Journal of Experimental Psychology:
Human Perception and Performance
2006, Vol. 32, No. 3, 535–557

0096-1523/06/$12.00 DOI: 10.1037/0096-1523.32.3.535

Contrasting Five Different Theories of Letter Position Coding: Evidence From Orthographic Similarity Effects

Colin J. Davis
Macquarie University

Jeffrey S. Bowers
University of Bristol

Five theories of how letter position is coded are contrasted: position-specific slot-coding, Wickelcoding, open-bigram coding (discrete and continuous), and spatial coding. These theories make different predictions regarding the relative similarity of three different types of pairs of letter strings: substitution neighbors, neighbors-once-removed, and double-substitution neighbors. In Experiment 1, we used an illusory word paradigm and found that neighbor-once-removed similarity contexts resulted in fewer illusory word reports than substitution neighbors but more illusory words than double-substitution neighbors. In Experiments 2 and 3, we used a masked form priming technique with a lexical-decision task. The pattern of facilitation was as predicted by spatial coding but was incompatible with slot-coding, Wickelcoding, and both versions of open-bigram coding. These results provide further support for the SOLAR (self-organizing lexical aquisition and recognition) model of visual word identification.

Keywords: visual word recognition, reading, orthographic input coding, computational models, lexical decision, masked printing, letter migration

A basic question that must be addressed in any theory of visual word identification is how the position of the letters within a word is coded. The fact that readers do code this information is apparent from their ability to distinguish anagrams like *garden* and *danger*. The development of a coding scheme that deals with this issue appropriately is of considerable theoretical importance, not least because the choice of coding scheme greatly affects the performance of computational models of reading. As Plaut, McClelland, Seidenberg, and Patterson (1996) noted, the use of an inappropriate coding scheme prevented the Seidenberg and McClelland (1989) model from learning to generalize adequately, which illustrates how the choice of input and output coding schemes influences the difficulty of the learning process in computational models. Moreover, the manner in which letter strings are coded determines the similarity between different letter strings, which consequently affects a model's ability to explain priming relationships and interactions among lexical competitors. For these reasons, the nature of letter position coding has become a topic of considerable theoretical interest during the last few years (e.g., Davis, 1999, 2006; Davis & Bowers, 2004; Davis & Taft, 2005; Schoonbaert & Grainger, 2004; Grainger, Granier, Farioli, Asche, & van Heuven, in press; Grainger & van Heuven, 2003; Grainger & Whitney, 2004; Perea & Lupker, 2003a, 2003b; Peresotti & Grainger, 1999; Whitney, 2001). In the following section, we review five different schemes that have been proposed for coding letter position.

Slot-Coding

The most common approach to the problem of coding letter position is to assume separate slots of position-specific letter codes, that is, one slot for each possible letter position. For example, the word *cat* would be coded by activating the three letter codes C_1, A_2, and T_3, whereas the word *act* would be coded as A_1, C_2, and T_3 (where the subscript indexes letter position). This type of slot-coding approach is used in the interactive activation model (McClelland & Rumelhart, 1981), the dual-route cascaded model (Coltheart, Rastle, Perry, Langdon, & Ziegler, 2001), the activation-verification model (Paap, Johansen, Chun, & Vonnahme, 2000; Paap, Newsome, McDonald, & Schvaneveldt, 1982), and some parallel-distributed processing models (e.g., Harm & Seidenberg, 1999; Hinton & Shallice, 1991).

Wickelcoding

An alternative to position-specific coding is to code letter order in terms of local context. For example, the *A* in *cat* can be coded by noting that it has a *C* to its left and a *T* to its right. One scheme that relies on this contextual coding approach is *Wickelcoding*, named after Wickelgren (1969), who was an early proponent of the idea of using local context to avoid explicit coding of serial position. Schemes of this sort have been adapted for use in a number of connectionist models of word processing (Rumelhart & McClelland, 1986; Seidenberg & McClelland, 1989). Coding a

Colin J. Davis, Macquarie Centre for Cognitive Science, Macquarie University, Sydney, New South Wales, Australia; Jeffrey S. Bowers, Department of Experimental Psychology, University of Bristol, Bristol, United Kingdom.

This research was supported by a grant from the Australian Research Council awarded to Colin J. Davis and by Grant 7/S17491 from the Biotechnology and Biological Sciences Research Council awarded to Jeffrey S. Bowers and Markus Damian. We thank Derek Hanley for help with data collection and Sachiko Kinoshita for collecting the data at Macquarie University. We are grateful to Jonathan Grainger and Ken Paap for their constructive criticism of a draft of this article.

Correspondence concerning this article should be addressed to Colin J. Davis, who is now at the Department of Experimental Psychology, University of Bristol, 8 Woodland Road, Bristol BS8 1TN, United Kingdom. E-mail: colin.davis@bristol.ac.uk

word using Wickelcoding requires activating a set of units that represent letter triples. For example, the word *stop* would be coded as the set of Wickelfeatures {_st, sto, top, op_}, where _ indicates a word boundary.

Open-Bigram Coding

In open-bigram coding schemes, a letter string is coded in terms of all of the ordered letter pairs that it contains (Grainger & van Heuven, 2003; Grainger & Whitney, 2004; Schoonbaert & Grainger, 2004; Whitney, 2001; Whitney & Berndt, 1999). For example, the word *clam* would be coded by the set {*cl*, *ca*, *cm*, *la*, *lm*, *am*}. Two slightly different versions of open-bigram coding have been proposed. In the version proposed by Grainger and van Heuven (2003; see also Schoonbaert & Grainger, 2004) open-bigram units are associated with discrete patterns of activation. Thus, in the case of *clam*, the units *cl*, *ca*, *cm*, *la*, *lm*, and *am* would each have activities of 1, whereas all other bigram units would be coded by activities of 0. In this version, open-bigram units are only activated for letter pairs that are contiguous or separated by one or two letters (e.g., the unit *cm* is activated by the word *clam*, but not by the word *claim*), although this detail of the coding scheme is not relevant to the present research, because the experiments described here focused on the interior positions of 4- and 5-letter stimuli (and hence critical letter pairs never spanned more than 2 intervening positions).

In the version of open-bigram coding that is used in the SERIOL (sequential encoding regulated by inputs to oscillations within letter units) model (Whitney, 2001; Whitney & Berndt, 1999), bigram units are associated with continuous activities, and these activities are used to code information about letter position and letter contiguity. To distinguish these two versions of open-bigram coding, the following discussion will use the labels *discrete open-bigram* coding and *continuous open-bigram* coding. As we will discuss herein, the two versions of open-bigram coding make some different empirical predictions.

Spatial Coding

A different approach to encoding letter order, called *spatial coding*, has been used in the SOLAR model (Davis, 1999, 2006). In this scheme, all letter units are independent of position context. That is, a node that codes the letter *A* is activated when the input stimulus contains an *A*, irrespective of the serial position in which this letter occurs, or the surrounding context. The relative order of the letters in a letter-string is encoded by the relative pattern of activities across letter nodes. Different letter orderings result in different spatial patterns of activity (hence the term "spatial coding"; note that the word "spatial" does not refer to visuospatial coordinates). This method of coding order has its origins in Grossberg's (1978) use of spatial patterns of node activity to encode temporal input sequences; more recently, similar coding schemes have been used by Page (1994) in a model of melody perception and by Page and Norris (1998) in their primacy model of serial recall. Some examples of spatial coding are shown in Figure 1.

As can be seen in the figure, letter strings that share common letters are coded by relatively similar patterns, even if the common letters are found in different serial positions. In the top example, the word *stop* is encoded by a pattern in which the *S* unit has the largest activity, the *T* unit has the second largest, and so on; thus left-to-right order is coded by a monotonically decreasing sequence of activities. The word *post* (the second example in the figure) is encoded by activating exactly the same set of letter units, but with a very different pattern of activities (i.e., one in which the *P* unit had the largest activity, followed by the *O* unit, and so on). The codes for *stop* and *soap* are quite similar: there is overlap in the letter unit activities for the letters *S*, *O*, and *P*, although the magnitude of activity in the *O* unit differs slightly in the two codes. The codes for *stop* and *slap* are less similar, overlapping for only two letters. The SOLAR model assumes a set of word detectors, each of which computes the match between the word that it codes and the current input stimulus, based on the degree of pattern overlap among the respective spatial codes. A letter that is common to the two codes will therefore contribute to the match computation even if it occurs in different serial positions in the input stimulus and the word coded by the word detector.

Figure 1. Examples of spatial coding for the words STOP, POST, SHOP, SOAP, and SLAP.

It is important to note that the method of spatial coding used in the SOLAR model does not imply that letters in later positions are perceived less well or are assigned less weight in the similarity computation. This is because the mechanism that codes letter position is completely independent of the mechanism for coding how well the individual letters in a word are perceived. Indeed, the basic spatial coding model assumes that all of the letters in a word are coded with equivalent signal strengths and, hence, that each letter position contributes equally to the computation of similarity between the sensory input and previously learned codes (Appendix A describes the details of this similarity computation). Davis

(2006) discusses potential modifications to the model to accommodate the possibility that exterior letters are assigned greater weight in orthographic similarity calculations (e.g., Forster, 1976; Humphreys, Evett, & Quinlan, 1990; McCusker, Gough, & Bias, 1981; Perea, 1998). This possibility is not critical for the present work, which focuses on the similarity of pairs that differ with respect to their interior letters. However, in Experiments 2 and 3, we test the assumption that all interior letters are assigned equal weight in the similarity computation process.

Evidence From Illusory Word Phenomena

In a recent article, we exploited the phenomenon of illusory word perception to investigate letter position coding (Davis & Bowers, 2004). When readers attempt to attend to two-letter strings that are presented simultaneously at different locations, they occasionally perceive an illusory word that combines letters from the two strings (Davis & Bowers, 2004; Davis & Coltheart, 2002; McClelland & Mozer, 1986; Shallice & McGill, 1978). For example, if the pair of words *line* and *love* are presented very briefly and then followed by a pattern mask, readers will occasionally report that one of the words was "live" or "lone." In these examples, a letter apparently "migrates" between words while maintaining its within-word position (e.g., the letter *O* remains in position 2). The apparent tendency of letters to preserve within-word position has led some authors to cite this illusory word phenomenon as evidence in favor of position-specific letter coding (e.g., Ellis, Flude, & Young, 1987; Hinton & Shallice, 1991). For example, Ellis et al. (1987) noted that "letters will only migrate perceptually if they can maintain the same within-word position in the error as in the target word from which they originate" (p. 457) and thus concluded that letter representations are coded for their specific position.

This conclusion was challenged by Davis and Bowers (2004), who suggested that the tendency for letters to preserve serial position in illusory word reports reflects lexical and orthotactic constraints (e.g., a strong preference not to report letter strings that are nonwords or unpronounceable, such as "lvne"), rather than position-specific letter coding. We used a partial report paradigm in which participants are required to report just one of the two words in a briefly presented display; the word that they report is referred to as the *target* word, while the other word is referred to as the *context* word. In the examples given here, the target word is indicated by italicization (e.g., *STEP* SHOP); in the experiments, a poststimulus cue was used to indicate which of the two words was the target. The results of three separate experiments showed that illusory words in which letters migrate to different serial positions *did* occur when lexical and orthotactic constraints were removed. For example, the pair of words *STEP* SOAP led to reports of the illusory word "stop," even though the letter *O* occurs in position 2 in the word *soap* but position 3 in the word *stop*. Furthermore, in one of these experiments we found that letters could also migrate across two-letter positions, as in *ABIDE* ARISE → "aside." Thus illusory word phenomena cannot be considered to provide evidence in support of position-specific letter coding—to the contrary, such phenomena appear to contradict position-specific coding schemes.

Evidence From Masked Form Priming Studies

Much of the evidence concerning the nature of orthographic input coding has come from studies of orthographic similarity effects in the masked form priming paradigm (Evett & Humphreys, 1981; Forster & Davis, 1984). The most common form of masked priming technique is the three-field procedure. In this procedure, a lower-case prime is presented very briefly (typically for around 50 ms), preceded by a forward mask (usually a row of # symbols) and immediately followed by an upper-case target, with all three stimuli appearing in the same location. Participants are unable to report the identity of the prime and are often not even aware of its presence in the display; any impact of the prime is therefore caused by automatic rather than strategic processes. Masked priming effects are considerably greater for word targets than for nonword targets (many studies have failed to find priming effects for nonword targets, although the bulk of the evidence now suggests that a small priming effect can be obtained for nonword targets preceded by identity primes), which supports the conclusion that the locus of masked priming is lexical, rather than sublexical (e.g., Bowers, 2003; Grainger & Jacobs, 1999; but see Masson & Bodner, 2003, for an alternative perspective).

Masked priming experiments have established that preceding a target word with an orthographically similar letter string can result in facilitatory priming of responses to the target, relative to targets that are preceded by unrelated letter strings (e.g., Ferrand & Grainger, 1992, 1993; Forster, Davis, Schoknecht, & Carter, 1987; Forster & Veres, 1998; Perea & Lupker, 2003a, 2003b; Perea & Rosa, 2000; Schoonbaert & Grainger, 2004). In particular, previous studies of masked form priming have demonstrated two distinct forms of orthographic similarity that lead to facilitatory priming. First, many studies have observed facilitatory priming effects when a word target is preceded by a nonword that differs from the target with respect to the substitution of a single letter, for example, *wold-WORD* (e.g., Ferrand & Grainger, 1992, 1993; Forster et al., 1987; Forster & Veres, 1998; Perea & Rosa, 2000), which is consistent with evidence from other paradigms that highlights the orthographic similarity of *substitution neighbors* (*SN*s; e.g., Andrews, 1997; Coltheart et al., 1977; Grainger, O'Regan, Jacobs, & Segui, 1989; Segui & Grainger, 1990). Second, facilitatory priming effects are observed when a word target is preceded by a nonword that differs from the target with respect to the transposition of 2 adjacent letters, for example, *wrod-WORD* (Forster et al., 1987; Perea & Lupker, 2003a; Schoonbaert & Grainger, 2004). This priming effect is significantly greater than that which is obtained when the target word is preceded by a nonword that differs from the target with respect to the substitution of two letters, for example, *wuld-WORD*. The perceptual similarity of *transposition neighbors* (*TN*s) in masked form priming experiments is consistent with evidence from other paradigms (e.g., Andrews, 1996; Chambers, 1979; Davis & Andrews, 2001; Perea & Lupker, 2003b; Taft & van Graan, 1997). The relative perceptual similarity of SNs (e.g., *wold-word*), double-substitution neighbors (*DSN*s; e.g., *wuld-word*), and TNs (e.g., *wrod-word*) provides critical constraints on orthographic input coding schemes.

How Do the Different Coding Schemes Explain the Similarity of TNs?

The aforementioned coding schemes (slot-coding, Wickelcoding, open-bigram coding, and spatial coding) differ with respect to their ability to explain the similarity of TNs. Slot-coding has difficulty explaining the facilitatory effects of primes that are TNS of the target word (e.g., *wrod-WORD*). According to slot-coding, TNs like *word* and *wrod* share only two common letter units (W_1 and D_4), and hence pairs like this are no more similar than DSN pairs like *word* and *weld*, and less similar than SN pairs (e.g., *word* and *ward*). However, Perea and Lupker (2003a) found that target words primed by TNs were classified significantly faster than the same targets primed by DSNs; for example, lexical decision latencies for trials like *uhser-USHER* were 30 ms faster than those for trials like *ufner-USHER*. This contradicts position-specific coding models, which predict that these two conditions should not differ. Furthermore, effects of TN similarity that have been reported in tasks that do not involve priming also pose problems for position-specific coding models (Andrews, 1996; Chambers, 1979; Davis & Andrews, 2001; Taft & van Graan, 1997).

One also has difficulty in using Wickelcoding to account for the perceptual similarity of TNs, because TN pairs like *word* and *wrod* do not share any common Wickelfeatures: the transposition of adjacent letters greatly changes the Wickelfeatures in a word. Thus, the predictions made by slot-coding and Wickelcoding are rather similar, despite their very different approaches to coding letter position.

Open-bigram coding schemes are able to explain the similarity of TNs, because this form of orthographic similarity results in relatively similar bigram codes. For example, transposition neighbors like *word* and *wrod* share five of six open-bigram units {*wo*, *wr*, *wd*, *od*, and *rd*}. By contrast, SNs like *word* and *wold* share only three of six open-bigram units {*wo*, *wd*, and *od*}. Thus, open-bigram coding is able to explain the evidence suggesting that TN similarity is greater than SN similarity (e.g., Andrews, 1996; Chambers, 1979; Forster et al., 1987). This suggests that open-bigram coding is a promising alternative to position-specific slot-coding.

One also can use spatial coding to explain the similarity of TNs, because this form of orthographic similarity results in relatively similar spatial codes: all of the same letter units are active in the two spatial codes, with only slightly different patterns of activity (the relative activities of the transposed letters are reversed). The exact mechanism for quantifying this similarity is described in Appendix A; for present purposes, we simply note that the SOLAR model predicts that TNs are more similar than SNs. Thus spatial coding and open-bigram coding are both able to provide an account of the similarity of TNs, whereas slot-coding and Wickelcoding have difficulty explaining this form of similarity.

Contrasting the Predictions Made by Five Different Coding Schemes

The above review illustrates how studying one particular form of orthographic similarity—transposed neighbor similarity—has helped to distinguish and test the predictions made by different input coding schemes. The resulting data present a strong challenge to the two coding schemes (slot-coding and Wickelcoding) that have been used most often, and in the most influential models of visual word recognition (e.g., Coltheart et al., 2001; Grainger & Jacobs, 1996; McClelland & Rumelhart, 1981; Paap et al., 1982, 2000; Seidenberg & McClelland, 1989). However, transposed neighbor similarity does not help to distinguish the predictions of open-bigram coding and spatial coding, because both types of schemes predict (a) that TNs are highly similar and (b) that TNs are more similar than SNs. A different form of orthographic similarity is required to distinguish between these two approaches to coding letter position.

Fortunately, there is a form of orthographic similarity that is able to differentiate the predictions of open-bigram coding and spatial coding. This distinction is made possible by exploiting the differential importance of letter contiguity in the two approaches. It is important, here, to distinguish between the two versions of open-bigram coding that have been proposed. In the discrete form of open-bigram coding, letter contiguity has no influence on the coding of four-letter strings (it only becomes a factor for letter pairs that are separated by more than two serial positions). In the continuous form of open-bigram coding used in the SERIOL model, letter contiguity has an effect in that bigram units that code contiguous letter pairs are activated more strongly than those that code noncontiguous pairs. As we shall see, this results in a somewhat-counterintuitive prediction. Spatial coding, however, makes a much more obvious prediction: it is sensitive both to letter identity overlap and to relative position overlap, and hence preserving letter contiguity optimizes the match between letter strings.

These abstract differences can be made more concrete by considering the predicted similarity of letter strings in which one of the common letters is displaced by one serial position. An example is the pair *stop* and *soap*—the letter *O* occurs in both of these, but in different serial positions. We will refer to letter strings that have this form of similarity as *neighbors-once-removed*; pairs of stimuli that exhibit this similarity relationship will be referred to as *N1R* pairs. This type of orthographic similarity is ideal for distingushing between slot-coding, discrete open-bigram coding, continuous open-bigram coding, and spatial coding, as each of these schemes makes different predictions concerning the relative similarity of N1R pairs compared with SN pairs and DSN pairs. The similarity of a pair of letter strings x and y can be quantified by computing a normalized match value $M(x, y)$ that lies between 0 and 1, where $M(x, y) = 0$ indicates that there is no overlap between x and y, and $M(x, y) = 1$ indicates a perfect match (i.e., this is the case when x and y are the same letter string). In the following analysis, we determine the match values for each of the 3 forms of orthographic similarity that are predicted by each of the five coding schemes.[1] For consistency, we use the same example stimuli throughout, in which the word *stop* is compared with an SN (*shop*), a neighbor once-removed (*soap*), and a DSN (*snap*).

Calculating the match between same-length stimuli is straightforward for slot-coding: it is just the proportion of shared letter units in the 2 codes. For example, SNs like *stop* and *shop* share three of four letter units, resulting in a match value of $M(stop, shop) = 0.75$, whereas DSNs like *stop* and *snap* share two of four letter units, and hence $M(stop, snap) = 0.5$. The critical comparison, for present purposes, concerns N1R pairs like *stop* and *soap*. According to a coding scheme based on absolute letter position, the *O* in position 2 of *soap* has no relationship to the *O* in position 3 of *stop*, and hence *stop* and *soap* share only two common letter units (S_1 and P_4). Thus, $M(stop, soap) = 2/4 = 0.5$, that is, N1R are no more similar than DSNs. Thus, the ordering of similarity relations predicted by slot-coding is $M(stop, shop) > M(stop,$

[1] A program for computing match values for each of the coding schemes described here can be obtained by contacting the first author.

soap) = *M*(*stop*, *snap*). As we discuss later, similar predictions are derived from variants of slot-coding that are not based on absolute letter position.

The predictions that arise from Wickelcoding are similar to slot-coding with respect to the relative match values that are predicted, although they differ with respect to absolute similarity. As noted already, neighbors like *stop* and *shop* share only a single Wickelfeature (*op#*), so that *M*(*stop*, *shop*) = 1/4 = 0.25. Slightly greater overlap is predicted in the case of neighbors that differ at an exterior letter position (e.g., *stop* and *stow* share two Wickelfeatures), but we do not include this type of neighbor similarity in the experiments reported here. DSN pairs like *stop* and *slap* do not share any common Wickelfeatures, and nor do N1R pairs like *stop* and *soap*. The reason that leftward displacement has this effect is that it removes the contiguity of the letters *O* and *P*, thereby eliminating the *op*# Wickelfeature. It follows, therefore, that the members of an SN pair are slightly similar to each other, but that the members of N1R pairs and DSN pairs are completely dissimilar (i.e., match values of 0 in both cases): The ordering of similarity relations predicted by Wickelcoding is *M*(*stop*, *shop*) > *M*(*stop*, *soap*) = *M*(*stop*, *snap*).

Unlike slot-coding, open-bigram and spatial coding predict that a displaced letter can still give rise to a similar code (i.e., that there is something in common about the letter O in *stop* and *soap*), and this enables these schemes to predict that N1R pairs are more similar than DSN pairs. They differ, however, with respect to their predictions concerning the relative similarity of N1R pairs and SN pairs. According to the discrete form of open-bigram coding, the *so* open-bigram contained in *soap* is indistinguishable from the *so* open-bigrams in *stop* and *silo*: the fact that the *so* letter pair is contiguous in the first example but not in the other two is not taken into consideration. It follows that N1Rs are just as similar to each other as SNs. For example, *stop* and *soap* overlap with respect to exactly the same set of open-bigrams (*so*, *sp*, and *op*) as *stop* and *shop*; in both cases there are three of six shared bigrams, that is, *M*(*stop*, *shop*) = *M*(*stop*, *soap*) = 0.5. DSNs like *stop* and *snap*, on the other hand, share only one open-bigram unit (*sp*), and hence *M*(*stop*, *snap*) = 1/6 = 0.17. Thus the ordering of similarity relations predicted by discrete open-bigram coding is *M*(*stop*, *shop*) = *M*(*stop*, *soap*) > *M*(*stop*, *snap*).

For the continuous version of open-bigram coding used in the SERIOL model, the prediction is slightly more complicated. The level of bigram activation in this coding scheme is determined by both letter contiguity and the serial position of the initial letter of the bigram. The formulae for calculating bigram activity are described in Whitney and Berndt (1999). For contiguous letters, bigram activity is set to 0.6^{pos-1}, where *pos* denotes the position of the initial letter of the bigram. Thus, when coding the word *stop*, the activities of the contiguous bigrams are *st* = 1, *to* = 0.6, and *op* = 0.36. For open-bigrams consisting of noncontiguous letters, the activation is set to 0.6^{pos}. The exception is the open-bigram formed by the initial and final letters, for which the activation is 1.0 – 0.01n, where n is the length of the stimulus. Thus the activities of the noncontiguous bigrams in *stop* are *so* = 0.6, *tp* = 0.36, and *sp* = 0.96. Whitney and Berndt (1999) also describe how to compute the input to word nodes based on the pattern of activity at the bigram nodes: “The weight vector for each word node was set to the bigram activation vector corresponding to that word. The activation of a word node was calculated as the dot product of its weight vector and the input vector” (p. 156). Thus the same formula for setting bigram activities is used to set the weights between bigram nodes and word nodes, for example, the connection weight between the *st* and *stop* nodes is 1, between the *to* and *stop* nodes is 0.6, and so on. The use of dot-product matching of the input and weights is a standard neural network approach.

These specifications for the SERIOL model explain how to compute the match between SNs, N1R, and DSNs. The calculations are shown in Table 1. The bigram activities for each stimulus are multiplied by the corresponding weights (which are equal to the top row of numbers in the table), and then summed. To obtain a match value that is on the same scale as for the other schemes (where a value of 1 indicates a perfect match), the sum of the products has been divided by 3.90. As can be seen, *M*(*shop, stop*) > *M*(*stop*, *snap*), that is, SN pairs are more similar than DSN pairs, and *M*(*stop*, *soap*) > *M*(*stop*, *snap*), that is, N1R pairs are also more similar than DSN pairs. However, SERIOL’s coding scheme also predicts that *M*(*stop*, *soap*) > *M*(*stop*, *shop*), that is, that N1R pairs like *stop* and *soap* are more similar than SN pairs like *stop* and *shop*. The reason for this rather counterintuitive prediction can be understood by considering the one difference between the second and third rows of Table 1: the *so* bigram is coded by an activity of 1.0 in *soap* (because it is an initial contiguous bigram), compared to an activity of 0.6 in *shop* (where it is not contiguous). The presence of a larger activity in the code for *soap* means that it computes a larger match with *stop* than does *shop*, even though the *so* bigram is contiguous in the word *soap* but not in *stop* or *shop*. Note also that the *op* bigram has the same value in the codes for *shop* and *soap*, though for different reasons: it would have a lower activity in *soap*, where this letter pair is noncontiguous, but this is balanced by the fact that the letter *O* occurs earlier in *soap* than in *shop*. Thus the ordering of similarity relations predicted by the model is *M*(*stop*, *soap*) > *M*(*stop*, *shop*) > *M*(*stop*, *snap*).

Intuitively, looking at the examples in Figure 1, and the degree of overlap in the patterns of activities across the *S*, *O* and *P* units, it is not hard to see that spatial coding predicts the ordering relation *M*(*stop*, *shop*) > *M*(*stop*, *soap*) > *M*(*stop*, *snap*). The first part of this inequality obtains because the spatial codes for *stop* and *shop* overlap perfectly for three of the four letters, whereas the overlap between the spatial codes for *stop* and *soap* is not quite as good: although these codes also share three of the four letter units, the letter activity differs between the two codes for one of these three. But this overlap is still greater than that between *stop* and *snap*, which have overlapping patterns for only two of the four letters. This ordering reflects the fact that the matching mechanism used by word detectors in the SOLAR model is sensitive both to the

Table 1
Bigram Activations and Match Values for the Inputs STOP, SHOP, SOAP, and SNAP Given the Assumptions of the SERIOL Model

Similarity	Input	ST	SO	SP	TO	TP	OP	Match
Identity	STOP	1.0	0.6	0.96	0.6	0.36	0.36	1.00
SN	SHOP	0	0.6	0.96	0	0	0.36	0.49
N1R	SOAP	0	1.0	0.96	0	0	0.36	0.57
DSN	SNAP	0	0	0.96	0	0	0	0.32

presence of common letters *and* the relative position of these common letters. The specific match values depend upon the nature of the matching mechanism; the calculations for computing a match in the SOLAR model are described in Appendix A, which shows how the values $M(stop, shop) = 0.75$, $M(stop, soap) = 0.70$, and $M(stop, snap) = 0.5$ are computed.

To summarize, each of the five coding schemes that have been discussed make differing predictions regarding the relative similarity of SNs, once-removed neighbors, and DSNs. The match values predicted by each of the five schemes are depicted in Figure 2.

Testing Letter Position Coding Schemes

The goal of the experiments reported in this work is to test the different predictions regarding the orthographic similarity between letter strings made by the five different letter position coding schemes discussed above, that is, slot-coding, Wickelcoding, the two variants of open-bigram coding, and spatial coding. To perform this model comparison, we used two different experimental paradigms: Experiment 1 used the illusory word paradigm (cf. Davis & Bowers, 2004), whereas Experiments 2 and 3 used a masked priming methodology in conjunction with a standard lexical-decision task (LDT; cf. Forster & Davis, 1984; Grainger et al., 1989).

Experiment 1

In this experiment, we refined a technique based on the phenomenon of illusory word perception that we have previously used to investigate letter position coding (Davis & Bowers, 2004). Previous research has provided strong evidence that illusory word phenomena have a lexical locus. An alternative theoretical explanation was explored by Treisman and Souther (1986), who suggested a feature integration account of illusory word phenomena. According to this account, when attention is overloaded, letters are sometimes identified but not localized, allowing letters to be recombined incorrectly to form illusory words. However, this account is unable to explain the surround-similarity effect: the incidence of illusory word report is greater than chance when the two-letter strings are orthographically similar (e.g., *RAGE* RICE → "race"), but not when they are dissimilar (e.g., *RAGE* LOCK), even though the context LOCK contains an *C* that could migrate to the target to form the illusory word "race" (Davis & Bowers, 2004; McClelland & Mozer, 1986; Shallice & McGill, 1978). This similarity effect is not based on physical similarity: case differences between the target and context words (*rage*-RiCe or *rage*-RICE) do not decrease the likelihood of illusory word report (McClelland & Mozer, 1986; Shallice & McGill, 1978).

These aspects of illusory word report strongly suggest that the locus of the phenomenon is lexical, rather than a prelexical failure of letter localization. In particular, it appears that the tendency to report illusory words stems from the fact that both of the letter strings in the display converge on the representation of the illusory word; for example, given the word pair *RAGE* RICE, both stimuli partially activate the representation of the word *race*. By contrast, in the case of a dissimilar pair like *RAGE* LOCK, only one of the words activates the representation of "*race,*" and hence this illusory word will be reported no more often than for a control display like *RAGE* MONK (i.e., the presence of a *C* in position 3 of the word *lock* has no effect on illusory word report). In passing, we note that this interpretation of the phenomenon underlies our preference for the term *illusory word report*, rather than *letter migration* (cf. Ellis, Flude, & Young, 1987; Hinton & Shallice, 1991; McClelland & Mozer, 1986; Treisman & Souther, 1986).

The aforementioned lexical account suggests that the illusory word phenomenon can be exploited as a tool for measuring the relative perceptual similarity of letter strings. For instance, Davis and Bowers' (2004) finding that the illusory word "stop" was reported more often given the display *STEP* SHOP than the display *STEP* SNAP implies that *shop* is more similar to *stop* than is *snap*, that is, $M(stop, shop) > M(stop, snap)$. This finding is not at all surprising: all of the coding schemes that have been reviewed predict that SN pairs are more similar than DSN pairs. A more important aspect of Davis and Bowers' findings was that the illusory word "stop" was also reported significantly more often given the display *STEP* SOAP than the display *STEP* SNAP. This implies that *soap* is more similar to *stop* than is *snap*, that is, $M(stop, soap) > M(stop, snap)$. In other words, the illusory word phenomenon provides evidence that N1R pairs are more similar than DSN pairs. This result is not predicted by either slot-coding or Wickelcoding, but is correctly predicted by open-bigram coding and spatial coding.

For present purposes, the critical comparison is between the relative similarity of SN pairs and N1R pairs, as this distinguishes spatial coding and the two versions of open-bigram coding. The data of Davis and Bowers (2004) are somewhat indeterminate in this respect. In two of the three experiments, we reported there was a numerical difference between these conditions, such that illusory words were reported more often for displays containing a SN than for displays containing a neighbor once-removed (in the remaining experiment there was no difference at all between these conditions). The numerical difference was greatest in Experiment 3, in which illusory word reports occurred on 15.3% of trials when the context word was a SN (e.g., *SMACK* SHARK → "*shack*"), compared with 11.0% of trials when the context word was a

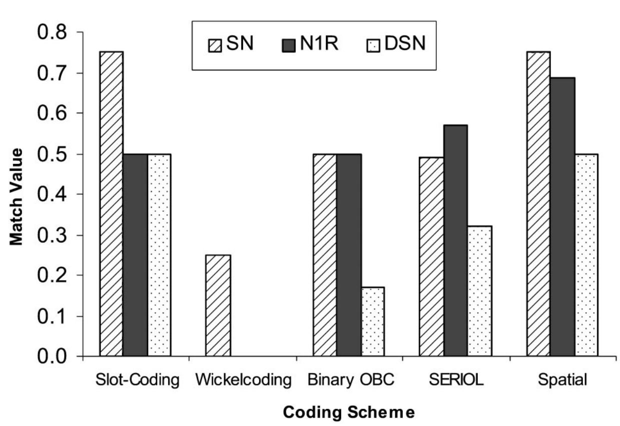

Figure 2. Predictions of the five different coding schemes for the match between substitution neighbors (SN), neighbors-once-removed (N1R), and double-substitution neighbors (DSN). Note: Discrete OBC = Discrete Open-Bigram Coding, SERIOL refers to the Continuous Open-Bigram Coding employed in the SERIOL model, and Spatial refers to the spatial coding scheme used in the SOLAR model.

neighbor once-removed (e.g., *BLOCK* BLEAK → "*black*"). However, the numerical differences in these experiments did not attain statistical significance. One factor that may have contributed to the difficulty in observing a significant difference was that the different conditions used different target stimuli (as well as different potential illusory words), thereby introducing additional variance. The use of different targets across conditions is unavoidable in two-word displays, because of the scarcity of words that possess a SN, a neighbor once-removed, and a DSN, all of which preserve the same exterior letters.

However, it *is* possible to simultaneously match the identity of both the target word and the potential illusory word across conditions if the context stimulus is a nonword rather than a word. For example, the target word *SINK* can be paired with either (a) the nonword context SELK (an SN of the illusory word "silk"), or (b) the nonword context SLEK (an N1R of "silk"), or (c) the nonword context SORK (a DSN of "silk"). Therefore, the use of nonword context stimuli in the present experiment allows us to directly compare 3 critical conditions (SN, N1R, and DSN pairs) using a fixed set of word targets. It is also worth noting that previous research has shown that the lexical status of the context stimulus does not affect the likelihood of illusory word report (McClelland & Mozer, 1986; Treisman & Souther, 1986), a finding that we replicated in a pilot experiment in our own laboratory. Thus, we can expect the illusory word "silk" to be reported just as often for displays like *SINK* SELK as for displays like *SINK* SULK. We hoped that the use of nonword context stimuli would be a methodological improvement over Davis and Bowers (2004), and that this would increase the sensitivity to detect subtle differences in orthographic similarity.

Different Predictions Made by the Five Coding Schemes

The predictions for this experiment are based on the match values derived in the Introduction (Figure 2). To be concrete, we focus on the example in which the target is the word *SINK*, the potential migration response is "silk," and the three possible nonword contexts are the SN SELK, the once-removed neighbor SLEK, and the DSN SORK. Slot-coding predicts that the illusory word SILK will be reported more often for the SN context (SELK) than for the other two contexts, which should not differ (because SLEK and SORK are equally similar to SILK, according to slot-coding). The same prediction follows from Wickelcoding, because the context SELK shares a Wickelfeature with SILK, whereas SLEK and SORK share no features with the illusory word. The version of open-bigram coding proposed by Grainger and colleagues (Grainger & van Heuven, 2003; Schoonbaert & Grainger, 2004) predicts that both the SN and the N1R contexts should produce more illusory word reports than the DSN context and that the former two conditions should not differ (because SELK and SLEK are equally similar to SILK). By contrast, the SERIOL version of open-bigram coding predicts that the N1R context SLEK will result in more illusory word reports than the SN context (because SLEK is more similar to SILK than is SELK), and that both of these conditions will result in more illusory word reports than the DSN context. Finally, spatial coding predicts that neighbor contexts should produce more illusory word reports than N1R contexts (because the similarity of SELK and SILK is greater than that of SLEK and SILK), and that N1R contexts should produce more illusory word reports than DSN contexts (because the similarity of SLEK and SILK is greater than that of SORK and SILK). The goal of Experiment 1 was to test these different predictions.

Method

Participants. Thirty-two undergraduates from Macquarie University participated in the experiment in return for course credit. All participants were native speakers of English and had normal or corrected-to-normal vision.

Stimuli and design. A partial report methodology was used in which participants were shown a pair of letter strings, followed by a cue that indicated which of the 2 letter strings to report. In the following, we refer to the cued stimulus as the target and the noncued stimulus as the context; when describing stimulus pairs, the target stimulus is italicized (e.g., *SEND* SALD), although of course the target was not in italics when presented to the participants. The stimulus pair always consisted of a word and a nonword, either of which was equally likely to be the target (i.e., the identity of the target was not cued by lexical status). There were 60 targets in all: 30 words and 30 nonwords. Target words were of relatively high frequency (between 20 and 300 counts per million in the CELEX database, median frequency = 54 per million) and were mostly from dense orthographic neighborhoods (median $N = 10$); N values and frequency estimates were obtained using the N-Watch software (Davis, 2005). One of the neighbors of each target served as the potential illusory word, for example, the illusory word associated with the target SEND was SAND. Illusory words were also of relatively high frequency (between 20 and 500 counts per million in the CELEX database, median frequency = 65 per million); 14 of the 30 illusory words were of higher frequency than the target word.

Each target word was paired with three different nonword contexts, which varied according to the similarity relationship between the context and the illusory word: (a) an SN context, for example, *SEND* SALD → SAND; (b) an N1R context, for example, *SEND* SLAD → SAND; and (c) a DSN context, for example, *SEND* SLUD. Context stimuli in the first two conditions were selected such that only one of their internal letters could replace an internal letter of the target word to form a legal word; contexts in the third condition were selected such that neither of their internal letters could replace an internal letter of the target word to form a legal word. In the first two conditions, in which the context stimulus contained an internal letter of the illusory word, this letter was the second letter for half of the items and the third letter for the remaining half.

A similar manipulation was performed for the nonword targets, except that different nonword targets were employed for the SN and N1R conditions. Each nonword target was paired with two different word contexts: one which contained an internal letter common to the illusory word (i.e., either an SN context or an N1R context) and one in which neither of the internal letters was shared with the illusory word (i.e., the DSN context). For example, the nonword HONT was paired with the SN context HURT, which allows the illusory word response *hunt*; the DSN context in this case was the word HEAT, for which neither of the internal letters is shared with the illusory word *hunt*. Meanwhile, the nonword TROE was paired with the N1R context TUNE, which allows the illusory word response *true*; the DSN context in this case was the word TAME. Table 2 shows examples

Table 2
Examples of Target and Context Stimuli in Experiment 1

Lexical status of target	Similarity condition	Target	Illusory word	SN context	NIR context	DSN context
Word	~	SEND	SAND	SALD	SLAD	SLUD
Nonword	SN	HONT	HUNT	HURT	—	HEAT
	N1R	TROE	TRUE	—	TUNE	TAME

of the stimuli. The full set of experimental stimuli can be found in Appendix B.

Procedure. Participants were tested in a quiet room in groups of up to four. They were told that they would see two-letter strings (one word and one nonword), followed by a cue that indicated which of these two strings to report. The instructions stressed response accuracy rather than speed. There was a block of 16 practice trials before the experiment proper began. The stimuli for these trials were selected subject to the constraint that their combinations of initial and final letters did not match those of the experimental stimuli.

The sequence of events was as follows. A fixation point appeared on the center of the screen for 1000 ms, followed by a blank screen for 500 ms. A pair of four-letter stimuli was then displayed for 67 ms (this corresponded to four screen refreshes on the testing computers). The stimulus pair was centered on the screen, with the two-letter strings separated by one character width (i.e., the blank space was in the same physical position where the fixation point had previously appeared). The total width of the stimulus pair subtended a visual angle of approximately 3.3 degrees, whereas the height was approximately 0.4 degrees. The letter strings were then replaced by two rows of hash characters (i.e., #### ####). After 200 ms, the mask disappeared and a cue consisting of a horizontal line appeared approximately 1 degree below the position where the target stimulus had previously appeared. The participant then reported the identity of the probed stimulus by typing their response on the computer keyboard. Each participant saw each of the target stimuli twice: once paired with a context that contained an internal letter of the illusory word (either the SN context or the N1R context), and once with a DSN context. For example, half of the subjects saw the target word SEND paired with the SN context SALD, whereas the other half saw this target paired with the N1R context SLAD, but all subjects saw this target paired with the DSN context SLUD. Four separate stimulus lists were prepared to counterbalance the pairing of target and migration contexts and target position (left or right). The order of experimental trials was randomized individually for each participant.

Results

Each response was categorized as either a correct response, an illusory word, or as an "other" error (examples of the latter are *SEND* SLUD → "stud" and SINK SLEK → "sunk"). Two of the 32 subjects had very low levels of reporting accuracy (reporting less than 7% of targets correctly), indicating that they had great difficulty in seeing the stimuli at the very brief exposure durations used in this experiment; subsequently, these two participants were excluded from the analyses. To balance the design (so that we had an equal number of subjects in each of the four stimulus lists) we excluded two other subjects; for this purpose, we selected the two subjects who had the highest accuracy rates in their respective lists (these participants' mean accuracy to report word targets was 92% and 70%, respectively; the exclusion of these participants had no effect on the pattern of results), leaving 28 subjects (i.e., 7 per stimulus list). The overall accuracy across lists was very similar, ranging between 33% and 39%, with an average of 36%.

Not surprisingly, analyses of the response accuracy data showed a large word superiority effect, that is, participants reported word targets much more accurately (M = 53%) than nonword targets (M = 19%), $F1(1, 27) = 389.43$, $F2(1, 56) = 89.78$, both ps $<$ .001. Because of the different way in which the context variables were manipulated, we analyzed word and nonword targets separately.

Word targets. The results for the word targets are shown in Table 3. There were significantly more illusory word reports in the SN condition than in the DSN condition, $F1(1, 27) = 47.43$, $p <$ .001, $F2(1, 29) = 25.76$, $p < .01$. Similarly, the rate of illusory word reports in the N1R condition was higher than that in the DSN condition, $F1(1, 27) = 13.65$, $p < .001$, $F2(1, 29) = 17.75$, $p <$ .01. Critically, the number of illusory word reports in the SN condition significantly exceeded that in the N1R condition, $F1(1, 27) = 7.71$, $p < .01$, $F2(1, 29) = 5.25$, $p < .05$.

Table 3
Percentage of Correct and Illusory Word Responses to Word Targets in Experiment 1 as a Function of Context Condition

	Context Condition		
Response type	SN	N1R	DSN
Correct	46.0	56.0	55.1
Illusory word	12.9	7.4	2.1
Other	41.1	36.6	42.8
Total	**100.0**	**100.0**	**100.0**

Nonword targets. The results for the nonword targets are shown in Table 4. The analysis of nonword targets showed a similar pattern of results to that obtained for word targets. The incidence of illusory word reports for the SN and N1R contexts exceeded that for the DSN contexts, $F1(1, 54) = 38.04$, $p < .001$, $F2(1, 26) = 23.92$, $p < .001$. The difference was significantly higher in the SN condition (M = 11.7%) than in the N1R condition (M = 3.1%), $F1(1, 27) = 13.81$, $p < .001$, $F2(1, 26) = 8.22$, $p <$ .01. Further tests showed that the greater incidence of illusory word reports in the SN condition compared to its DSN control condition was significant in both analyses, $F1(1, 27) = 38.92$, $p <$ .001, $F2(1, 13) = 23.16$, $p < .001$, whereas the greater incidence of illusory word reports in the N1R condition compared to its DSN control condition was significant in the analysis over participants, and bordered on significance in the analysis over items, $F1(1, 27) = 4.23$, $p < .05$, $F2(1, 13) = 2.93$, $p < .06$.

Discussion

The results of this experiment showed clear differences in the frequency of illusory word reports across the three nonword context conditions, supporting one of the five theories of letter position coding that we tested, and providing evidence against the other four. First, the finding of more frequent illusory word reports in the N1R condition than in the DSN condition replicates findings recently reported by Davis and Bowers (2004), and provides further evidence against position-specific letter coding (slot-coding) and Wickelcoding. Second, the finding of a significant difference in the frequency of illusory word reports in the SN and N1R conditions provides evidence against discrete open-bigram coding (Grainger & van Heuven, 2003; Schoonbaert & Grainger, 2004), according to which these two conditions should not differ. Third, the fact that the observed difference was in the direction of more illusory word reports in the SN condition than in the N1R condition provides evidence against continuous open-bigram coding (Whitney, 2001; Whitney & Berndt, 1999), which predicts a difference in the opposite direction. Finally, the observed difference agrees with the prediction made by the spatial coding scheme employed in the SOLAR model (Davis, 1999).

Table 4
Percentage of Correct and Illusory Word Responses to Nonword Targets in Experiment 1 as a Function of Context Condition

Response type	Context Condition			
	SN	DSN control	N1R	DSN control
Correct	16.3	18.7	20.4	22.2
Illusory word	18.6	6.9	8.4	5.4
Other	65.1	74.3	71.2	72.4
Total	**100.0**	**100.0**	**100.0**	**100.0**

An interesting aspect of the results was that the difference between the SN and N1R conditions was present for all three response categories (i.e., correct responses, illusory word reports, and other errors).[2] A plausible outcome of this experiment would have been for differences in illusory word reports to be mirrored in the differences in other incorrect responses, with the percentage of correct reports more or less equivalent across the three context conditions. However, the rate of correct reports was actually much lower for the SN condition (46%) than for the N1R condition (56%); a post hoc test showed that this difference was highly significant, $F1(1,27) = 12.42$, $p < .01$, $F2(1,29) = 6.08$, $p < .05$, indicating that SN contexts interfere with correct identification of the target even when this does not lead to illusory word reports. We interpret this outcome as evidence supporting competitive network models of recognition—in such models, simultaneous activation of both the target and one of its close competitors can result in neither representation exceeding the identification threshold.

The nonword targets showed an identical pattern of results as was observed for the word targets: N1R contexts (e.g., *TROE* TUNE → "true") resulted in more illusory word reports than DSN contexts (e.g., *TROE* TAME → "true"), but fewer illusory word reports than SN contexts (e.g., *HONT* HURT → "hunt"). The difference that was observed for word targets is more compelling, however, because the three context conditions involved exactly the same target words and illusory words, thereby eliminating a potential confound in the interpretation of the results.

In summary, the results of Experiment 1 replicate our previous findings in showing that both SN contexts and N1R contexts produce more illusory word reports than DSN contexts (Davis & Bowers, 2004). More importantly, they also extend these findings by showing significantly more illusory word reports for SN contexts than for N1R contexts. It is probable that the absence of a significant difference between the SN and N1R conditions in our previous work was the result of confounds with characteristics of the target, context, or illusory words (however, this does not invalidate the main conclusions of these experiments, which were not designed to test the comparison between SN and N1R contexts). According to the lexical account of the illusory word phenomenon (which is supported by a body of independent evidence), the likelihood of illusory word report depends on the orthographic similarity of both the target and the context stimulus to the illusory word. By this logic, the present findings should be sufficient to reject the slot coding, Wickelcoding, and open bigram coding schemes, in favor of the spatial coding scheme. Nevertheless, there is some reason to be cautious regarding this conclusion. Data-limited paradigms such as the illusory word paradigm leave open the possibility that participants' responses are influenced by slow inferential processes that are outside the realm of normal word perception. Furthermore, the exact mechanisms underlying performance in this divided attention procedure are not well understood. Although we do not believe that these considerations mitigate our conclusions, it would clearly be highly desirable to obtain converging evidence for the critical difference between the SN and N1R conditions in a conventional reaction time task. This was the goal of the following experiments.

Experiment 2

In this experiment, we used the masked priming LDT paradigm to compare the orthographic similarity of different pairs of letter strings. As noted, previous masked priming experiments have established that preceding a target word with an orthographically similar letter string can result in facilitatory priming of responses to the target, relative to targets that are preceded by unrelated letter strings (e.g., Ferrand & Grainger, 1992, 1993; Forster et al., 1987; Forster & Veres, 1998; Perea & Rosa, 2000). However, two qualifications to this conclusion must be noted. The first qualification is that there is evidence of a *prime lexicality effect* (e.g., Davis & Lupker, in press; Forster & Veres, 1998; Segui & Grainger, 1990): whereas facilitatory priming is observed for nonword primes, inhibitory or null effects often are observed for word primes. In particular, several studies have found that priming a low-frequency target word with an orthographically similar word of higher frequency results in an inhibitory effect (e.g., Davis & Lupker, in press; de Moor & Brysbaert, 2000; Drews & Zwitserlood, 1995; Segui & Grainger, 1990). This finding lends support to competitive network models of visual word recognition, in which orthographically similar words compete with each other during the recognition process (e.g., Davis, 1999; Grainger & Jacobs, 1996; McClelland & Rumelhart, 1981). Because we wished to measure orthographic similarity in terms of facilitatory priming effects, we used primes that were nonwords.

The second qualification is that facilitatory effects of masked form priming appear to depend upon the neighborhood characteristics of the target word. Forster (1987; Forster et al., 1987) found that facilitatory priming effects from nonword neighbor primes were not robust for four-letter words. Further experiments led him to suggest that the failure of facilitatory priming for these targets was a consequence not of length per se, but of the high neighborhood density of (most) short target words; he referred to this as the *density constraint* (e.g., Forster & Taft, 1994; Forster et al., 1987; Perea & Rosa, 2000). Van Heuven, Dijkstra, Grainger, and Schriefers (2001) suggested a refinement of this conclusion, according to which the critical variable is not the overall number of orthographic neighbors of the target, but rather the number of neighbors of the target word that are also neighbors of the prime stimulus. A similar conclusion follows from the results of a masked priming study reported by Hinton, Liversedge, and Underwood (1998), and further evidence of the importance of the shared neighborhood of prime and target has recently been re-

[2] We thank Ken Paap for drawing our attention to this aspect of the results.

ported by Davis and Lupker (in press), in a series of masked priming experiments investigating inhibitory priming from word neighbor primes.

Thus, to maximize our chance of observing strong facilitatory form priming effects, we selected word targets that had relatively few lexical neighbors ($N \leq 3$), and which did not share any neighbors with their primes. Satisfying these criteria necessitated the use of 5-letter stimuli, as it is relatively difficult to find form-related primes and targets that do not share neighbors when four-letter stimuli are used.

Contrasting the Predictions of the Open-Bigram and Spatial Coding Models

There is now an abundance of evidence that challenges slot coding and Wickelcoding (e.g., the results of Experiment 1, as well as Davis & Bowers, 2004; Grainger et al., in press; Perea & Lupker, 2003a, 2003b; Schoonbaert & Grainger, 2004), and we therefore chose to focus our attention in the present experiment on the remaining three coding schemes: discrete open-bigram coding, continuous open-bigram coding, and spatial coding. For this reason, we did not include DSN primes in the present experiment, as all three coding schemes predict that DSN pairs are less similar than either SN or N1R pairs. Instead, we focus on the critical comparison of SN versus N1R pairs. For example, we compared classificiation latencies for the target *SALON* when it was primed by the SN prime *szlon* versus the N1R prime *slzon*.

The three schemes each make different predictions regarding the relative effectiveness of SN primes and N1R primes. The slight increase in the length of the stimuli from Experiment 1 does not (qualitatively) affect the predictions discussed in the Introduction. Discrete open-bigram coding predicts that these two types of prime should have equivalent effects (it is worth noting that, because we once again restricted ourselves to substitutions involving interior letters of a word, the limit on the number of intervening letters in an open-bigram is not relevant). Both SN and N1R primes share five of the nine bigrams in a five-letter target, and hence the match value is equal to 0.56 in both cases; the only bigrams that differ between the coding of *szlon* and *slzon* are *zl* and *lz*, and these have no bearing on similarity to the target *SALON*. Continuous open-bigram coding predicts that the N1R prime *slzon* should have a greater facilitatory priming effect than the SN prime *szlon*. Finally, spatial coding predicts that SN primes should have a greater facilitatory priming effect than N1R primes, because the SN prime is more similar to the target than the N1R prime: $M(salon, szlon) = 4/5 = 0.8 > M(salon, slzon) = 3.77/5 = 0.75$.

Serial Position Effects

A further question that we investigated in this experiment was whether orthographic similarity effects are modulated by the position of the substituted letter. We felt that it was important to address this issue to avoid the possibility that any difference between the SN and N1R conditions was specific to a particular serial position. The question of whether there are interactions between orthographic similarity and serial position for interior letters is a theoretically interesting one because such an interaction is a direct prediction of serial input models (e.g., Whitney's, 2001, SERIOL model). We are aware of only one published study that has systematically examined the effect of the serial position at which SNs differ (Perea, 1998). Perea used a perceptual identification paradigm in which five-letter word targets were primed by either word neighbors (e.g., *bride-bribe*) or unrelated words. He found that neighbor primes had an inhibitory effect on target word identification when the prime and target differed at the third or fourth position, but not when they differed at the first, second, or fifth positions. The contrast between prime-target pairs that differ with respect to an exterior, as opposed to an interior letter is consistent with evidence from other priming studies suggesting that exterior letter overlap is particularly important (e.g., Forster, 1976; Humphreys et al., 1990; McCusker et al., 1981). That is, neighbor pairs that differ with respect to an exterior letter (e.g., *lemon-demon*, *grass-grasp*) may be less similar than pairs that differ with respect to an interior letter (e.g., *bride-bribe*).

However, the exterior–interior letter distinction does not explain why Perea (1998) did not observe priming for pairs like *start-smart*, which differ at the second position: This suggests that some other factor could affect the interaction of orthographic similarity and serial position for interior positions. A serial, left-to-right process is one candidate. For example, Whitney's (2001) SERIOL model predicts a match value of 0.79 for pairs like *bride* and *bribe*, compared to a match value of only 0.54 for pairs like *start* and *smart*. An alternative possibility is that the second position of five-letter words is particularly salient, because it corresponds to the optimal viewing position or preferred viewing location (both of which are slightly to the left of the center of a word; e.g., O'Regan & Jacobs, 1992; Rayner, 1979); a difference at this position may be more noticeable, leading pairs like *start-smart* to be perceived as less similar than pairs like *bride-bribe*. Before investigating such explanations in detail, though, it is necessary to establish the robustness of the phenomenon. As we note in the discussion, the inhibitory priming procedure used by Perea may not be the optimal technique for investigating serial position effects. Experiment 2 used a facilitatory priming paradigm, which we will argue is more suited to examining interactions between serial position and orthographic similarity.

In Experiment 2, half of the SN primes were created by substituting letters at the second position (as in the *szlon* example); we refer to these as SN_2 primes. The remaining SN primes were created by substituting letters at the fourth position (e.g., *salzn*); we refer to these as SN_4 primes. According to the continuous open-bigram coding scheme used in the SERIOL model, early positions carry greater weight (e.g., the bigram formed by the letters in positions 1 and 2 is coded by a greater activity than the bigram formed by the letters in positions 3 and 4). Therefore, this model predicts that replacing the second letter of the target should result in a less similar prime than replacing the fourth letter (i.e., SN_2 primes should be less effective than SN_4 primes); the predicted match values are 0.54 and 0.79, respectively. By contrast, the other four coding schemes reviewed in the introduction predict no difference between SN_2 and SN_4 primes. We also included a similar manipulation for the N1R primes: half of these were formed by transposing the second and third letters of a corresponding SN_2 prime (e.g., transposing the *Z* and the *L* in the SN_2 prime *szlon* produces the $N1R_2$ prime *slzon*), and the remaining half were formed by transposing the third and fourth letters of a corresponding SN_4 prime (e.g., transposing the *Z* and the *L* in the SN_2 prime *salzn* produces the $N1R_4$ prime *sazln*). The SERIOL model predicts that $N1R_4$ primes should be more effective than $N1R_2$ primes. The other models, by contrast, predict no difference. Indeed, most models implicitly adopt a *symmetry premise* (Davis, 2006), according to which displacement of a letter has equivalent effects whichever direction it is shifted in, that is, backward, as in $N1R_2$ primes, or forward, as in $N1R_4$ primes.

Method

Participants. Thirty-five undergraduates from the University of Bristol participated in the experiment in return for course credit. All participants were native speakers of English and had normal or corrected-to-normal vision.

Stimuli and design. The experiment consisted of 120 word targets and 120 nonword targets. Each target was paired with five different nonword primes: (a) an SN_2 prime, which was a neighbor of the target that differed at position 2 (e.g., *pxlar-POLAR*), (b) an SN_4 prime, which was a neighbor of the target that differed at position 4 (e.g., *polxr-POLAR*), (c) an $N1R_2$ prime, which was a N1R of the target formed by transposing the second and third letters of the corresponding SN_2 prime (e.g., *plxar-POLAR*), (d) an $N1R_4$ prime, which was a N1R of the target formed by transposing the third and fourth letters of the corresponding SN_4 prime (e.g., *poxlr-POLAR*), and (e) an unrelated prime (e.g., *gxief-POLAR*). Five different counterbalanced versions of the experiment were designed, so that each participant saw a given target only once, paired with one of its five primes.

The stimuli were selected as follows. First, we selected a set of five-letter words with CELEX written frequencies of between 2 and 30 counts per million and no more than three neighbors. We excluded plurals, past tense forms, proper nouns, and words that seemed relatively unfamiliar, as well as words that included repeated letters (because letter repetition could interfere with the examination of letter position effects). We then wrote a computer program that determined the set of all possible primes for each of the four related prime conditions. Thus, for each target, a set of primes of the form 1*d*345, 123*d*5, 13*d*45, and 124*d*5 was computed, where the string 12345 refers to the letters of the target, and *d* is a letter not contained in the target. The program excluded primes that were themselves words, or primes that had neighbors (or N1R, or TN) other than the target. Words for which no possible primes could be found were excluded from being potential targets; this resulted in a set of 120 potential targets. The program then computed the (length and position-specific) summed log bigram frequency (SLBF) for each possible prime. Finally, for each target it selected a single substitution letter that minimized the difference in SLBF between the SN and N1R conditions (e.g., for the target *POLAR*, the substitution letter that the program selected was *x*, resulting in the set of primes *pxlar*, *polxr*, *plxar*, and *poxlr*). This method of selecting primes was designed to try to ensure that the SN and N1R primes were closely matched with respect to their wordlikeness. The goal of matching with respect to SLBF was satisfied: the mean SLBF was 5.4 for both SN_2 primes and $N1R_2$ primes, and 5.7 for both SN_4 primes and $N1R_4$ primes. This method of selecting primes also avoided the possibility of unconscious biases in stimulus selection (Forster, 2000). Note that SN primes had exactly one SN (the target) and no N1R (or TNs), whereas N1R primes had exactly one N1R (the target) and no SNs (or TNs). There were a small number of cases in which the prime had a deletion neighbor (e.g., Davis & Taft, 2005), usually of low frequency. For example, the SN_2 prime selected for the target *ANKLE* was *axkle*, which has the deletion neighbor *axle*. In most cases when this happened, the letters of the deletion neighbor were non-contiguous within the target, and the same deletion neighbor was possessed by one of the SN conditions and one of the N1R conditions (e.g., the N1R-prime for *ANKLE* was *akxle*, which also has the DN *axle*). However, there were five cases in which only one of the prime conditions had a DN (e.g., the SN_2 prime that was originally selected for the target *VAULT* was *vcult*, which has the DN *cult*; by contrast, *cult* is not a DN of the other primes selected for this target, i.e., *vauct*, *vuclt*, and *vacut*). For these cases, we selected the substitution letter that ranked second for minimizing the SLBF difference (e.g., the primes for *VAULT* became *vkult* etc.). The same procedure was used to replace one case in which the SN_4 prime initially selected (*hinje*) was a pseudohomophone of the target (*HINGE*). Unrelated primes were chosen by pseudorandomly pairing primes from the related conditions (30 from each of the four related conditions) with targets, such that the resulting prime-target combinations shared at most one letter; when there was a common letter, it occupied a different position in the target and its unrelated prime.

Each of the nonword targets was selected by changing the third letter of a five-letter large-N ($N = 5$) word. The primes for the nonword targets were constructed in the same way as for the word targets, except that the constraints on neighbors were relaxed (i.e., primes were allowed to have neighbors other than the target). The full set of stimuli for this experiment can be found in Appendix B.

Procedure. Participants were tested in a quiet room either individually or in groups of two or three. They were told that words and nonwords would be displayed on the monitor in front of them, and that they should press one of two buttons to indicate whether each stimulus was a word or a nonword. Word responses were made with the participant's right hand, and they were instructed to respond as rapidly as possible while maintaining a reasonable level of accuracy. Participants were initially presented with 16 sample trials consisting of eight words and eight nonwords. The experiment proper consisted of 240 experimental trials, the order of which was randomized for each participant. Stimuli were presented using the DMDX software for stimulus display (Forster & Forster, 2003), on Windows PCs with a refresh rate calibrated to 13.3 ms. A standard masked priming methodology was followed: each trial consisted of a mask stimulus (#####) that was displayed for 500 ms, followed by a lower case prime stimulus that was displayed for 67 ms (i.e., 5 screen refreshes), followed by an upper case target which remained visible until the subject responded.

The prime and target were displayed using different font sizes: primes were displayed in 12-point Courier New, whereas targets were displayed in 16-point Courier New. The aim of this was to ensure that any advantage for SN primes relative to N1R primes was not due to visual overlap in cases where the lower and upper case versions of the same letter was similar.[3]

Results and Discussion

Latencies greater than 1,500 ms (0.4% of the data) or less than 300 ms (0%) were excluded from the analysis of reaction times. Five word targets (*bison*, *brute*, *dogma*, *pluck*, and *voter*) were classified as nonwords by more than 25% of the participants and were therefore excluded from the analyses. Mean reaction times (RTs) and error rates across conditions are shown in Table 5.

Mean RT for the SN prime conditions was 11 ms faster than for the N1R prime conditions. We analyzed this difference using a one-tailed test, in which the null hypothesis was that there was no difference and the alternative hypothesis was that SN primes were more effective than N1R primes. The difference was significant in both the participant and item analyses, $F1(1, 34) = 3.92$, $p < .05$; $F2(1, 114) = 5.43$, $p < .05$. (Although a one-tailed test is the more appropriate analysis, we note in passing that, in a two-tailed test, the difference by items is still significant and the difference by participants is marginally significant, with $p = .056$). Thus the observed difference agrees with the predictions of the SOLAR model, but disagrees with the prediction of open-bigram coding.

There was no effect of the position of the replaced letter: the mean latency was 594 ms when the letter in position 2 was

[3] We thank Jonathan Grainger for suggesting this method of presentation.

Table 5
Mean Reaction Times and Error Rates Across Prime Conditions in Experiment 2

Prime condition	Example	RT (ms)	ER (%)	Priming effect (ms)
SN_2	axkle-ANKLE	592	6.6	25
SN_4	ankxe-ANKLE	587	5.5	30
$N1R_2$	akxle-ANKLE	596	5.7	20
$N1R_4$	anxke-ANKLE	603	6.0	14
Unrelated	wgzon-ANKLE	617	6.3	

Note. SN_2 = Substitution Neighbor formed by making a substitution at position 2; SN_4 = Substitution Neighbor formed by making a substitution at position 4; $N1R_2$ = Neighbor-Once-Removed formed by making a substitution at position 2 and then transposing the second and third letters; $N1R_4$ = Neighbor-Once-Removed formed by making a substitution at position 4 and then transposing the third and fourth letters.

replaced and 595 ms when the letter in position 4 was replaced (both Fs < 1). Nor was there any interaction between type of orthographic similarity and replacement position, $F1(1, 34) = 1.23$, $p > .05$; $F2(1, 114) = 1.58$, $p > .05$. This agrees with the predictions of the SOLAR model and the discrete open-bigram model, but disagrees with the SERIOL model, which predicts that replacing the letter in position 4 of the target should result in a much more similar prime than replacing the letter in position 2.

Pairwise comparisons using Dunnett's test showed that the unrelated prime condition produced significantly longer RTs than the SN_2 condition, $t1(34) = 3.07$, $p < .05$; $t2(114) = 3.50$, $p < .01$; the SN_4 condition, $t1(34) = 4.75$, $p < .01$; $t2(114) = 4.50$, $p < .01$; and the $N1R_2$ condition, $t1(34) = 2.63$, $p < .05$; $t2(114) = 2.94$, $p < .05$. The difference between the $N1R_4$ condition and the unrelated condition did not attain significance, $t1(34) = 1.80$, $p > .05$; $t2(114) = 1.80$, $p > .05$. An analysis of error rates showed no effect of type of orthographic similarity, no effect of position, and no interaction between these factors (all Fs < 1). The mean correct RT for nonword targets was 667 ms, and the mean error rate was 8.8%.[4]

In summary, the results of Experiment 2 show a clear difference between SN primes and N1R primes, enabling us to reject the null hypothesis of no difference between these conditions.[5] The greater effectiveness of SN primes is consistent with the prediction of the spatial coding scheme used in the SOLAR model, but is inconsistent with open-bigram coding schemes. Note also that the difference between SN and N1R primes cannot be attributed to low-level visual factors (i.e., a greater physical overlap between the prime and the target in the case of SN primes) because the use of different font sizes for primes and targets eliminated any close physical overlap between the corresponding letters of the prime and the target.

Another interesting aspect of the results was the absence of any serial position effect. There was only a 1-ms difference between the RTs for primes formed by replacing the second letter of a word target and primes formed by replacing the fourth letter of the target. When only SN primes are considered, the difference increases to 5 ms (in the direction of greater priming for SN_4 primes), but is nevertheless far from significant ($p > .4$). This aspect of our results differs from the findings of Perea (1998), who observed (inhibitory) priming for SN_4 primes (e.g., *bride-bribe*), but not SN_2 primes (*start-smart*). There are a number of relevant differences between the experiments. In our experiment, the word targets were the same for both position conditions (thereby ruling out the possibility that the position factor was confounded with some aspect of the target words), whereas the methodology of Perea's experiment made it necessary to use different targets for each position condition. Furthermore, the primes in our experiment were nonwords that had no neighbors (or N1R) other than the target, ruling out the possibility that the manipulation of position was confounded with prime frequency or neighborhood properties of the prime. The latter possibilities do affect the interpretation of Perea's experiment, and masked inhibitory priming can be affected by both relative prime-target frequency and the shared neighborhood of the prime and target (e.g., Davis, 2003; Davis & Lupker, in press). Recent evidence from other masked priming experiments also has failed to find evidence for serial position effects (Grainger et al., in press). The null effect of serial position is consistent with the SOLAR model, which assigns equal weight to all interior positions (Davis, 2006), and also with parallel input models like the dual-route cascaded model and M-ROM. However, it is inconsistent with left-to-right serial input models, like SERIOL, which predict that primes formed by replacing the fourth letter of a word target should be considerably more effective than primes formed by replacing the second letter of the target (because the fourth letter carries less weight than the second letter).

Experiment 3

In Experiment 3, we sought to replicate the findings of Experiment 2 and also to establish that the difference between SN and N1R primes is the result of orthographic rather than phonological processes. A number of experiments have investigated the relative time course of orthographic and phonological processing by comparing orthographically similar pseudohomophone primes (e.g., *mayd-MADE*), orthographically similar nonpseudohomophone primes (e.g., *mard-MADE*), and unrelated controls (e.g., *filb-MADE*; Ferrand & Grainger, 1992, 1993; Perfetti & Bell, 1991). Ferrand and Grainger (1992) found orthographic facilitation at a prime duration of 32 ms (i.e., orthographically related primes resulted in faster RTs than unrelated primes) but no phonological facilitation (i.e., the magnitude of priming was identical for orthographically similar pseudohomophonic and nonpseudohomophonic primes). Perfetti and Bell (1991) observed the same pattern at a prime duration of 35 ms, with phonological effects only starting to emerge at 45 ms. Likewise, Brysbaert (2001) observed phonological priming with a prime duration of 43 ms but not with a

[4] The design of the primes was the same for the nonword targets as for the words, simply so that the relationships between prime and target were approximately matched for word and nonword targets, ensuring that this was not a cue to the lexical status of the target. However, the related prime conditions were not matched with respect to orthographic variables such as N or bigram frequency (the SN primes had systematically more neighbors than the N1R primes), and so analyses of the orthographic similarity and position factors would not be meaningful for the nonword targets.

[5] It is perhaps worth noting that Experiment 2 showed only a 4-ms difference between the SN2 and N1R2 conditions. We return to this comparison in Experiment 3.

duration of 29 ms. A recent set of masked priming experiments report by Grainger et al. (in press) showed that (when number of shared letters was controlled), there was a small-but-significant correlation between the degree of phonological overlap between prime and target and the magnitude of form priming at a prime duration of 50 ms (r = .17, p < .01), but not at a prime duration of 33 ms (r = .02). In particular, Grainger et al. concluded that when the shorter prime duration is used, with no forward mask, form priming effects are robust, and reflect orthographic rather than phonological processing. Thus, a variety of masked priming experiments, using English, French, and Dutch stimuli, have all converged on a prime duration of around 33 ms as one at which orthographic effects should be present, whereas phonological effects will not yet have emerged (e.g., Ferrand & Grainger, 1992, 1993; Grainger et al., in press; Perfetti & Bell, 1991). For this reason, Experiment 3 used exactly the same materials as Experiment 2, but with the methodology of Grainger et al. (in press, Experiment 5), using a prime duration of 33 ms. We therefore expected that any difference between the SN and N1R conditions would reflect orthographic, rather than phonological processing.

Method

Participants. Thirty participants were drawn from the same population as in Experiment 2. They were either paid for their participation or received course credit for participating.

Stimuli and design. The stimuli used in this experiment were identical to those used in Experiment 2.

Procedure. The procedure for this experiment was identical to that of Experiment 2, except that the forward mask was eliminated and the prime duration was decreased to 33 ms (corresponding to 4 refreshes on computers that had been calibrated to have screen refresh rates of exactly 8.32 ms). Thus, the display sequence consisted of a blank screen for 500 ms, followed by a fixation point for 500 ms, followed by a blank screen for 500 ms, and then the prime (for 33 ms, in lower-case 12 point Courier New font), followed by the target (in upper-case 16 point Courier New font), which remained visible until the participant responded. We checked that the prime was not visible when this display procedure was used; when asked, none of the participants reported being aware of a stimulus preceding the target.

Results and Discussion

Latencies greater than 1,500 ms (0.6% of the data) or less than 300 ms (0%) were excluded from the analysis of reaction times. One participant exhibited a strong bias to respond "yes" (resulting in an error rate of 30% for nonword stimuli), as well as a large number of slow RTs (11.2% of this participant's correct responses were excluded as slow outliers), and we therefore excluded this participant from the analyses (this didn't affect the pattern of significant results). Five word targets (*bison*, *brute*, *pluck*, *voter*, and *ratio*) were classified as nonwords by more than 25% of the participants, and were therefore excluded from the analyses (four of these were also excluded in Experiment 2). Mean RTs and error rates across conditions are shown in Table 6.

Latencies for the SN prime conditions were 14 ms faster than latencies for the N1R prime conditions. An analysis of variance performed on mean correct RTs revealed a significant main effect of type of orthographic similarity, $F1(1, 24) = 9.22$, $p < .005$; $F2(1, 110) = 8.93$, $p < .005$. It is also worth noting that the difference between the SN2 and N1R2 prime conditions was relatively large in this experiment (18 ms), suggesting that the small difference in Experiment 2 masked a true difference between these two conditions.

Table 6
Mean Reaction Times and Error Rates Across Prime Conditions in Experiment 3

Prime condition	Example	RT (ms)	ER	Priming effect (ms)
SN_2	axkle-ANKLE	575	2.7	34
SN_4	ankxe-ANKLE	577	3.9	32
$N1R_2$	akxle-ANKLE	593	4.5	16
$N1R_4$	anxke-ANKLE	587	4.3	22
Unrelated	wgzon-ANKLE	609	5.2	

Note. SN_2 = Substitution Neighbor formed by making a substitution at position 2; SN_4 = Substitution Neighbor formed by making a substitution at position 4; $N1R_2$ = Neighbor-Once-Removed formed by making a substitution at position 2 and then transposing the second and third letters; $N1R_4$ = Neighbor-Once-Removed formed by making a substitution at position 4 and then transposing the third and fourth letters.

RTs were once again unaffected by the position of the replaced letter: the mean latency was 584 ms when the letter in position 2 was replaced and 582 ms when the letter in position 4 was replaced. Thus, there was no effect of position of the changed letter, $F1(1, 24) = 0.17$; $F2(1, 110) = 0.12$, nor was there an interaction of prime type and letter position, $F1(1, 24) = 0.06$; $F2(1, 110) = 0.26$. Pairwise comparisons using Dunnett's test showed that the unrelated prime condition produced significantly longer RTs than the SN conditions: for the SN_2 condition, $t1(24) = 4.30$, $p < .01$; $t2(110) = 3.94$, $p < .01$; for the SN_4 condition, $t1(24) = 4.73$, $p < .01$; $t2(110) = 4.49$, $p < .01$. The differences between the N1R conditions and the unrelated condition were not quite as robust: for the $N1R_2$ condition, $t1(24) = 2.80$, $p < .05$; $t2(110) = 1.79$, $p > .05$; for the $N1R_4$ condition, $t1(24) = 3.18$, $p < .05$; $t2(110) = 2.78$, $p < .05$.

The mean error rate for the SN conditions was 3.3%, compared with a mean of 4.4% for the N1R conditions. This difference approached significance in a one-tailed test, $F1(1, 24) = 1.84$, $p <$ .10; $F2(1, 110) = 2.11$, $p < .08$. Error rates were unaffected by the position of the substituted letter, nor was there any interaction between type of similarity and position (all Fs < 1). One-tailed pairwise comparisons using Dunnett's test showed that there were significantly fewer errors for the SN2 condition than for the unrelated prime condition, $t1(24) = 2.33$, $p < .05$; $t2(110) = 2.79$, $p < .05$; there were no other significant differences in the error rates between conditions. The mean correct RT for nonword targets was 662 ms, and the mean error rate was 6.6%.

In summary, the results of Experiment 3 replicate those of Experiment 2. Word targets preceded by SN primes showed significantly greater facilitatory priming than the same targets that were preceded by N1R primes. This agrees with the prediction of the spatial coding model. Because the present findings were obtained with a relatively short prime duration (33 ms), the observed difference is very unlikely to reflect phonological processes (cf. Brysbaert, 2001; Ferrand & Grainger, 1992, 1993; Grainger et al., in press; Perfetti & Bell, 1991).

The other respect in which this experiment replicates the previous experiment is in the absence of any effect of the position of the substituted letter. This agrees with models that assign equal weight to each of the interior letters, such as the SOLAR model, but is contrary to the prediction of the SERIOL model, which assigns greater weight to earlier interior letters.

Experiments 2 and 3 demonstrate that the masked priming paradigm, given a sufficiently powerful design, provides strong support for the claim that SNs are more similar than N1R. This finding may not seem especially surprising—indeed, we expect that it would accord with most readers' intuitions. Nevertheless, of the five coding schemes reviewed in the Introduction, only the spatial coding predicts this outcome.

General Discussion

The present experiments were designed to test between five different theories of letter position coding. The results provide support for one of these theories, and evidence against the other four. Experiment 1 obtained evidence that is inconsistent with slot-coding and Wickelcoding. In their standard form, these schemes predict that N1R pairs like *stop* and *soap* are no more similar to each other than DSN pairs like *stop* and *snap* (indeed, Wickelcoding predicts that N1R pairs are not at all similar, e.g., that *stop* and *soap* are no more similar to each other than *stop* and *beer*). The results of Experiment 1 disconfirm this prediction: Participants were significantly more likely to report an illusory word if the display included a context stimulus that was an N1R of that illusory word than if the context was a DSN of the illusory word. Thus, our data contradict the two coding schemes that have been used most often in computational models of reading. This replicates our previous findings (Davis & Bowers, 2004), and adds to evidence derived from other paradigms, using other forms of orthographic similarity, such as the similarity of TNs (Andrews, 1996; Perea & Lupker, 2003a, 2003b; Schoonbaert & Grainger, 2004; Taft & van Graan, 1998), and addition/deletion neighbors (Davis & Taft, 2005; de Moor & Brysbaert, 2000; Schoonbaert & Grainger, 2004).

The major theoretical contribution of the present work is its empirical comparison of three coding schemes that have been proposed as alternatives to slot-coding and Wickelcoding: continuous open-bigram coding (Whitney, 2001), discrete open-bigram coding (Grainger & van Heuven, 2003), and spatial coding (Davis, 1999). This comparison depends on the different predictions made by these three schemes regarding the relative similarity of SNs and N1R. The continuous open-bigram coding used in the SERIOL model makes the somewhat counterintuitive prediction that SNs like *stop* and *shop* are less similar than N1R like *stop* and *soap*. All three of our experiments falsified this prediction: Experiment 1, using the illusory word paradigm, showed a significant difference in the opposite direction; likewise, Experiments 2 and 3, using a masked priming LDT paradigm, both showed significant differences in the opposite direction. These findings are also inconsistent with discrete open-bigram coding, which predicts that SNs and N1R should be equally similar. By contrast, the results of all three experiments are consistent with the spatial coding scheme employed in the SOLAR model (Davis, 1999, 2006).

Could a Variant of Slot-Coding Explain the Present Data?

Most of the computational models of reading that employ slot-coding have assumed a simple form of slot-coding in which letter units are based on absolute position (Coltheart et al., 2001; Grainger & Jacobs, 1996; McClelland & Rumelhart, 1981). Our test of slot-coding in the present experiments was based on the match values that are predicted by this type of scheme. It is appropriate, though, to consider whether variations on this scheme could do a better job of accounting for the data. A number of simple variations to straightforward slot-coding have been suggested in which letters are assigned to units on the basis of their position relative to other letters, rather than their absolute position. However, none of these are able to accommodate the present data. Jacobs, Rey, Ziegler, and Grainger (1998) proposed a coding scheme in which the exterior letters of the stimulus are always coded in the outer slots (slots one and eight), and other letters are positioned relative to these; for example, a four-letter word would be coded by activating units in slots one, two, seven, and eight. This implies that N1R pairs like *silk* and *slek* have corresponding letters in slots one and eight, but different letters in slots two and seven, and hence the match value is exactly the same as for a slot-coding scheme based on absolute letter position.

Another variant of slot-coding, suggested by Harm and Seidenberg (1999; cf. Zorzi et al., 1998), assumes that letter slots are vowel-centered. The first vowel is always assigned to slot four, and other letters are arranged relative to this letter. For example, the word *silk* would be coded by the units S_3, I_4, L_5, and K_6, whereas the nonword *slek* would be coded by the units S_2, L_3, E_4, and K_5. Thus *silk* and *slek* do not share any common features according to this scheme, that is, $M(silk, slek) = 0$. However, a DSN pair like *silk* and *sork* share two common features (S_3 and K_6), that is, $M(silk, sork) = 0.5$. This leads to an incorrect prediction about the direction of the difference between the N1R and DSN conditions in Experiment 1. Like the absolute-position form of slot-coding, then, vowel-centered coding is unable to explain our data.

Plaut et al. (1996) proposed a different form of slot-coding that partitions syllables into onset, vowel, and coda slots. For example, the word *blind* would be coded by activating the *B* and *L* onset units, the *I* vowel unit, and the *N* and *D* coda units. When there are multiple letters activated in a single subsyllabic slot, the relative order of these letters is determined by graphotactic constraints on the structure of English orthography: There are no English words that begin with the letter sequence *lb*, and so the coactivation of *B* and *L* in the onset must indicate an initial *bl* cluster. The model's knowledge of graphotactic constraints is encoded via a left-to-right ordering of graphemes within each slot (e.g., within the onset slot the grapheme *B* is listed before the grapheme *L*). This scheme is unable to accommodate the illusory word phenomena observed in Experiment 1. For example, the *L* in *silk* is coded by a different unit (the *L* unit in the coda slot) than the *L* in *slek* (the *L* unit in the onset slot). Thus the codes for N1R pairs like *silk* and *slek* are no more similar than the codes for DSN pairs like *silk* and *sork* (in both cases the common units are the *S* in the onset slot and the *K* in the coda slot). Like the simpler slot-coding scheme, this fails to explain the difference between the N1R and DSN conditions in our data.

A more general problem with this method of coding letter position is that it cannot code position veridically, because the order of letters within a slot cannot be adequately represented by relying on graphotactic constraints. For example, the nonword *lbidn* would be coded by activating the *B* and *L* onset units, the *I* vowel unit, and the *N* and *D* coda units: that is, this letter string is coded in exactly the same way as the word *blind*. Indeed, the model proposed by Plaut et al. (1996) would pronounce *lbidn* as "blind." It is perhaps not surprising, then, that subsequent parallel-distributed processing models of reading have used position-specific letter units, in which letter order is coded unambiguously (e.g., Harm & Seidenberg, 1999).

A similar problem affects the "two-slot" scheme discussed by Shillcock and colleagues (Shillcock, Ellison, & Monaghan, 2000; Shillcock & Monaghan, 2001). The basis of this approach is that the orthographic input code is split into two distinct hemifields, and that this split enables words to be recognized satisfactorily even if the order of the letters in each hemifield is not coded. For example, Shillcock and Monaghan (2001) note that, "if the left hemifeld contains *a*, *c*, and *r* and the right hemifeld contains *e*, *p*, and *t*, then the word must be *carpet*" (p. 1194). Shillcock et al. (2000) argue that "existing models of visual word recognition lose letter-hemifield information in requiring the process to begin with a single representation of the whole word and must retrieve the information by imposing order on the letters. . . . [We have] shown the informativeness of simply being able to specify the position of each letter relative only to a fixation point somewhere near the middle of the word" (p. 841). As in the case of Plaut et al.'s (1996) scheme, the problem with this logic is that there is no guarantee that orthographic inputs will be restricted to familiar words. In the above example, the stimulus could be *carpet*, but it could equally well be *crapte*; the two-slot model has no means of telling the difference. This is a critical flaw in this model, not only because it prevents veridical coding of letter order in nonwords, but because it prevents learning of new words. For example, how could the word *item* be learned if the word recognition system has previously learned that, "if the left hemifeld contains *i* and *t* and the right hemifeld contains *e* and *m*, then the word must be *time*"?

A different implementation of the split model was described by Shillcock and Monaghan (2001). This model has eight slots (four slots for each hemifield). The authors trained the model on a set of 60 4-letter words, with each word presented in each of five possible positions (i.e., with the first letter in slot 1, 2, 3, 4, or 5). The goal of the model was to learn to map inputs that were split across two hemifields onto a four-slot output code that is independent of visual field. That is, the structure of the model's output is equivalent to the letter-slot input coding of the original interactive activation model. The present data and other data reviewed in the Introduction argue strongly against this approach.

"Sloppy" Slot-Coding

The introduction of position uncertainty to slot-coding may help it to overcome some of its problems. For example, the stimulus *slek* might result in the activation of not only the *L* unit in position 2 but also the partial activation of the *L* units in positions 1 and 3. This would increase the match between *silk* and *slek*, enabling this model to predict that $M(silk,\ slek) > M(silk,\ sork)$. A coding scheme along these lines has recently been discussed by Perea, Gomez, and Ratcliff (2003). It is worth noting that most of the models that have used slot-coding (e.g., Coltheart et al., 2001; Grainger & Jacobs, 1996; McClelland & Rumelhart, 1981) have assumed that inhibitory signals are passed from active letter nodes to incompatible word nodes, and hence the introduction of noise to the slot-based code would require some significant changes to these models (see Davis & Bowers, 2004, for further discussion). However, not all models that have assumed slot coding have incorporated letter-word inhibition (e.g., Paap et al., 1982, 2000).

Our present data cannot rule out the noisy version of slot coding. However, it strikes us as somewhat peculiar to make a theoretical commitment to position-specific representations and then introduce noise to these representations in an attempt to capture data that imply nonspecific letter position coding. Furthermore, we believe that this solution is ultimately not a very satisfactory one to the problems associated with slot-coding, because the addition of location noise addresses some of the problems related to position-specificity without tackling the more basic problems introduced by this method of letter position coding. Foremost among these is the *alignment problem* (Bowers, 2002; Davis, 1999, 2006). This problem arises whenever familiar patterns are presented in unfamiliar positions. For example, if the word *stop* is coded as a pattern consisting of the letter *S* in the first position, *T* in the second position, and so on, how is this word recognized when it occurs in a complex context such as *backstop*? It is clear that readers are able to recognize familiar words in unfamiliar words: this is the basis of the ability to read novel compound words like *buckstop*, which are the most common form of neologisms in the English language. The addition of location uncertainty does not help to explain how this might be achieved; it would be implausible (and not viable) to suppose that location coding is so noisy that a letter *S* in position five can activate a word beginning with *S*. Grainger et al. (in press) recently have reported some masked form priming data that illustrate this alignment problem. They found that form primes containing four of the seven letters of a target were effective primes irrespective of whether these four letters were the first four letters, the final four letters, or the first, third, fifth, and seventh letters of the target. It is very hard to see how a noisy slot-coding model could accommodate this finding. The fundamental nature of the alignment problem argues strongly against the use of position-specific letter coding in models of reading. Note that spatial coding is not subject to this problem, because the same relative pattern of activities is employed to code a word irrespective of its serial position (e.g., the pattern for coding *stop* is the same in *stop* and *buckstop*).

Could a Variant of Context-Coding Explain the Present Data?

It is also possible to ask whether a modified version of open-bigram coding or Wickelcoding could explain the present results. Although both of these schemes encode a letter's position in terms of its surrounding letter context, rather than its absolute position (i.e., they are context-coding schemes), the reasons for their failure to accommodate the present data are quite different, and hence different sorts of modifications to these schemes seem to be required. The fundamental problem with the open-bigram coding schemes is that they do not assign sufficient weight to letter contiguity. In the case of discrete open-bigram coding, contiguity information is dismissed altogether for four-letter strings; thus, the *sl* in *slek* is indistinguishable from that in *silk* or *soil*. As we have seen, this leads to the incorrect prediction that N1R pairs are just as similar as SN pairs. The only way to rectify this problem is to incorporate letter contiguity information, so that the coding scheme is (in principle) capable of distinguishing letter pairs that match perfectly (e.g., the *so* in *stop* and *shop*) from letter pairs that are only an approximate match (e.g., the *so* in *stop* and *soap*).

The version of open-bigram coding posited in the SERIOL model does incorporate some degree of letter contiguity information; this is achieved by assigning greater activities to bigram units representing contiguous letter pairs than to those which represent noncontiguous letter pairs. However, this is insufficient to capture the influence of letter contiguity; indeed, in its present form, the model makes a prediction that goes in the opposite direction to that observed in the data. This is the result of two separate problems with this coding scheme. The first problem is that bigram activity—the property that codes letter contiguity—also has the responsibility for coding the effects of serial position on perceptibility. Consequently, a bigram's activity can be smaller than 1 either because it is noncontiguous or because its initial letter occurs in a medial position, or both: the recipient word node has no way to distinguish among these possibilities. Similarly, a bigram's activity can be close to 1 either because it is the initial contiguous bigram of a letter string, or because it is the open-bigram formed by the combination of the initial letter and the final letter (i.e., the "least" contiguous, but the most perceptible). It is this confound between the two types of information coded by bigram activities that causes the SERIOL model to make a prediction in the opposite direction to the data. Although it is desirable to code the effects of serial position on a letter's perceptibility, problems of this sort indicate the necessity of disentangling this factor from letter contiguity.

The confound between the coding of contiguity and letter perceptibility is not the only problem for continuous open-bigram coding, however. If bigram activities did not code for the effects of letter position on perceptibility (e.g., by setting bigram activities to 1 for all contiguous letter pairs and 0.6 for all noncontiguous pairs), continuous open-bigram coding would make exactly the same (incorrect) prediction as discrete open-bigram coding, that is, that N1R pairs and SN pairs are equally similar. Although the bigram activities encode information about letter contiguity, this information is not exploited by the matching mechanism, which is insensitive to contiguity. The SERIOL model adopts the standard assumption that matches between input signals and connection weights are computed by summing the products of open-bigram activities and corresponding weights. But multiplying the activity of a noncontiguous unit by the weight associated with an noncontiguous unit will result in a smaller product than multiplying the same weight by the activity of a contiguous unit (i.e., 0.6×0.6 is smaller than 1×0.6), even though the incongruence between the value of the activity and the weight indicates a contiguity mismatch. Thus, it is not sufficient to rely on the model's activities and weights to distinguish between contiguous and noncontiguous letter pairs: the mechanism that matches these activity patterns and weight vectors must also be modified so as to be sensitive to this variable (Davis, 2006, describes a matching mechanism that could be employed for this purpose).

These considerations indicate the properties that are necessary for open-bigram coding to be able to accommodate the data presented here: (a) Letter contiguity must be explicitly coded (as in the continuous version of open-bigram coding, but not the discrete version), (b) Letter contiguity and serial position information must be dissociated (this requires modification to continuous open-bigram coding), and (c) The mechanism that matches input codes against previously learned codes must be sensitive to incongruities in letter contiguity. It seems likely that the capabilities of an open-bigram coding scheme that implements these modifications will be the subject of future research.

In contrast to the open-bigram coding schemes, the problem encountered by Wickelcoding is that strict letter contiguity is weighted too heavily. Consequently, a disruption of contiguity at the letter level exerts a large effect on the lexical matching process, due to the disruption to letter triples. For example, the mismatch in the contiguity of the letters *L* and *K* means that *slek* does not share any Wickelfeatures with *silk*, despite the fact that the two letter strings share the same exterior letters and a common medial letter. This extreme commitment to letter contiguity could be weakened through the introduction of location noise. For example, the letter string *slek* could activate both *ek#* and (to a lesser extent) *lk#*, and the partial activation of the latter unit would result in a small degree of overlap between the coding of *slek* and *silk*.

A more sophisticated version of Wickelcoding was developed by Mozer (1991). Mozer's scheme allows for Wickelcodes that code three letters in four consecutive positions, including wildcard characters. For example, the code *S***OP* would be used to code both *stop* and *shop* (and any other letter string where some letter is preceded by the letter *S* and followed by *OP*). He also includes codes for just the initial letter or final letter (e.g., _S or P_). The addition of these extra units results in a considerable increase in the number of codes that are required: all told, a full implementation requires 56,966 possible letter cluster units! A more important characteristic of the model, for present purposes, is its assumption of position uncertainty, which leads to a nonveridical pattern of activity across the Wickelcoding units. In principle, this could enable Mozer's scheme to accommodate our results, due to the partial activation of incorrect units (e.g., the S*OP unit, which is part of the representation of *stop*, may be partially activated by the stimulus *soap*). However, it is difficult to quantify this prediction. Very few simulations of this model have been reported, in large part due to the huge computational burden imposed by its implementation. Davis (2006) considers this coding scheme in more detail and notes some other potential problems that the model may have in accommodating empirical data.

The aforementioned considerations suggest some ways in which the two types of context-coding schemes could potentially be modified so as to attain a better fit to the behavioral data. Considerations of parsimony may come into play here: context-coding schemes require vastly more coding units than spatial coding. For example, a completely general implementation of Wickelcoding requires tens of thousands of Wickelfeatures. When continuous open-bigram coding is used, a seven-letter string can be represented by activating 21 bigram units, but this number increases factorially if uncertainty about letter identity is introduced (e.g., to code *OF* it would be necessary to activate not just the *OF* unit, but also bigrams involving visually similar letters, e.g., *OE*, *QE*, *QF*, etc.). Another reason for preferring to avoid context-coding schemes, which cannot be addressed by straightforward modifications, is the *dispersion problem* (Plaut et al., 1996). Satisfactory generalization is very difficult to achieve when Wickelcoding or open-bigram coding is employed, because spelling-sound correspondences (e.g., the sound associated with the letter *p*) are dispersed over an extremely large number of local contexts (e.g., #*pa*, *elp*, *op*#, etc.). This problem critically affected the nonword naming performance of Seidenberg and McClelland's (1989) model (Besner et al., 1990; Plaut et al., 1996). By contrast, the position and context-independence of letter units in spatial coding is ideal for learning regular associations between letters and phonemes (e.g., learning that $p \rightarrow$ /p/ enables appropriate generalisation even to contexts entailing very unfamiliar bigram combinations).

Serial Position Effects

A second issue of interest in the present experiments was the interaction of serial position and orthographic similarity. The default assumption adopted by the SOLAR model is that all interior positions contribute equally to the computation of similarity; similar assumptions are implicit in many other models (e.g., Coltheart et al., 2001; Grainger & Jacobs, 1996; Paap et al., 1982, 2000). A completely different assumption is adopted by Whiteney's (2001) SERIOL model, as a consequence of its serial, left-to-right encoding assumptions. That is, the SERIOL model predicts that neighbors that differ at the fourth position should be more similar than neighbors that differ at the second position. Our results offer no support for this prediction. Instead, it appears that the default assumption—that serial position does not matter for interior letters—is correct.

The present results do not address the question of whether exterior letters are assigned greater weight than interior letters in the computation of orthographic similarity. Several findings, from a variety of experimental paradigms, suggest that this is indeed the case (e.g., Forster, 1976; Humphreys et al., 1990; McCusker et al., 1981; Perea, 1998; Perea & Lupker, 2003a, 2003b; Schoonbaert & Grainger, 2004), and modelers of visual word identification have discussed a number of possible explanations of this effect (e.g., Davis, 2006; Grainger, O'Regan, Jacobs, & Segui, 1992; Rumelhart & McClelland, 1982; Paap et al., 1982). It seems most likely that such an effect reflects the fact that exterior letters are perceived more accurately than interior letters (Estes, Allmeyer, & Reder, 1976; Mewhort, Campbell, Marchetti, & Campbell, 1981). However, it is interesting to note that Grainger et al. (in press) have recently reported a failure to find any interaction between serial position effects and orthographic similarity. Thus, both the existence of the putative exterior-interior letter difference and the best means of accommodating such a difference within existing models remains a subject for further research.

In conclusion, the results of the present experiments support a form of letter position coding that is not tied to absolute position but that is sensitive both to the relative position of letters and to the contiguity of letters. The best candidate appears to be a form of letter position coding that encodes sequence information across a set of position- and context-independent letter units. This approach is the one taken in the spatial coding scheme used in the SOLAR model (Davis, 1999, 2001, 2004, 2006). This coding scheme was designed to address some of the fundamental limitations of position-specific coding schemes. In addition to the letter migration data that we have reported here and elsewhere (Davis & Bowers, 2004), spatial coding has been used to explain a broad variety of empirical data, especially the effects of orthographic similarity, in studies that have examined SNs (Davis, 1999), transposed neighbors (Davis & Andrews, 2001; Davis, 1999), addition and deletion neighbors (Bowers, Davis, & Hanley, 2005; Davis & Taft, 2005), and the automatic segmentation of novel compounds (Andrews & Davis, 1999; Davis, 1999). The present study has extended this previous work by suggesting another form of orthographic similarity—N1R—and demonstrating that spatial coding correctly predicts the similarity of N1R pairs relative to other forms of orthographic similarity, in contrast to other coding schemes. This predictive success, together with the ability to solve critical problems such as the alignment and dispersion problems, suggests that spatial coding provides a close approximation of the way in which readers code letter position.

References

Andrews, S. (1996). Lexical retrieval and selection processes: Effects of transposed-letter confusability. *Journal of Memory and Language, 35,* 775–800.

Andrews, S. (1997). The effect of orthographic similarity on lexical retrieval: Resolving neighborhood conflicts. *Psychonomic Bulletin & Review, 4,* 439–461.

Andrews, S., & Davis, C. J. (1999). Interactive activation accounts of morphological decomposition: Finding the trap in mousetrap? *Brain & Language, 68,* 355–361.

Besner, D., Twilley, L., McCann, R. S., & Seergobin, K. (1990). On the association between connectionism and data: Are a few words necessary? *Psychological Review, 97*, 432–446.

Bowers, J. S. (2002). Challenging the widespread assumption that connectionism and distributed representations go hand-in-hand. *Cognitive Psychology, 45,* 413–445.

Bowers, J. S. (2003). An abstractionist account of masked and long-term priming. In S. Kinoshita & S. J. Lupker (Eds.), *Masked priming: The state of the art* (pp. 39–55). Philadelphia: Psychology Press.

Bowers, J. S., Davis, C. J., & Hanley, D. A. (2005). Automatic semantic activation of embedded words: Is there a 'hat' in 'that'? *Journal of Memory and Language, 52,* 131–143.

Brysbaert, M. (2001). Prelexical phonological coding of visual words in Dutch: Automatic after all. *Memory & Cognition, 29*, 765–773.

Chambers, S. M. (1979). Letter and order information in lexical access. *Journal of Verbal Learning and Verbal Behavior, 18,* 225–241.

Coltheart, M., Davelaar, E., Jonasson, J. T., & Besner, D. (1977). Access to the internal lexicon. In S. Dornic (Ed.), *Attention and performance VI* (pp. 535–555). New York: Academic Press.

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-organising lexical acquisition and recognition (SOLAR) model of visual word recognition (Doctoral dissertation, University of New South Wales, Sydney, New South Wales, Australia, 1999). *Dissertation Abstracts International, 62,* 594. (Available on the Web at www.maccs.mq.edu.au/~colin)

Davis, C. J. (2001, December). *A novel method of parallel matching in neural network models of recognition*. Paper presented at the 10th Australasian Mathematical Psychology Conference, Newcastle.

Davis, C. J. (2003). Factors underlying masked priming effects in competitive network models of visual word recognition. In S. Kinoshita & S. J. Lupker (Eds.), *Masked priming: The state of the art* (pp. 121–170). Philadelphia: Psychology Press.

Davis, C. J. (2004). Phase coding: A method for simultaneously encoding order and signal strength in a neural network model of recognition. *Proceedings of the Eighth International Conference on Cognitive and Neural Systems,* 21.

Davis, C. J. (2005). N-Watch: A program for deriving neighborhood size and other psycholinguistic statistics. *Behavior Research Methods, 37,* 65–70.

Davis, C. J. (2006). *The SOLAR (Self-Organizing Lexical Acquisition & Recognition) model of visual word identification, Part I: Orthographic* *input coding and lexical matching.* Manuscript submitted for publication.

Davis, C. J., & Andrews, S. (2001). Inhibitory effects of transposed-letter similarity for words and non-words of different lengths. *Australian Journal of Psychology, 53,* 50.

Davis, C. J., & Bowers, J. S. (2004). What do letter migration errors reveal about letter position coding in visual word recognition? *Journal of Experimental Psychology: Human Perception and Performance, 30,* 923–941.

Davis, C. J., & Coltheart. M. (2002). Paying attention to reading errors in acquired dyslexia. *Trends in Cognitive Sciences, 6,* 359–361.

Davis, C. J., & Lupker, S. J. (in press). Masked inhibitory priming in English: Evidence for lexical inhibition. *Journal of Experimental Psychology: Human Perception and Performance.*

Davis, C. J., & Taft, M. (2005). More words in the neighborhood: Interference in lexical decision due to deletion neighbors. *Psychonomic Bulletin & Review, 12,* 904–910.

de Moor, W., & Brysbaert, M. (2000). Neighborhood-frequency effects when primes and targets are of different lengths. *Psychological Research, 63,* 159–162.

Drews, E., & Zwitserlood, P. (1995). Morphological and orthographic similarity in visual word recognition. *Journal of Experimental Psychology: Human Perception and Performance, 21,* 1098–1116.

Ellis, A. W., Flude, B. M., & Young, A. W. (1987). "Neglect dyslexia" and the early visual processing of letters in words and nonwords. *Cognitive Neuropsychology, 4,* 439–464.

Estes, W. K., Allmeyer, D. H., & Reder, S. M. (1976). Serial position functions for letter identification at brief and extended exposure durations. *Perception & Psychophysics, 19,* 1–15.

Evett, L. J., & Humphreys, G. W. (1981). The use of abstract graphemic information in lexical access. *Quarterly Journal of Experimental Psychology, 33A,* 325–350.

Ferrand, L., & Grainger, J. (1992). Phonology and orthography in visual word recognition: Evidence from masked nonword priming. *Quarterly Journal of Experimental Psychology, 45A,* 353–37.

Ferrand, L., & Grainger, J. (1993). The time course of orthographic and phonological code activation in the early phases of visual word recognition. *Bulletin of the Psychonomic Society, 31,* 119–122.

Forster, K. I. (1976). Accessing the mental lexicon. In E. C. J. Walker & R. J. Wales (Eds.), *New approaches to language mechanisms* (pp. 257–287). Amsterdam: North-Holland.

Forster, K. I. (1987). Form-priming with masked primes: The best-match hypothesis. In M. Coltheart (Ed.), *Attention and Performance XII* (pp. 127–146). Hillsdale, NJ: Erlbaum.

Forster, K. I. (2000). The potential for experimenter bias effects in word recognition experiments. *Memory & Cognition, 28,* 1109–1115.

Forster, K. I., & Davis, C. (1984). Repetition priming and frequency attenuation in lexical access. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 10,* 680–698.

Forster, K. I., Davis, C., Schoknecht, C., & Carter, R. (1987). Masked priming with graphemically related forms: Repetition or partial activation? *Quarterly Journal of Experimental Psychology, 39A,* 211–251.

Forster, K. I., & Forster, J. C. (2003). DMDX: A Windows display program with millisecond accuracy. *Behavior Research Methods, Instruments & Computers, 35*, 116–124.

Forster, K. I., & Taft, M. (1994). Bodies, antibodies, and neighborhood-density effects in masked form priming. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 20*, 844–863.

Forster, K. I., & Veres, C. (1998). The prime lexicality effect: Form-priming as a function of prime awareness, lexical status, and discrimination difficulty. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 24*, 498–514.

Grainger, J., Granier, J. P., Farioli, F., Van Assche, E., & van Heuven, W. (in press). Letter position information and printed word perception: The relative-position priming constraint. *Journal of Experimental Psychology: Human Perception and Performance*.

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

Grainger, J., & Jacobs, A. M. (1999). Temporal integration of information in orthographic priming. *Visual Cognition, 6,* 461–492.

Grainger, J., O'Regan, J. K., Jacobs, A. M., & Segui, J. (1989). On the role of competing word units in visual word recognition: The neighborhood frequency effect. *Perception & Psychophysics, 45,* 189–195.

Grainger, J., O'Regan, J. K., Jacobs, A. M., & Segui, J. (1992). Neighborhood frequency effects and letter visibility in visual word recognition. *Perception & Psychophysics, 51,* 49–56.

Grainger, J., & van Heuven, W. J. B. (2003). Modeling letter position coding in printed word perception. In P. Bonin (Ed.), *Mental lexicon: "Some words to talk about words"* (pp. 1–23). New York: Nova Science.

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

Grossberg, S. (1978). A theory of human memory: Self-organization and performance of sensory-motor codes, maps, and plans. In R. Rosen & F. Snell (Eds.), *Progress in theoretical biology* (pp. 233–374). New York: Academic Press.

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

Hinton, J., Liversedge, S. P., & Underwood, G. (1998). Neighborhood effects using a partial priming methodology: Guessing or activation? *Journal of Experimental Psychology: Learning, Memory, and Cognition, 24,* 1294–1305.

Hinton, G. E., & Shallice, T. (1991). Lesioning an attractor network: Investigations of acquired dyslexia. *Psychological Review, 98,* 74–95.

Humphreys, G. W., Evett, L. J., & Quinlan, P. T. (1990). Orthographic processing in visual word identification. *Cognitive Psychology, 22,* 517–560.

Jacobs, A. M., Rey, A., Ziegler, J. C., & Grainger, J. (1998). MROM-p: An interactive activation, multiple readout model of orthographic and phonological processes in visual word recognition. In J. Grainger & A. M. Jacobs (Eds.), *Localist connectionist approaches to human cognition* (pp. 147–188). Mahwah, NJ: Erlbaum.

Masson, M. J., & Bodner, G. E. (2003). A retrospective view of masked priming: Toward a unified account of masked and long-term repetition priming. In S. Kinoshita & S. J. Lupker (Eds.) *Masked priming: The state of the art* (pp. 57–94). Philadelphia: Psychology Press.

McClelland, J. L., & Mozer, M. C. (1986). Perceptual interactions in two-word displays: Familiarity and similarity effects. *Journal of Experimental Psychology: Human Perception and Performance, 12,* 18–35.

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.

McCusker, L. X., Gough, P. B., & Bias, R. G. (1981). Word recognition inside out and outside in. *Journal of Experimental Psychology: Human Perception and Performance, 7,* 538–551.

Mewhort, D. J. K., Campbell, A. J., Marchetti, F. M., & Campbell, J. I. D. (1981). Identification, localization, and "iconic memory": An evaluation of the bar-probe task. *Memory & Cognition, 9,* 50–67.

Mozer, M. C. (1991). *The perception of multiple objects: A connectionist approach*. Cambridge, MA: MIT Press.

O'Regan, J. K., & Jacobs, A. M. (1992). Optimal viewing position effect in word recognition: A challenge to current theory. *Journal of Experimental Psychology: Human Perception and Performance, 18,* 185–197.

Paap, K. R., Johansen, L. S., Chun, E., & Vonnahme, P. (2000). Neighborhood frequency does affect performance in the Reicher task: Encoding or decision? *Journal of Experimental Psychology: Human Perception and Performance, 26,* 1691–1720.

Paap, K. R., Newsome, S. L., McDonald, J. E., & Schvaneveldt, R. W. (1982). An activation-verification model for letter and word recognition: The word-superiority effect. *Psychological Review, 89,* 573–594.

Page, M. P. A. (1994). Modelling the perception of musical sequences with self-organizing neural networks. *Connection Science: Journal of Neural Computing, Artificial Intelligence and Cognitive Research, 6,* 223–246.

Page, M. P. A., & Norris, D. (1998). The primacy model: A new model of immediate serial recall. *Psychological Review, 105,* 761–781.

Perea, M. (1998). Orthographic neighbors are not all equal: Evidence using an identification technique. *Language and Cognitive Processes, 13,* 77–90.

Perea, M., Gómez, P., & Ratcliff, R. (2003). *A model of encoding letter positions: The overlap model*. Paper presented at the 12th Congress of the European Society for Cognitive Psychology, September, Granada, Spain.

Perea, M., & Lupker, S. J. (2003a). Does jugde activate COURT? Transposed-letter similarity effects in masked associative priming. *Memory & Cognition, 31,* 829–841.

Perea, M., & Lupker, S. J. (2003b). Transposed-letter confusability effects in masked form priming. In S. Kinoshita & S. J. Lupker (Eds.), *Masked priming: The state of the art* (pp. 97–120). Philadelphia: Psychology Press.

Perea, M., & Rosa, E. (2000). Repetition and form priming interact with neighborhood density at a brief stimulus onset asynchrony. *Psychonomic Bulletin & Review, 7*, 668–677.

Peressotti, F., & Grainger, J. (1999). The role of letter identity and letter position in orthographic priming. *Perception and Psychophysics, 61,* 691–706.

Perfetti, C. A., & Bell, L. (1991). Phonemic activation during the first 40 ms of word identification: Evidence from backward masking and priming. *Journal of Memory & Language, 30*, 473–485.

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.

Rayner, K. (1979). Eye guidance in reading: Fixation locations within words. *Perception, 8*, 21–30.

Rumelhart, D. E., & McClelland, J. L. (1986). On learning the past tenses of English verbs. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel distributed processing, Vol 2: Explorations in the microstructure of cognition* (pp. 216–271). Cambridge, MA: MIT Press.

Schoonbaert, S., & Grainger, J. (2004). Letter position coding in printed word perception: Effects of repeated and transposed letters. *Language & Cognitive Processes, 19,* 333–367.

Segui, J., & Grainger, J. (1990). Priming word recognition with orthographic neighbors: Effects of relative prime-target frequency. *Journal of Experimental Psychology: Human Perception and Performance, 16,* 65–76.

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

Shallice, T., & McGill, J. (1978). The origins of mixed errors. In J. Requin (Ed.), *Attention and performance VII* (pp. 193–208). Amsterdam: North-Holland.

Shillcock, R., Ellison, T. M., & Monaghan, P. (2000). Eye-fixation behaviour, lexical storage and visual word recognition in a split processing model. *Psychological Review, 107,* 824–851.

Shillcock, R. C., & Monaghan, P. (2001). The computational exploration of visual word recognition in a split model. *Neural Computation, 13,* 1171–1198.

Taft, M., & van Graan, F. (1998). Lack of phonological mediation in a semantic categorization task. *Journal of Memory and Language, 38,* 203–224.

Treisman, A. & Souther, J. (1986). Illusory words: The roles of attention and of top-down constraints in conjoining letters to form words. *Journal of Experimental Psychology: Human Perception and Performance, 12,* 3–17.

Van Heuven, W., Dijkstra, T., Grainger, J., & Schriefers, H. (2001). Shared neighborhood effects in masked orthographic priming. *Psychonomic Bulletin & Review, 8,* 96–101.

Whitney, C. (2001). How the brain encodes the order of letters in a printed word: The SERIOL model and selective literature review. *Psychonomic Bulletin & Review, 8,* 221–243.

Whitney, C., & Berndt, R. S. (1999). A new model of letter string encoding: Simulating right neglect dyslexia. In J. A. Reggia, E. Ruppin, & D. Glanzman (Eds.), *Progress in brain research* (Vol. 121, pp. 143–163), Amsterdam: Elsevier.

Wickelgren, W. A. (1969). Auditory or articulatory coding in verbal short-term memory. *Psychological Review, 76,* 232–235.

Zorzi, M., Houghton, G., & Butterworth, B. (1998). Two routes or one in reading aloud? A connectionist dual-process model. *Journal of Experimental Psychology: Human Perception and Performance, 24*, 1131–1161.

(*Appendixes follow*)

Appendix A

Match Calculations in the SOLAR Model

Spatial Codes

A spatial code can be written as a vector consisting of n elements, where n is the number of letters in the input string and the values in the vector represent the activities of the corresponding letter nodes. Spatial codes always use a monotonically descending series to code letter position. Suppose, for simplicity, that a four-letter word is coded by the set of activities {4, 3, 2, 1}; for example, in the word *STOP*, the *S* letter node is coded by an activity of 4, the *T* letter node by an activity of 3, and so on.

Equilibrium Values of the Weights Connecting Letter and Word Nodes

We suppose that one effect of learning is that each word node "knows" which letters to attend to, that is, which letters make up the particular word that it codes. Thus the word node that codes *STOP* only considers inputs from the letter nodes for S, T, O, and P. For the ith word node, this set of letters is denoted L_i, and the number of letters in this set (i.e., the length of the word) is denoted l_i (e.g., L_{STOP} = {S, T, O, P} and $l_{STOP} = 4$). The weight between a letter node and a word node is equivalent to the value of that letter node's activity in the spatial code for that word; for example, the weight from the S letter node to the STOP word node is $z_{\mathrm{S,STOP}} = 4$. Davis (1999) describes how the SOLAR model is able to self-organize so as to learn appropriate weights following exposure to a vocabulary.

Computation of Match Values

Each word node computes a match value that describes the degree to which the word that it codes matches the current input stimulus. The method we describe here is called superposition matching, and is described in more detail in Davis (submitted).[A1] The first step involves computing a set of signal-weight differences. For each of the elements in the set L_i a difference d_{ji} is computed by subtracting from s_j (the activity of the jth letter node) the corresponding weight z_{ji}, i.e.,

$$d_{ji} = s_j - z_{ji} \tag{1}$$

Each signal-weight difference is then associated with a continuous function $f_{ji}(x)$ that is symmetrical around $x = d_{ji}$:

$$f_{ji}(x) = e^{(-(D_{ji}-x)^2/\sigma)} \tag{2}$$

The parameter σ in (2) controls the width of the difference function and can be interpreted as a measure of letter position uncertainty (a default value of $\sigma = 3$ is assumed for this parameter). Then the superposition of these functions is:

$$F_i(x) = \Sigma_{j \in L_i} f_{ji}(x) \tag{3}$$

(where the set L_i refers to the set of comparison letters). A match value M_i can then be found by dividing the peak of the superposition function by the number of comparison letters (l_i), that is,

$$M_i = \frac{\max(F_i(x))}{l_i} \tag{4}$$

The set of equations (1) through (4) produce a match value that lies between 0 and 1.

To illustrate these computations, consider the match values that are computed by the STOP word node for the inputs *stop*, *shop*, *soap* and *snap*. A perfect match (*stop*) results in four signal-weight difference functions that are all aligned around 0, and thus the peak of the superposition function is 4, and the match value is 4/4 = 1. In the case of a SN like *shop* there are three signal-weight difference functions that are all aligned around 0, and thus the peak of the superposition function is 3, and the match value is 3/4 = .75. Similarly, for a DSN like *snap* there are two signal-weight difference functions, both aligned around 0, and thus the peak of the superposition function is 2, and the match value is 2/4 = .5

In the case where the input stimulus (*soup*) is a neighbor once-removed of the comparison word (*stop*), the difference functions for the letters *S* and *P* will be aligned around the modal difference of 0. The difference function for the letter *O* is misaligned with these two, although it is close enough to affect the peak of the superposition function (i.e., the letter *O* "counts" in the computation of similarity, even though it occurs in a different position in *soap* and *stop*). The exact value of this peak will lie somewhere between 2 and 3, depending on the width of the difference functions. Given the setting $\sigma = 3$, the peak of the superposition function is 2.79, resulting in a match value of 2.79/4 = .70. This is a much better match than for a DSN like *snap*, but a poorer match than for a SN like *step*, just as is suggested by the behavioral data discussed in the text.

[A1] The appendix of Davis and Bowers (2004) describes an alternative method which is slightly simpler, but which results in the same predictions for SNs, DSNs, and N1Rs. Davis (1999b) discusses reasons to prefer superposition matching.

Appendix B

Stimuli in Experiment 1

Word Targets

Target word	Illusory word	SN context	N1R context	DSN context
port	part	palt	plat	pult
burn	born	boin	bion	blin
send	sand	sald	slad	slud
bend	bond	bord	brod	burd
word	ward	waud	wuad	wuld
read	road	roid	riod	riud
fill	fall	farl	fral	frul
call	cell	cerl	crel	cirl
form	farm	falm	flam	felm
cast	cost	cort	crot	crit
rang	ring	rieg	reig	relg
miss	mess	meus	mues	murs
ease	else	elne	enle	enve
post	past	pait	piat	plit
hung	hang	haig	hiag	hilg
soup	soap	siap	saip	simp
soil	soul	swul	surl	swal
cake	care	cere	cree	cele
term	team	twam	tawm	twim
pace	pale	pule	plue	pume
sink	silk	selk	slek	sork
lord	loud	leud	lued	lind
sake	same	sume	smue	sube
town	torn	tern	tren	teln
land	laid	luid	liud	lowd
fond	ford	fard	frad	fawd
film	firm	furm	frum	fusm
five	fire	fure	frue	fube
held	head	hiad	haid	huid
bent	belt	balt	blat	baft

Nonword Targets

SN Context Condition

Nonword	Illusory word	SN context	DSN context
hont	hunt	hurt	heat
cire	care	case	clue
wure	wore	woke	wake
cipe	cape	cage	cube
pock	pack	park	pork
rale	role	rose	rare
doal	dual	dull	doll
wace	wade	wide	wife
foem	form	firm	film
bamk	bank	bunk	bulk
weft	went	want	what
mibe	mile	male	mode
pife	pipe	pope	pose
cabe	cane	cone	cute

N1R Context Condition

Nonword target	Illusory word	N1R context	DSN context
gofs	goes	gets	guys
borl	boil	bill	bell
foyl	foul	fuel	feel
loid	load	laid	lied
coul	coal	call	cell
troe	true	tune	tame
fike	file	flee	fake
fust	fast	flat	foot
bued	bred	bird	band
wurm	worm	whom	whim
pist	post	plot	putt
bick	back	beak	book
smap	soap	shop	step
smot	slot	salt	seat

Stimuli in Experiment 2

Target word	SN prime	N1R prime	DSN prime	Unrelated prime
BULB	belb	bleb	bemb	wamf
WOLF	worf	wrof	werf	bimb
DUMB	duob	doub	drob	senf
DEBT	deat	daet	duat	crom
SELF	silf	slif	sinf	norb
GENE	gine	gnie	gire	drib
NUMB	nuab	naub	narb	sier
BUZZ	bulz	bluz	blez	krat
CHUM	clum	culm	clim	blaz
DRUM	drem	derm	delm	gleb
DRUG	drog	dorg	dolg	spon
PLUS	prus	purs	pers	bour
GLAD	gnad	gand	gond	pirs
CLUB	cleb	celb	cerb	jonz
GRUB	glub	gulb	glab	dilm
SIGN	sian	sain	shan	calb
JAZZ	jaez	jeaz	jerz	dulg
BLUR	baur	buar	biar	dest
WRAP	wrup	wurp	wump	guld
STIR	sair	siar	saer	wulp

Stimuli Used In Experiments 3 and 4

Word targets

Target	SN2	SN4	N1R−	N1R+	Unrelated
ACUTE	ayute	acuye	auyte	acyue	yeqld
ADOPT	ajopt	adojt	aojpt	adjot	wuisk
AGONY	ajony	agojy	aojny	agjoy	brupe
ALBUM	asbum	albsm	absum	alsbm	yczht
ALOUD	ayoud	aloyd	aoyud	alyod	wifgh
ANKLE	axkle	ankxe	akxle	anxke	wgzon
AWOKE	ayoke	awoye	aoyke	awyoe	wdvth
BACON	bmcon	bacmn	bcmon	bamcn	rmlax
BASIN	bdsin	basdn	bsdin	badsn	vyter
BISON	bvson	bisvn	bsvon	bivsn	tdval
BOXER	bhxer	boxhr	bxher	bohxr	yaust
BRUTE	bpute	brupe	bupte	brpue	vlzid
CABIN	cdbin	cabdn	cbdin	cadbn	vkdeo
CAMEL	cgmel	camgl	cmgel	cagml	viors
CHALK	cgalk	chagk	caglk	chgak	vernm
CHORD	cqord	choqd	coqrd	chqod	tpgic
CIDER	ckder	cidkr	cdker	cikdr	vaukt
CIGAR	ctgar	cigtr	cgtar	citgr	telmo
CLERK	cverk	clevk	cevrk	clvek	tordh
COBRA	cybra	cobya	cbyra	coyba	tigdr
CRUEL	cyuel	cruyl	cuyel	cryul	thyub
CRUSH	ciush	cruih	cuish	criuh	thxif

(*Appendix continues*)

Target	SN2	SN4	N1R−	N1R+	Unrelated
DELAY	dxlay	delxy	dlxay	dexly	tgorn
DEMON	dcmon	demcn	dmcon	decmn	tfxic
DENIM	dpnim	denpm	dnpim	depnm	vbcar
DEPTH	dvpth	depvh	dpvth	devph	scajf
DEVIL	dtvil	devtl	dvtil	detvl	sygrp
DISCO	dlsco	dislo	dslco	dilso	rfyle
DOGMA	dtgma	dogta	dgtma	dotga	ocexn
DRAFT	dqaft	draqt	daqft	drqat	slzon
DRIFT	djift	drijt	dijft	drjit	vonda
DWARF	djarf	dwajf	dajrf	dwjaf	sozbr
ELBOW	egbow	elbgw	ebgow	elgbw	qoyta
ETHIC	eahic	ethac	ehaic	etahc	rugmy
FETCH	fxtch	fetxh	ftxch	fexth	ropbn
FIBRE	fibre	fibte	fbtre	fitbe	rmval
FILTH	fzlth	filzh	flzth	fizlh	wajle
FLUID	fjuid	flujd	fujid	fljud	risxy
FOCUS	fvcus	focvs	fcvus	fovcs	rinje
FRAUD	fzaud	frazd	fazud	frzad	qoyte
FREAK	fxeak	frexk	fexak	frxek	pilqt
FROZE	fqoze	froqe	foqze	frqoe	mblon
GHOST	gzost	ghozt	gozst	ghzot	rflic
GLORY	gxory	gloxy	goxry	glxoy	sauoe
GRAPH	guaph	grauh	gauph	gruah	pyuck
GRAVY	gqavy	graqy	gaqvy	grqay	pulke
GRIEF	gxief	grixf	gixef	grxif	poxlr
GUEST	gkest	guekt	gekst	guket	pnjic
HAUNT	hdunt	haudt	hudnt	hadut	orgdn
HINGE	hxnge	hinxe	hnxge	hixne	cgalk
HUMID	hgmid	humgd	hmgid	hugmd	pedgl
IMPLY	ibply	impby	ipbly	imbpy	sirfn
INDEX	ipdex	indpx	idpex	inpdx	orvbt
IRONY	iuony	irouy	iouny	iruoy	lecmn
IVORY	ijory	ivojy	iojry	ivjoy	onqst
JUICE	jaice	juiae	jiace	juaie	soqrd
LEMON	lcmon	lemcn	lmcon	lecmn	rabto
LOGIC	lpgic	logpc	lgpic	lopgc	juiae
LYRIC	ldric	lyrdc	lrdic	lydrc	medgl
MAYOR	mzyor	mayzr	myzor	mazyr	hinxe
MEDAL	mgdal	medgl	mdgal	megdl	nopbe
MELON	mblon	melbn	mlbon	mebln	lgpic
MERIT	mvrit	mervt	mrvit	mevrt	panml
NOBLE	npble	nobpe	nbple	nopbe	ldric
OCEAN	oxean	ocexn	oexan	ocxen	guket
ONSET	oqset	onsqt	osqet	onqst	irouy
ORBIT	ovbit	orbvt	obvit	orvbt	inpdx
ORGAN	odgan	orgdn	ogdan	ordgn	dilso
PANEL	pmnel	panml	pnmel	pamnl	fqoze
PANIC	pjnic	panjc	pnjic	pajnc	ijory
PEDAL	pgdal	pedgl	pdgal	pegdl	hgmid
PIANO	pfano	piafo	pafno	pifao	mrvit
PILOT	pqlot	pilqt	plqot	piqlt	fbtre
PLUCK	pyuck	pluyk	puyck	plyuk	mazyr
POLAR	pxlar	polxr	plxar	poxlr	gxief
PULSE	pklse	pulke	plkse	pukle	gqavy
QUOTA	qyota	quoya	qoyta	quyoa	goxry
QUOTE	qyote	quoye	qoyte	quyoe	ctgar
RATIO	rbtio	ratbo	rtbio	rabto	fzaud
RELAX	rmlax	relmx	rlmax	remlx	fvcus
RELIC	rflic	relfc	rlfic	reflc	pifao
RIFLE	ryfle	rifye	rfyle	riyfe	dotga
RINSE	rjnse	rinje	rnjse	rijne	flzth
RISKY	rxsky	risxy	rsxky	rixsy	fexak
RIVAL	rmval	rivml	rvmal	rimvl	fetxh
ROBIN	rpbin	robpn	rbpin	ropbn	cagml
RUGBY	rmgby	rugmy	rgmby	rumgy	ckder
SALON	szlon	salzn	slzon	sazln	hdunt
SAUCE	souce	sauoe	suoce	saoue	dxlay
SCARF	sjarf	scajf	sajrf	scjaf	ibply
SIREN	sfren	sirfn	srfen	sifrn	drqat
SOBER	szber	sobzr	sbzer	sozbr	decmn
SWORD	sqord	swoqd	soqrd	swqod	dnpim
SYRUP	sgrup	syrgp	srgup	sygrp	djift
TEMPO	tlmpo	temlo	tmlpo	telmo	eahic
THIEF	txief	thixf	tixef	thxif	acyue
THORN	tgorn	thogn	togrn	thgon	devph
THUMB	tyumb	thuyb	tuymb	thyub	cadbn
TIDAL	tvdal	tidvl	tdval	tivdl	grauh
TIGER	tdger	tigdr	tgder	tidgr	cruyl
TOPIC	tgpic	topgc	tpgic	togpc	elgbw
TORCH	tdrch	tordh	trdch	todrh	dwjaf
TOXIC	tfxic	toxfc	txfic	tofxc	agojy
VALID	vzlid	valzd	vlzid	vazld	ghozt
VAULT	vkult	vaukt	vuklt	vakut	cevrk
VENOM	vrnom	venrm	vnrom	vernm	cbyra
VICAR	vbcar	vicbr	vcbar	vibcr	devtl
VIDEO	vkdeo	vidko	vdkeo	vikdo	bohxr
VIRUS	vorus	viros	vrous	viors	ayoke
VOCAL	vgcal	vocgl	vcgal	vogcl	fujid
VODKA	vndka	vodna	vdnka	vonda	bisvn
VOTER	vyter	votyr	vtyer	voytr	bcmon
WAGON	wzgon	wagzn	wgzon	wazgn	coqrd
WEIGH	wfigh	weifh	wifgh	wefih	ayoud
WHALE	wjale	whaje	wajle	whjae	ciush
WHISK	wuisk	whiuk	wiusk	whuik	aojpt
WIDTH	wvdth	widvh	wdvth	wivdh	albsm
YACHT	yzcht	yaczt	yczht	yazct	akxle
YEAST	yuast	yeaut	yaust	yeuat	vocgl
YIELD	yqeld	yieqd	yeqld	yiqed	basdn

Nonword targets

Item	SN2	SN4	N1R−	N1R+	UR
BETCH	bwtch	betwh	btwch	bewth	triul
BILCH	bwlch	bilwh	blwch	biwlh	paery
BLONK	bbonk	blobk	bobnk	blbok	wadtn
BLORE	bqore	bloqe	boqre	blqoe	trxid
BLUND	bmund	blumd	bumnd	blmud	todmr
BOACH	bmach	boamh	bamch	bomah	trhee
BOMER	btmer	bomtr	bmter	botmr	wudth
BRELD	bveld	brevd	bevld	brved	swaae
BREWN	boewn	breon	beown	broen	thjoe
BROAK	bboak	brobk	bobak	brbok	tauiy
BRONE	bxone	broxe	boxne	brxoe	swiag
BULTY	bdlty	buldy	bldty	budly	swfal
CADER	ctder	cadtr	cdter	catdr	stxag
CANCH	cvnch	canvh	cnvch	cavnh	stzek
CARDY	cvrdy	carvy	crvdy	cavry	slyoe
CARTE	cxrte	carxe	crxte	caxre	pnjch
CHACK	ctack	chatk	catck	chtak	stver
CHENK	ctenk	chetk	cetnk	chtek	sttue
CHISE	ctise	chite	citse	chtie	stjok
CHONE	cnone	chone	conne	chnoe	stgak
CLASK	cjask	clajk	cajsk	cljak	sexrn
CLECK	cxeck	clexk	cexck	clxek	srjut
CLOAT	cyoat	cloyt	coyat	clyot	spvue
CLONK	cfonk	clofk	cofnk	clfok	styae
COGER	ciger	cogir	cgier	coigr	sifpe
COICH	cyich	coiyh	ciych	coyih	speie
CRAWN	cqawn	craqn	caqwn	crqan	sodme
CRECK	cbeck	crebk	cebck	crbek	snuie
CRELK	cgelk	cregk	ceglk	crgek	staie
CRINK	caink	criak	ciank	craik	slyeh
CRITE	czite	crize	cizte	crzie	slyak
DARLY	djrly	darjy	drjly	dajry	sltok
DINTY	dpnty	dinpy	dnpty	dipny	slgoe

Item	SN2	SN4	N1R−	N1R+	UR
DITER	dpter	ditpr	dtper	diptr	spjoe
EAKER	ecker	eakcr	ekcer	eackr	sixll
FERLY	fsrly	fersy	frsly	fesry	ctise
FIMER	fumer	fimur	fmuer	fiumr	spoal
FINTH	fbnth	finbh	fnbth	fibnh	savbe
FOUTH	fyuth	fouyh	fuyth	foyuh	rvjen
FURLY	fxrly	furxy	frxly	fuxry	riqse
GINER	gvner	ginvr	gnver	givnr	cxeck
GLAPE	ghape	glahe	gahpe	glhae	rnfer
GOUND	gjund	goujd	gujnd	gojud	risdy
GRICE	gfice	grife	gifce	grfie	lavjr
GRIDE	ggide	grige	gigde	grgie	pufnt
GUILY	gnily	guiny	ginly	guniy	prjch
HALDY	hxldy	halxy	hlxdy	haxly	rloer
HALTE	hvlte	halve	hlvte	havle	pnjer
HETCH	hptch	hetph	htpch	hepth	pkder
HINCH	hfnch	hinfh	hnfch	hifnh	pmjer
HOTER	hbter	hotbr	htber	hobtr	pibry
LANCH	lqnch	lanqh	lnqch	laqnh	pipty
LAVER	ljver	lavjr	lvjer	lajvr	piink
LIBER	lmber	libmr	lbmer	limbr	podne
LIREN	ldren	lirdn	lrden	lidrn	pgxht
LORSE	lirse	lorie	lrise	loire	pagin
LOULY	leuly	louey	luely	loeuy	wirvr
LUTER	ltter	luttr	ltter	luttr	sixme
MAUSE	mfuse	maufe	mufse	mafue	ginvr
MIDER	mtder	midtr	mdter	mitdr	halve
MORSE	mmrse	morme	mrmse	momre	luttr
MOTER	mzter	motzr	mtzer	moztr	lanqh
MUEKY	mreky	muery	merky	murey	grige
PAIRY	pbiry	paiby	pibry	pabiy	treek
PAITY	ppity	paipy	pipty	papiy	lorie
PAKER	pdker	pakdr	pkder	padkr	lirdn
PANCH	pjnch	panjh	pnjch	pajnh	libmr
PANER	pjner	panjr	pnjer	pajnr	fouyh
PAUNT	pfunt	pauft	pufnt	pafut	hinfh
PEARY	peary	peaey	paery	peeay	hotbr
PIGHT	pxght	pigxt	pgxht	pixgt	merky
PIRCH	pjrch	pirjh	prjch	pijrh	louey
PLINK	piink	pliik	piink	pliik	halxy
PLONE	pdone	plode	podne	pldoe	eakcr
POMER	pjmer	pomjr	pmjer	pojmr	guiny
PRAIN	pgain	pragn	pagin	prgan	hetph
RAIDY	rsidy	raisy	risdy	rasiy	mrmse
RILER	roler	rilor	rloer	riolr	goujd
RINER	rfner	rinfr	rnfer	rifnr	glahe
ROISE	rqise	roiqe	riqse	roqie	cyoat
ROMER	rtmer	romtr	rmter	rotmr	furxy
ROVEN	rjven	rovjn	rvjen	rojvn	fimur
SATER	svter	satvr	stver	savtr	finbh
SCRUT	sjrut	scrjt	srjut	scjrt	grife
SHABE	svabe	shave	savbe	shvae	mtzer
SHERN	sxern	shexn	sexrn	shxen	midtr
SHILL	sxill	shixl	sixll	shxil	ditpr
SHIME	sxime	shixe	sixme	shxie	dinpy
SHIPE	sfipe	shife	sifpe	shfie	darjy
SHOME	sdome	shode	sodme	shdoe	ctenk
SLANK	syank	slayk	saynk	slyak	cyich
SLESH	syesh	sleyh	seysh	slyeh	cxrte
SLOCK	stock	slotk	sotck	sltok	fersy
SLOTE	syote	sloye	soyte	slyoe	cvrdy
SLOVE	sgove	sloge	sogve	slgoe	cvnch
SNIRE	suire	sniue	siure	snuie	ctack
SPALL	soall	spaol	saoll	spoal	cnone
SPILE	seile	spiee	siele	speie	ctder
SPOCE	sjoce	spoje	sojce	spjoe	rmter
SPURE	svure	spuve	suvre	spvue	cjask
STANG	sxang	staxg	saxng	stxag	crize
STAPE	syape	staye	saype	styae	boewn
STASK	sgask	stagk	sagsk	stgak	ciger
STECK	szeck	stezk	sezck	stzek	cqawn
STITE	saite	stiae	siate	staie	cgelk
STONK	sjonk	stojk	sojnk	stjok	maufe
STULE	stule	stute	sutle	sttue	cbeck
SWALL	sfall	swafl	safll	swfal	caink
SWANG	siang	swaig	saing	swiag	btmer
SWARE	saare	swaae	saare	swaae	bdlty
TAIRY	tuiry	taiuy	tiury	tauiy	bwlch
THONE	tjone	thoje	tojne	thjoe	bveld
TOMER	tdmer	tomdr	tmder	todmr	cfonk
TRECE	thece	trehe	tehce	trhee	bmund
TRECK	teeck	treek	teeck	treek	bxone
TRIND	txind	trixd	tixnd	trxid	bqore
TRULL	tiull	truil	tuill	triul	bmach
WATEN	wdten	watdn	wtden	wadtn	bbonk
WIVER	wrver	wivrr	wvrer	wirvr	bwtch
WUTCH	wdtch	wutdh	wtdch	wudth	bboak

Received April 6, 2004
Revision received March 22, 2005
Accepted August 29, 2005 ■