How Can *I* Know What *You* Think?: Assessing Representational Similarity in Neural Systems

Aarre Laakso (AARRE@UCSD.EDU)
Department of Philosophy
University of California, San Diego
La Jolla, CA 92093

Garrison W. Cottrell (GARY@CS.UCSD.EDU)
Institute for Neural Computation
Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093

Abstract

How do my mental states compare to yours? We suggest that, while we may not be able to compare experiences, we can compare neural representations, and that the correct way to compare neural representations is through analysis of the distances between them. In this paper, we present a technique for measuring the similarities between representations at various layers of neural networks. We then use the measure to demonstrate empirically that different artificial neural networks trained by backpropagation on the same categorization task, even with different representational encodings of the input patterns and different numbers of hidden units, reach states in which representations at the hidden units are similar.

Introduction

Many psychologists have postulated models of semantic memory that identify semantic similarity with proximity in a high-dimensional space of concepts. Some use techniques from psychophysics, performing multidimensional scaling on large numbers of human similarity judgements (Rips et al., 1973). Others have extracted cooccurrence matrices with semantic properties from large text corpora (Lund et al., 1995). Paul Churchland has argued that the activation state space of neural networks captures semantic similarity by the proximity of activation vectors (Churchland, 1989). This view has much to be said for it. It explains the psychometric data that human judgements of similarity tend to be robust across subjects; it explains the data from lexical decision experiments that semantically-related pairs of words presented sequentially are identified more quickly than non-related words; and it explains many prototypicality effects in categorization.

Calling Churchland's view "state-space semantics", however, Jerry Fodor and Ernest Lepore have mounted a powerful argument against it (Fodor and Lepore, 1992). They argue that state-space semantics entails semantic holism, and that semantic holism is intolerable. *Semantic holism* is the view that the content (meaning) of each one of a particular person's thoughts depends on the content of every other one of that person's thoughts (what I mean when I think *That's a dog* depends on what I mean when I think *That's a lightbulb*, and so on). Fodor and Lepore argue that semantic holism is intolerable on the grounds that it would entail that communication, language learning, psychological explanation, and scientific progress are impossible[1]

[1] We will not go into the details of their arguments against holism, because we intend to show that state-space semantics is *not* perniciously holistic. See (Fodor and Lepore, 1992, pp. 8-16) for the details of their arguments.

Fodor and Lepore argue that state-space semantics is perniciously holistic because it postulates: (1) that the meanings of representations in semantic space are determined by their relations to other representations in semantic space; and (2) that we could never determine whether two semantic spaces exhibit identical, or even similar, sets of relations. We agree with (1) but deny (2). In this paper, we present a concrete method for measuring the similarity between two semantic spaces and we demonstrate empirically that the semantic spaces of different neural networks trained on similar problems are often similar, even when their input encodings or number of hidden units are different.

Fodor and Lepore state their argument as follows:

> What Churchland has on offer is the idea that two concepts are similar insofar as they occupy similar positions *in the same state space*. The question thus presents itself: *When are S_1 and S_2 the same state space?* When, for example, is your semantic space a token of the same semantic space state type as mine? Well, clearly a necessary condition for the identity of state spaces is the identity of their dimensions; specifically, identity of their semantic dimensions, since the current proposal is that concepts be located by reference to a space of *semantically relevant properties*. We are thus faced with the question of when x and y are the same semantic dimensions....But this is surely just the old semantic identity problem back again (Fodor and Lepore, 1992, pp. 197-8).

Putting the argument explicitly in terms of neural networks, it goes like this: suppose you have two networks, possibly with different numbers of nodes (i.e., different dimensionality of the activation space) and differently-weighted connections (i.e., different dimensions in the activation space). How then can you tell when the networks are representing their inputs *the same way*?

We propose the following answer: because state-space semantics claims that the *proximities* of points in activation space capture semantic content, points in two different semantic spaces represent the same thing just in case they conform to the same set of distance relations. Thus, we needn't

determine whether the *dimensions* of the spaces are the same, but only whether points in two spaces have the same distances *relative* to each other. The question then becomes: how do we determine whether two different activation spaces have the same set of distance relations among their points, regardless of the number of dimensions they may have or the interpretations of those dimensions?

Assuming that we have *labeled points* (that is, that we can label each representation by the stimulus that induced it), one approach is to use *cluster analysis* to visualize the relationships between points. In the application of cluster analysis to networks, patterns of activation at the hidden units are measured for each input; the patterns are then progressively matched with each other according to their proximity. The result is a dendogram, or tree structure, which shows the proximities of the input patterns as they are represented at the hidden layer. In the first application of cluster analysis to representation in artificial neural networks, Sejnowski and Rosenberg showed that similarities among hidden-layer representations in their NETTalk network matched the phonological similarities that humans perceive in spoken phonemes. For example, hard-'c' and 'k' sounds were grouped together, and at the highest level, consonants were grouped together, as were vowels. (Sejnowski and Rosenberg, 1987). Given two networks, then, one could do cluster analyses of the same inputs to each and compare the resulting dendograms. This is fine if all we want to do is "eyeball" the similarity, but it does not yield a *number* that tells us how similar the two representations are. We know of no accepted way of rigorously assessing the similarity of different dendograms.

Hence, instead of using cluster analysis, we propose a different method of measuring representational similarity: correlation between inter-point distances in the respective networks. We start by computing the Euclidean distance between pairs of points in each of the two activation spaces. By comparing only distances between points, we achieve invariance to uniform global translation, rotation, and reflection of representational space.

Next, we calculate correlation between the two sets of distances. Suppose the rows of a matrix X are the activation patterns generated by various stimuli in Network X. X_i designates the pattern of activation (row of the matrix) corresponding to stimulus i as it is represented by Network X. Likewise, suppose the representation of the same stimuli from a different network is encoded in the matrix Y. Then for m stimuli, we calculate the distances between the hidden activation vectors in the two networks, giving two length $m(m-1)/2$ vectors of distances. We than compute similarity between the two representations by calculating correlation between these two vectors. Taking correlation also gives us one more invariance: correlation is invariant to differences in scale between the two spaces.

We believe that calculating correlation between distances among points in the hidden-unit representations in two neural networks gives us a number which tells us how similar the "semantic structures" are in the two networks. Correlating distances between points answers Fodor and Lepore's challenge: we needn't know the meanings of the dimensions in the two networks and, indeed, the number of dimensions need not even be the same so long as they preserve the distances between points.

Experiment 1

As an example of how our technique for measuring similarities in network representations can be used, we modeled color categorization in artificial neural networks using a variety of input encodings. The different encodings might be thought of as ways in which different "species" encode the impact of various wavelengths of light on their sensory systems. We were interested in two questions. First, to what extent would different "species" agree about their internal representations of the concepts when they all carved up the world in the same way (i.e., all agreed about the color labels)? Second, to what extent would members of the *same* species agree, given that they may have different numbers of neurons (while all had sufficient numbers to do the task)?

Procedure

We trained a number of networks on a color categorization task; inputs were based on spectrophotometer readings, and outputs were localist representations of 5 color categories (red, yellow, green, blue, and purple). For inputs, we used a database of spectrophotometer readings from color samples (anonymous, 1995). The original data were 61-element vectors of integers between 0 and 4095, read at 5nm intervals of the visual spectrum between 400nm and 700nm. We used the red, yellow, green, blue, and purple patterns, scaled the inputs to 0-255 integers, and selected every 5th field from the original data, leaving 12 input elements. We used a localist encoding for the outputs (red = 1 0 0 0 0, yellow = 0 1 0 0 0, and so on).

From this base data set, we created four different encodings of the input patterns to be used in training the networks: The *real* encoding was formed by scaling the 0-255 integer inputs to decimal representations between 0 and 1. Thus, each pattern had 12 input elements in the real encoding, each element a rational number between 0 and 1. For example, the real representation of the first pattern was <0.827451 0.835294 0.827451 0.815686 0.796078 0.827451 0.874510 0.874510 0.862745 0.847059 0.827451 0.815686>. The *binary* encoding was formed by representing the 0-255 integer inputs as 8-bit binary numbers. Thus, each pattern had 96 (=12x8) input elements in the binary encoding, each element valued either 0 or 1. For example, the binary representation of the first pattern was <1 1 0 1 0 0 1 1 1 1 0 1 0 1 0 1 1 1 0 1 0 0 1 1 1 1 0 1 0 0 0 0 1 1 0 0 1 0 1 1 1 1 0 1 0 0 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 0 0 1 1 0 1 1 0 0 0 1 1 0 1 0 0 1 1 1 1 0 1 0 0 0 0>. The *gaussian* encoding was formed by dividing the interval between 0 and 255 into quarters, and using five units to represent the endpoints of the intervals. A particular value was coded as a Gaussian "bump" on this interval, with a standard deviation of 32 and mean at the point to be represented. The input gaussians were thus centered at 0, 63.75, 127.5, 191.25 and

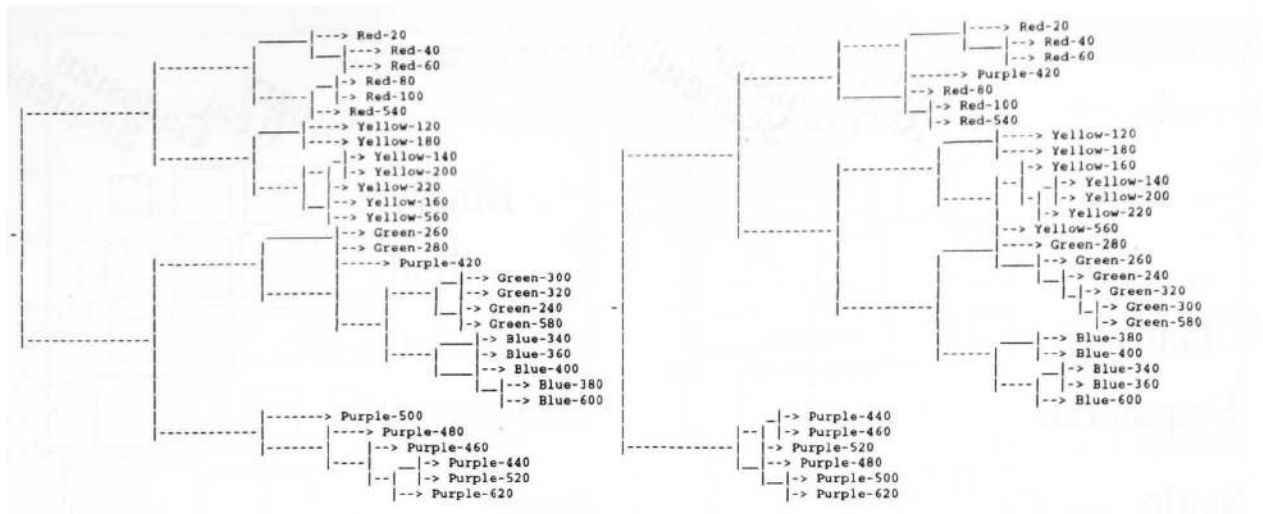

Figure 1: Representative clusterings of hidden-unit activations from two of the five networks trained on the "real" encoding (31 of 627 patterns shown).

255. For example, the gaussian representation of the first pattern was <0.000000 0.000004 0.003024 0.193481 0.976358 ... 0.000000 0.000006 0.004143 0.228310 0.992218>. The *sequential* encoding was formed by numbering the patterns sequentially with 3-digit decimal numbers from 001 to 627. Each 3-digit number was then represented by a single unit with an activation between 0 and 1. For example, the sequential representation of the first pattern was <0.0 0.0 0.1>. Because the input patterns in the data set were ordered by color, this representation makes more sense than it may appear at first.

From each representation, we selected every sixth line for the holdout set (104 patterns) and left the rest for the training set (523 patterns). Because we were not exploring generalization in this experiment, we did not use a testing set.

Using backpropagation, we trained 3-layer networks, each with 3 hidden units, on each input encoding for a maximum of 10,000 cycles using a learning rate of 0.25. Training was stopped before epoch 10,000 if the root mean-squared error of the holdout patterns had not declined in as many epochs as taken to reach the previous low. For example, if a minimum root mean-squared error was reached after epoch 2,500 and no subsequent epoch had a lower error, then training would be stopped after epoch 5,000. For each encoding, the experiment was repeated with 5 networks, each starting with a different set of initial random weights.

Using the best learned weights (the ones between the beginning and end of training with the best error on the holdout set), we computed activations at the hidden nodes on each input pattern, thereby obtaining each network's representation of the input patterns at its hidden layer. We then computed the Euclidean distances, for each activation matrix, between each pattern and each other pattern in that matrix. We then computed correlation between each set of distances and every other set of distances.

Results

In the input encodings, the clustering of the intensities of light at various wavelengths do not match very well with our qualitative perceptions of color similarities. Hence, the cluster diagrams for the real, binary, and gaussian input patterns appear disorganized, in the sense that colors that we would group together (e.g., greens) were interspersed with other colors (data not shown). Thus, we expected the different input encodings to not be very highly correlated. Contrary to our expectations, the binary, real and gaussian input encodings were highly correlated with each other (see Figure 2, part a). The correlation between the real and gaussian encodings was nearly 1, and the binary encoding had a correlation of about 0.6 with both the real and the gaussian encodings. The sequential encoding, on the other hand, was almost completely uncorrelated with the other encodings.

The measured difference between the sequential input encoding and the other input encodings may be due to the fact that the original data were grouped by color. That is, the first 115 patterns were reds, the next 120 patterns were yellows, and so on. Because the patterns to which the sequential encoding was applied were ordered by their color category, the sequential numbers with which they are encoded contain some information about their category. Most colors that should be categorized together are nearby in the input pattern space, but there are two kinds of exceptions. The first is that patterns differing in the ordering by as much as 100 can be as close together as patterns differing by only one in the ordering. For example, pattern 345 (represented as <0.3, 0.4, 0.5>) is as close to pattern 245 (<0.2, 0.4, 0.5>) as 245 is to 244 (<0.2, 0.4, 0.4>).

The second exception is caused by the fact that all neighbors in the ordering are 0.1 apart in the encoding *except* points with a 0 element. Each pattern with a 0 element in the sequential encoding comes right after one with a 0.9 element (and hence the two are at least 0.9 units apart). For example,

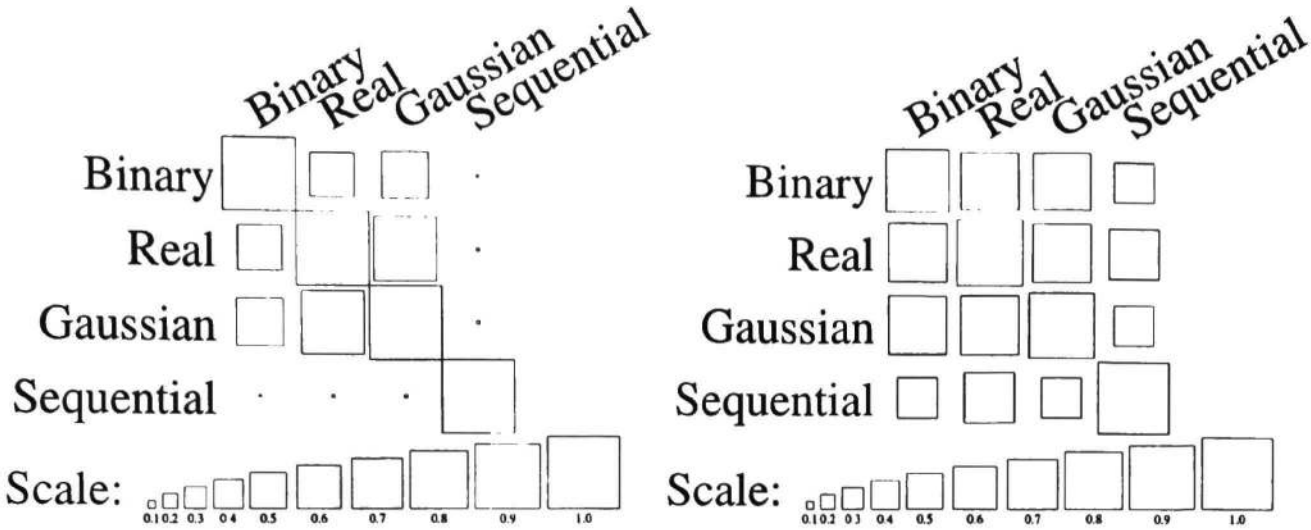

Figure 2: Hinton diagrams showing correlation among input patterns (Part (a), on the left), and among hidden unit activations (Part (b), on the right). Part (a) shows correlation among the input patterns used in training the networks. Part (b) shows mean correlation between hidden unit activations of 5 networks trained on each encoding and hidden unit activations of 5 networks trained on each other encoding (e.g., binary vs. real), as well as mean correlation between hidden unit activations among the 5 networks trained on each encoding (e.g., binary vs. binary). The sides of the boxes are proportional to the values.

although patterns 458, 459, and 460 are right next to each other in the data set, the sequential representation of pattern 459 (<0.4, 0.5, 0.9>) is much closer to that of pattern 458 (<0.4, 0.5, 0.8>), than it is to that of pattern 460 (<0.4, 0.6, 0.0>).

Correlations between the hidden unit activations for each of the five networks trained on the same representations were all greater than 0.87. Different networks, starting from different random initial weights, found similar solutions to the color categorization problem for each input encoding. The similarities are reflected in their cluster diagrams, which show colors grouped in human-like ways (see Figure 1). Even more striking, the hidden unit representations of networks trained on *different* input representations were also highly correlated (see Figure 2 part b). Correlations between hidden unit activations of networks trained on the binary, gaussian, and real input encodings are all greater than 0.8, while correlations of these networks with networks trained on the sequential encoding are somewhat lower.

Experiment 2

We also conducted a second set of experiments, varying the numbers of hidden units in the networks, and using only the real encoding and a variation on the sequential encoding, in order to determine whether networks with different numbers of hidden units would develop similar representational structures.

Procedure

We used the same color categorization task for the second experiment as for the first, but the input representations were slightly different. For the real encoding, we used all 61 elements of the original dataset, rather than sampling at 12 evenly-spaced intervals. We did not use the gaussian or binary encodings, because they had proved in the first experiment to be highly similar to the real encoding. We also randomized the order of presentation of the patterns during each training epoch, and used a separate testing set in addition to the training and holdout sets of the first experiment.

The most important difference between the first and second experiments, however, was that we varied the number of hidden units in the second experiment. Whereas in the first experiment, all of the networks had 3 hidden units, in the second experiment, the number of hidden units was varied from 1 to 10. For each of the two input encodings (real and sequential), we trained 3-layer networks with 1 to 10 hidden units. Each network was trained a minimum of 500 epochs, and training was stopped after the 500th epoch whenever the root mean-squared error on the holdout set had not decreased in 50 epochs. We also replicated the training regime on 10 additional networks with 5 hidden units each, in order to demonstrate that the results with networks with different initial random weights were robust and to compare the new procedures with the previous ones.

Results

Networks with 1 and 2 hidden units failed to learn, and so will not be considered further. Networks with 3 to 10 hidden

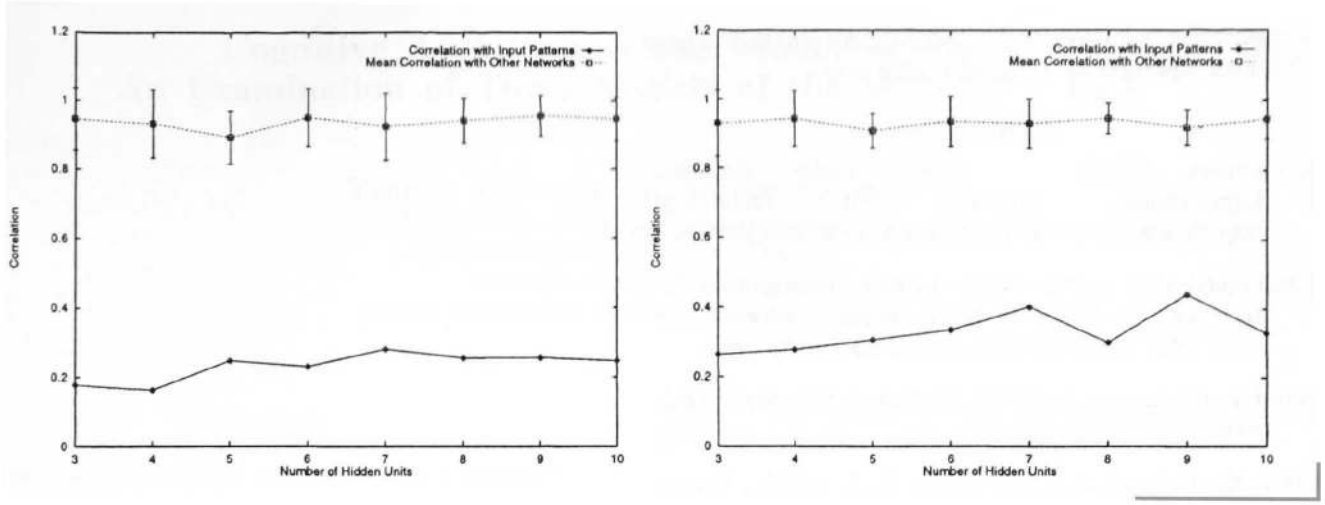

Figure 3: Number of hidden units versus correlation to input patterns and mean correlation to networks with different numbers of hidden units for networks trained on the "real" encoding (Part (a), on the left) and for networks trained on the "sequential" encoding (Part (b), on the right).

units trained on the real input encoding again learned hidden-layer representations that were very similar to each other, *regardless of the number of hidden units in the network.* Correlations between hidden-unit activations and input patterns were low (mean=0.232, sd=0.040), but average correlations between hidden-unit activations over networks with different numbers of hidden units were high (mean=0.934, sd=0.020).

The same was true of networks trained on the sequential encoding (see Figure 3). Correlations between hidden-unit activations and input patterns were low, although higher than they were for the "real" encoding (mean=0.333, sd=0.061), but average correlations between hidden-unit activations over networks with different numbers of hidden units trained on the "sequential" encoding were still high (mean=0.934, sd=0.013). The correlation between the hidden unit activations of networks trained on the real encoding and networks trained on the sequential encoding, however, was even lower than in the first experiments reported here (about 0.3), most likely due to the reduced amount of training in this experiment.

For networks with 5 hidden units, 10 replications starting from different initial random weights confirmed that networks with different weights trained on the same encoding found very similar solutions to the problem. Average correlation among the 10 different networks trained on the real encoding was 0.929, while average correlation among the 10 different networks trained on the sequential encoding was also 0.944, demonstrating that networks with different weights trained on the same encoding found very similar solutions to the problem regardless of which encoding they used. Average correlation between the hidden unit activations of the 10 5-unit networks trained on the sequential encoding and the sequential encoding itself was 0.325, whereas average correlation between the hidden unit activations of the 10 5-unit networks trained on the real encoding and the real encoding itself was 0.202, demonstrating that the hidden unit representations, while not completely unrelated to the input patterns, were not simply copies of the input patterns.

Discussion

It is well known that different networks trained on the same problem may partition their activation spaces in similar ways. We have presented a way to measure this. Our results indicate that it is also possible for networks from different "species" (i.e., trained from different input encodings) to partition their activation spaces in similar ways. Even though we trained our networks on different input representations, the high correlations between their hidden-layer representations show that they partition their activation spaces similarly. Evidently, it is possible for the representational states of two individuals who categorize their inputs the same way to be similar even though they have different "sensory systems" (i.e., input encodings) and different numbers of units. Finally, it is remarkable that individuals from the same "species", with *different numbers of hidden units*, all achieve essentially identical representational structures when they agree on the category structure. There is some hope for communication between us, even in the world of state space semantics.

Conclusions

In response to Fodor and Lepore's challenge to state-space semantics, we have argued that representational similarity can be measured by correlation between inter-point distances in any two activation state spaces. Moreover, we have given a technique for measuring representational similarity. Our measure is a robust criterion of content similarity, of just the sort that Fodor and Lepore demanded in their critique of Churchland. It can be used to measure similarity of internal representations regardless of how inputs are encoded, and regardless of number of hidden units. Furthermore, we have used our measure of state-space similarity to demonstrate empirically that different individuals, even individuals

with different "sensory organs" and different numbers of neurons, may represent the world in similar ways.

References

anonymous (1995). *Kuopio Color Database*. Lappeenranta University of Technology, http://www.lut.fi/ltkk/tite/research/color/lutcs_database.html.

Churchland, P. M., editor (1989). *A Neurocomputational Perspective: The Nature of Mind and the Structure of Science*. MIT Press/Bradford Books, Cambridge, MA.

Fodor, J. and Lepore, E. (1992). *Holism: A Shopper's Guide*. Blackwell, Cambridge, MA.

Lund, K., Burgess, C., and Atchley, R. A. (1995). Semantic and associative priming in high-dimensional semantic space. Number 17 in Proceedings of the Annual Conference of the Cognitive Science Society.

Rips, L. J., Shoben, E. J., and Smith, E. E. (1973). Semantic distance and the verification of semantic relations. *Journal of Verbal Learning & Verbal Behavior*, 12(1):1–20.

Sejnowski, T. J. and Rosenberg, C. R. (1987). Parallel networks that learn to pronounce english text. *Complex Systems*, 1:145–168.