Learning Representations of Orthographic Word Forms

Daragh E. Sibley (dsibley@gmu.edu)
Department of Psychology, George Mason University
Fairfax, VA 22030-4444 USA

Christopher T. Kello (ckello@gmu.edu)
Department of Psychology, George Mason University
Fairfax, VA 22030-4444 USA

Abstract

Presented is an extension of the simple recurrent network (SRN), termed the sequence encoder, which learns fixed-width representations of variable-length sequences. This architecture was used to learn orthographic representations for nearly 75,000 English words, of which nearly 69,000 were multisyllabic. Analyses showed that sequence encoder representations are shaped by the dependencies among letters in English word forms that reflect orthographic structure. The model was used to predict participant ratings of the orthographic legality of pseudowords, and results showed that the model accounted for a substantial amount of variance in the ratings.

Introduction

Orthographic structure in English word forms is reflected in complex dependencies among letters and positions within each word. The probability of a given letter being present in a word may depend upon the presence of other letters, resulting in common clusters like GH. The probability of finding a given letter may vary by position; for instance, the letter Z is rarely found at the end of English words. A representational scheme intended to capture the structure of English word forms must be sensitive to these as well as other kinds of dependencies. Ideally, a representational scheme should be sufficiently sensitive to the structure in its corpus to differentiate word forms by their legality, where legality is determined by a word form's conformity to standard English dependencies.

Representational schemes have been used to build models that capture the structure of English word forms, notably those used in Plaut, McClelland, Seidenberg, and Patterson (PMSP) and the dual route cascaded (DRC) models of lexical processing (Plaut et al., 1996; Coltheart et al., 2001). From the representational schemes used in these two models it was possible to extract a substantial amount of information about the structure in English word forms. However, these representational schemes have restricted models to processing only monosyllabic word forms. This is due to difficulties inherent in representing the position of letters in multisyllabic word forms, arising from the increased variability in length of multisyllabic word forms.

The difficulty with variable-length word forms can be described as a problem of alignment. Choosing how to align word forms of variable-length involves defining how the positions in one word relate to the positions in another. A good alignment scheme should capture information about the similarities between a pair of word forms. To illustrate the problem, if the two orthographically similar sequences JUMP and JUMPS are aligned by their first letter then they share four of the same letters in the same positions; thus for these two sequences, alignment by the first letter suggests they are very similar. However, when aligned by their first letter the two orthographically similar sequences BACK and ABACK share 0 letters in the same position. If BACK and ABACK are instead aligned by their last letter, they share four letters in the same position. However, when aligned by their last letter, JUMP and JUMPS share no letters. Essentially, aligning word forms by either their first or last letter, or even around vowels (Daugherty & Seidenberg, 1992), fails to represent the similarities in various word forms.

The desire to represent multisyllabic word forms motivated our efforts to develop a mechanism for learning representations of sequences that is based on the well-known Serial Recurrent Network (SRN) architecture (Elman, 1990; Jordan, 1986). The sequence encoder (Kello et al., 2004) learns such representations in the service of learning to encode and decode sequences of variable-length. By virtue of being based on the SRN architecture, the sequence encoder representations are shaped by dependencies (e.g., conditional probabilities) that exist in its training corpus of sequences. Thus the sequence encoder may be used to learn representations of both mono- and multisyllabic word forms.

The sequence encoder, shown in Figure 1, is created by conjoining two SRNs. The first, called the encoding SRN, receives a sequence of input letters and encodes them into a single distributed representation, called the bridge representation. The second, called the decoding SRN, decodes the bridge representation into a sequence of output letters.

By conjoining these two SRNs, the sequence encoder can be trained to code all the letters of a word in order. By contrast, a standard SRN that is presented with letters one at a time and trained to simply predict each subsequent letter will learn to code only the information necessary to minimize the prediction error. For many kinds of lexical corpora, it is either not necessary or not useful for the SRN to code information about all, or even many, of the letters in order to minimize the prediction error. The consequence

is that the learned representations will not be shaped to code all the letters and their positions.

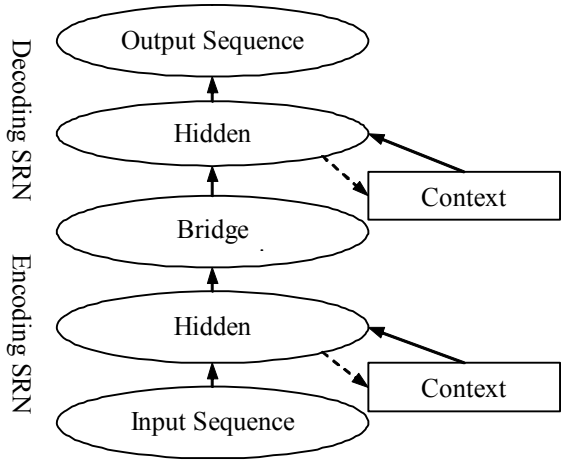

Figure 1: The sequence encoder architecture

Most generally, the sequence encoder can be trained to learn bridge representations for any variable-length input/output mapping. In other words, input sequences do not have to match output sequences in length or content. However, the task of learning word forms is most directly implemented by training the sequence encoder to reproduce verbatim each input sequence as an output sequence. This "auto-encoding" task forces the bridge representations to code each letter in its position. It has also been shown that connectionist auto-encoder models exploit the statistical structure of their inputs (Bishop, 1995).

The remainder of this paper examines the sequence encoder's ability to learn representations that are sensitive to the structure of both mono- and multisyllabic word forms. To this end, a sequence encoder was trained on 75,265 English orthographic word forms, varying in length from 1 to 18 letters. The model's sensitivity to the dependencies in English word forms was assessed by testing its ability to account for participants' ratings of the orthographic legality of pseudowords.

Modeling Method

Training corpus. Representations were learned for 74,265 English words, 68,945 of which were multisyllabic. These words were chosen by intersecting the CMU pronunciation dictionary (available at http://www.speech.cs.cmu.edu/cgi-bin/cmudict) and the Wall Street Journal Corpus (Marcus et al., 1993), then discarding homographs (for purposes outside the scope of the work discussed here) and words with more than 18 letters.

Input and output representations. Each letter in a sequence was coded by activating a single unit in two different banks of nodes. The first bank consisted of 26 units, each representing a single letter (A through Z) with one additional unit representing the end of a sequence. The second bank consisted of two units which denoted whether the current letter was a consonant or vowel (the vowel letters were A, E, I, O, U and Y).

Model architecture. The sequence encoder, as shown in Figure 1, was built from 7 layers of connectionist units. The first layer received an input sequence on a set of 29 localist nodes. Activation for each letter in the sequence was propagated to hidden and context layers, producing distributed representations along each layers' 500 nodes. Activation flowed from these representations to the Bridge layer, where they were accumulated to generate a single 500 dimension distributed representation for the entire input sequence. Activation was then propagated thorough a set of hidden and context layers each having 500 nodes, to produce a sequence of localist representations along the 29 localist nodes of the output layer.

The representational layers of the sequence encoder were connected as shown in Figure 1. Any two layers linked by a solid arrow were fully connected; an independent connection weight extended from each unit in the sending layer to each unit in the receiving layer. Before training, these weights were initialized to small random values. The dashed arrows represent a copy function; the activation pattern on the sending layer is copied onto the receiving layer at the end of each time step.

The sequence encoder processed each sequence as a number of discrete time steps, first encoding then decoding a sequence. On each encoding time step, the input units representing a single letter were activated and the encoding SRN run. Upon completion of the last encoding time step, a final bridge representation was generated. Then, for each decoding time step the bridge representation was run through the decoding SRN, producing a pattern of activation on the output layer. On each decoding time step, the most active letter unit and consonant/vowel unit was taken as the model's response. For a sequence to be correct, every letter and consonant/vowel value had to be produced in the proper order.

The net input to each hidden and output unit was calculated as the dot product of the incoming weight and activation vectors. The activation of each hidden unit was computed as the hyperbolic tangent of its net input. This function is sigmoidal with asymptotes at -1 and 1. The activation of each output unit was computed as the exponential of its net input, normalized across the bank of output units. This normalized exponential function is appropriate because letters were coded as localist representations and so could be interpreted as the probability of the output of each letter (Rumelhart, 1995).

Weight updates were made after every batch of 50 training examples chosen at random from the corpus. For each element of each sequence, error derivatives were calculated and back-propagated up to the bridge representation. This error signal was used to train the decoding SRN, with a learning rate of 0.00005. Next, the error signal accumulated on the bridge layer, for the entire

length of the sequence, was back-propagated through time while the sequence was reset onto the input layer. Weight updates were then made on the encoding SRN with a learning rate of 0.00000001. A smaller learning rate was needed for the encoding SRN because of the accumulation of error derivatives on the bridge units. Learning appeared to reach asymptote after 250,000 epochs of training, and was halted.

Training and testing the sequence encoder. A coarse test of the sequence encoder's sensitivity to structure was achieved by examining the model's ability to auto-encode 3 different types of sequences. These three types of sequences embody 3 different levels of legality, so a model sensitive to the structure of English word forms should process these different sequences with different degrees of success.

First, sensitivity to the structure of English should result in correct auto-encoding of the trained words. Second, to demonstrate that the model discovered general structure instead of memorizing the specific sequences in the training corpus, the model was assessed for its ability to correctly auto-encode "Legal Nonwords". A legal nonword is a sequence of letters that conform to the structure of English word forms, but are not in the training corpus (i.e., pseudowords). Correct performance of these items required the model to generalize information about the dependencies in the training corpus. Third, to test whether the model learned about the structure of English word forms and not how to auto-encode arbitrary sequences, we assessed its performance on sequences of letters that did not conform to the structure of English, here called "illegal nonwords". To achieve this analysis, we needed to generate legal nonwords and illegal nonwords.

To create legal nonwords, we took each trained word form and replaced one of its letters. These replacements yielded new sequences which were not in the training corpus. To make the new sequences legal, each replacement letter was chosen from a word that shared the letters surrounding the replaced letter. To illustrate, the word SPORTY could be changed into SPARTY by replacing the sequence SPORT with SPART, where SPART occurs in the word SPARTAN. In this example, a letter is replaced by a new legal letter, determined by matching the 4 flanking letters. For each of the replacements used to create a legal nonword, a single letter was chosen at random and replaced with another letter of the same class, vowel or consonant. Three letters to either side were matched unless the beginning or end of the sequence was encountered.

To create illegal nonwords, the letters in each of the legal nonwords were scrambled. An inspection of the resulting illegal nonwords revealed that this method generally produced sequences that violated the structure of English word forms.

Model Results

The sequence encoder correctly processed 89% of the words in its training corpus, including 88% of the 68,945 multisyllabic words. This performance demonstrated the sequence encoder's ability to generate representations in the service of auto-encoding both monosyllabic and multisyllabic word forms. The sequence encoder correctly processed 76% of the legal nonwords, including 75% of the multisyllabic legal nonwords. Finally, the model correctly auto-encoded only 24% of the illegal nonwords. The poor performance of the model on illegal nonwords relative to legal nonwords is clear evidence that the auto-encoding task drove the model to discover and exploit dependencies among letters that reflect the structure of English word forms.

The results of this simulation show that the sequence encoder provides a viable means of learning representations for large numbers of monosyllabic and multisyllabic word forms. Representations were shaped by the structure of English word forms, as evidenced by the selective generalization to legal but not illegal nonwords. This shaping occurred through the learning of dependencies among letters and positions within the sequences. This sensitivity to dependencies enabled the model to differentiate between words based on their legality, despite never having been explicitly trained to perform this task.

Pseudoword Legality Ratings

The previous analysis suggests that the sequence encoder was able to learn some amount of structure in both monosyllabic and multisyllabic word forms. However, the analysis provided little information about the extent to which the bridge representations became sensitive to different sorts of dependencies. Further, it did not show whether the dependencies captured by the sequence encoder corresponded to those to which real language users are sensitive. These questions were addressed by evaluating the model's sensitivity to orthographic structure relative to behavioral data. In particular, we tested whether the model's success at processing novel word forms predicted participants' judgments about the legality of a sequence of letters. We then examined the dependencies to determine which might give rise to the sequence encoder's predictive power.

The sequence encoder's sensitivity to the structure of English word forms can be assessed by the model's ability to auto-encode novel sequences. Successful auto-encoding of a novel word form requires the model to generalize information about the structure of the learned word forms. A failure to auto-encode a novel sequence reflects the model's inability to represent the novel sequence using the information it acquired about the structure of English. As such, the model's success in processing each novel sequence should reflect the legality of the sequence. The following analysis tested the extent to which the sequence

encoder's performance predicted participants' judgments of orthographic legality.

The comparison of the model and behavioral data was performed with word forms within a restricted range of legality; none of the sequences in this analysis were either typical words (like FOOTBALL) or completely illegal nonwords (like TLLBAOOF). Because of the restricted range of legality, the model needed to differentiate between subtle differences in legality, in order to account for any variance in behavioral responses. This produced a more rigorous test of the sequence encoder's abilities.

Method

The sequence encoder's ability to generalize was compared to participants' judgments about the legality of 600 nonwords. The nonwords for this analysis were developed using the previously described method for generating legal nonwords. To generate word forms distinct from real English, the procedure previously discussed was run iteratively, 5 times on each word. This resulted in 600 relatively legal nonwords, each of which corresponded to a real word with up to 5 of its letters replaced.

As would be expected, the nonwords created by replacing up to 5 letters in each word tended to be less legal than those produced by replacing a single letter. This resulted in decreased performance on these new legal nonwords relative to those previously discussed - 71% instead of 76% correct, respectively. Also, the shorter word forms created by this procedure tended to be more legal than the longer word forms. The final 600 sequences were chosen by randomly selecting 150 of the new legal nonwords from the 4 most frequent lengths; 5, 6, 7, and 8 letters. Length was restricted to this range to reduce any confounding length effects introduced by the procedure used to make nonwords. This was necessary because iteratively replacing 5 letters would have a different effect on a word form composed of 3 letters than one composed of 13 letters. Of the final 600 new legal nonwords, 571 were multisyllabic. A representative sample of the word forms used in this analysis is shown in Table 1.

Worst performance			Best performance	
0	1	2	3	4
sriteanf	roadham	wingins	broins	roins
voruranc	baclater	ronte	shiriner	teres
aterkan	dertlee	prare	macee	lanter
mislerl	naintira	battly	barss	cears
crultie	crerley	tonkies	bardey	stors
dexcuges	bencham	hantend	gaged	funter
debsane	overetle	dleised	cerel	lired
lotingke	gertisos	stadio	penscs	panen
echoc	corleder	boalans	jurded	rones
intail	tatzir	mause	sloners	beins

Table 1: Legal nonwords for which *P*, a measure of the sequence encoder's performance, was closest to each of the integers; 0, 1, 2, 3, and 4.

To formulate a more sensitive measure of the model's generalization it was necessary to measure the model's performance beyond a binary (correct or incorrect) distinction. Instead, a continuous measure of the sequence encoder's performance was generated for each of the 600 words. This was possible because the model tended to activate target nodes less than the maximum allowed.

One conceptual interpretation of a connectionist model's output is that the activation of each output node reflects the probability (nodes assume a value between 0 and 1) of producing the represented feature. So the probability of producing a fully correct sequence could be calculated as the product of the activation on each of the target nodes. However, because many of the model's outputs were asymptotically correct, a logarithmic transform was used to provide a more sensitive measure. A continuous measure of the model's performance was created

$$P = -\log(1 - (\Pi a_i)),$$

where P represents the model's performance on a given sequence and a_i is the activity on target node i in the sequence (the activation on a perfectly produced target node was 1). For the nonwords used in this analysis, P assumed a value between 0 and 4, (for all but 3 discarded outliers) with 0 being the worst produced word form and 4 being the best produced.

Nine participants judged each of the 600 legal nonwords for their conformity to the structure of English word forms. Participants were instructed to consider the likelihood that each sequence was an English word they were unfamiliar with. Legality judgments for each of the 600 word forms were made on a 5 point scale.

Results

Table 1 shows the 10 word forms for which P was closest to each of the integers; 0, 1, 2, 3, and 4. Inspection of this table yields two observations, which are corroborated by later analysis. First, the model generated a reasonable estimation of the legality of nonwords. That is, word forms in the left most column were less regular than those in the right most column. Second, the length of a sequence interacted with the model's proxy for legality - sequence length accounted for 17.2% of the variance in P. However, sequence length also accounted for 5.1% of the variance in the participants' average responses. Because the model and participants responded differentially to items of different length, we can partially attribute the length by legality interaction to an anomaly in the procedure used to generate the legal nonwords.

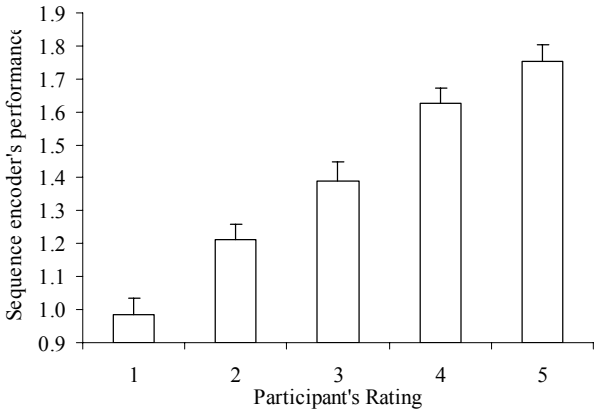

Figure 2: The sequence encoder's performance on word forms receiving different legality ratings from participants

To check the experimental paradigm, Cronbach's alpha was calculated for the 9 participants' responses. At 0.87, it shows the participants' judgments were largely internally consistent. We then tested the correspondence between the model's performance on novel sequences (*P*) and the participants' ratings for each word form. This was achieved by calculating the model's average performance on each of the 5 ratings provided by each participant. Figure 2 was created by averaging this quantity across participants. As depicted in this figure, each of the 5 different ratings provided by a participant tended to be associated with the production of a different response pattern from the model, $F(4, 32) = 38.64$, $p < .001$. This statistically significant capacity of the sequence encoder to account for behavioral data suggests it has developed a sensitivity to the structure of mono and multisyllabic English word forms. Further, the sequence encoder's sensitivity to dependencies is in some way similar to the sensitivity of native English readers.

To depict the model's ability to predict human behavior, we used a regression analysis to compare the model's performance on each of the legal nonwords with the average of the 9 participants' response to each item. Figure 3 is a scatter plot showing the relationship between the sequence encoder's proxy for legality and the average of the participants' judgments of legality, by item. As depicted, the model predicted 14.5% of the variance in the participants' judgments. This demonstrates that the model became sensitive to some dependencies to which the participants were also sensitive. This suggests that the sequence encoder discovered some structure in English orthography similar to that discovered by a skilled English reader.

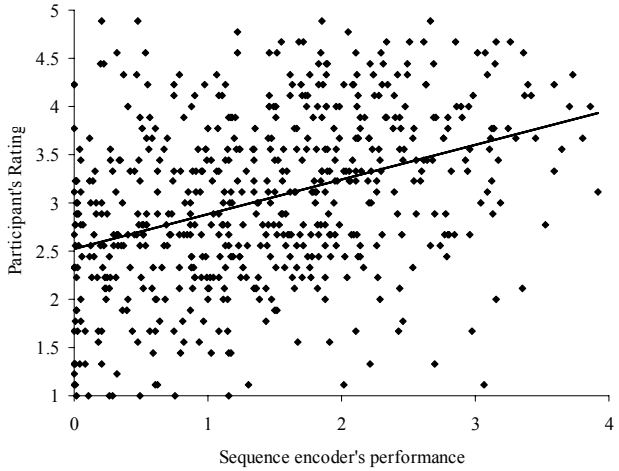

Figure 3: Scatter plot of participant and model responses

Conclusions

The present work examines the sequence encoder's ability to generate representations that are sensitive to the structure of orthographic word forms. The sequence encoder learned representations for more than 75,000 English words, varying in length from 1 to 18 letters, with nearly 69,000 of the learned words being multisyllabic.

In processing multisyllabic words, the sequence encoder overcame problems associated with representing positions in variable-length sequences. While other connectionist systems have been used to process sequences, this sequence encoder is particularly suited to learning representations of lexical stimuli, an ability which may be useful in future models of lexical processing. To elaborate on this point, we will briefly contrast the sequence encoder with three related connectionist systems.

The sequence encoder is an extension of the SRN, which was designed to process sequences while performing the prediction task (Elman, 1990; Jordan, 1986). These SRNs maintain a representation of prior elements in order to predict an upcoming element. Prediction is accomplished by using whatever conditional probabilities exist in the trained sequences (Elman, 1995). The sequence encoder inherits the SRN's sensitivity to conditional probabilities, but goes beyond the standard SRN by creating a task that explicitly forces representations to code all the elements of a sequence along with their positions. The sequence encoder thus becomes sensitive to conditional probabilities among all the letters of a word, which is also generally true of skilled readers (see McClelland & Rumelhart, 1981).

Another connectionist system for processing variable-length sequences is the recurrent auto-associative memory (RAAM; Pollack, 1990). In RAAMs, connectionist units are trained to pack and unpack representations in a strictly recursive fashion. By varying the number of recursive steps, the model can input or output a variable-length sequence. However, the strictly recursive nature of this process makes it ill-suited to the problem of learning word

forms. The RAAM architecture imposes a fixed hierarchical structure on each sequence, while English words are characterized by a more rough hierarchical organization that is free to vary from one word to the next (Andrews et al., 2004).

A recently-developed connectionist system uses fully recurrent networks to learn representations of variable-length sequences (Botvinick & Plaut, in press). In this architecture, a sequence is input to the model one element at a time, much like the sequence encoder, until an output cue triggers the model to reproduce the sequence. This architecture was developed as a model of serial recall, a task with notable differences from lexical processing, and is therefore not well-suited to our current needs. In particular, the model requires extensive training, and it is unclear how well it would scale and generalize in much larger linguistic domains like the one discussed here.

Current models of the lexical system incorporate only monosyllabic words, because their representational schemes cannot easily incorporate multisyllabic words (Coltheart et al., 2001; Plaut et al., 1996). The sequence encoder's ability to learn representations for large numbers of multisyllabic words could be exploited to develop new models of impaired and unimpaired lexical processing (see Kello, in press). These models could process a number of words approaching an adult's vocabulary and could address large quantities of behavioral data regarding multisyllabic words (Balota et al., 2002). These are some of the directions that are currently being pursued with the sequence encoder.

Acknowledgements

This work was funded in part by NIH Grant MH55628, and NSF Grant 0239595. The computational simulations were run using the Lens network simulator (version 2.6), written by Doug Rohde (http://tedlab.mit.edu/~dr/Lens). We thank David Plaut for collaborations on precursors to this work.

References

Andrews, S., Miller, B. & Rayner, K. (2004) Eye fixation measures of morphological segmentation of compound words: There is a mouse in the mousetrap. European Journal of Cognitive Psychology, 16(2), 285-311.

Balota, A. A., Cortese, M. J., Hutchison, K. A., Neely, J. H., Nelson, D., Sompson, 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:

Bishop, C. M. (1995). *Neural networks for pattern recognition. Oxford*: Oxford University Press.

Botvinick, M., & Plaut, D. C. (in press). Short-term memory for serial order: A recurrent neural network model. *Psychological Review*.

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.

Daugherty, K., & Seidenberg, M. S. (1992). Rules or connections? The past tense revisited. *Proceedings of the 14th Annual Confrence of the Cognitive Science Society.* Hillsdale, NJ: Lawrence Erlbaum Associates. Pp. 259-264.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2), 179-211.

Elman, J. L. (1995). Language as a dynamical system. In R.F. Port & T. van Gelder (Eds.), *Mind as Motion: Explorations in the Dynamics of Cognition.* Cambridge, MA: MIT Press. Pp. 195-223.

Jordan, M. I. (1986). *Serial order: A parallel distributed processing approach* (No. 8604 ICS Technical Report): University of California at San Diego, La Jolla, CA.

Kello, C. T. (in press). Considering the junction model of lexical processing. To appear in S. Andrews (Ed.), *All about words: Current research in lexical processing*. Sydney: Psychology Press.

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.

Marcus, M., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19, 313-330.

McClelland, J. L., & Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception: I. An account of basic findings. *Psychological Review, 88*, 375-407.

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. 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(1), 56-115.

Pollack, J. B. (1990). Recursive distributed representations. *Artificial Intelligence, 46*, 77-105.

Rumelhart, D. E., Durbin, R., Golden, R., & Chauvin, Y. (1995). Backpropagation: The basic theory. In Chauvin, Yves (Ed); Rumelhart, David E (Ed) (1995) *Backpropagation: Theory, architectures, and applications Developments in connectionist theory* (pp 1-34).