For Better or Worse: Modelling Effects of Semantic Ambiguity

Jennifer Rodd (jrodd@csl.psychol.cam.ac.uk)
Centre for Speech and Language, Department of Experimental Psychology
Cambridge University
Cambridge, UK
Gareth Gaskell (g.gaskell@psych.york.ac.uk)
Department of Psychology
University of York, York, UK
William Marslen-Wilson (william.marslen-wilson@mrc-cbu.cam.ac.uk)
MRC Cognition and Brain Sciences Unit
15 Chaucer Road, Cambridge, UK

Abstract

Several studies have reported an advantage in lexical decision for words with multiple meanings. More recently, Rodd, Gaskell, and Marslen-Wilson (in press) have reported a more complex pattern of ambiguity effects. While there is a processing advantage for words that have many highly related word senses (e.g., *twist*), there is a disadvantage for words that have more than one meaning (e.g., *bark*). Here we show that these two apparently opposite effects of ambiguity can both emerge from the competition to activate a coherent semantic representation in an attractor network. Ambiguity between unrelated meanings delays recognition because of interference between the two possible stable patterns of semantic activation, that correspond to separate attractor basins. In contrast, the patterns of semantic activation that correspond to different senses of the same word meaning all lie within a single attractor basin, and the semantic flexibility associated with these words results in a widening of the attractor basin, thus produces a processing advantage relative to unambiguous words.

The Ambiguity Disadvantage and Sense Benefit

Models of word recognition often make the simplifying assumption that each word in the language has a single, well-defined meaning. However, many words refer to more than one concept. For example, *bark* can refer either to a part of a tree, or to the sound made by a dog. Other words, such as *twist*, have a range of systematically related dictionary definitions including to *make into a coil or spiral*, *to operate by turning*, *to alter the shape of*, *to misconstrue the meaning of*, *to wrench or sprain*, and *to squirm or writhe*. To understand such words, we select the appropriate interpretation, normally on the basis of the context in which the word occurs. In this paper we review the literature on how semantic ambiguity affects the recognition of single words, and report a series of network simulations that examine the implications of these results for models of word recognition

Several studies in the literature report faster lexical decision times for ambiguous words, compared with unambiguous words (Azuma & Van Orden, 1997; Borowsky & Masson, 1996; Millis & Button, 1989). There have been various explanations for why it might be easier to recognise words with multiple meanings. Typically it is assumed that ambiguous words benefit from having more than one competitor in the race for recognition. More recently, this view that there is a simple advantage for semantic ambiguity has been challenged. Rodd et al. (in press) argue that a distinction should be made between the accidental ambiguity of words like *bark* which, by chance, have two unrelated meanings, and the systematic ambiguity of words that have multiple senses. For example, although there are important differences between what it means to *twist an ankle* compared with to *twist the truth*, these different senses of the word *twist* are closely related to each other, both etymologically and semantically. This relationship is quite unlike the ambiguity for a word like *bark*.

All standard dictionaries respect this distinction between word meanings and word senses; lexicographers routinely decide whether different usages of the same spelling should correspond to different lexical entries or different senses within a single entry. However, although this distinction appears easy to formulate, people will sometimes disagree about whether two usages of a word are sufficiently related that they should be taken as senses of a single meaning rather than different meanings. However, even if there is not always clear distinction between these two different types of ambiguity, it is important to remember that words that are described as ambiguous can vary on a continuum between these two extremes.

Rodd et al. (in press) support the psychological importance of this distinction in a set of lexical decision experiments which show that while multiple related word senses do produce a processing advantage, multiple unrelated meanings delay recognition. Here we report a series of simulations which investigate whether these two apparently opposite effects of ambiguity can both emerge from the competition to produce a coherent distributed semantic representation within an attractor network.

Semantic Competition Models of the Ambiguity Advantage

Joordens and Besner (1994) and Borowsky and Masson (1996) have tried to model effects of ambiguity using a two-layer Hopfield network (Hopfield, 1982) to learn the mapping between orthography and semantics. The models show an advantage for words that are ambiguous between unrelated meanings. The authors argue that this advantage arises because, when the orthography of a word is presented to the network, the initial state of the semantic units is randomly determined. The network must move from this state to a valid finishing state corresponding to the meaning of the word. For ambiguous words there are multiple valid finishing state, and on average, the initial state of the network will be closer to one of these states than for an unambiguous word, where there is only one valid finishing state. However, as discussed above, it is now apparent that ambiguity between unrelated meanings produces an disadvantage, so that there is a discrepancy between the data and the behaviour of these models.

One limitation of these models is that their performance on the task was surprisingly poor. Joordens and Besner (1994) report an error rate of 74%. These errors often result from the network settling into blend states, which are a mixture of the word's meanings. Gaskell and Marslen-Wilson (1999) have shown that blends between unrelated semantic representations can be relatively meaningless, and may be closer to a different word in the lexicon than to either of the components of the blend. In the Borowsky and Masson (1996) study, these blend states are not considered to be errors; the authors argue that to perform lexical decision it is not necessary to resolve the ambiguity successfully in order for there to be sufficient familiarity to make a successful lexical decision. Although this approach may be appropriate for modelling the specific task of lexical decision, this would severely limit the model in being extended to be a more general model of word recognition. It is the case that, given an ambiguous word in isolation, we are able to retrieve one of its meanings. In contrast, the model would predict that without a contextual bias to direct us to one of the meanings we would get stuck in a blend state that may be quite unlike either of the meanings.

It is possible that the observed ambiguity advantage may be an artefact of this tendency to settle into blend states. Indeed, Joordens and Besner (1994) report that as the size of their network is increased, and performance improves, the ambiguity advantage is eliminated. However, even in these larger networks, the problem of blend states is still present; Joordens and Besner (1994), report a maximum performance level of 48.8% for ambiguous words. In the following simulation, we attempt to improve the overall performance of the network, and investigate how ambiguity affects performance in a network that is able to successfully retrieve the meanings of ambiguous words.

Simulation 1: The Ambiguity Disadvantage

Introduction

While Hopfield networks are known to have limited capacity, the networks discussed above are performing well below the theoretical capacity limit. Hopfield (1982, pg 2556) stated that *"About 0.15 N states can be simultaneously remembered before error in recall is severe"*, where N is the number of units in the network. Therefore the Joordens and Besner (1994) network should be able to learn 45 patterns, and yet the network cannot reliably learn 4 words. This poor performance is because the patterns corresponding to the different meanings of ambiguous words share the orthographic part of their pattern. Hopfield (1982) noted that these networks have a particular difficulty with correlated patterns. Therefore, the simple Hebbian learning rule, which captures the correlational structure of the training set, may not be suitable for learning ambiguous words.

Simulation 1 uses instead the least mean-square error-correcting learning algorithm, which adjusts the weights between units to reduce any error in the activation patterns produced by the current sets of weights. This may therefore alleviate the problem of blend states, as the learning algorithm will change the weights such that these states are not stable.[1]

Method

Network Architecture The network has 300 units: 100 (orthographic) input units and 200 (semantic) output units. The network is fully connected; each unit is connected to all other units. All units are bipolar; they are either on $[+1]$ or off $[-1]$.

Learning Algorithm All connection strengths were initially set to 0. During each learning trial, the network was presented with a single training pattern, and an error-correcting learning algorithm was used to change the connection strengths. The change in connection strength from a given unit i to a unit j is given by

$$\Delta w_{ij} = x_i(x_j - \sum_k w_{kj}x_k)/3n, \quad i \neq j, \qquad (1)$$

where x_i is the activation of unit i and n is the number of units in the network. The learning rate parameter, $1/3$, was selected to provide good performance after a relatively small amount of training.

Training Unambiguous word representations were created by randomly assigning values of $+1$ and -1 to each of the 100 input and 200 output units, such that half the units in each part of the network were assigned $+1$, and the other half -1. For the ambiguous words, a single, randomly generated input pattern was paired with two different output patterns.

In the Joordens and Besner (1994) simulations, the network was trained on only one unambiguous word and one ambiguous word. Here the training set varies between 1 and 16 pairs of ambiguous and unambiguous words, (i.e., 2 to 32 words or 3 to 48 unique semantic patterns). The number of times that each word was presented to the network was varied between 2 and 64 times. The unambiguous and ambiguous words were matched for overall frequency of the orthographic pattern. For each combination of training set size and length of training, the network was trained, and its performance was tested on 200 independent passes; for each pass, a different, independently generated set of training items was used.

Testing Each input pattern was presented to the network, and the output units were all set randomly to $[+1]$ or $[-1]$. Retrieval of the semantic patterns was the result of an asynchronous updating procedure. A unit was selected at random, and its activation was updated by summing the weighted input to that unit. If this input was greater than zero, then the unit was set to $+1$, otherwise the unit was set to -1. This updating continued until a sequence of 1500 updates produced no change in the state of any unit. The network was considered to have settled correctly only if the activation of all its units was correct when it reached a stable state.

Results

For the unambiguous words, the network settled into the correct semantic pattern for over 99.8% of the words, for all the levels of training and set sizes. For the ambiguous words, performance was more variable. The percentage of trials on which the network settled into a correct training pattern for these words is shown in Figure 1 for different amounts of training, and for different sizes of the training set. Importantly, under some conditions, the network was able to settle

[1] Kawamoto, Farrar, and Kello (1994) used this algorithm to learn ambiguous words, but they do not report error rates.

correctly into the semantic pattern corresponding to one of the word's two meanings on 98% of the trials. Therefore, the LMS error-correcting algorithm performed substantially better than the Hopfield algorithm on this task.

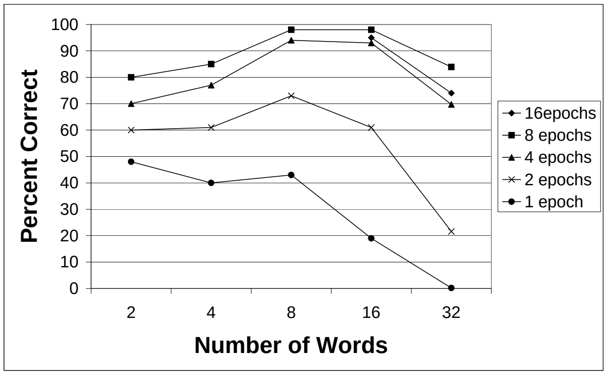

Figure 1: Simulation 1, Performance for Ambiguous Words

Despite this improvement, the ambiguous words were still difficult to learn, compared with the unambiguous words. For the unambiguous words, the network always reached near-perfect performance, having been presented with the training set only once. For the ambiguous words, only when the training set had been presented to the network four times, did performance ever rise above 90%. The number of cycles taken by the network to settle was also generally greater for the ambiguous words than for the unambiguous words. Table 1 shows the difference between the settling times for the two types of words; positive numbers indicate faster settling for the unambiguous words. For the smallest training-set size, the difference between the two types of words was small and variable, but for larger training sets, a consistent ambiguity disadvantage emerges. Crucially, for all the networks where performance on the ambiguous words was greater than 90%, there was a significant ambiguity disadvantage (all significant using the Bonferroni correction for multiple comparisons).

Table 1: Simulation 1, Percentage Benefit in Settling Times for Unambiguous Words

	2	4	8	16	32
Training	Words	Words	Words	Words	Words
2	-1	4	38	87	-
4	0	6	23	65	105
8	-1	3	16*	41*	94
16	2	5	11*	25*	64
32	-2	3	12*	32*	57
64	-1	4	11*	32*	49

Notes. * performance on ambiguous words exceeded 90%,

The change in the performance as a function of the size of the training set was somewhat surprising. At all levels of training, the network settled more quickly when it had been trained on fewer patterns. However, the effect of training-set size on error rates for the ambiguous words is more complex (see Figure 1). It is not altogether clear why the network performs so poorly for a very small training sets. It is possible that the increased error produced by the other words in the training set results in the error-correcting learning algorithm operating more effectively; alternatively, the number of spurious stable attractors may increase for small training sets because of the small number of learned attractor basins.

There is an interesting effect of training on performance: initially, training improves performance, in terms of both error rates, and settling times. However, for some training-set sizes, performance reduces if the training set is presented more than 16 times. This suggests that over-learning of the training set produces poor performance for the test items (in which only a subset of the training features are activated).

Discussion

This simulation shows that the introduction of an error-correcting learning algorithm improved performance on the ambiguous words to a level where it is reasonable to investigate the effects of ambiguity on performance. In all conditions where performance exceeded 90%, there was a significant *disadvantage* for the ambiguous words in terms of the number of cycles taken for the network to settle.

Therefore, this simulation suggests that the ambiguity advantage found by Joordens and Besner (1994) is atypical for a network of this type. When performance is improved such that the network reliably settles into a stable semantic representation that corresponds to one of the word's meanings, the interference between the multiple patterns of ambiguous words delays their recognition, relative to unambiguous words. Therefore, a simple semantic competition network of this type can simulate the ambiguity disadvantage seen by Rodd et al. (in press). The question that remains is whether this type of network can also produce the benefit for words with multiple, related word senses.

Simulation 2: Word Senses as Random Noise

Introduction

We have now shown that semantic competition between word meanings delays the settling of the network for ambiguous words, relative to unambiguous words. How then are we to explain the advantage reported by Rodd et al. (in press) for words with multiple senses? One difference between these two forms of ambiguity is the degree of semantic overlap between the alternative semantic patterns. However, although an increase in the similarity of the two meanings of an ambiguous words may reduce the level of semantic competition (and therefore the ambiguity disadvantage), this can only improve performance to the level of the unambiguous words; it cannot produce a benefit.[2]

In this simulation, we explore the hypothesis that the variation in the meanings of words such as *twist* and *flash*, which are listed as having many word senses, should be viewed not in terms of ambiguity, but in terms of flexibility. We assume that the multiple senses of these words are not distinct, but that their meaning is flexible or vague, such that it has a slightly different interpretation in different contexts. In particular, we assume that these words can be represented as having a single base pattern that represents the core meaning of the word. Then, every time this pattern is presented to the

[2] This has been confirmed in a set of simulations identical to Simulation 1 except that the semantic relationship between the meanings of the words was systematically varied (Rodd, 2000).

network, random noise is added to this base pattern, such that each time the network sees the word, it is slightly different from other instances of the word.

Although this idea that words with many senses should be characterized as words whose meanings are flexible about a core meaning does not reflect how these words are listed in dictionaries, there is support for this idea that the classification of the meanings of such words into distinct senses is artificial. For example, Sowa (1993) states that “for polysemous words, different dictionaries usually list different numbers of meanings, with each meaning blurring into the next”.

The reason that we might expect this characterization of word senses to produce the processing benefit seen in the human data is that, as we saw in Simulation 1, if an identical pattern is repeatedly presented to the network, it can develop a very deep attractor basin that can be difficult for the network to settle into when it is given only the orthographic input. It is possible that adding a small amount of noise to the network might prevent this over-learning, and might allow the network to develop broader attractor basins, that are easier for the network to enter.

Method

Network Architecture, Learning Algorithm and Processing The architecture and learning algorithm used in this simulation were identical to those used in Simulation 1. However, to reduce the length and variability of the settling times, a different updating procedure was used. Updating now consisted of a series of update sequences in which all the semantic units were updated once in random order.

Training The networks were each trained on 64 words. Half these words were unambiguous, and were presented to the network in exactly the same form on each presentation. The other words had noise added to them; each time these words were presented to the network, a small number of the semantic units were randomly changed from the original base pattern. The number of units that were changed varied from 1 to 5 across different simulations. The number of times that these words were presented to the network was varied from 16 to 128. For each level of training and noise, 100 networks were trained on independently generated sets of patterns.

Results

For the unambiguous words, the network settled correctly in over 99.5% of trials, in all conditions. For the words that had noise added to them, it is less clear what it means for the network to settle correctly; as the level of noise increased, the percentage of trials on which the network settled into the base pattern decreased. However, those trials on which the network did not settle into the base pattern should not all be considered as errors. If the network settles into a pattern that does not differ from the base pattern by more than the amount of noise that was added to the patterns during training, this can be thought of as the network settling into one of the word’s senses rather than the core meaning, and should not be considered to be an error. Using this approach, the percentage correct for these words was always above 99.5%

Table 2 shows settling times for the unambiguous words and the words with the added noise, and the differences between these scores. Positive numbers reflect a disadvantage for the noisy patterns. These data show complex interactions between the effects of noise and training, but crucially, while low levels of noise have no stable influence on performance, as the level of noise increases, a reliable disadvantage for noise emerges. This disadvantage for noise is greatest at low levels of training, and increases with the level of noise.

Table 2: Simulation 2, Cycles to Settle for Unambiguous and Noisy Words

Units		Training Presentations			
Changed		16	32	64	128
1	Unambiguous	603	617	615	588
1	Noisy	611	622	614	593
1	Difference	+8	+5	-1	+5
3	Unambiguous	587	598	564	515
3	Noisy	617	614	583	539
3	Difference	+30	+16	+19	+24
5	Unambiguous	578	573	536	481
5	Noisy	623	609	568	509
5	Difference	+45	+36	+32	+28

Discussion

Contrary to the idea that noise during training might improve performance, the network was slower to settle into those training patterns that had noise added to them during training, compared with the unambiguous patterns. Therefore, this simulation suggests that even if we characterize the ambiguity between multiple senses as being noise about a base pattern, the ambiguity still produces a processing disadvantage. This is, of course, the reverse of the pattern seen in the human data.

In this simulation, the noise that was added to the semantic representations was random; on each training trial, the activation of a given number of units in the semantic pattern was changed. However, this is not a realistic characterization of how the senses of words differ; it is not the case that a new sense of a word can be created from the core meaning of the word by simply changing arbitrary features. Rather, it is that case that these words have sets of possible semantic features which are sometimes, but not always, present. Therefore, rather than modelling word senses as the addition of random noise, it might be better to assume that each word has a range of possible semantic features, but that not all these features are always turned on. For example, the word *twist* may in some contexts not activate the features relating to pain, but it will never arbitrarily gain a feature such as *has legs*.

To model word senses in this way, we need to move away from semantic representations in which half the units are set to $+1$ and the other half set to -1. Instead, for any given semantic representation, most of the units will be turned off, and only a subset will be turned on. Noise is added such that only those features that should be turned on may be turned off, but there is no arbitrary addition of semantic features.

Simulation 3: Sparse Representations

Method

Network Architecture, Learning Algorithm and Processing The architecture used in this simulation was identical to that used in Simulations 1 and 2. The training patterns used were sparse, such that only 10% of the units were set to $+1$ and the remainder were set to 0. The change in connection strength from a given unit i to a unit j is given by

$$\Delta w_{ij} = 5x_i(x_j - \sum_k w_{kj}x_k)/n, \quad i \neq j \tag{2}$$

Training As in Simulation 2, the network was trained on 64 words; half of the words had noise added to the semantic representations during training, and the other half did not. Again, the number of units that were changed for the noisy words varied from 1 to 5; however noise was added only to units that were set to $+1$ in the base pattern.

Results

Figure 2 shows the performance of the network at different levels of training and noise.[3] At all levels of noise, performance is better for the words that had noise added during training; the network is able to correctly produce the semantic representations for these words at lower levels of training.

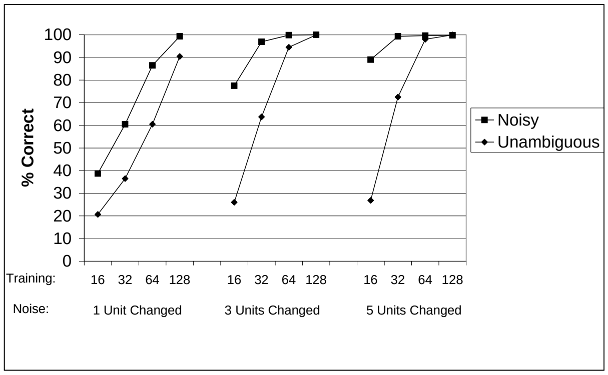

Figure 2: Simulation 3, Error rates

We then looked in detail at the settling behaviour of the network, which was presented with each word 128 times, with a level of noise of 5 units. This network successfully retrieved the meaning in over 99.7% of trials for both types of words. This network settled significantly more quickly for the unambiguous words than for the noisy words; the unambiguous words took on average 407 updates before they were stable, the noisy words took 435 ($t(99) = 8.9, p < .001$). However, a more interesting picture emerges if we look at how the activation of the semantic representations built up over time for this network. Figure 3 shows the total number of semantic units that are switched on at the end of each update of the 200 semantic units. If the network activates 20 units, this corresponds to the activation of a complete semantic pattern. For the noisy words, however, the network tends to activate only a subset of the 20 units; this corresponds to the activation of a sense of the word that does not contain all the possible semantic features for that word.

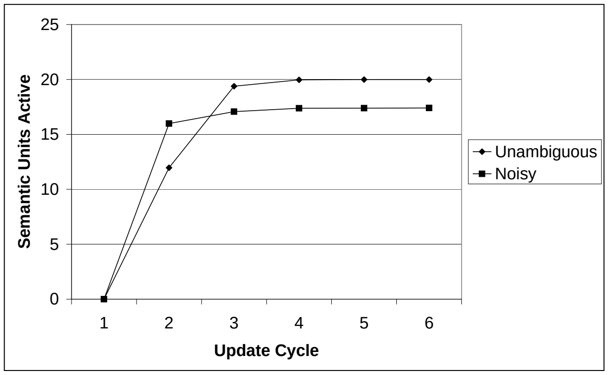

Figure 3: Simulation 3, Activation of Semantic Units

Interestingly, at the end of the first update cycle, the network is significantly more active for the noisy words ($p < .001$). For the unambiguous words, on average 12 units are switched on; for the noisy words, 16 are activated. Therefore, if we assume that lexical decisions are made before the activation of the semantic units has become completely stable, there will be an advantage for the noisy words. It is worth noting that, if this network is presented with a novel word that was not in the training set, the activation of the semantic units very rarely rises above 10. If an activation threshold were set at this level, there would be an advantage for the noisy words.

The later advantage for unambiguous words reflects an assumption built into the training set that the total number of semantic features that are ever activated for words with many senses is equivalent to the total number of features for the unambiguous words. In other words, we have assumed that the individual senses of words with many senses have fewer semantic features than those with only a single sense. This assumption is probably incorrect; it is more likely that words with many senses have a larger set of possible semantic features than words with few senses. It may have been more realistic to assume that the groups of words should be equated on the average number of features that are activated for each individual sense. If this had been the case then the two types of words would settle to the same mean activation level, and the noise advantage would be larger, and extend later in the settling of the network.

Discussion

This simulation shows that if the activation of the semantic features is used as a metric of lexical decision, then there is an advantage for words to which noise is added during training. The advantage is only present early in the settling of the network. This suggests that, as predicted, the noise acts to ensure that the attractor basins are sufficiently wide to allow the activation of the networks to enter the basin quickly. Later in the processing, however, there is a disadvantage for these words; this may be because of competition between multiple stable states (within the large attractor) that correspond to the different senses of the words.[4]

[3] Unambiguous words are considered to have settled correctly if they settled into the exact training pattern. Noisy words are considered to have settled correctly if they do not differ from their base pattern by more than the amount of noise that was added during training. In a separate analysis, not reported here, this tolerance was also used for the unambiguous patterns; In this analysis, no the error rates differed from those reported here by more than 0.2%.

[4] Additional simulations, not reported here, show that the low error rates for ambiguous words and ambiguity disadvantage seen in Simulation 1 is also seen in the rate of activation of semantic units when these sparse representations are used (Rodd, 2000).

Conclusions

The simulations reported here show that networks using the same architecture and learning rule can accommodate the two apparently opposite effects of semantic ambiguity reported by Rodd et al. (in press). While the semantic competition associated with the ambiguity between unrelated meanings delays recognition, the flexibility around the base pattern seen in words with many senses can produce a benefit.

The ambiguity disadvantage shown in Simulation 1 is important because previous simulations of ambiguity effects using networks of this type have shown an ambiguity advantage. We argue that these earlier results were atypical, and relied on using networks that were not able to disambiguate between the different meanings of ambiguous words. Simulations 2 and 3 show that a network of this type can also show a benefit for words whose meanings are flexible between different word senses, but only when their semantic features vary within a limited set of possible features. This limitation fits in with our intuitions about how the semantic representations of words senses vary.

These contrasting effects of ambiguity can best be viewed in terms of the attractor structure of the network. The delay in activating the meaning of an ambiguous word is due to competition between the two stable attractors that correspond to the two different meanings of the word. The initial state of the semantic units produced by the orthographic input will correspond to an unstable blend of the two meanings; the attractor structure of the network will then move the activation of the units away from this blend state towards one of the stable attractors. This disambiguation process takes time, and is responsible for the observed ambiguity disadvantage. In contrast, the different senses of a words all lie within a single attractor basin. Further, the semantic flexibility associated with these words results in a widening of the attractor basin, thus producing a processing advantage relative to unambiguous words. There may, however, be a disadvantage for these words later in processing, due to the existence of multiple stable attractors within the large basin that corresponds to the set of different senses of the word.

In summary, these simulations show that it is possible that the pattern of ambiguity effects reported by Rodd et al. (in press) can be explained in terms of the effects of these two types of ambiguity on the competition to activate a coherent semantic representation within an attractor network. The next stage is to determine whether these explanations are correct.

First, we have assumed that words with many senses should be characterised as words whose meanings are flexible about a core meaning; this assumption must be validated on the basis of detailed analysis of the stimuli used in the experiments. Second, it needs to be confirmed that flexibility is the key property responsible for the sense benefit. As noted by Rodd et al. (in press), words with many senses differ from words with few senses on a range of dimensions, including semantic richness and contextual predictability.

Finally, these simulations investigate two extreme cases of ambiguity; we have compared words with two completely unrelated meanings, with words whose different senses correspond to all the possible combinations of a set of permitted features. This is clearly unrealistic - most words with multiple senses do have some level of structure, and the variation in word senses is often systematic across words. Although the simulations reported here demonstrate important principles about how extreme forms of ambiguity can affect processing, further work needs to be done using more realistic semantic representations. These issues are important if we are to fully understand the implications of ambiguity effects for theories about the representation and access of word meanings.

References

Azuma, T., & Van Orden, G. C. (1997). Why safe is better than fast: The relatedness of a word's meanings affects lexical decision times. *Journal of Memory and Language*, *36*, 484–504.

Borowsky, R., & Masson, M. E. J. (1996). Semantic ambiguity effects in word identification. *Journal of Experimental Psychology: Learning Memory and Cognition*, *22*, 63–85.

Gaskell, M. G., & Marslen-Wilson, W. D. (1999). Ambiguity, competition and blending in speech perception. *Cognitive Science*, *23*, 439–462.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America–Biological Sciences*, *79*(8), 2554–2558.

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.

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 and Performance*, *20*, 1233–1247.

Millis, M. L., & Button, S. B. (1989). The effect of polysemy on lexical decision time: now you see it, now you don't. *Memory & Cognition*, *17*, 141–147.

Rodd, J. M. (2000). *Semantic representation and lexical competition: Evidence from ambiguity.* Unpublished doctoral dissertation, University of Cambridge.

Rodd, J. M., Gaskell, M. G., & Marslen-Wilson, W. D. (in press). Making sense of semantic ambiguity: Semantic competition in lexical access. *Journal of Memory and Language*.

Sowa, J. F. (1993). Lexical structure and conceptual structures. In J. Pustejovsky (Ed.), *Semantics and the lexicon* (pp. 223–262). Dordrecht/Boston/London: Kluwer Academic Publishers.