of item variance in naming latencies and 41.6% of item variance in lexical decision latencies (Sibley, 2008). This model had some capacity to generate pseudoword pronunciations (37.0% of the monosyllabic nonwords used by Seidenberg et al. (1994) and 25.1% of the mono- and multisyllabic nonwords used by Sibley et al. (2008)), but not at a level approaching a skilled reader.

These large-scale simulations demonstrated that learned orthographic and phonological representations could be integrated into a model of word reading and recognition. These representations helped break long-standing barriers to simulating naming and lexical decision on a scale that approaches the vocabulary of skilled English readers. Finally, these models accounted for substantial amounts of item variance in naming and lexical decision data from the ELP database using relatively few mechanisms (e.g., sigmoidal and radial basis processing units, SRNs, backpropagation learning, and a single processing pathway between orthography

and phonology) and even fewer free parameters (e.g., two scaling exponents, one for converting word frequencies to error scalars, the other for converting output activations to reaction times).

The primary question raised by these large-scale simulations was whether representations learned by sequence encoders could support pseudoword naming. Poor generalization performance could have resulted from the use of lexical nodes, because novel inputs lack dedicated nodes. Or poor generalization could result from the staged development of the model, because learning of orthographic and phonological representations occurred prior to, and independent of learning the mapping between them. Or the shear scale of the model could have caused difficulties in generalization.

The two simulations reported herein explored whether sequence encoders could simulate speeded naming data on a relatively small scale (about 5000 monosyllabic words in Simulation 1), including pseudoword naming, and whether this modeling approach can scale up to about 13,000 mono- and bisyllabic words. Both simulations used learned, distributed representations to map orthographic representations onto their phonological counterparts. Also, learning of graphotactic and phonotactic information occurred while the models learned to map between orthography and phonology. This allowed us to specifically test the ability of sequence encoders to support pseudoword naming.

Simulation 1: Monosyllabic Word Naming

In Simulation 1, a sequence encoder model was trained to map sequences of letters onto sequences of phonemes for a corpus of monosyllabic words. The corpus was very similar

to one used for the CDP+ model so that Simulation 1 results could be directly compared with the extant model that accounts for the most variance in monosyllabic naming latencies in the ELP database.

Method

Model Architecture and Representations. In general, the groups of units and their connectivity matched the architecture shown in Figure 1, and described in detail in Sibley et al. (2008). There were 250 units in each context group and hidden group, and 500 units in the sequence representation group. The only qualitative difference between the sequence encoder architecture in Figure 1 and the present model was in the input and output representations. Inputs and outputs were the same in the original sequence encoder, whereas in the present model, inputs were orthographic and outputs were phonological.

Another difference was in the way that letters and phonemes were coded. In previous sequence encoders and large-scale models, input and output groups consisted of 26 letter units or 39 phoneme units, plus an "end-sequence" unit. Input and output sequences were processed one letter or one phoneme at a time, for words from 1 to 10 letters (1 to 13 phonemes) in length. Encoding-decoding accuracies fell off as a function of length, partly because error signals needed to propagate further back in time for longer sequences, and partly because the number of training words decreases as length increases beyond 7 letters. This effect of sequence length suggests that performance should improve if lengths are shortened.

One aspect of the present simulation that served to shorten sequence lengths was to assign sequence elements to vowel-consonant (VC) clusters, rather than individual letters or

phonemes. Parsing a sequence of letters or phonemes into VC clusters is a simple, unambiguous process, and it substantially reduces sequence lengths without restricting word lengths. Letter and phoneme sequences were parsed left-to-right, allowing for the possibility of no vowel at the beginning of a sequence and no consonant at the end of a sequence. To illustrate, SAVED was parsed as S-AV-ED (no initial vowel), AURA was parsed as AUR-A (no final consonant), /sAvd/ was parsed as /s-Avd/, and UPHILL was parsed as UPH-ILL. Comparing SAVED with /sAvd/ shows that VC clusters did not necessarily align between orthography and phonology, and the UPHILL example shows that VC clusters did not necessarily align with graphemic, phonemic, and morphemic structures.

We consider VC clusters to be convenient abstractions for purposes of implementation. They are not meant to correspond to eye fixations used to perceive written words, or articulatory sequences used to speak words. They are frames used to feed information into the sequence encoder so that representations can be learned. This learning should be affected by statistical dependencies among letters and phonemes, regardless of how the dependencies are parsed (Sibley et al., 2008). Ultimately, the sequence encoder learning task requires that all letters or phonemes and their positions are encoded in the learned representations, regardless of how sequences are parsed. Thus learned representations, rather than the input and output sequencing, carry the most theoretical weight in the model.

Orthographic VC input clusters consisted of 26 letter units plus 153 open bigram units, 25 of which were VV bigrams and the remaining 133 were CC bigram units. VC and CV bigrams were not included because they were redundant, i.e. VC clusters were uniquely determined on the basis of VV, CC, and letter units. For example, the orthography for EAST consisted of one

VC cluster that activated the E, A, E-A, S, T, and S-T input units. The orthographic sequence for EASTERN consisted of a second VC cluster ERN that activated the E, R, N, and R-N units, and the two VC clusters were input to the model in sequence. Phonemic VC clusters were created in the same way, except there were 39 phoneme units, 114 CC biphone units, and one end-sequence unit. There were no VV biphone units because diphthongs were coded as individual vowels, and any remaining adjacent vowels were parsed into separate VC clusters. For instance, the phonological wordform /plAR/ ("player" with a vocalic r) was parsed as /plA-R/. The end-sequence unit was activated simultaneously with the final VC cluster of a sequence (i.e., it was not parsed into its own VC cluster).

Training Corpus and Procedure. A total of 6,116 English monosyllabic words were chosen for the training corpus. This corpus was created by intersecting the 7,441 words in the CDP+ training corpus with the CMU pronunciation dictionary (for phonemic transcriptions) and the Wall Street Journal corpus (for word frequency estimates). Words ranged from 1 to 7 letters and 2 to 7 phonemes in length.

The procedure for presenting the model with a given input-output pair from the training corpus is outlined in the caption of Figure 1, and described in detail in Sibley et al. (2008). The forward propagation of activation and backpropagation of error was governed by connectionist algorithms used in many previous models of word naming: For all hidden units and output units (including sequence representation units), net inputs were computed as the dot product of their incoming activation vector and weight vector, $I_j = \sum a_i w_{ij}$, and activations were computed as a sigmoidal function of their net inputs (i.e., the logistic for output units, and the hyperbolic tangent for hidden units). Activation vectors over context units were set equal to

activation vectors of their corresponding hidden units from the previous time step of processing.

Error between output unit activations and their targets was computed using two different metrics. Early in training, squared error ($e_j = [a_j - t_j]^2$) was computed and the backpropagation algorithm (Rumelhart, Durbin, Golden, & Chauvin, 1995) was used to calculate weight derivatives. Cross entropy error ($e_j = t_j \log a_j + [1 - t_j] \log[1 - a_j]$) was computed later in training to increase pressure for outputs to be close to their targets. Also, errors were not calculated early in training when outputs were within 0.2 of their targets, and this "zero error radius" was removed later in training to refine outputs. Throughout training, errors were scaled by the square root of the printed frequency of the word as estimated in the Wall Street Journal Corpus.

Each input-output pair was sampled randomly from the training corpus, input unit activations were propagated forward, and output unit errors were backpropagated to calculate unit and weight derivatives (see Sibley et al., 2008). Connection weights were initially set to values sampled randomly from a flat distribution between -0.1 and 0.1, except for weights projecting from input units, for which the range was -0.4 to 0.4. This larger range ensured different inputs lead to different patterns of activation over the sequence representation units at the beginning of training. Weight derivatives were accumulated every 1000 samples and then applied to weights after being scaled by a learning rate parameter than ranged from 5e-07 at the beginning of training, down to 1e-07 by the end of training. Training was halted after 20,000 weight updates, at which point learning asymptoted.

Simulation 1 Results and Discussion

To assess performance, a given words orthographic sequence was input by the model, and phonemic VC clusters were generated until activation of the end-sequence unit exceeded 0.5. Each VC cluster was converted into a phoneme sequence by choosing the most highly activated vowel unit (unless all vowel unit activations were < 0.05, in which case no vowel was chosen), and the target consonant vector (including the null vector for no consonants) that most closely matched the output consonant vector. Matching was based on the square root of activation values in order to give more weight to weakly activated units. This weighting scheme was used because any given consonant unit or consonant bigram unit was activated for only a small proportion of VC clusters, which biased their activation values towards zero. Phoneme sequences were concatenated across VC cluster sequences to generate a complete phonemic output sequence for a given orthographic input.

A given phonemic output sequence was judged to be correct only if it perfectly matched its target sequence. At the end of training, the model produced correct sequences for 97.4% of the words in the training corpus. Naming latency for a given output sequence *w* was computed as $RT_w = \sqrt{\sum E(1 - a_k)}$, where a_k is activation of an activated output unit (a_k > 0.05) on sequence step *k*, *E*() is the average of all activated units on step *k*, and summation is across steps. This measure was designed to estimate the "confidence" of model outputs (i.e. strength of activation) as a proxy for naming latency.

Simulated latencies were regressed against mean naming latencies from the ELP database for two different subsets of words, and the resulting R^2 values are shown in the Simulation 1 column of Table 1. The monosyllabic subset was the intersection of our training

corpus with the naming latencies present in the ELP database, and a monomorphemic subset used by Yap (2007) in regression analyses of the ELP database. By comparison R^2 values for the CDP+ model, also shown in Table 1, are slightly higher. However, this contrast should be interpreted carefully as the intent and scope of these models are quite different. Perry et al. (2007) present CDP+ as a computationally implemented theory of word reading and recognition behaviors. Simulation 1 is only intended to explore the sequence encoder's ability to learn about orthographic and phonological representations and the mapping between them. Simulation 1 does not, for instance, include word recognition capabilities that presumably affect word naming performance. In contrast, CDP+ is designed to simulate word recognition and other behaviors like priming. Nonetheless, CDP+ offers a useful baseline for interpreting Simulation 1's ability to address several behavioral phenomena.

We also tested whether our sequence encoder model can generate acceptable pronunciations of untrained letter sequences, i.e., pseudoword naming. Monosyllabic pseudowords from the naming experiments reported by Seidenberg et al. (1994) were used to test the model. Again, as shown in Table 1, Simulation 1's performance was comparable to, but slightly lower than, performance of the CDP+ model. A more challenging test of generalization abilities was offered by the stimuli presented in Rastle & Coltheart (1998). Each of these "whammy" pseudowords includes a digraph, which means the model must associate two letters with a single phoneme in a novel context. For these more challenging pseudowords, Simulation 1, produced 83.3% acceptable pronunciations, while Perry et al. (2007) report that CDP+ produced 91.7% acceptable pronunciations.

Models of word naming are also evaluated in terms of their ability to simulate the effects of lexical variables known to correlate with naming latency. Table 2 displays bivariate effect sizes, in terms of R^2, between each individual psycholinguistic variables and naming latencies, for monosyllabic words from three different sources: The ELP database, the CDP+ model, and Simulation 1 (statistically significant effects, $p < .05$, are denoted with an *). Results show that Simulation 1, CDP+, and behavioral (ELP) latencies are all correlated with measures of frequency, length, neighborhood, and consistency.

Directions of the effects just listed were mutually consistent across latency sources. The CDP+ model generated an overly strong frequency effect and Simulation 1 generated a weak length effect, both relative to ELP latencies. The effects of orthographic and phonological neighborhood were assessed with 4 different terms. Coltheart's orthographic and phonological N is calculated as the number of English words a given string can be transformed into, by changing a single letter or phoneme. Levenshtein distance is the average of the minimum number of letters or phonemes that must be added, removed, or substituted to transform a word into its nearest 20 neighbors (Yarkoni et al., submitted). Again, Simulation 1 and CDP+ exhibit effects of these variables that are similar to ELP latencies, though both are overly sensitive to phonological neighborhood size. Consistency was assessed using 3 different measures, provided by Yap (2007). The first two measures compute the ratio of a words friends (similarly spelt words, receiving similar pronunciations) to its total number of orthographic neighbors, with respect to either a words onset or rhyme. Levenshtein consistency is calculated as the ratio of a word's Levenshtein orthographic distance to its Levenshtein phonological distance, where less consistent words tend to have different orthographic and phonological

neighborhood sizes. CDP+ underestimates the effects of onset consistency, while Simulation 1 overestimates the effect of Levenshtein consistency.

We also tested five interaction effects and found a more qualitative distinction between the models. We calculated variables for the interactions of frequency with length, neighborhood, and consistency by multiplying the respective variables. As suggested by Cohen et al (2003) and applied by Yap (2007), interactions were tested using a hierarchical regression model. The two main effect variables were first entered into the model, followed by the interaction term. Differences in R^2 between the first and second steps are used to estimate the interaction effect size. Results showed Simulation 1 and behavioral latencies produce similar trends for all five interaction variables, while CDP+ and behavioral latencies only correspond for three interactions.

In summary, Simulation 1 established that the sequence encoder has several desirable qualities for modeling word and pseudoword naming. Using very few qualitatively distinct mechanisms and free parameters, model performance in Simulation 1 was comparable to the CDP+ model. In particular, the sequence encoder comprises the following standard connectionist mechanisms: sigmoidal units, connection weights, a sequencing mechanism, an error-driven learning mechanism, representations of letters and phonemes, word frequencies, and an algorithm for converting model outputs to phonemic responses and naming latencies. The free parameters are numbers of hidden units and the activation threshold of 0.05, and only the latter was tuned to maximize performance.

By comparison, the CDP+ model has its own version of all of the above mechanisms in its assembly route alone. The model also includes additional mechanisms for graphemic

parsing, numerous parameters on its lexical route, plus mechanisms for coordinating the two routes. The consequence is at least 25 free parameters that must be tuned, in addition to numbers of hidden units. As for the ability of the two models to account for benchmark effects in word naming, the effect size analyses indicate again that Simulation 1 is comparable to the CDP+ model. This suggests that a scheme for learning orthographic and phonological representations, like the Sequence Encoder could be very usefully integrated into a more general model of the lexical system.

Simulation 2: Mono and Bisyllabic Word Naming

Perhaps the biggest advantage of the sequence encoder model over previous models of word naming is its ability to scale up to process multisyllabic words without adding new mechanisms. Here we report the results of a sequence encoder model trained on a corpus of monosyllabic and bisyllabic words (longer words were excluded to minimize computational demands), and we compare results (when possible) with Simulation 1 and the CDP+ model.

Method

Model architecture and representations were the same as in Simulation 1, with the following exceptions. The number of open bigram and biphone units was increased to cover the expanded space of possibilities (totals were 256 bigram units and 266 biphone units), and two stress units were added to the phonemic output group. Stress units applied to the single vowel per VC cluster and represented levels of primary, tertiary, and no stress (two, one, and

zero units activated, respectively). Also, there were 300 units in each context group and each hidden group, and 600 units in the sequence representation group.

The training corpus included all monosyllabic words from Simulation 1, plus 8,000 bisyllabic words. The latter were chosen by taking all bisyllabic words in the ELP database less than 9 letters in length, intersecting them with the CMU pronunciation dictionary, and choosing the 8,000 most frequent words according to the Wall Street Journal corpus. Of the remaining 2165 lowest frequency words, 1845 were used as pseudowords by withholding them from the training corpus. The remaining lower frequency words were discarded because they had unusual spellings in terms of trigram frequencies. The model was trained for 30,000 weight updates, at which point learning asymptoted.

Results

Model outputs were converted into phonemic responses using the same procedures as in Simulation 1; with the addition of converting stress unit activations to stress levels (stress units were also included in the naming latency measure). The model generated correct responses for 99.8% of the words in the training corpus, and Table 1 shows percentages correct for the Seidenberg et al. (1994) monosyllabic pseudowords and our proxy corpus of bisyllabic pseudowords (i.e., untrained words). Performance was slightly lower on monosyllabic pseudowords compared with Simulation 1, and 19.3 percentage points lower on bisyllabic pseudowords compared with monosyllabic pseudowords.

Table 1 also shows percentages of ELP naming latency variance accounted for by Simulation 2. R^2 values were again comparable to, but slightly lower than those of Simulation 1

for the two monosyllabic word sets. R^2 was slightly highly for the full bisyllabic set compared with the monosyllabic sets, and 8.6 percentage points higher for the monomorphemic bisyllabic word set compared with the full bisyllabic word set. Finally, the same lexical variable analysis was conducted as in Simulation 1, with the addition of stress typicality and syllabic length variables. Stress typicality values were set equal to the probability that a given word would take on its stress pattern, given its grammatical category (grammatically ambiguous words were arbitrarily assigned a single category). Effect sizes for each variable were statistically significant and directionally consistent for the two sources of latency data, except for the frequency by consistency effects which did not reach statistical significance for simulated of behavioral latencies. Notably, Simulation 2 tended to overestimate the effects of most variables, relative to the ELP data. This could occur because human latencies include many sources of error variance (i.e., individual differences and measurement error), which do not contribute to the simulated latencies.

In summary, the results of Simulation 2 showed that the sequence encoder can be used to simulate word and pseudoword naming data for tens of thousands of monosyllabic and bisyllabic words, including the effects of multisyllabic variables like syllabic length and stress typicality. The ability to simulate these two effects of bisyllabic naming, are notable as we did not include any new mechanisms specifically for this purpose. As a result, these models stand as counter examples to claims that syllabic length effects imply a functional role for syllables (e.g., New, Ferrand, Pallier, & Brysbaert, 2006) and notions that stress assignment requires complex special purpose mechanisms (e.g., Rastle & Coltheart, 2000). Pseudoword naming accuracy decreased for longer words, and such length effects are standard in speeded naming

tasks. However, it is likely that performance would not decrease as much as in the present simulation if skilled readers were to name bisyllabic pseudowords from our corpus. Thus an important task for future modeling work will be to investigate methods of improving pseudoword naming.

General Discussion

The simulations reported herein demonstrated how orthographic and phonological representations can be learned in models of monosyllabic and bisyllabic naming. This work represents a step towards understanding how learning about orthographic and phonological forms of words can be integrated with learning about reading. Further, these models addressed substantial amounts of behavioral data. This includes phenomena specific to multisyllabic word naming, in particular stress assignment and the effect of syllabic length.

Our prior large-scale models utilizing this representational scheme accounted for substantial variance in naming and lexical decision data for nearly 30,000 words, but the mapping from spelling to sound did not generalize well to pseudowords. The present work showed that this lack of generalization was not due to the sequence encoder, in spite of recent criticism that sequence encoders do not effectively solve problems with slot-based codes (Bowers & Davis, in press). Pseudoword naming was successfully simulated in that the sequence encoder generated acceptable pronunciations for most novel monosyllabic and bisyllabic inputs.

The sequence encoder scaled up from monosyllabic to bisyllabic naming with no additional assumptions, mechanisms, or parameters. The same model architecture and procedure could also be applied to multisyllabic words of arbitrary lengths, but current and previous findings (Sibley et al., 2008) indicate the ability to read pseudowords would degrade as their length increased. The problem is that there are few long words in any given corpus relative to the exponential growth in possible letter sequence space as length increases. We briefly outline three possible approaches to this issue.

One approach is to abandon vector-based representations in favor of structured representations (Markman, 1999). This approach would require a theoretical framework for modeling the learning of structured representations, and the mapping of one kind (orthography) to another (phonology). A second approach would be to modify the sequence encoder in way that effectively shrinks the sequence space, and/or more fully samples from this space. The use of VC clusters is an example of this approach, and one could also imagine a hierarchy of sequence encoders in which models at higher levels learn to encode sequences of representations learned at lower levels. A third approach would be to claim that sequence encoders apply only to letter strings perceived in a single eye fixation, which would limit the length of sequences to be processed (see Plaut, 1999). Letter strings that require multiple fixations would require an additional assembly process of some kind.

Another issue raised by our findings is whether sequence encoder models can be extended to simulate both naming and lexical decision data. The present models distinguish words from pseudowords to slight degree in that simulated word responses are faster and more accurate, on average, compared with pseudowords. Lexical decisions, however, require nearly perfect discriminations between known and novel stimuli. In PDP models like the sequence encoder, the ability to generalize training on words to pseudowords runs directly counter to the discrimination of words from pseudowords. The simulation of this and other behaviors require additional mechanisms, for instance an implemented semantic layer of representations.

As discussed earlier, sequence encoders can be incorporated into larger models of word reading that also include lexical and/or semantic pathways of processing, in which case the

latter can simulate lexical decisions (Coltheart et al., 1993, 2001; Perry et al., 2007; Plaut et al., 1996; Seidenberg & McClelland, 1989). The alternative is to posit a lexical pathway that can simulate both lexical decision and word naming tasks (Glushko, 1979; Kello, Sibley, & Plaut, 2005). Large-scale models implementing this alternative successfully simulated lexical decisions (Kello, 2006; Sibley, 2008), but not pseudoword naming. The present simulations indicate that the problem with pseudoword naming was not in using sequence encoders. Further work is necessary to determine how models of orthographic and phonological learning, like the sequence encoder, can be best integrated with more complete theories of the lexical system.

References

Balota, A. A., Cortese, M. J., Hutchison, K. A., Neely, J. H., Nelson, D., Simpson, G. B., et al.(2002). The English lexicon project: A web-based repository of descriptive and behavioral measures for 40,481 English words and nonwords. http://elixicon.wustl.edu, Washington University.

Bailey, T.M., & Hahn, U. (2001). Determinants of wordlikeness: Phonotactic or lexical neighborhoods? *Journal of Memory and Language, 44*, 568-591.

Balota, D. A., Cortese, M. J., Sergent-Marshall, S. D., Spieler, D. H., & Yap, M. (2004). Visual word recognition of single-syllable words. *Journal of Experimental Psychology: General, 133*, 283-316.

Bowers, J., & Davis, C. (in press). Learning representations of wordforms with recurrent networks: Comment on Sibley, Kello, Plaut, & Elman. To appear in *Cognitive Science*.

Coltheart, M., Curtis, B., Atkins, P., & Haller, M. (1993). Models of reading aloud: Dual-route and parallel-distributed-processing approaches. *Psychological Review, 100*, 589-608.

Coltheart, M., Rastle, K., Perry, C., Langdon, R. & Ziegler, J. (2001). DRC: A dual route cascaded model of visual word recognition and reading aloud. *Psychological Review, 108*, 204-256.

Davis, C. J. (1999). The self-organizing lexical acquisition and recognition (SOLAR) model of visual word recognition. Unpublished doctoral dissertation. University of New South Wales, Australia.

Dell, G. S., Juliano, C., & Govindjee, A. (1993). Structure and content in language production: A theory of frame constraints in phonological speech errors. *Cognitive Science, 17*, 149-195.

Elman, J.L. (1990). Finding structure in time. *Cognitive Science, 14*, 179-211.

Glushko, R.J. (1979). The organization and activation of orthographic knowledge in reading aloud. *Journal of Experimental Psychology: Human Perception and Performance, 5*, 674-691.

Grainger, J., & Whitney, C. (2004). Does the huamn mind raed wrods as a wlohe? *Trends in Cognitive Sciences, 8*, 58:59.

Grainger, J., & Jacobs, A. M. (1996). Orthographic processing in visual word recognition: A multiple read-out model. *Psychological Review, 103*, 518-565.

Harm, M. W., & Seidenberg, M. S. (1999). Phonology, reading acquisition, and dyslexia: Insights from connectionist models. *Psychological Review, 106*, 491-528.

Harm, M. W., & Seidenberg, M. S. (2004). Computing the meanings of words in reading: Cooperative division of labor between visual and phonological processes. *Psychological Review, 111*, 662-720.

Kello, C. T. (2003). The emergence of a double dissociation in the modulation of a single control parameter in a nonlinear dynamical system. *Cortex, 39*, 132-134.

Kello, C. T. (2006). Considering the junction model of lexical processing. In S. Andrews (Ed.), From inkmarks to ideas: Current issues in lexical processing. Sydney: Psychology Press.

Kello, C. T. & Plaut, D. C. (2003). Strategic control over rate of processing in word reading: A computational investigation. *Journal of Memory & Language, 48*, 207-232.

Kello, C. T., Sibley, D. E., & Colombi, A. (2004). Using simple recurrent networks to learn fixed-length representations of variable-length strings. In *Proceedings of the AAAI Symposium on Compositional Connectionism*. Washington, DC.

Kello, C. T., Sibley, D. E., & Plaut, D. C. (2005). Dissociations in performance on novel versus irregular items: Single-route demonstrations with input gain in localist and distributed models. *Cognitive Science, 29*, 627-654.

Levy, S.D. (2007) Changing semantic role representations with holographic memory. In Computational Approaches to Representation Change during Learning and Development: Papers from the 2007 AAAI Symposium. Technical Report FS-07-04, AAAI Press.

Markman, A. (1999). *Knowledge Representation*. Mahweh, NJ: Lawrence Erlbaum.

McClelland, J. L., & Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception, Part 1: An account of basic findings. *Psychological Review, 88*, 375-405.

New, B., Ferrand, L., Pallier, C., & Brysbaert, M. (2006). Re-examining word length effects in visual word recognition: New evidence from the English Lexicon Project. *Psychonomic Bulletin & Review, 13,* 45-52.

Perry, C., Ziegler, J. C., & Zorzi, M. (2007). Nested incremental modeling in the development of computational theories: The CDP+ Model of reading aloud. *Psychological Review, 114*, 273-315.

Plate, T. A. (1994). Distributed representation and nested compositional structure. Department of Computer Science, University of Toronto.

Plaut, D. C. (1999). A connectionist approach to word reading and acquired dyslexia: Extension to sequential processing. *Cognitive Science, 23*, 543-568.

Plaut, D. C. & Gonnerman, L. M. (2000). Are non-semantic morphological effects incompatible with a distributed connectionist approach to lexical processing? *Language and Cognitive Processes, 15*, 445-485.

Plaut, D. C., & Kello, C. T. (1999). The emergence of phonology from the interplay of speech comprehension and production: A distributed connectionist approach. In B. MacWhinney (Ed.), *The Emergence of Language* (pp. 381-415). Mahwah, NJ: Erlbaum.

Plaut, D. C., McClelland, J. L., Seidenberg, M. S., & Patterson, K. (1996). Understanding normal and impaired word reading: Computational principles in quasi-regular domains. *Psychological Review, 103*, 56-115.

Pollack, Jordan. (1990). Recursive Distributed Representations. *Artificial Intelligence, 46*, 77−105.

Rastle, K., & Coltheart, M. (1998). Whammy and double whammy: Length effects in nonword naming. *Psychonomic Bulletin and Reviews, 5*, 277-282.

Rastle, K., & Coltheart, M. (2000). Lexical and nonlexical print-to-sound translation of disyllabic words and nonwords. *Journal of Memory & Language, 42,* 342-364.

Rumelhart, D. E., Durbin, R., Golden, R., & Chauvin, Y. (1995). Backpropagation: The basic theory. In Y. Chauvin and D. E. Rumelhart (Eds), *Backpropagation: Theory, Architectures, and Applications* (pp 1-34).

Saffran, J. R., & Sahni, S.D. (2007). Learning the sounds of language. In M. Joanisse, M. Spivey, and K. McCrae (Eds.), Cambridge Handbook of Psycholinguistics, Cambridge University Press.

Seidenberg, M. S., Plaut, D. C., Petersen, A. S., McClelland, J. L., & McRae, K. (1994). Nonword pronunciation and models of word recognition. *Journal of Experimental Psychology: Human Perception and Performance, 20*, 1177–1196.

Seidenberg, M. & McClelland, J. (1989). A distributed developmental model of word recognition and naming. *Psychological Review, 96*, 523-568.

Sibley, D., E. (2008). Large scale modeling of single word reading and recognition. Unpublished PhD thesis, George Mason University.

Sibley, D. E., Kello, C. T., Plaut, D. C., & Elman, J. L. (2008). Large-scale modeling of wordform learning and representations. *Cognitive Science, 32*, 741 -754.

Sibley, D. E. & Kello, C. T. (2004). Computational explorations of double dissociations: Modes of processing instead of components of processing. *Cognitive Systems Research, 6*, 61-69.

Wickelgren, W. A. (1969). Context-sensitive coding, associative memory, and serial order in (speech) behavior. *Psychological Review, 76*, 1-15.

Yap, M. (2007). Visual word recognition: Explorations of megastudies, multisyllabic words, and individual differences. Unpublished PhD thesis, Washington University.

Yarkoni, T., Balota, D. A., & Yap, M. J. (submitted). Levenshtein distance measures of orthographic and phonological similarity in visual word recognition.

Word Sets	N	Percent Item Variance Accounted For		
		CDP+	Simulation 1	Simulation 2
Monosyllabic	5,191	18.4%	16.4%	14.7%
Monomorphemic Monosyllabic	3,141	22.0%	19.1%	14.4%
Mono & Bisyllabic	13,191	NA	NA	16.6%
Monomorphemic Mono & Bisyllabic	5,718	NA	NA	25.2%

Nonword Sets	N	Percent Acceptable Pronunciations		
		CDP+	Simulation 1	Simulation 2
Monosyllabic (Seidenberg et al., 1994)	589	93.8%	86.8%	84.7%
Bisyllabic (withheld from training)	2,164	NA	NA	65.0%

Table 1. Model performance for words and nonwords

Lexical Variable	ELP	CDP+	Sim. 1
Frequency	.191*	.429*	.203*
Orthographic length	.131*	.150*	.057*
Coltheart's Orthographic N	.112*	.109*	.116*
Coltheart's Phonological N	.046*	.078*	.087*
Levenshtein Orthographic Distance	.149*	.154*	.158*
Levenshtein Phonological Distance	.125*	.142*	.193*
Onset Consistency	.031*	.004*	.068*
Rhyme Consistency	.000	.000	.008
Levenshtein Consistency	.001*	.005*	.031*
Frequency * Orthographic length	.019*	.001	.014*
Frequency * Coltheart's N	.013*	.000	.014*
Frequency * Onset Consistency	.000	.000	.000
Frequency * Rhyme Consistency	.000	.000	.000
Frequency * Levenshtein Consistency	.000	.000	.000

Table 2: Effect sizes (R^2) of lexical variables for monosyllabic ELP latencies, CDP+ latencies, and Simulation 1 latencies

Lexical Variable	ELP	Sim. 2
Frequency	.216*	.268*
Orthographic length	.146*	.212*
Syllabic length	.081*	.207*
Coltheart's Orthographic N	.130*	.258*
Coltheart's Phonological N	.093*	.293*
Levenshtein Orthographic Distance	.177*	.342*
Levenshtein Phonological Distance	.172*	.379*
Onset Consistency	.046*	.086*
Rhyme Consistency	.030*	.096*
Levenshtein Consistency	.063*	.194*
Stress typicality	.039*	.114*
Frequency * Orthographic length	.017*	.013*
Frequency * Coltheart's N	.023*	.020*
Frequency * Onset Consistency	.000	.000
Frequency * Rhyme Consistency	.000	.000
Frequency * Levenshtein Consistency	.000	.000

Table 3: Effect sizes (R^2) of lexical variables for bisyllabic ELP latencies and Simulation 2 latencies

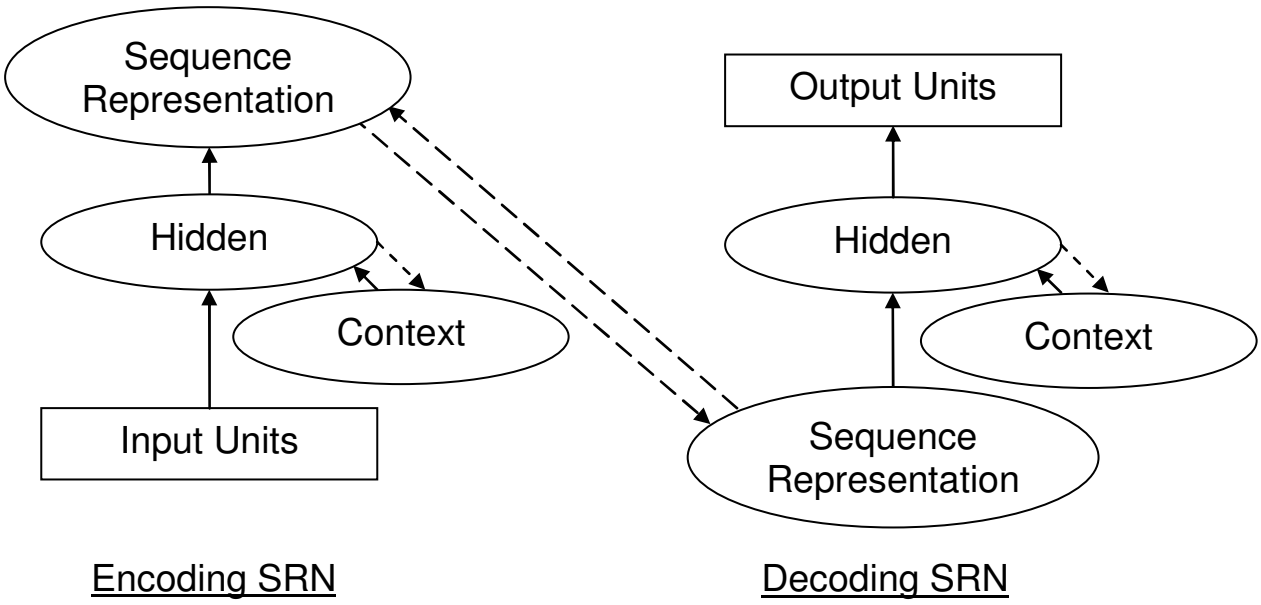

Figure 1. General sequence encoder architecture. Inputs patterns are presented sequentially and integrated at the first sequence representation group to generate a learned representation at the end of the input sequence. The sequence representation is then copied to the second sequence representation group and used as a plan representation to generate an output sequence. Error between output and target sequence is backpropagated and summed at the sequence representation units. This summed error is used as a target signal while the input sequence is replayed through the Encoding SRN, and error is backpropagated from the sequence representation units. Context units and hidden units are computed as in the standard SRN architecture.