A Simple Co-Occurrence Explanation for the Development of Abstract Letter Identities

Thad A. Polk
Department of Psychology
University of Pennsylvania
3815 Walnut Street
Philadelphia, PA 19104-6196
polk@psych.upenn.edu

Martha J. Farah
Department of Psychology
University of Pennsylvania
3815 Walnut Street
Philadelphia, PA 19104-6196
mfarah@psych.upenn.edu

Abstract

Evidence suggests that an early representation in the visual processing of orthography is neither visual nor phonological, but codes *abstract letter identities* (ALIs) independent of case, font, size, etc. How could the visual system come to develop such a representation? We propose that, because many letters look similar regardless of case, font, etc., different visual forms of the same letter tend to appear in visually similar contexts (e.g., in the same words written in different ways) and that correlation-based learning in visual cortex picks up on this similarity among contexts to produce ALIs. We present a simple self-organizing Hebbian neural network model that illustrates how this idea could work and that produces ALIs when presented with appropriate input.

Abstract Letter Identities

A growing body of evidence suggests that an early processing stage in reading involves the computation of *abstract letter identities* (ALIs), that is, a representation of letters that denotes their identity but that abstracts away from their visual appearance (uppercase vs. lowercase, font, size, etc.) (Coltheart, 1981; Besner, Coltheart, & Devalaar, 1984; Bigsby, 1988; Mozer, 1989; Prinzmetal, Hoffman, & Vest, 1991). It appears that this representation is not a phonological code, but is computed much earlier by the visual system itself (Bigsby, 1988). For example, even when subjects are asked to classify pairs of letter strings as same or different based purely on physical criteria, letter strings that differ in case but that share the same letter identities (e.g., HILE/hile) are distinguished less efficiently than are strings with different spellings but the same phonological code (e.g., HILE/hyle) (Besner et al., 1984). Similarly, just as subjects tend to underestimate the number of letters in a string of identical letters (e.g., DDDD) compared with a heterogenous string (e.g., GBUF), subjects also tend to misreport the number of Aa's and Ee's in a display more often when uppercase and lowercase instances of a target appear than when one of each target appears (Mozer, 1989). The repetition of ALIs must be responsible for this effect because visual forms were not repeated in the mixed-case displays. Notice that none of these effects can be due simply to the visual similarity of different visual forms for a given letter because they also show up for letters whose visual forms are *not* visually similar (e.g., A/a, D/d, E/e, etc.).

The evidence then is that a relatively early representation in the visual processing of orthography (before lexical access or the representation of phonology) does not reflect fundamental visual properties such as shape—stimuli that have little or no visual similarity (e.g., "A" and "a") are represented with the same code. This is quite surprising. After all, how could the visual system come to develop such a representation? Reading and writing are relatively recent developments in evolutionary terms so a genetic explanation seems implausible. But what learning mechanisms would cause the visual system to develop such a representation?

The Co-Occurrence Hypothesis

We propose that the statistics of co-occurrence among letters in words interact with correlation-based Hebbian learning in the visual system of the brain to produce ALIs. Specifically, because many letters look similar regardless of case, font, etc., we assume that there is often a high degree of visual similarity among the contexts in which different visual forms of the same letter appear, and that the visual system picks up on this correlation and produces representations corresponding to ALIs as a result.

During reading, people are exposed to the same word written in a variety of different ways: all caps and lowercase, in different fonts, in different sizes, etc. As a result, the contexts in which one visual form of a letter appear are similar to the contexts in which other visual forms of that same letter appear. For example, if the visual form "a" occurs in a given context (e.g., between "c" and "p" in "cap", before "s" in "as"), then the visual form "A" almost always occurs in a similar context (between "C" and "P" in "CAP", before "S" in "AS"). What is critical, according to our explanation, is not that these contexts involve the same letters, but rather that these contexts are *visually* similar ("c-p" and "-s" *look* similar to "C-P" and "-S" respectively). Of course, the different visual forms of some letters are fairly different (e.g., "D" vs. "d") and so there will be some contexts that are unique to one visual form of a letter (e.g., "a" but not "A" occurs in the context "d-d"). But given that 18 out of 26 letters have a fair degree of similarity in their uppercase and lowercase forms (the obvious exceptions are: Aa, Bb, Dd, Ee, Gg, Nn, Qq, and Rr) and that letters in different fonts and sizes tend to look similar, these cases should be the exception rather than the rule.

A Neural Network Model

How could the visual system pick up on this correlation in the environment to produce representations corresponding to ALIs? Figure 1 presents a simple and natural mechanistic model that demonstrates one possibility. The model is a 2-layer neural network that uses a Hebbian learning rule to modify the weights of the connections between the input and output layers. Hebbian learning is a neurophysiologically plausible mechanism that generally corresponds to the following rule: If two units are both firing then their connection is strengthened, if only one unit of a pair is firing then their connection is weakened (Hebb, 1949). The input layer represents the visual forms of input letters using a localist representation (each unit represents a visual form, similar visual forms (e.g., "C" and "c") are represented by the same unit). This representation does not code letter position as it does not play a role in our explanation (the same mechanism would work for a representation that did code letter position). Initially, the output layer does not represent anything (since the connections from the input layer are initially random), but with training it should self-organize to represent ALIs. Neighboring units in the output layer are connected via excitatory connections and units further away are connected via inhibitory connections, in keeping with the general pattern found in human cortex. The appendix describes the model's details.

The figure illustrates the model's behavior when the first word (say "cap") is initially presented. The pattern of output activity is initially random (Figure 1, left), reflecting the random initial connection strengths. The short-range excitatory connections lead to clusters around the most active units (Figure 1, middle) and these, in turn, drive down activity elsewhere via the longer-range inhibitory connections leading to a single cluster (or a small number) (Figure 1, right). The Hebb rule then strengthens the connections to this active cluster from the active input units (the letters "c", "a", and "p"), but weakens the connections from other (inactive) inputs as well as the connections from the active input to the inactive output units. Because of these weight changes, "c", "a", and "p" will subsequently be biased toward activating units in that cluster while other inputs (e.g., "d") will be biased away from that cluster. So, for example, if the word "dog" was presented next, then it would tend to excite units outside the cluster excited by "cap".

Now suppose we present the same word, but now in uppercase ("CAP", Figure 2). Because "C" and "P" are visually similar to "c" and "p" their input representations will also be similar (in this simple localist model, that means they excite the same units; in a more realistic distributed model, the representations would share many

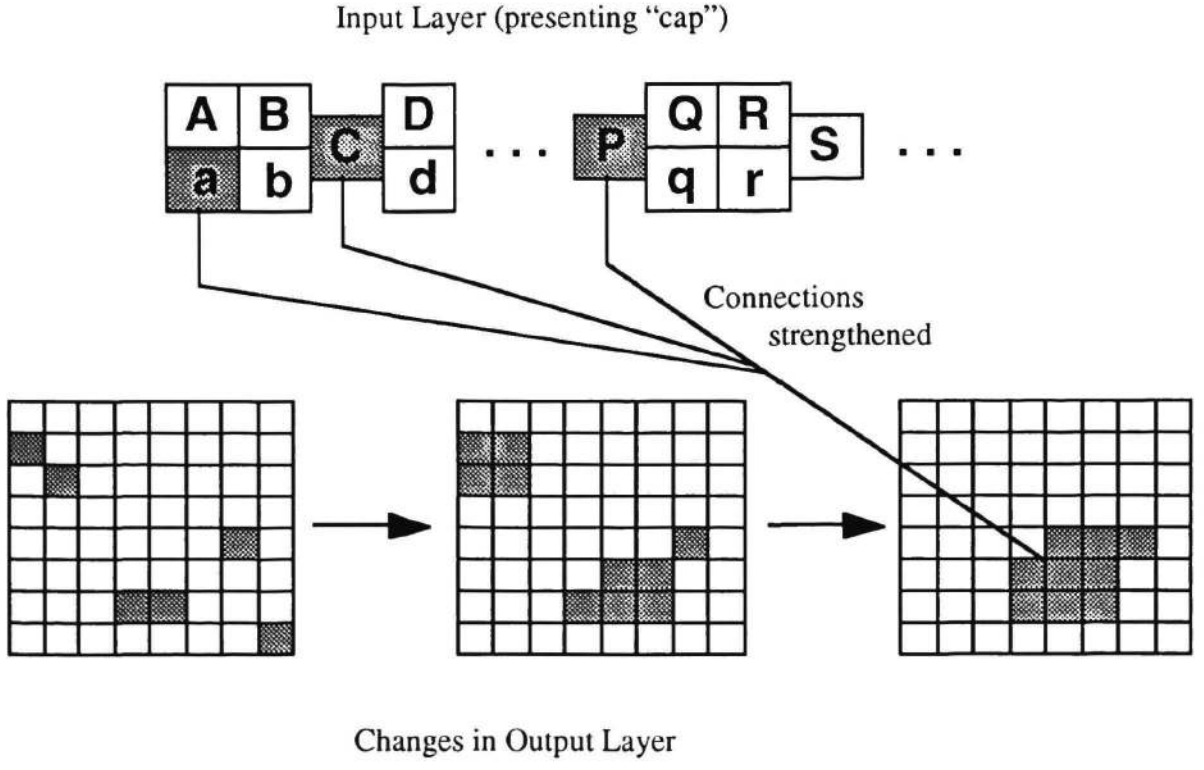

Figure 1. A neural network model of the development of abstract letter identities. The input layer (top) represents the visual forms of input letters, but does not code their position. Letters whose uppercase and lowercase forms are visually similar are represented by a single unit (e.g., C, P, S). In this example, the word "cap" is presented. Initially, the output representation is random (left), but eventually a cluster of activity develops (right) and Hebbian learning strengthens the connections to it from the input letters.

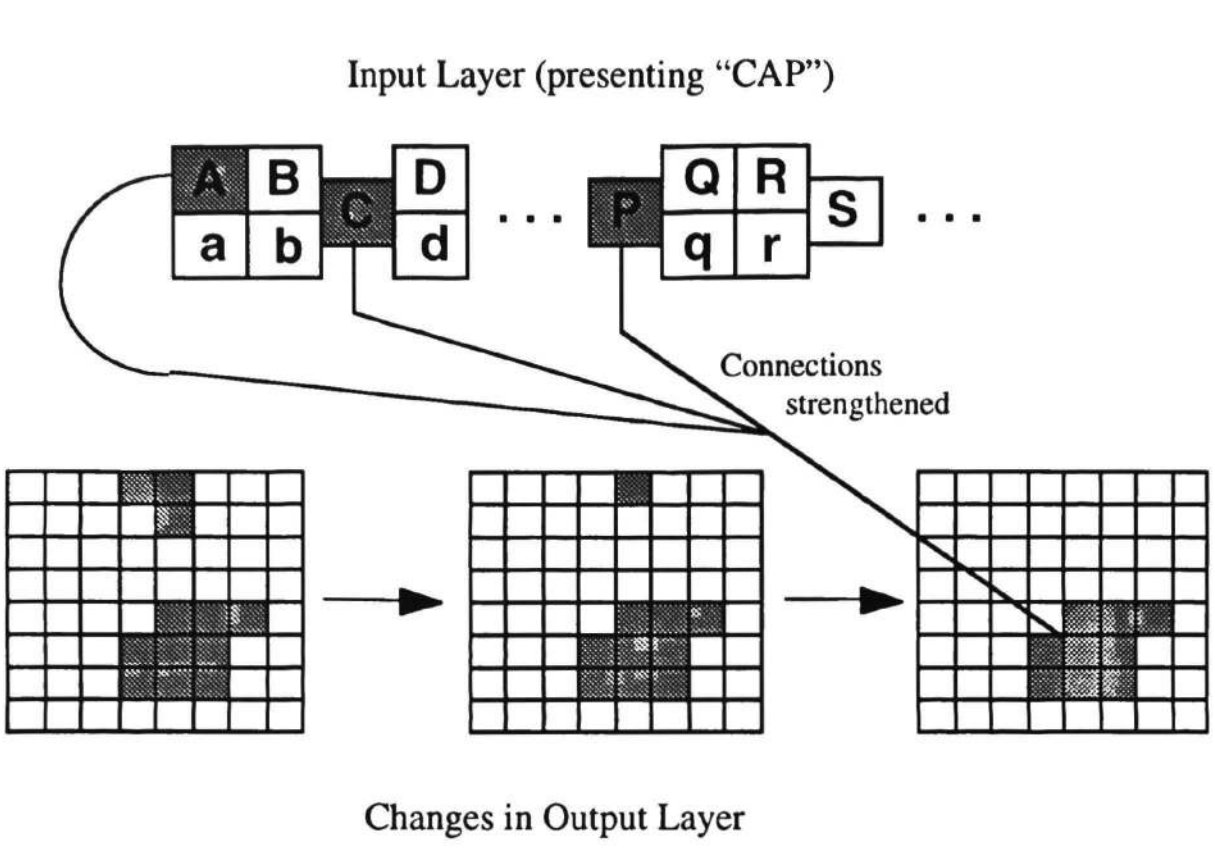

Figure 2. The behavior of the network from Figure 1 when presented with the word "CAP". "C" and "P" excite the previous cluster because of their visual similarity to the "c" and "p" in "cap", but at first "A" excites a distinct set of units (top of the output layer). These two regions of activity compete until the previous cluster wins out. Hebbian learning then strengthens the connections from all the active inputs (including "A") to the cluster, biasing "a" and "A" toward exciting nearby units.

units rather than being identical, but the same process should work). As a result, "C" and "P" will be biased toward exciting some of the same output units that "cap" excited. The input "A", however, has no such bias. Indeed, its connections to the "cap" cluster would have been weakened (because it was inactive when the cluster was previously active) and it excites units outside this cluster (top of the output layer at the left of Figure 2). The cluster inhibits these units via the long-range inhibitory connections and it eventually wins out (right of Figure 2). Hebbian learning again strengthens the connections from the active inputs (including "A") to the cluster. The result is that "a" and "A" are biased toward exciting nearby units despite the fact that they are visually dissimilar and initially excited quite different units. An ALI has emerged.

If this were the whole story then one might expect all letters to converge on the same output representation. But in addition to strengthening the connections between correlated units, the learning mechanism also weakens connections between anti-correlated units. So when two visual forms occur in different contexts (e.g., "a" in "cap" and "d" in "dog"), the network will be biased toward using distinct output units to represent them. Of course, the same is true of the different visual forms of any given letter: Because the two forms appear in many different contexts (e.g., "a" in "cap" vs. "A" in "BAT"), there is pressure on the network to represent them using different output units. Indeed, given that the network will be exposed to far more pairs of different words than pairs of words that differ only in visual appearance (e.g., uppercase vs. lowercase), one might think that the bias toward using distinct output units will far outweigh any tendency toward ALIs. But because the network has a limited amount of space in the output layer in which to represent the visual forms, it will be forced to put some of these representations closer together than others. The critical question is which representations will the network prefer to put nearby, given that it cannot keep them all widely separated. The answer is that it will tend toward moving together those visual forms whose contexts are the most similar. And those visual forms, for the reasons discussed above, will correspond to the same abstract letter identity.

Simulation Results

Figure 3 shows the results of presenting a network like this with simple stimuli that satisfy the constraints outlined above. The stimuli consisted of a random sequence of 36 3-letter words—12 in uppercase and 24 in lowercase (the same 12 words that appeared in uppercase were presented in

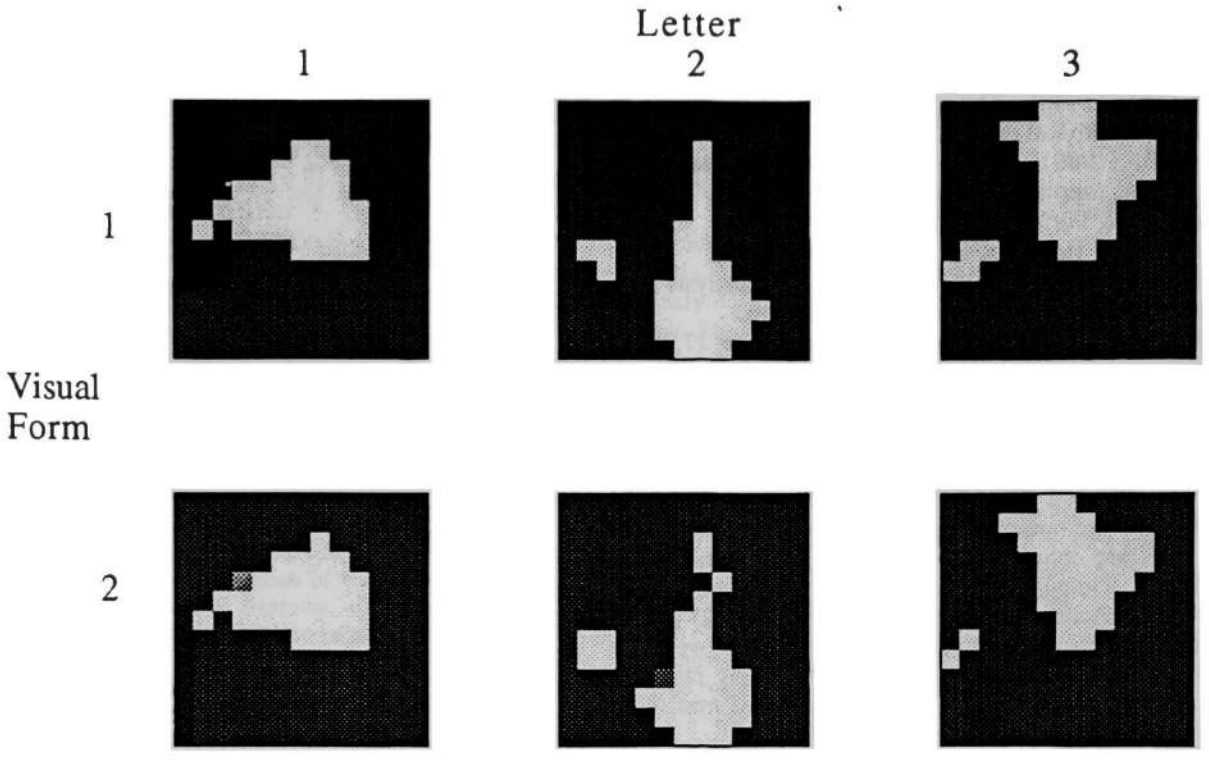

Figure 3. The patterns of output activity when presenting two different visual forms for each of three letters to the trained network model. In all three cases, the two different visual forms activate virtually identical output representations, that is, ALIs. Also note that the three letters each have distinct representations.

lowercase twice in the random sequence)[1]. Each word contained one letter from a set of three ALI candidate letters—each of these letters had two possible visual forms (e.g., uppercase and lowercase)—and each such letter appeared in four of the 12 words. The other two letters were randomly chosen from a set of 20 whose visual forms were similar in uppercase and lowercase. The figure shows the output activity when presenting each of the three candidate ALI letters in each of their visual forms. In all three cases the output representations for the two different visual forms are virtually identical—the output representation corresponds to an ALI for the letter. Also note that the output representations of the three different letters were distinct.

Discussion

This model is relatively simple and that is both an advantage and a disadvantage. On the positive side, it is easy to understand and hence illustrates the theory much more clearly than a more complicated model would. But it also has a number of limitations that future models will need to address: It uses a simple, localist input representation rather than a more realistic distributed code, it only addresses a specialized subset of words (three-letter words in which one letter is an ALI candidate and the other two are not), and it does not model the positions of letters within words. The theory itself does not depend on these assumptions, however, and they could thus presumably be relaxed in future work. In any case, the model provides the first hypothesis to account for the development of ALIs, namely, that they arise from an interaction between the co-occurrence of letters in words and a correlation-based learning mechanism.

1 The two visual forms were presented with different frequency in order to prevent the two forms from balancing out and possibly canceling the effects of competition (e.g., if "cap" and "CAP" occur with the same frequency, then the network will not be biased toward one cluster). Of course, lowercase words occur far more frequently than uppercase words in real text as well.

Acknowledgements

Max Coltheart made helpful comments on this research and we gratefully acknowledge his assistance. This research was supported by a grant-in-aid for training from the McDonnell-Pew program in cognitive neuroscience to T.A.P. and by grants from the NIH, ONR, Alzheimer's Disease Association and the Research Association of the University of Pennsylvania.

References

Besner, D., Coltheart, M., Davelaar, E. (1984), Basic processes in reading: Computation of abstract letter identities, Canadian Journal of Psychology, 38, 126-134.

Bigsby, P. (1988), The visual processor module and normal adult readers, British Journal of Psychology, 79, 455-469.

Coltheart, M. (1981), Disorders of reading and their implications for models of normal reading, Visible Language, 15, 245-286.

Hebb, D.O. (1949), The organization of behavior, a neuropsychological theory, New York, Wiley.

Mozer, M. (1989), Types and tokens in visual letter perception, Journal of Experimental Psychology: Human Perception & Performance, 15, 287-303.

Prinzmetal, W., Hoffman, H., and Vest, K. (1991), Automatic processes in word perception; An analysis from illusory conjunctions, Journal of Experimental Psychology: Human Perception & Performance, 17, 902-923.

Appendix: Details of the Neural Network

The network has 195 total units. 26 are inputs (3 ALI candidate letters x 2 visual forms each + 20 other letters) and the other 169 are outputs in a 13x13 2-D arrangement. All input units are connected to all output units with plastic connections. Output units are connected to neighbors (in 2-D) by fixed excitatory connections (weight = 0.2) and to other output units by fixed inhibitory connections (weight = -0.02). The minimum and maximum unit firing rates are fixed at 0.0 and 100.0 while the minimum and maximum connection weights are fixed at 0.0 and 3.0.

Initially, the activity of output units is uniform random between 0.0 and 10.0 and the connection weights from inputs to outputs is uniform random between 0.0 and 0.6.

The following Hebbian learning rule based on firing rate is used after every cycle to update connection strengths between input and output units: If both pre- and post-synaptic units are firing above threshold (50.0), increase connection weight by 0.08; if both units are below threshold, make no change; otherwise, decrease the connection weight by 0.02.

The output units use a sigmoid transfer function:

$$\text{output} = \frac{100.0}{1 + e^{-(input - 40.0)}}$$

The total input to each output unit is multiplied by a 0.9 gain factor before passing through the transfer function. The input units are clamped to their values and do not decay.