Proceedings of the Fourth Neural Computation and Psychology Workshop, pp. 213-226. London: Springer-Verlag, 1997.

Modelling Lexical Decision Using Corpus Derived Semantic Representations in a Connectionist Network

John A. Bullinaria & Christopher C. Huckle
Department of Psychology, Birkbeck College
London, UK

Abstract

Connectionist models of the mapping from orthography or phonology to random binary semantic vectors allow the simulation of lexical decision with reaction times that show patterns of semantic and associative priming similar to those found experimentally with human subjects. The co-occurrence statistics of words in large corpora allow the generation of vectors whose distribution correlates with the perceived semantic relatedness of the words. Here we discuss the use of these more realistic corpus derived semantic representations in connectionist models of lexical decision. We find lexical decision priming that correlates with distances in the semantic vector space, but the reaction times are very noisy. Averages over many words and/or many networks are required for the relationships to become clear. The question of associative priming remains open.

1 Introduction

Connectionist models of many language processing tasks have now reached a level of sophistication whereby reasonably realistic distributed representations of semantics are required. Lexical decision (i.e. the task of deciding whether a given string of letters or phonemes is a real word) is widely used in psychological experiments to investigate the processes and representations employed in basic human language processing [1, 2, 3]. Of particular interest is the study of priming (i.e. the effect by which response is speeded through prior presentation of certain related words). The fact that we observe facilitation by semantically related words (e.g. 'jump' primes 'leap') suggests that the lexical decision process taps into some underlying semantic representation and that lexical decision experiments can be designed to explore these representations. However, priming is also found to be produced by words that are associated but not semantically related (e.g. 'pillar' primes 'society'). It is not yet clear if such associative priming needs to be explained by a different mechanism to semantic priming or if it is inherent in the properties of the same semantic representations.

Recently, connectionist models of the lexical decision process have been constructed that account for many aspects of semantic and associative priming [4,

5]. The semantic priming arises due to the overlap of distributed semantic vectors and the associative priming arises due to word co-occurrence during learning. However, these models have been based on hand-crafted and/or random binary semantic vectors. Other researchers [6] have derived semantic vectors from large text corpora and have suggested that the distances between words in this semantic vector space can account for the experimental priming results. In this paper we put these two approaches together and investigate the properties of connectionist lexical decision models based on corpus derived semantic representations. In particular, we question whether it is really possible to obtain useful results *without* considering the two approaches together.

2 Modelling Lexical Decision

Given the experimental evidence that semantics has an effect on lexical decision reaction times, it is natural to assume that the time taken to activate the appropriate semantic representation provides at least one factor in the lexical decision process. Within the conventional connectionist framework we model this by choosing (simplified) representations for the orthography/phonology and semantics of words and setting up a network to map between them. Plaut [4] chose to do this with a recurrent network trained with continuous back-propagation through time. Bullinaria [5] used a cascaded feed-forward network trained in a similar manner. Both approaches led to similar (though experimentally distinguishable) patterns of reaction times and priming. Here we shall adopt the Bullinaria [5] approach. The idea is that we explore these simple models first and only add in complicating factors as they are required by the experimental data.

For simplicity, we restrict ourselves to mono-syllabic words and represent phonology by having one unit for each possible onset, vowel and offset phoneme cluster. Each word then has three phonological input units activated. Since the phonology to semantics mapping in English is essentially random (ignoring morphological effects) it is not unreasonable to represent the semantics by random binary vectors with the interpretation that activated units correspond to the small number of relevant semantic micro-features [7]. The network will then require a sufficiently large layer of 'hidden units' in order to handle the random and non-linearly separable associations between this phonology and semantics.

Since we are aiming to model reaction times (RTs), it makes sense to think in terms of activation cascading through the network [8] as in recurrent networks rather than the typical one pass approach of standard feed-forward networks. To simulate this we discretize the time and at each time slice *t* take:

$$Out_i(t) = Sigmoid(Sum_i(t))$$

$$Sum_i(t) = Sum_i(t-1) + \lambda \sum_j w_{ij} Prev_j(t) - \lambda Sum_i(t-1)$$

with the output $Out_i(t)$ of each unit *i* the usual sigmoid of the sum of the inputs into that unit at that time. The sum of inputs $Sum_i(t)$ is given by the existing sum at time

t–1 plus the additional weight w_{ij} dependent contribution fed through from the activation $Prev_j(t)$ of the previous layer and a natural exponential decay of activation depending on some time scale λ.

There are now two broad approaches to training the network. The quick and easy way is to note that, as long as we have static inputs, the asymptotic state of the above equations reduce to:

$$Out_i(t_\infty) = Sigmoid(Sum_i(t_\infty)) \quad , \quad Sum_i(t_\infty) = \sum_j w_{ij} Prev_j(t_\infty)$$

which are the equations for a standard non-cascaded feed-forward network. It follows that, if we only require the network to produce correct outputs for individual words, we can simply train on this asymptotic state using a standard gradient descent algorithm (such as back-propagation). The resultant trained network can then be used in a cascaded fashion to extract the RTs. If, however, we want the network to respond efficiently to sequences of words, we need to train *during* the cascading process so that the network can *learn* to make quick transitions from one set of activations to another. The network can still be trained using a standard gradient descent procedure to modify the weights w_{ij} iteratively so that the output activation errors are reduced. However, for each input word, we now need to repeat this process over many time slices as the network settles into a stable state. If we present the training words in random order, and keep the time parameter λ and learning rate η sufficiently small that large fluctuations in the weights and activations do not occur, then the network eventually learns to produce the correct outputs for any input word without any resetting of the network after the previous word.

Reaction times can then be defined in terms of time slices in a number of ways. We could simply take the time required for the network to settle into a stable output semantic state (as used by Plaut, [4]). Alternatively, we could attempt to be more explicit about modelling the lexical decision process by timing the consistency checking between the input phonology and the phonology produced by allowing activation to flow from phonology to semantics and back to phonology. This simple 'activate and check' mechanism was shown by Bullinaria [5] to be able to provide a reliable method for performing lexical decision in this kind of model, whereas details of the pattern of semantic activation alone were not sufficient. Finally, we could argue that the semantic output activations need to drive some later decision process, and that we can ignore the details of this process and take the time required for the integrated output activations to reach some threshold. This may be feasible if all the semantic vectors had equal numbers of fully activated units, but if different words have unequal numbers of activated units (e.g. to represent word concreteness [7]) or if they have non-binary activations (as we shall consider later), then this approach makes less sense. In the following we shall consider both the settling and consistency checking times, and compare their results. In each case we first activate the network for the prime input word and then, without resetting the other network activations, present the target input word and measure the RT.

In this framework, semantic priming arises naturally due to overlap of the semantic vectors. If the network activations due to the prime word are already close to that which will be activated for the target word, then it will take fewer time slices

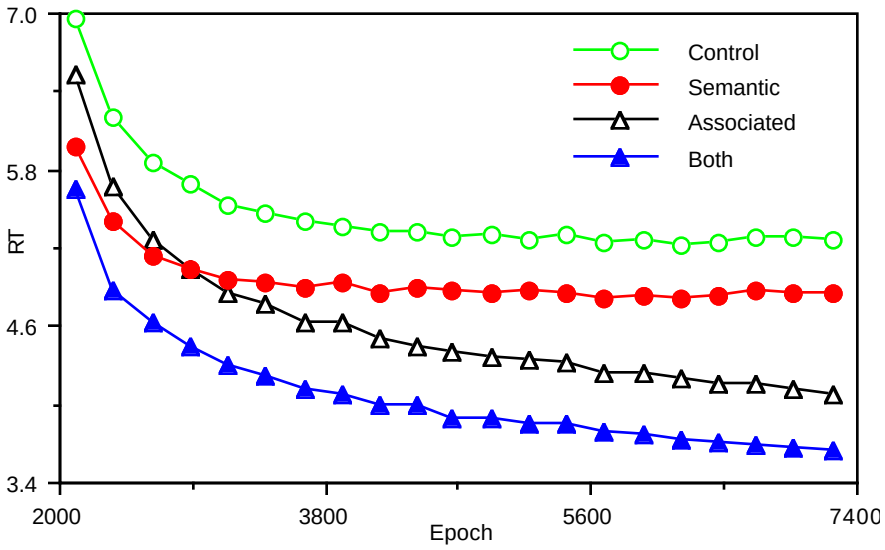

Figure 1: Semantic and associative RT priming effects during the training process.

for the target word to be activated from this state than if it were starting from the activation pattern of some unrelated control word. Associative priming may be caused by properties of the semantic vectors in a similar way, but it has been shown explicitly in connectionist models [9, 4, 5] how the facilitation can also arise purely due to co-occurrence of words in the training data. If, for example, 'society' follows 'pillar' much more often during training than would be expected by chance, then it is not surprising that an efficient learning system will be able to make use of this fact to speed its response times.

To confirm these ideas explicitly, such a phonology to semantics to phonology network with 50 phonological units, 400 hidden units and 27 semantic units was trained by back-propagation throughout the cascading process on 200 mono-syllabic words split into 40 sets of five [5]. Each set consisted of one target word, one unrelated control prime word, one semantically related prime word (with two out of three activated semantic units in common), one associated prime word (that preceded the target word 25% of the time during training) and one prime word that was both semantically and associatively related. Figure 1 shows how the mean primed target word consistency RTs varied for the four prime types during training from the point at which the network had learnt to perform the lexical decision task reliably. We find clear and highly significant ($p < 0.0001$) semantic and associative priming, but there is still a large overlap between individual RTs for the different prime types, for example at epoch 5000 we have mean RTs and standard deviations of 5.28 ± 0.30 for the control primes, 4.84 ± 0.40 for semantically related primes, 4.33 ± 0.47 for associated primes and 3.90 ± 0.32 for semantically related and associated primes. Models of this type also allow us to easily investigate various more detailed RT effects such as prime duration, target degradation, priming spanning unrelated items and mediated priming [5].

3 Corpus Based Semantic Vectors

As noted above, there has been considerable interest recently in using the vectors of statistics that describe the contexts in which words occur in large text corpora to provide a representation of their semantics. The idea is that words which occur in similar contexts will naturally tend to have similar meanings. The obvious advantages of this approach include the elimination of the need for semantic features to be generated randomly (e.g. as done by Plaut [4] and Bullinaria [5]) or by hand (e.g. as done by Plaut & Shallice [7]) and the parsimony of being able to use that statistical structure latent within the language itself. Work of this type has often been pursued from the perspective of computational linguistics [10], but Huckle [11] and Landauer & Dumais [12] have recently noted that it is also of importance for exploring psychological questions concerning human acquisition and representation of word meanings, and have discussed such methods in conjunction with the use of neural network models.

The general approach is to represent each word's distributional context using a vector of probabilities obtained by 'reading' large samples of natural language text. For every 'target word' $word_p$ being represented, each component of its semantic vector contains the probability that some other 'context word' $word_q$ will occupy a particular relationship to $word_p$ in the text. This relationship is typically one which concerns the physical distance between the two words, such as one word occurring within a window of a particular size around the other. Once vectors of this general type have been obtained, the distances between them can be calculated using various metrics to reveal similarities between word meanings and suitable transformations may be applied to provide useful semantic representations for neural network models (e.g. as discussed by Schütze [13]). Alternatively, the inter-word distances in the semantic space can be used directly as a basis for investigating psychological phenomena such as semantic priming (e.g. as suggested by Lund et al. [6]).

For reliable statistics to be obtained, an appropriately large corpus must be used, (e.g. [14]). It was convenient here to use a 10 million word corpus taken from issues of the Wall Street Journal published in 1988 and 1989. Since Zipf's law [15] informs us that the probabilities for less frequent words in a corpus of this size will rapidly become less reliable, we restricted ourselves to using only the highest frequency items. We actually used the most frequent 1000 words in the corpus as our target words and the most frequent 200 words as our context words.

This still left us with many possibilities for deriving useful semantic vectors and choosing different values for the various parameters that specify the vector extraction process will affect the quality of the vectors obtained [14]. For the purposes of this paper, however, it was not crucial that we used the best vectors possible, as long as their distributional properties were sufficiently typical. For each of the 1000 target words, we calculated the 200 dimensional vector in which each component contained the probability that a particular context word would occur within a window of two words to the left of the target word in the corpus, and a similar vector for a window of two words to the right. The window length of two words was chosen following exploratory work which suggested this captured the words' semantics reasonably well. Finally, we concatenated our left and right

context vectors to give a 400 dimensional vector of probabilities for each of the target words. This approach to representing the target words' semantics is thus similar to that adopted by Lund et al. [6].

4 The Combined Model

Unfortunately, using our corpus derived vectors in the existing lexical decision model described above was not a totally straightforward matter. Our first problem was that, if we are to train our networks in a reasonable amount of time, we need to keep the networks as small as possible, which in turn means minimising the dimensionality of our semantic vector space. For our purposes, principal component analysis proved to be a convenient procedure. The 400 dimensional semantic vectors were projected onto the 30 dimensional sub-space containing the maximum variance. This provided much lower dimensional vectors with relatively little loss of information and had the added advantage that we lost a lot of the noise in the process. The inter-word distances in the original space and the sub-space correlated well (Pearson $r = 0.94$). The psychological relevance of this dimensional reduction procedure for corpus derived semantic representations has been discussed recently by Landauer & Dumais [12].

The second problem arose with the standard use of sigmoidal activation functions in our network. For our non-binary outputs it was appropriate to use a linear activation function for the network outputs rather than sigmoids. Since the distribution of vector components was rather skewed towards small values anyway, which corresponds to the central linear region of the sigmoid rather that the saturated extremes, this difference is probably not crucial.

What might be more crucial however, is the decision to use the non-binary components that come naturally out of the corpus analysis rather than attempting to convert them into the binary form commonly used in connectionist systems. Since we already know that we can get semantic and associative priming using random binary semantic vectors [4, 5], it seemed appropriate go on to investigate the more ambitious case of non-binary vectors. Moreover, in the human case, it would be natural to consider our semantic representations to be 'hidden representations' that are learnt by the brain, and it is rare to find hidden representations developing purely binary values in connectionist models. We shall see later how our networks behave rather differently when based on real, rather than binary, outputs.

The next thing we have to consider is that the Wall Street Journal is a rather atypical source of the English language. Certain word pairs (such as 'wall street' and 'dow jones') occur together much more frequently than in normal English, and other words (such as 'bush' and 'ford') are often used in atypical ways. In general, it is interesting and important to study such words, since they are simply extreme examples of what does happen in real language, but to avoid possible artefacts that these words may cause in the current study, we simply removed them from our network training set. We also removed 16 homographs (e.g. 'lead' and 'row') which would clearly have problematic semantic vectors.

In the original model reviewed in Section 2, Bullinaria [5] simulated mappings between phonology and semantics. Here (as Plaut did in his model [4]) we consider

the mappings between orthography and semantics. Given our simplified abstract input representations and the regularity of the orthography to phonology relationship, the distinction is unlikely to be crucial, but it should be kept in mind. As with other models that map to semantics, the randomness of the mapping means that in order to train the network in a reasonable amount of time, we need to restrict the number of training words, which also allows us to use less hidden units. It then follows that we have to reduce the size of the input space so that the word distribution does not become unnaturally sparse. To this end we restricted ourselves to monosyllables with orthography made up of the most common onset consonant clusters (30), vowel clusters (18) and offset consonant clusters (38) plus two units to code for the presence or absence of a final 'e'. Our highest frequency set of 1000 target words contained 270 words consistent with all the above restrictions. These all occurred at least 1297 times in the corpus and were within the most frequent 993 words. Finally, in an attempt to keep the weights and weight changes at values comparable to the binary output networks, we set the arbitrary scale and origin of the semantic vector space so that the mean activation of each semantic unit was zero and the overall standard deviation was 0.14, with maximum component 1.95 and minimum component –1.27.

Since all our 270 words are classed as high frequency in the experimental studies, we did not attempt to impose a word frequency structure on our network training regime. We trained our main network for 75000 epochs on these words using back-propagation on the asymptotic output patterns (with 270 hidden units, sum squared error measure, learning rate $\eta = 0.01$, no momentum). By this point the network was able to perform reliable lexical decision by input-output orthography consistency checking. The RTs were then extracted as above, except that we used a reduced $\lambda = 0.01$ to give a more accurate approximation to the continuous process. The settling RTs were defined as the number of time slices required for all output activation changes per time slice to fall below 0.0001. The consistency RTs were defined as the number of time slices required for the total difference between the input and output orthography activations to fall below 0.1. Note that the precise RT criteria are somewhat arbitrary. The way to proceed is (as in [16]) to vary the various parameters and criteria and show that the resultant RTs are highly correlated over a wide range of reasonable values, and then pick central values within that range. The values used here were also chosen to result in similar RT means and standard deviations for the two RT approaches.

Finally, we also attempted to train an identical network on the same words throughout the cascading process with no resetting of activation between words (150 time slices per word, $\lambda = 0.1$, sum squared error measure, learning rate $\eta = 0.0001$, no momentum), though, as we shall discuss in Section 6, the network never managed to learn to perform reliable lexical decision.

5 Semantic Priming

Needless to say, we checked that our closest semantic vectors (defined in terms of Euclidean distances) did actually correspond to words that were semantically related (the shortest distances were 21 between 'will' and 'would', and 21 between 'three'

Prime Set	Distances	Settling Times	Consistency Times
C1	561 (42)	669 (44)	709 (77)
C2	459 (43)	655 (58)	699 (75)
C3	399 (53)	638 (47)	676 (83)
P3	96 (54)	591 (72)	586 (91)
P2	86 (52)	586 (71)	569 (93)
P1	71 (44)	581 (78)	570 (101)

Table 1. The simulated primed RTs compared with distances in semantic space.

and 'four') and the most distant words really were unrelated (the longest distances were 737 between 'lot' and 'past', and 646 between 'same' and 'try') though in this paper we shall not attempt to match our network results to real lexical decision experiments. To do that reliably we would need to train a much larger number of considerably more realistic networks on many more words and use semantic vectors derived from significantly more representative corpora.

As we discussed in Section 2, semantic priming has been found before in network models such as ours [5]. In these random binary target networks, the semantic Euclidean distances were all $\sqrt{2} \sim 1.41$ between semantically related words and $\sqrt{6} \sim 2.45$ between un-related words, yet there was still a large distribution of RTs and varying degrees of priming. Clearly factors other than simple semantic distances are determining these results. Inevitably, each RT will be determined by the unit i that is the slowest to update its activation. Looking at the above cascade equations we see that, to first approximation, the number of time slices required to move from the prime output to the settled target output will be given by the difference of the final $Sum_i(t_\infty)$ between the prime and target divided by the average step size which will be related to the $Sum_i(t_\infty)$ for the target. Unfortunately, we cannot use this as an easy way to predict the RTs, because determining the average step size is harder than actually running the network, but it does tell us something useful about the RTs. The problem with sigmoids and binary output targets is that the sigmoids saturate, which means that very large random variations may arise in the Sum_i's during learning without affecting the actual network outputs very much and hence without being constrained by the training algorithm. Given that the values of the Sum_i's have such a big influence on the RTs, it is no wonder that the RTs are so noisy. With our non-binary semantic vectors we have no output sigmoids and the output activations are the Sum_i's themselves which are learnt to be particular values. We may thus expect to suffer less seriously from random effects than the binary case. But is this true (given that we still have sigmoids at the hidden layer) and does it really allow the simulated priming results to correlate better with the distances between our semantic vectors?

There are many ways in which we can illustrate our network priming results. One useful approach is to look at the RTs for each of our 270 words when primed by the three closest words in semantic space (sets P1, P2, P3) compared to the RTs when primed by the three furthest words (sets C1, C2, C3). The mean distances and RTs are shown in Table 1 (with standard deviations in brackets). The first thing to notice is that both forms of simulated RT show faster times for the prime sets (P's)

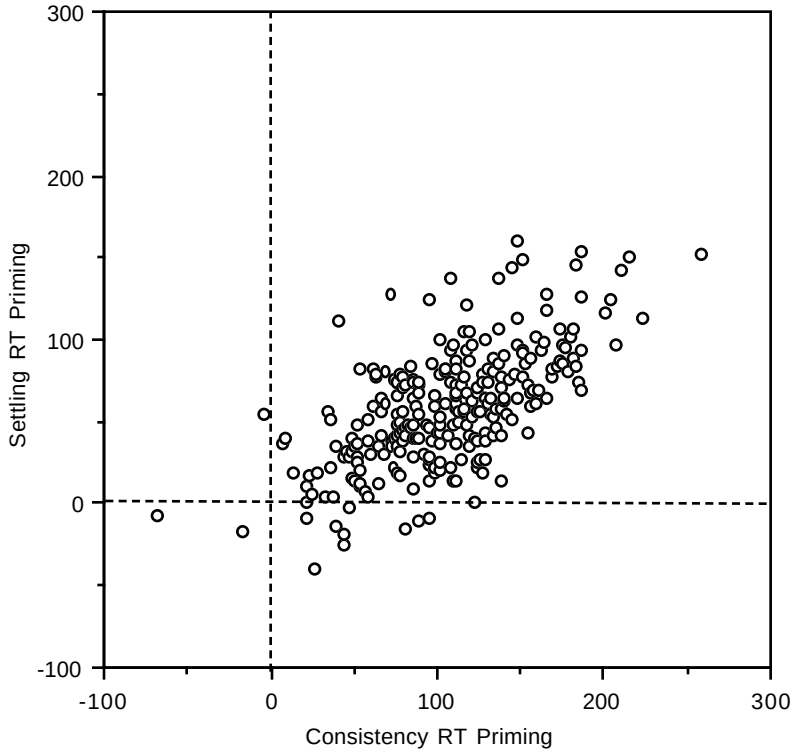

Figure 2: Comparison of priming results for settling and consistency RTs.

than the control sets (C's), so we do find semantic priming (and this is highly significant, $p < 0.0001$). However, it is clear from the standard deviations that the RTs are very noisy and that, as we also found in the binary target networks, there is no simple relationship between the Euclidean distances and the RTs. Indeed, if we take what we would expect to be the maximum priming result, i.e. the differences in RTs between control prime set C1 and semantically close prime set P1, we find the effect is not always even in the right direction. Not only do both RT approaches show some negative priming, but they also fail to agree on which words this happens for. For the settling RTs we have a mean priming of 88 (standard deviation 76), minimum –72, maximum +293. For the consistency RTs we have 140 (101), –135, +429. If we average over all three sets of primes and control primes, the priming is only slightly more reliably positive: 68 (48), –69, +205 for settling and 119 (61), –38, +322 for consistency. At least every time we test a given trained network on a particular prime-target pair, we get the same RT. In experiments on human subjects, noise arises even here (e.g. due to effects such as concentration, tiredness, episodic memory recall, and so on) to give a distribution of RTs. Since distributions of priming results are also found in humans, there is no real problem for our models in that respect, but it does make it rather difficult to uncover the precise relationship between the semantic space distances and the network priming results.

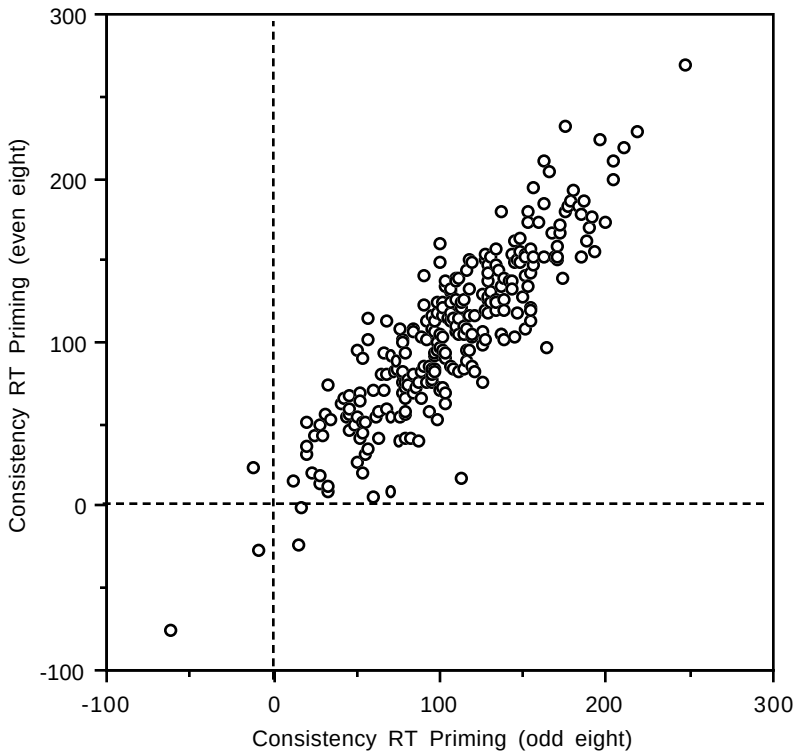

Figure 3: Reliability of consistency RTs for individual words.

To see past the random noise and to reveal the reliable underlying effects we really need to look at many networks in the same way that experimental results are averaged over many human subjects. We therefore trained another fifteen networks with exactly the same architecture and words, but with different random initial connection weights and learning rates, and obtained results similar to those of our original network. The individual word primings did not correlate particularly well between the networks – we typically found Pearson r ~ 0.4 to 0.5 for consistency priming and Pearson r ~ 0.5 to 0.6 for settling priming across networks. This was comparable to the correlation between the consistency and settling priming within an individual network – typically Pearson r ~ 0.45 to 0.55. However, each network did show a similar, and highly consistent, average semantic priming effect for both settling and consistency RTs. Averaging over all sixteen networks and the three primes and controls per word gave a settling priming of 58 (36), –40, +160 and a consistency priming of 106 (47), –69, +257. Figure 2 shows the actual distributions of average priming and compares the settling and consistency results. The correlation between the two approaches has now risen to Pearson r = 0.65.

We can get a fair indication of the reliability of these average priming results by comparing the results obtained from two disjoint random subsets of eight of our sixteen networks. Figure 3 shows how well the consistency primes correlate

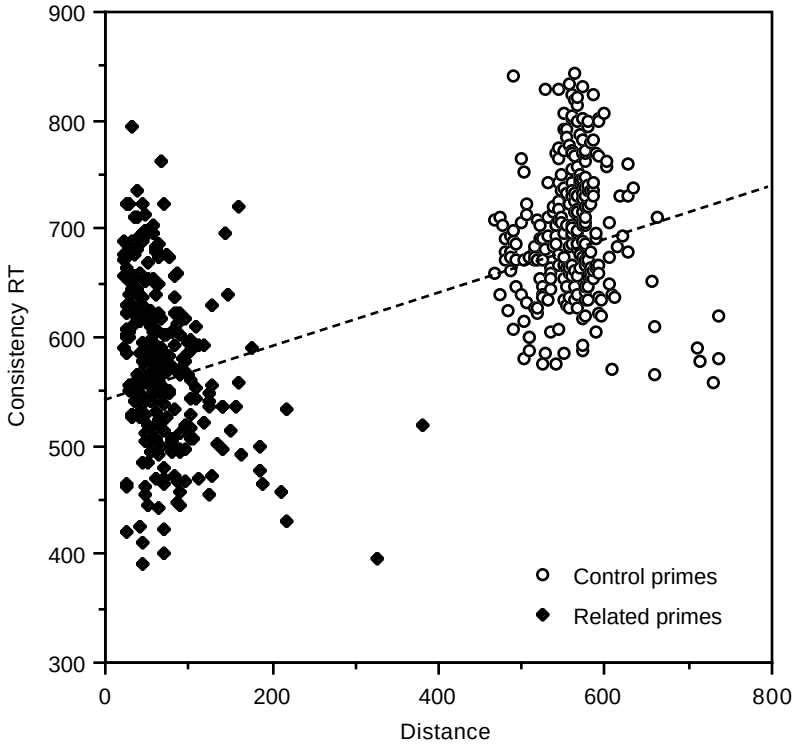

Figure 4: Comparison of the semantic vector space distances with consistency RTs showing a significant, but very noisy, semantic priming effect.

(Pearson r = 0.88) and we get a very similar pattern for the settling primes (Pearson r = 0.91). These results suggest that averages over sufficiently many networks and/or primes can cancel out the noise to leave reliable semantic relatedness effects together with any other (e.g. orthographic, consistency, neighbourhood, RT approach) effects not inherent in individual semantic vectors themselves. It is unfortunate, given the computational resources necessary to train the networks to map between the orthographic and semantic representations, that so many networks are required in order to obtain a clear picture of what is going on.

The final question we need to ask is what exactly is causing the spread of priming that is becoming clear in plots such as Figure 3. The natural first guess is that it is simply the Euclidean distances between the words in our semantic vector space [6]. Figure 4 shows how these distances for individual word pairs (the P1's and C1's) compare with the consistency RTs averaged over all sixteen networks. We get a similar pattern for the settling RTs. Once again we can see that there is more involved than simple Euclidean distances. It may well turn out that city block distances or particular information theoretic measures can provide a better indicator of priming, but, since these are not preserved by the principal component dimensional reduction, it is rather difficult to investigate this reliably.

In the long term, we really need to investigate the networks' performance much more closely. One approach is to look at the words which stand out as being reliably good or bad primers and see what is special about them. Unfortunately, the variability between networks makes this rather difficult and we often find that a given RT is determined by one particular output unit that has no obvious significance. Another problem is that the current networks are still far too small scale. Despite our restrictions on the orthographic space, there are still some graphemes that only occur in two of the training words, whilst others occur up to 57 times. Similarly some words occur in sparsely populated areas of semantic space whereas others have several close neighbours. These factors, which presumably contribute to the spread of priming results, may well disappear in more realistic networks. It is clear that there is much more work to be done in this area.

6 Associative Priming

As we discussed in Section 2, for random binary semantic vectors, we find that training our networks throughout the cascading process allows them to use any frequent word co-occurrences that arise during training to speed their RTs, and hence they automatically exhibit associative priming [5]. This is largely because such sigmoidal networks with binary output targets have an inbuilt flexibility that enables them to make the weight adjustments appropriate for an advantage in RT with little increase the output error. For example, weight changes could be made that result in a large shift (from –15.0 to –5.0 say) in a $Sum_i(t_\infty)$, with only a small increase (~0.007) in the activation error $Out_i(t_\infty)$.

In our linear non-binary output networks it is much harder to gain an RT advantage in this way, since any significant change to an output $Sum_i(t_\infty)$ will directly introduce a significant error into the $Out_i(t_\infty)$. When using the standard cascaded learning approach described in Section 2 in this case, the output errors that inevitably occur during the word transitions cause weight changes that disrupt the networks' output performance to such an extent that it is very difficult for the networks to learn accurate semantic representations (as checked by testing them on the orthography to semantics mapping) and they are hence unable to perform reliable lexical decision by input-output orthography consistency checking. However, if we artificially reduce the disruption by increasing λ (to 0.5), the network *is* able to learn sufficiently well to end up performing lexical decision reasonably reliably, and the RTs do exhibit semantic and associative priming. Both types of priming are significant but, compared with human experimental results, the associative priming is too small in comparison to the semantic priming.

It thus remains a rather open question as to whether human brains are able to exploit this word co-occurrence mechanism with fixed semantic representations (such as derived from corpus statistics) and hence exhibit associative priming. It is possible that the semantic representations must themselves be subject to adjustment during training in order to show this effect. It is also possible that the experimental associative priming is already inherent in the 'semantic' vectors without the need for any additional training effects. It may even be possible that additional connections between associated words (i.e. their semantic micro-feature units) are employed.

Clearly, many more network simulations are required to resolve this matter. Of course, it should be noted that the corresponding experimental results are far from clear cut either. Separate claims have been made that all associative priming is really a form of semantic priming (e.g. by Lund et al. [6]), that all semantic priming is actually associative priming (e.g. by Shelton & Martin [2]), and that both forms of priming exist in their own right (e.g. by Moss et al. [3, 9]).

7 Conclusions

The use of a cascaded activation approach in neural network models of simple mappings provides a natural procedure for simulating realistic distributions of RTs including priming and speed-accuracy trade-off effects [16]. Previously this approach has been used to show how connectionist models of the mappings between simplified representations of orthography/phonology and random binary vector representations of semantics can account for many experimental results concerning semantic and associative priming of lexical decision RTs [5]. In this paper we have investigated the use in these lexical decision models of non-binary semantic vectors derived from the word co-occurrence statistics of large text corpora. Our main conclusion is that there *is* a significant relationship between the prime-target distances in the underlying corpus derived semantic vector spaces and the priming in the connectionist networks that use these vectors, *but* the relationship is very noisy. In the same way that experiments need to control for many other factors and average over many test words and many subjects to show clear priming results, so do the connectionist simulations.

We believe that the simplified and small scale connectionist lexical decision models presented here show much promise and already exhibit many of the essential features required of a complete model of this task. We have also identified and solved a number of problems inherent in this class of model. However, it is clear from the preliminary results presented here that further investigations employing larger and more representative corpora and many larger and more realistic networks will be required before we can be confident of the precise relationships between the various corpus, neural network and experimental results.

References

1. Neely JH. Semantic Priming Effects in Visual Word Recognition: A Selective Review of Current Findings and Theories. In Besner D& Humphreys GW (Eds) Basic Processes in Reading: Visual Word Recognition. Erlbaum, Hillsdale NJ, 1991, pp 264-336
2. Shelton JR & Martin RC. How Semantic is Automatic Semantic Priming? Journal of Experimental Psychology: Learning, Memory and Cognition 1992; 18:191-210
3. Moss HE, Ostrin RK, Tyler LK & Marslen-Wilson WD. Accessing Different Types of Lexical Semantic Information: Evidence From Priming. Journal of Experimental Psychology: Learning, Memory and Cognition 1995; 21:1-21

4. Plaut DC. Semantic and Associative Priming in a Distributed Attractor Network. Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society. Erlbaum, Mahwah NJ, 1995, pp 37-42
5. Bullinaria JA. Modelling Lexical Decision: Who Needs a Lexicon? In Keating JG (Ed) Neural Computing Research and Applications III. St. Patrick's College, Maynooth Ireland, 1995, pp 62-69
6. Lund K, Burgess C & Atchley RA. Semantic and Associative Priming in High-Dimensional Semantic Space. Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society. Erlbaum, Mahwah NJ, 1995, pp 660-665
7. Plaut DC & Shallice T. Deep Dyslexia: A Case Study of Connectionist Neuropsychology. Cognitive Neuropsychology 1993; 10:377-500
8. McClelland JL. On the Time Relations of Mental Processes: An Examination of Systems of Processes in Cascade. Psychological Review 1979; 86:287-330
9. Moss HE, Hare ML, Day P & Tyler LK. A Distributed Memory Model of the Associative Boost in Semantic Priming. Connection Science 1994; 6:413-427
10. Brown PF, Della Pietra VJ, deSouza PV, Lai JC & Mercer RL. Class-based n-gram models of natural language. Computational Linguistics 1992; 18:467-479
11. Huckle C. Grouping Words Using Statistical Context. Proceedings of the Seventh Conference of the European Chapter of the Association for Computational Linguistics. Morgan Kaufmann, San Francisco CA, 1995, pp 278-280
12. Landauer TK & Dumais ST. A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction and Representation of Knowledge. Psychological Review 1997; 104: 211-240
13. Schütze H. Word Space. In Hanson SJ, Cowan JD & Giles CL (Eds) Advances in Neural Information Processing Systems 5. Morgan Kauffmann, San Mateo CA, 1993, pp 895-902
14. Patel M, Bullinaria JA & Levy JP. Extracting Semantic Representations from Large Text Corpora. Proceedings of the Fourth Neural Computation and Psychology Workshop. Springer Verlag, London, 1997
15. Zipf GK. The Psycho-biology of Language. Houghton Mifflin, Boston, 1935
16. Bullinaria JA. Modelling Reaction Times. In Smith LS & Hancock PJB (Eds), Neural Computation and Psychology. Springer, London, 1997, pp 34-48