

UC Merced

Proceedings of the Annual Meeting of the Cognitive Science Society

Title

Processing Ambiguous Words: Are Blends Necessary for Lexical Decision?

Permalink

<https://escholarship.org/uc/item/9bn3k2rr>

Journal

Proceedings of the Annual Meeting of the Cognitive Science Society, 26(26)

ISSN

1069-7977

Authors

Medler, David A.
Piercey, C. Darren

Publication Date

2004

Peer reviewed

Processing Ambiguous Words: Are Blends Necessary for Lexical Decision?

David A. Medler (dmedler@mcw.edu)

Language Imaging Laboratory
Department of Neurology, Medical College of Wisconsin
Milwaukee, WI

C. Darren Piercey (piercey@unb.ca)

Department of Psychology, University of New Brunswick
Fredericton, NB

Abstract

A previous computational model (Joordens & Besner, 1994) has suggested that during lexical access, ambiguous words tend toward a blend state; that is, network activations settle into an incorrect state that is a mixture of the multiple representations of the ambiguous item. It has been suggested that this blend state actually aids lexical decision (LD) for ambiguous items as the blend state creates a larger “feeling of familiarity” which lexical decision may exploit. This theory, however, is based on the results of a computational model (a simple Hopfield network) in which multiple representations cannot be learned. Here we use a Symmetric Diffusion Network (SDN) to effectively learn and retrieve multiple mappings for a single input (i.e., ambiguous items). The model consists of three main processing regions—orthographics, phonology, and semantics—and is trained on a corpus of unambiguous items and ambiguous items that range in their degree of balance (probability distribution) between the multiple meanings. Following training, the SDN is able to reproduce the correct probability distributions for the ambiguous items; that is, it does not produce blend states. Furthermore, the model qualitatively captures the processing advantage for ambiguous items. Consequently, the notion of a blend state being used for LD is re-evaluated, and further assumptions about semantic processing are explored.

Introduction

From a computational perspective, we can break basic language processing into three main components: the semantic representation (what a word means), a phonological representation (the sound of a word), and an orthographic representation (the written form of a word) (e.g., Seidenberg & McClelland, 1989; Plaut, McClelland, Seidenberg, & Patterson, 1996; Harm & Seidenberg, 1999). The relationship between the phonological and semantic representations is initially established in early childhood, and then the mapping between the orthographic representation and the phonological representation (spelling-to-sound conversion) along with the mapping between the orthographic representation and the semantic representation (spelling-to-meaning conversion) is learned later in life (e.g., Harm & Seidenberg, *in press*).

Ideally, there would be a one-to-one mapping between any of the representations, such that one spelling would

correspond to one pronunciation, which would correspond to one meaning. Unfortunately, one-to-one mappings are far from the norm in English. That is, words that sound the same (homophones) can have different semantic representations (/flaɪ/: fly [insect]; fly [zipper]), different orthographic representations (/laɪt/: light [fewer calories]; lite [fewer calories]), or different semantic and orthographic representations (/beə/: bear [furry animal]; bare [naked]). Similarly, words that are spelled the same (have the same orthographic representation) can have different phonological representations (either: /'aɪ.Də/: [one or the other], /'I:Də/: [one or the other]), or phonological and semantic representations (wind: /waɪnd/ [twist]; /wɪnd/ [moving air]).

In fact, many words in English have polysemous or ambiguous semantics. For example, WordNet® (Fellbaum, 1998) lists a total of 146,350 noun, verbs, adjectives, and adverbs. Table 1 shows the percentage of unique and ambiguous words, as well as sense data. Whereas ambiguity is often defined as a word having multiple meanings across semantic categories or word classes, a word’s sense is defined as its meaning within a semantic category and can vary dramatically from the prior definition of ambiguity. For example, although Borowsky & Masson (1996) consider “deep” to be an unambiguous word, WordNet® lists “deep” with 3 noun senses, 15 adjective senses, and 3 adverb senses. It is clear that ambiguity is prevalent in English, and there is evidence that it has an effect on how we process words.

Table 1. Percentage of words having unique, ambiguous, and multisense meanings.

Word Class	Unique	Ambiguous	Senses
Noun	86.7	13.3	29.7
Verb	53.4	46.6	75.4
Adjective	74.5	25.5	48.7
Adverb	82.9	17.1	33.1

For example, the behavioral data from word ambiguity studies produces a paradox. In a lexical decision (LD) paradigm, ambiguity aids in word identification; that is, ambiguous words are identified as words more quickly and more accurately than unambiguous words (Gernsbacher,

1984; Borowsky & Masson, 1996). In contrast, in connected text studies (Rayner & Duffy, 1986; Rayner & Duffy, 1987; Duffy, Morris, & Rayner, 1988), ambiguous words are processed more slowly than unambiguous words. In other words, when semantic decisions (SD) (decisions on word meaning) are required, words with multiple meanings pose more difficulty than words with single meanings. This ambiguity paradox was illustrated in a single experiment in which participants first made a lexical decision, and then had to make a relatedness judgement on a subsequently presented word (Piercey & Joordens, 2000). In this study, participants showed an ambiguity advantage for lexical decision, and an ambiguity disadvantage on the subsequent relatedness decision. The importance of the ambiguity paradox lies in the fact that it leads directly to the question of how words are represented in the brain, and how we get access to these words. Any model of language will have to account for the ambiguity paradox if it is to be successful.

Previous models of the ambiguity advantage in LD, however, have shown mixed results. For example, Joordens & Besner (1994) trained a two layer Hopfield network consisting of 125 binary nodes (75 perceptual nodes and 50 conceptual nodes; activations of either +1 or -1). The perceptual nodes represented perceptual features and were never updated during retrieval (that is, they were clamped to a specific pattern). The conceptual nodes represented semantics, and the network was effectively fully connected. Learning was via a Hebbian learning algorithm.

They had two criteria for deciding if a PDP model could successfully account for the ambiguity effect; (a) the network had to retrieve one of the semantic patterns associated with the ambiguous words, and (b) the network had to retrieve ambiguous words faster than unambiguous words. Joordens and Besner (1994) were able to produce an ambiguity advantage within the *conceptual* nodes of their network when it into a stable pattern. This only occurred, however, when the network was relatively small and when the ambiguous meanings had equal probability. Most of the time (over 50% of the trials), their networks failed to settle into a correct pattern and formed a “blend” of the two learned meanings of the words over the conceptual units. Their initial conclusion from these simulations was that distributed models trained with Hebbian learning rule may not be suitable for capturing ambiguity effect.

In a different computational model, Kawamoto, Farrar & Kello (1994) trained a recurrent neural network with the Least Mean Square learning algorithm. Their model contained both “spelling” nodes and “meaning” nodes using a distributed representational coding scheme. During recall, the “spelling” nodes were given environmental activation, and the network was allowed to settle into a stable state. They found that they could produce an ambiguity advantage within the units representing “spelling”, but showed the opposite effect in units representing “meaning” (an ambiguity disadvantage in semantics?). It has been suggested, however, that Kawamoto et al.’s (1994) network

also settled into blend states in the meaning units (Kello, 2003, *Personal Communication*).

Although both of these models produced an ambiguity advantage (albeit in different processing regions), the networks failed to differentiate between the ambiguous items and produced blended representations. However, in later commentaries (Masson & Borowsky, 1995; Rueckl, 1995; Besner & Joordens, 1995), it was concluded that it may be possible for lexical decisions to be made prior to the network settling into these blend states. In other words, correct lexical decisions could be based on the “blend” states for ambiguous words resulting in a greater feeling of familiarity which could then be used to produce LD.

Using the model of Joordens and Besner (1994) as a basis for their theory, Piercey and Joordens (2000) developed the “efficient then inefficient” hypothesis for the processing of ambiguous words. They concluded that a lexical decision is made based on early processing and that a blend state (i.e., when all meanings of a word are simultaneously activated) produces an advantage for lexical decision but a disadvantage for the relatedness decision. That is, lexical decisions are made based on a feeling of familiarity that occurs during the early stages of processing, before a complete representation of the current item forms (i.e., efficient processing). Therefore, these decisions could be made regardless of an eventual blend state. However, when the participants need to determine which meaning of the word is appropriate to a particular context, processing slows down. The participant continues to process the ambiguous word and each of the word’s meanings compete with each other. That is, the participant needs to leave the blend state and choose a meaning for the item so that further semantic processing can occur. This disambiguation of the blend state is an inefficient process that unambiguous words do not share. It should be noted that this theory is based specifically on the fact that the model of Joordens and Besner (1994) produced blended states for ambiguous words.

In this paper, we readdress the ambiguity advantage for lexical decision using a computational model that is able to learn multiple mappings for a single input. These models do not produce blend states; therefore, if the ambiguity advantage can be reproduced, then the notion of blend states existing should be questioned.

Symmetric Diffusion Networks

Symmetric Diffusion Networks (SDNs) are a class of computational models based upon the principles of continuous, stochastic, adaptive, and interactive processing (Movellan & McClelland, 1993). From a computational perspective, SDNs can be viewed as a continuous version of the Boltzmann machine; that is, time is intrinsic to the dynamics of the network. Furthermore, SDNs embody Bayesian principles in that they develop internal representations based upon the statistics of the environment. One of the main advantages of SDNs is that they are able to

learn multiple mappings for a single concept, something previous models often have difficulties with. In other words, SDNs are able to learn ambiguous mappings.

Recent work (Medler & McClelland, 2001) has shown that when biologically inspired constraints (i.e. activations within the range [0,1], positive between layer projections, lateral inhibition) are applied to SDN's, their effective performance is increased substantially in terms of the number of patterns they can be trained on, the rate at which patterns are learned, and their ability to separate out independent sources in an unsupervised manner.

Network Dynamics and Learning

Network dynamics are based upon continuous activations that develop over time, and are governed by the following equation:

$$\Delta a_i(t) = \Delta t [net_i(t) - \hat{n}e\hat{t}_i(t)] + \sigma \cdot \sqrt{\Delta t} Z_i(t) \quad \text{Eq. 1}$$

where,

is the summed activation of all the activities coming into the unit—including its bias—passed through a squashing function

$$net_i = h\left(\sum_{j=1}^n a_j w_{ij}\right)$$

such as the logistic, $h(u) = 1 - \exp(-u)$, and

$$\begin{aligned} \hat{n}e\hat{t}_i &= 1/g_i \cdot f(a_i) \\ &= 1/g_i \cdot \log[(a_i - \min)/(\max - a_i)] \end{aligned}$$

represents the net input required to maintain an activation value of a_i . Here we use the inverse logistic, where min and max are the minimum and maximum activation bounds respectively. g_i is a gain function, and $Z_i(t)$ is the standard Gaussian variable with zero mean and unit variance. The last term in the equation adds stochasticity to the network, which allows it to learn multiple meanings for a single input.

SDNs are trained with the *Contrastive Hebbian Learning* (CHL) algorithm, which performs both supervised and unsupervised learning depending on the environmental inputs to the network. Basically, learning occurs by presenting a pattern to the network and letting it settle for a set number of cycles. During this positive phase, co-occurrence statistics are computed for all the units. A negative phase then follows where the pattern is removed, the network is allowed to re-settle, and co-occurrence statistics are collected once again. Weights are then adjusted using the difference between the negative phase statistics and the positive phase statistics.

$$\Delta w_{ij} = \varepsilon \left[\sum (a_i^+ a_j^+) - \sum (a_i^- a_j^-) \right] \quad \text{Eq. 2}$$

In essence, the CHL algorithm makes weight adjustments based upon subtracting out the statistics of the base activity of the network (negative phase) from the statistics of the environment plus base activity (positive phase). Weight adjustments in this model were computed after each pattern presentation, as opposed to batch learning

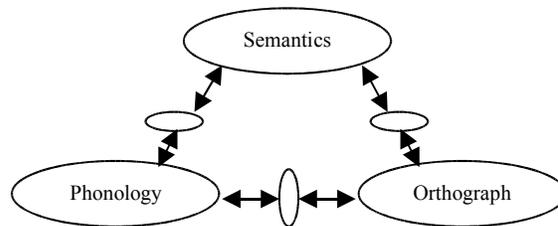


Figure 1. Network architecture showing the three main processing layers and connecting hidden layers.

which adjusts weights only after all patterns have been presented (Movellan & McClelland, 1993).

Network Architecture, Stimuli, & Training

In keeping with previous models of language (e.g., Harm & Seidenberg, 1999), the network consisted of three main processing layers: an “orthographic”, a “phonological”, and a “semantic processing” layer. To capture the gross relationship between semantics and the orthographic and phonology representation of words, there were twice as many units (10) in the semantic layer as in the orthographic and the phonology layers (5 units each). Each layer was connected to the other via a set of hidden layers (5 units). Between layer connections were excitatory, while within layer connections were inhibitory (Medler & McClelland, 2001).

Stimuli were arbitrary, distributed binary patterns [0,1] that encoded the orthography, phonology, and semantics of a given “word”. It is recognized that the abstract, distributed codes used in this simulation are not true representations of semantics, phonology, and orthography; however, future simulations using the same architecture will use more systematic encodings for these representations. Half of the training patterns (20) were unambiguous words, and half (20) were ambiguous words. In this model, only semantics had ambiguous patterns (as opposed to ambiguous orthography or phonology). Hence, ambiguous words had two possible meanings, and were selected with either a 70/30 distribution or a 50/50 distribution. Two representational training patterns are shown in Table 2; the presentation probability is the likelihood of that specific pattern being selected during the positive phase. Nonwords were simply random patterns across the orthographic and phonology units that had not been previously trained¹.

During training, the orthography and phonology units were clamped on, and the semantic and hidden units were modified during the positive and negative phases. Following training, the network was able to correctly produce the probability structure of the training stimuli. That is, the network was able to successfully recall the semantic patterns with the same probabilities that it was trained on. The

¹ As previous results have suggested that non-word foils need to be word-like for the ambiguity advantage to be stable (Borowsky & Masson, 1996), and we are assessing LD over the semantic units, we clamped both the orthographic and phonology units for the non-words.

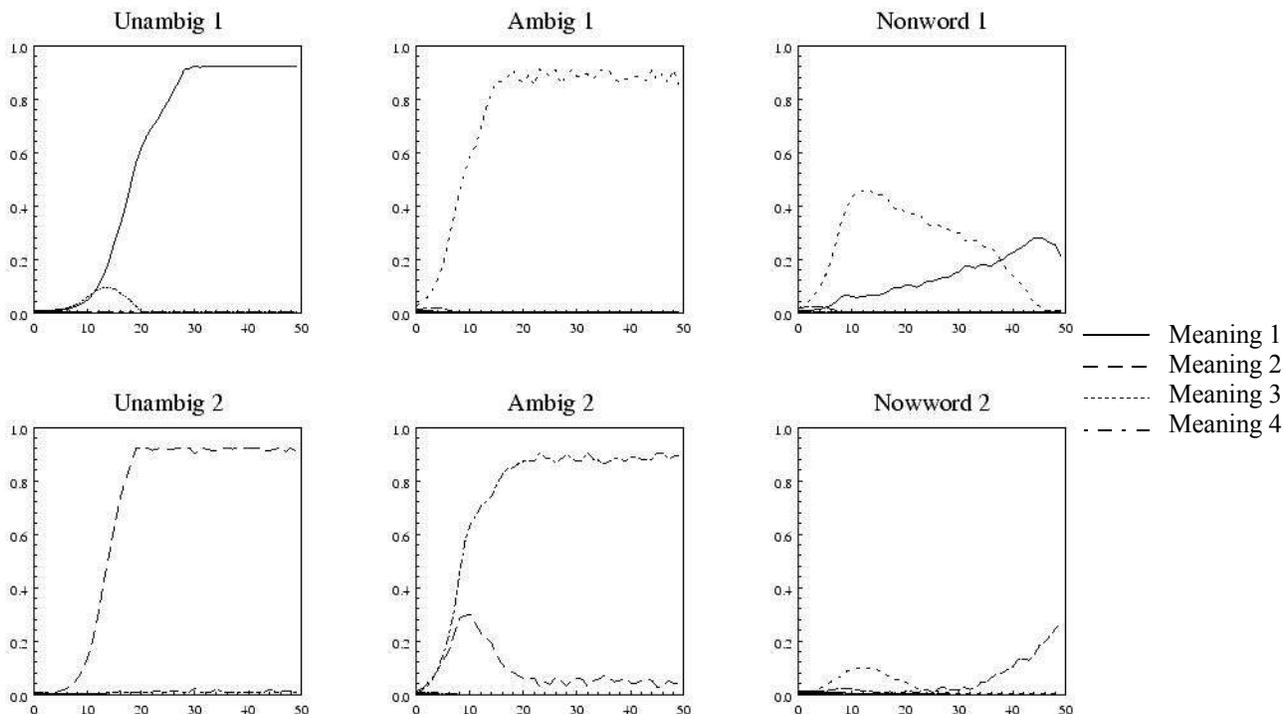


Figure 2. Sample differentiation scores for a subset of unambiguous, ambiguous, and non-words. Note that the first non-word is mistaken for a word at a criterion of 0.25.

network did not produce blend states for the ambiguous items.

Table 2. Sample Patterns Showing Positive and Negative Training Phases for Unambiguous and Ambiguous Words

Present. Prob.	Unambiguous		
	Orthography	Phonology	Semantics
+1.0	1 0 0 1 1	0 1 1 0 1	1 0 1 0 1 1 1 0 1 0
-1.0	1 0 0 1 1	0 1 1 0 1	* * * * * * * * * *
Ambiguous			
	Orthography	Phonology	Semantics
+0.7	0 1 1 1 0	1 1 0 0 0	0 1 1 0 0 1 1 0 0 1
+0.3	0 1 1 1 0	1 1 0 0 0	1 0 1 1 0 0 0 0 1 0
-1.0	0 1 1 1 0	1 1 0 0 0	* * * * * * * * * *

During testing, the orthography and phonology units were clamped on, and the semantic units were allowed to settle. Previous models waited for the networks to settle into a stable state, and took this measure as a reaction time. In our model, we assume a speeded decision based on a differentiation measure (McClelland & Chappell, 1998) computed over the known words, k :

$$diff_k = \prod_{i=1}^n |1 - P(T_i) - P(G_i)|$$

where G_i is the generated pattern, and T_i is a target. If a generated pattern does not match a target pattern, then the differentiated score should approach zero. A matched pattern, on the other hand, should produce a score that approaches one. If multiple patterns are partially activated (i.e., a blend), then several words should show a differentiation score that approaches a middle value.

When $diff_k$ exceeds a threshold (in this case, an arbitrary value of 0.25), a decision of “word” is made. If a word (or non-word) fails to reach the threshold within a certain time limit (an arbitrary point such as 20 time steps plus or minus some random time to introduce stochasticity in the response times), then a nonword decision is made. This nonword time limit can be adjusted to reflect task instructions (e.g., “respond as quickly as possible” vs. “respond as quickly and accurately as possible”). Consequently, we can produce both accuracy and reaction time measurements from our model.

Results

Figure 2 shows some representative differentiation scores for a sub-sample of the testing stimuli. As can be seen, no blends were formed (a single score tended towards one, whereas all other scores tended towards zero). Furthermore, the figure shows how using a threshold criterion of 0.25 for a speeded decision leads to the first non-word being misclassified as a word. Finally, it should also be noted that although the second ambiguous word looks like it is initially activating two word meanings (heading towards a blend

perhaps?), the second word meaning (i.e., the dashed line) is actually associated with the second unambiguous word.

In terms of reaction times, the network showed an ambiguity advantage. The network made a lexical decision for unambiguous items in an average of 13.0 time steps, whereas ambiguous items took 11.5 time steps. In contrast to previous empirical work (Piercey & Joordens, 2000), however, there was not a clear advantage of ambiguous items in terms of accuracy. Lexical decision for unambiguous items was approximately 97% correct, while ambiguous items were only 95% correct within the speeded decision.

One last note to make is that the final differentiation score for ambiguous items was often lower and more variable than for unambiguous words. The differentiation score averaged over the last ten time steps for the unambiguous items was 0.92 (var = 3.7×10^{-4}) whereas ambiguous items had an average differentiation score of 0.87 (var = 5.1×10^{-4}). This suggests that, for ambiguous items, the final settled state for the networks was more unstable than unambiguous items, and that if speeded decisions were not made, then the ambiguity advantage in reaction times may disappear.

Discussion

We have shown how a network trained with the *CHL* produces the ambiguity advantage over the semantic nodes based on speeded decision. Furthermore, the model was able to produce the approximate correct probability distributions of the training corpus, thereby avoiding “blend” states. Consequently, the theory of blend states having to exist to aid in LD for ambiguous items may have to be re-evaluated. Furthermore, the efficient-then-inefficient hypothesis of Piercey and Joordens (2000) may have to be recast.

The results from this network stimulation suggest an alternative theory as to why ambiguous items show an advantage for lexical decision. Given that there are multiple distinct attractor states in semantics for ambiguous items, and given a random start state, then the probability of starting near an attractor is greater for ambiguous items than unambiguous items. Consequently, if a decision is based on traveling toward an attractor basin, then ambiguous items should—on average—reach a basin sooner than unambiguous items. This is similar to the attractor basin theory proposed by Plaut and Booth (2000). Consequently, lexical decisions are efficient for ambiguous items because they have a higher probability of starting near an attractor basin.

Note that this theory would require lexical decisions to be made at the semantic level. That is, if LD could be completed at the orthographic level or at the phonological level (say by having non-words that either violated the orthographic rules or the phonological rules of English), then the ambiguity advantage would disappear (cf., Borowsky & Masson, 1996).

Interestingly, this theory would also explain the disadvantage seen for ambiguous items during semantic

decisions. If we assume that the network has settled into a stable state following the lexical decision (processing is automatic and continues even after the decision process), then both the unambiguous and ambiguous items will have activated a meaning in semantics. For unambiguous items, the semantic comparison would be relatively easy as there would only be one meaning to compare. For ambiguous items, however, the comparison becomes more unsettling. On some trials, the semantic decision would be relatively quick² as the network would be in the correct semantic attractor. On other trials, however, the network would be in an incorrect attractor, and would have to switch attractor states. Therefore, when trials are averaged, ambiguous items should show a disadvantage for semantic decisions. Consequently, semantic decisions are inefficient for ambiguous items because of the need to visit multiple attractor basins. Hence, this theory predicts that if we prime an ambiguous item towards one meaning or another, then the disadvantage should be lessened. Indeed, preliminary behavioral results show this to be the case (Piercey, Medler, & Hebert, 2003).

One area of potential criticism for the current model is that although it showed an ambiguity advantage for reaction times, it did not show an ambiguity advantage for accuracy. This discrepancy may be due to the choice of the differentiation score to evaluate network performance. This scoring mechanism assumes that the currently presented pattern is simultaneously compared to all learned words (thus, assuming that the learned patterns are stored somewhere exterior to the current model). Consequently, as unambiguous and ambiguous words are learned to criteria in the model, a decision based on the learned representations should show equal performance (where failure to recognize a word is based on a combination of the threshold criterion and the nonword decision time limit). One possible solution to this would be to use a different type of LD process, such as the harmony/referent model (Piercey, 2002; Joordens, Piercey, & Azarbeh, 2003) of lexical decision.

Future models will focus on training all processing levels (orthographic, phonological, and semantic) to address the theory of non-word background driving the ambiguity advantage in LD. As well, we will explicitly address the semantic relatedness decision issue to evaluate the theory predicted by the current simulations.

Reference List

- Besner, D., & Joordens, S. (1995). Wrestling with ambiguity--further reflections: Reply to Masson and Borowsky (1995) and Rueckl (1995). *Journal of Experimental Psychology: Learning Memory and Cognition*, 21, 515-519.

² It is unclear whether the decision would be as fast as the unambiguous trials, as simulation results show the final state for the ambiguous words to be less robust and more variable. Consequently, one might expect this variability to slightly slow decision processes.

- Borowsky, R., & Masson, M. E. J. (1996). Semantic ambiguity effects in word identification. *Journal of Experimental Psychology: Learning Memory and Cognition*, 22, 63-85.
- Duffy, S. A., Morris, R. K., & Rayner, K. (1988). Lexical ambiguity and fixation times in reading. *Journal of Memory & Language*, 27, 429-446.
- Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.
- Gernsbacher, M. A. (1984). Resolving 20 years of inconsistent interactions between lexical familiarity and orthography, concreteness, and polysemy. *Journal of Experimental Psychology: General*, 113, 256-281.
- Harm, M. W., & Seidenberg, M. S. (1999). Phonology, reading acquisition, and dyslexia: Insights from connectionist models. *Psychological Review*, 106, 491-528.
- Harm, M. W. & Seidenberg, M. S. Computing the Meanings of Words in Reading: Cooperative Division of Labor Between Visual and Phonological Processes. *Psychological Review*, (in press).
- Joordens, S., & Besner, D. (1994). When banking on meaning is not (yet) money in the bank: Explorations in connectionist modeling. *Journal of Experimental Psychology: Learning Memory and Cognition*, 20, 1051-1062.
- Joordens, S., Piercey, C. D., & Azarbeh, R. (2003). From word recognition to lexical decision: A random walk along the road of harmony. In *International Conference on Cognitive Modeling*.
- Kawamoto, A. H., Farrar, W. T., & Kello, C. T. (1994). When two meanings are better than one: Modeling the ambiguity advantage using a recurrent distributed network. *Journal of Experimental Psychology: Human Perception & Performance*, 20, 1233-1247.
- Masson, M. E. J., & Borowsky, R. (1995). Unsettling questions about semantic ambiguity in connectionist models: Comment on Joordens and Besner (1994). *Journal of Experimental Psychology: Learning Memory and Cognition*, 21, 509-514.
- McClelland, J. L., & Chappell, M. (1998). Familiarity breeds differentiation: A subjective-likelihood approach to the effects of experience in recognition memory. *Psychological Review*, 105, 724-760.
- Medler, D. A. & McClelland, J. L. (2001). Improving the performance of Symmetric Diffusion Networks via biologically inspired constraints. In K. Marko & P. Werbos (Eds.), *IJCNN'01: Proceedings of the INNS-IEEE International Joint Conference on Neural Networks* (pp. 400-405). Washington, DC: IEEE Press.
- Movellan, J. R., & McClelland, J. L. (1993). Learning continuous probability distributions with Symmetric Diffusion Networks. *Cognitive Science*, 17, 463-496.
- Piercey, C. D. (2002). *The Referent Model of Lexical Decision*. Unpublished doctoral dissertation, University of Alberta, Edmonton, Alberta, Canada,
- Piercey, C. D., & Joordens, S. (2000). Turning and advantage into a disadvantage: Ambiguity effects in lexical decision versus reading tasks. *Memory & Cognition*, 28, 657-666.
- Piercey, C. D., Medler, D. A., & Hebert, B. E. (Eds.). (2003). *Ambiguity Effects in Lexical Access: Do Blends Exist?* (Vol. 8).
- Plaut, D. C., & Booth, J. R. (2000). Individual and developmental differences in semantic priming: Empirical and computational support for a single-mechanism account of lexical processing. *Psychological Review*, 107, 786-823.
- 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., & Duffy, S. A. (1986). Lexical complexity and fixation times in reading: Effects of word frequency, verb complexity, and lexical ambiguity. *Memory & Cognition*, 14, 191-201.
- Rayner, K. & Duffy, S. A. (1987). Eye movements and lexical ambiguity. In J.K.O'Regan & A. Levy-Schoen (Eds.), (pp. 521-529). Amsterdam: North-Holland: Elsevier Science Publications.
- Rueckl, J. G. (1995). Ambiguity and connectionist networks: Still settling into a solution: Comment on Joordens and Besner (1994). *Journal of Experimental Psychology: Learning Memory and Cognition*, 21, 501-508.
- Seidenberg, M. S., & McClelland, J. L. (1989). A distributed, developmental model of word recognition and naming. *Psychological Review*, 96, 523-568.