Modeling form similarity in the mental lexicon with self-organizing feature maps

Wittenburg, Peter; Frauenfelder, Ulrich Hans

How to cite

WITTENBURG, Peter, FRAUENFELDER, Ulrich Hans. Modeling form similarity in the mental lexicon with self-organizing feature maps. In: Konvens 92: 1. Konferenz „Verarbeitung natürlicher Sprache“. Nürnberg (Germany). Berlin, Heidelberg : Springer, 1992. doi: 10.1007/978-3-642-77809-4_16

This publication URL: https://archive-ouverte.unige.ch//unige:84533
Publication DOI: 10.1007/978-3-642-77809-4_16

Modeling form similarity in the mental lexicon with self-organizing feature maps

Peter Wittenburg and Uli H. Frauenfelder
Max-Planck Institute for Psycholinguistics
Wundtlaan 1, 6525 XD Nijmegen, The Netherlands
pewi@mpi.nl or uli@mpi.nl

Abstract

This paper describes recent efforts to model the remarkable ability of humans to recognize speech and words. Different techniques for representing phonological similarity between words in the lexicon with self-organizing algorithms are discussed. Simulations using the standard Kohonen algorithm are presented to illustrate some problems confronted with this technique in modeling similarity relations of form in the human mental lexicon. Alternative approaches that can potentially deal with some of these limitations are sketched.

1. Introduction

The psychological processes underlying human behavior are extremely complex. To deal with this complexity, it has become increasingly common for psychologists to abandon simple verbal descriptions and to appeal to computer-implemented models. These models can translate unclear and complex ideas into more accessible and explicit hypotheses about processing and representation. These hypotheses can in turn be tested with psychological experiments. This interaction between simulation and experimentation can lead to the construction of computer models that capture systematically more of the human capacity. It should be obvious that what is important in this enterprise is not how well the models perform in the absolute, but how closely their structure and performance matches that of the human.

One area of psychological investigation where computer modeling has begun to have an impact is that of language processing and, in particular, speech and word recognition. Artificial neural network models have proven to be effective pattern recognizers that show properties similar to those of the human system. Moreover, these models may also have the capacity to store and represent information in ways that can be compared with human behavior.

In this paper we focus on the capacity of neural net models to handle lexical representation, and, in particular, to represent the form similarity (phonological and orthographic) between words. First, the problem of lexical representation and process will be identified. In so doing, the psychological evidence pointing to a lexical organization based on form similarity will be considered briefly. Then we sketch diverse connectionist techniques for representing similarity with local and distributed nets. Next, we discuss some attempts to model similarity with self-organizing feature maps (SOFM). We present some of our own simulation results that illustrate some problems associated with this approach for our purposes. Lastly, we consider the potential of some alternative algorithms to deal with the observed limitations of standard SOFMs for representing lexical similarity neighborhoods.

2. Lexical representation and process

The human ability to understand spoken language depends in large part upon the efficiency and rapidity with which words can be retrieved from the mental lexicon. By storing form and meaning information together, the lexicon solves the difficult problem of the arbitrary mapping from sensory input to a meaningful interpretation. In studying this ability, the issues of how words are represented in the lexicon and how they are recognized on the basis of sensory and contextual information must be addressed.

Most current psycholinguistic models describe spoken word recognition as a matching process in which some internal representation of the speech input, the **input representation,** is matched with internally stored **lexical representations.** Thus there are three basic components to be specified in defining these models: the input and lexical representations, and the mechanism for matching the two.

Considerable empirical research has been devoted to characterizing these three components. Experiments in speech perception [6] have aimed at determining which information is extracted from the acoustic signal and what types of internalized input representation are computed. The outcome of this research suggests an immediate sequential analysis of the signal, but is still inconclusive about the unit(s) representing this information (e.g., phoneme, syllable).

At the lexical level, there is an emerging consensus that listeners first activate a set of lexical candidates on the basis of the signal and then select the appropriate lexical entry from this set [8]. However, the precise definition of the lexical candidates and the way in which these competitors affect target recognition is still under intense investigation. Considerable disagreement persists as to what form properties the competitors must share with the target to be activated - for example, the onset or offset - and how the activated competitors influence target recognition - for example, through inhibition. What appears certain is that the time-course of word recognition can only be specified with reference to the set of lexical competitors from which a given target word must be discriminated.

In what follows we will examine how the representations produced by neural networks can shed light on the competitor set and its influence on word recognition.

3. Representation in connectionism

Although the focus of connectionist models has mainly been on the problem of pattern recognition, it has expanded more recently to issues of representation as well. This is partially in reaction to the serious criticism of the connectionist models by advocates of symbolic paradigm [2].

3.1 Representing words

Connectionist approaches to representation can be distinguished according to whether they represent specific hypotheses (i.e., words, concepts, etc..) in a **local** or **distributed** fashion. In the extreme form of localist models, each hypothesis is represented by the activity of a single unit. At the other extreme, distributed models assume that each hypothesis is represented by the pattern of activity across a number of different units and each unit is involved in the representation of more than one hypothesis.

There are a number of intermediary representational schemes that are combinations of local and distributed approaches. For example, in a locally distributed approach, a collection of units can serve to represent a hypothesis where the activated units are expected to be centered around the best matching unit. Each representation scheme has its advantages and disadvantages. The main interest of local representations is their transparency since each unit is labeled. Randomly distributed representations are harder to interpret but are more economical (more hypotheses can be stored in few units). Locally distributed representations are similar to local representations with respect to their transparency.

3.2 Representing similarity relations

Several ways of representing similarity exist within the connectionist framework. First, similarity can be expressed indirectly in local representations such as those assumed by interactive activation models. In the model Trace [10], for example, the similarity between two word units can be identified by looking at their level of activations as a function of the input. When a particular target word is presented as input to the model, other words sharing the same phonemes are activated in proportion to their match with the input. In this way the shared levels of high activation indicate form similarity.

Similarity between hypotheses can be identified in distributed networks more or less directly depending upon the representation assumed. Sharkey [15] makes the distinction between symbolic and subsymbolic micro-features. The number of shared symbolic micro-features provides a straightforward measure of similarity [11] since these micro-feature units are labeled and interpretable. For subsymbolic representation, the similarity is harder to detect since the hidden units do not have specific predetermined roles as is the case for symbolic micro-features. Nonetheless despite the fact that each processing unit is unlabeled, it is possible by means of different clustering techniques [4] to analyze the activation levels or the weights associated with the hidden units. Examples of such analyses can be found for visually presented words [17] and for phonemes [1].

Similarity can also be expressed spatially such that units (or collections of units) that are similar are located in physical proximity. An example of this approach can be found in self-organizing feature maps. Here, the similarity between two words can be expressed by the distance (e.g., Euclidean) separating the two corresponding units (or groups of units).

Since we are interested here primarily in how similarity can be expressed and how lateral and hierarchical interactions can be modeled, we will focus on the self-organizing systems with locally distributed representations. In what follows we will consider different attempts to model similarity with these models.

4. SOFMs and similarity

Several neural network models have been developed to represent different types of linguistic data spatially or topographically through an unsupervised learning process. Let us briefly consider research that investigates phonological and semantic relations below.

Kohonen [7] has produced two-dimensional phonotopic maps formed by self-organization. These maps display similarity relations between phonological units using short-time spectra extracted from speech as input. They partially reflect metric distance relations between these phonological units. This is possible since many phonemes can be characterized roughly by two dimensions (i.e., their first two formants). Words can be visualized in these maps as a trajectory through the two-dimensional space.

At a higher level, Ritter and Kohonen [13] extracted and organized words according to their meanings on the basis of sentence inputs. In the resulting semantotopic maps, words with similar meanings were placed in physical proximity, thereby creating regions in the map for words with shared semantic properties. As the authors themselves admit the input was extremely simple both in terms of syntactic structures and in the meaning of the words in the sentence. It remains an important question as to how well this approach can handle more complex problems.

An attempt to link form and meaning with SOFMs can be found in the work of Miikkulainen [12]. This model includes two different maps - one for form and the other for meaning representations. These two maps were linked with connections that were determined by the Hebb-rule. Within each map, the standard SOFM algorithm was used to achieve organization and express similarity. Since the inter-map links were derived from the activation patterns, the resulting organization and its limitations did not influence the quality of the links.

At a higher level, Scholtes [16] combined semantic and contextual maps to extract semantic relationships. To do so, he used recursive links that provide a short-term memory allowing the model to input and process words sequentially. To organize and express relations Scholtes also uses the standard SOFM algorithm.

The goal of each of these SOFM studies is not explicitly to deal with psychological constraints. Rather their primary objective was to show that the system could perform (i.e., accurate phoneme recognition).

5. Constraints on computational models

The psycholinguist's objective in modeling is to make explicit what is known about lexical processing in the form of a computer program and to use this model to make novel predictions that can be tested experimentally. It is important therefore to incorporate known psychological constraints as far as possible. There are, however, important implementational constraints that also must be reckoned with. These latter constraints are often in conflict with the psychological constraints forcing a trade-off or compromise. These compromises exist at each step in the modeling process starting with the definition of the input to the kinds of lexica presented to the model. We now consider these constraints.

Ideally, the computer model should take real speech as input. However, given the difficulty of automatic speech recognition [19], it is common practice in computer modeling of lexical processing to use **mock** input. This corresponds to some symbolic representation (i.e., string of phonemes or phones). It is important that this mock input render as closely as possible both the information extracted by the listener from the signal and the way in which the listener receives and processes this information (continuous and overlapping speech). Various types of mock input have been used in the literature.

An important aspect of lexical processing that needs to be expressed in the model is the word recognition process. It is of particular interest to be able to study the time-course of word recognition and to trace the changes in the activation states of the lexical entries across time as a function of the input. From a psychological perspective, it is desirable to model with a lexicon that is representative of a native speaker's lexical knowledge. However, complete lexica of 50,000 - 100,000 words is beyond the scope of most models. It is necessary to start with smaller lexica and to gradually scale up to larger ones. These lexica should constitute a representative sample. Another crucial aspect of modeling the lexicon is expressing the different types of relations between the individual lexical entries. In particular, it is important to be able to determine how the individual words are represented and how the relations between them can best be expressed.

6. Mapping lexical representation

In this section we present the results of several simulations that examine the spatial organization produced for two different lexica with SOFMs. For more details, we refer the reader to the study of Hogeweg [5]. In this mapping process, there are two important steps. In the first, the inventory of words, the word input space, must be coded in terms of vectors. In the next step the vector space is mapped onto feature maps. These two steps are shown in Figure 1.

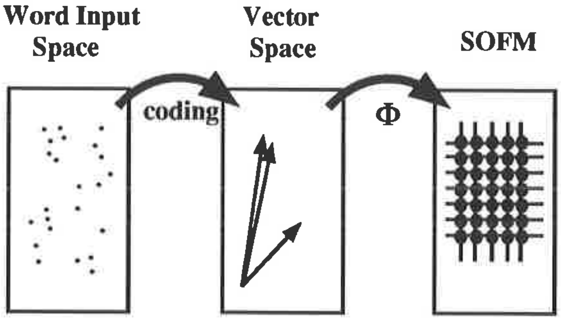

Figure 1: Transformations involved to map words to a SOFM

6.1 Simulation method

The Kohonen algorithm imposes several constraints upon the coding schemes. Each word must be represented by a unique vector. The vectors must also all be of the same dimension. Furthermore, the vector must also be presented simultaneously, unlike real speech which come in over time.

For the present purposes we used the following coding scheme. Each phoneme is coded as a 17-

dimensional vector where each dimension stands for one (binary) distinctive feature. Each word thus consists of a sequence of such 17-dimensional vectors, one for each phoneme. This coding approximates real speech in that distances between phonemes are expressed. However, there are no smooth transitions between phonemes as in real speech for which there is high auto-correlation. The code also does not code the internal linguistic structure (syllable structure) of the words directly.

Several different toy lexica (6) were selected from the CELEX database. Each lexicon contained words with different phonological relations. Here, we will only report about the results for a 24-words lexicon with several clusters of neighbors all of the same length and a 22-word lexicon with insertion neighbors and different lengths. These neighbors differed in one phoneme in any position. Similar effects as presented here have been found for the other lexica.

The map took the form of a two dimensional array of 100 (10 x 10) neurons. We used both circular and non-circular maps. In the circular maps a toroid-like continuous surface is modeled by connecting the sides of the map. Due to the high dimensionality of the input space these maps performed much better than non-circular maps. With the non-circular maps most of the words were represented by neurons at the edge of the maps even after many iterations.

Our experiments used the standard h(r,s) adaptation function in which exponential behavior determines the size of the affected area and the learning rate over time.

The vector word representations were presented to the network in random order with uniform frequency. We tested the behavior of the maps after 4000 and 100000 iterations with different random seed values for the initial weights. After the training phase, the words were presented once more. The neuron with the weight vector that best matched the input vector was labeled with that word. If there was more than one input vector in the receptive field of a single neuron, then this neuron represented two words.

6.2 Results

In this section, some selected results of the simulations are presented. To evaluate these results, we first can consider a map resulting from a simulation with 4000 iterations as shown in figure 2. If a neuron represents a word, then this word is printed on that lattice node.

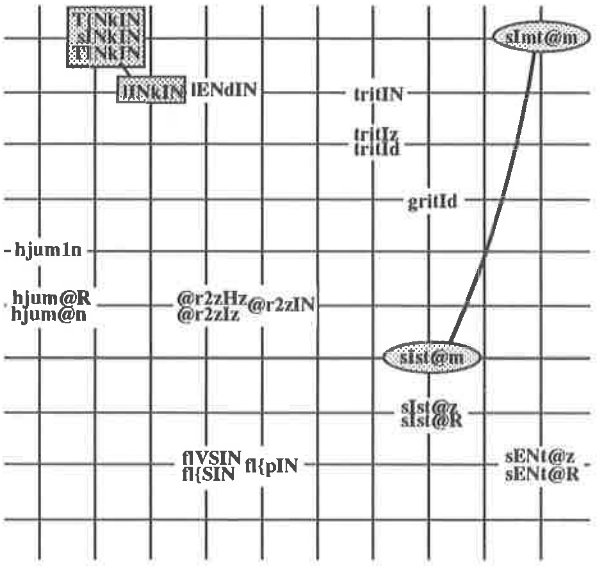

Figure 2: resultant SOFM for lexicon 1 after 4000 iterations, circular map used

This figure shows that the map clusters certain words that are neighbors according to the definition that we have adopted. However, in other instances lexical neighbors are not located close together in the map. It is of course impossible to consider all lexical relations here. Rather we will examine a few specific instances in which relations in word input space were not preserved in the resultant map. This will help us to identify the limitations of this approach.

Table 1:Words with inconsistent mappings

Word pair	distance	replacement
sINkIN-TINkIN	0	1
lINkIN-TINkIN	2	1
lINkIN-lEndIN	1	3
sImt@m-sIst@m	6	1
sIst@z-sEnt@z	3	2

Table 2: Words with inconsistent mapping

Word Pair	distance	insertion
@rEst - @rEsts	0	1
@rEst - @drEst	7	1

Table 1 gives some examples of words for which the distance in the map - in terms of number of intervening nodes - does not correspond to their phonological distance. This latter distance is computed as the minimum number of phonemes that need to be replaced or inserted to get from one word to the other. The first two word pairs with the same phonological distance (1 replacement) are at different distances in the map. Such inconsistencies in the map are even more salient for the second group of examples.

For another lexicon we obtained the mappings like those listed in table 2. These examples show the influence of the position of the inserted phoneme. Insertions near word onsets produce larger distances than those at offsets where there is no apparent effect of the inserted phoneme.

The results of other simulations with lexica of mixed length showed a general trend for words that were equal in length to be clustered together - even though they were not similar in their form properties. This is the result of the coding constraint imposed by the SOFM algorithm.

To assess the quality of the maps more globally, we also computed correlations between the distances between words in the input space, in vector space and in the resultant maps. We restrict ourselves here to the correlations between the vector space and the map distance for words within clusters and for all words. The maps produced high correlations of around 0.9 for words within the clusters. The correlation of distances between all words was lower (0.7).

Lastly, we also investigated the influence of the random values attributed to initial weights. The within and between cluster configurations varied dependent on the random seed used to set the initial random, weight values. Normally the set of members of clusters remained roughly constant. However their spatial configuration changed. The same hold for the spatial configuration of different clusters.

6.3 Discussion

The lexical organization resulting from our simulations with the standard SOFM algorithm proves not to be adequate for our purposes. A careful analysis of the resulting maps reveals their limitations even for this relatively simple problem. For example, the spatial organization obtained in the maps cannot be interpreted quantitatively as is required to introduce lateral effects between word units. This result raises the concern whether SOFMs are really able to represent similarities and extract generalizations as the language-oriented research described above requires.

Three major problems can be distinguished in the lexical modeling performance of SOFMs. The first has to do with the coding used, the second with the complexity of the problem and the third with the behavior of the algorithm itself. Here, we will only consider the last two.

The mappings performed in our simulations are very complex: reducing a high dimensional input space to a two-dimensional structure. The input code includes 17 binary values for each phoneme making up the word. It is impossible to obtain a topology preserving mapping with so much information reduction. As the maps show there are foldings and other non-linear transformations. These prevent us from being able to interpret the distances between word units quantitatively.

It is clear that using a higher dimensional map does not make sense in our application. The word input space is a complex non-Euclidean space with clusters with extreme differences in density and dimensionality. To solve the mapping problem, a brute force technical solution as SOFM is not appropriate. Rather one should abandon the fixed grid scheme to represent the similarity relations.

The central role of the h(r,s)-function in the standard SOFM algorithm leads to many problems. This function has to serve various conflicting purposes such as global organization, local organization, prototype vector distribution, and convergence. The larger and more complex the task, the more difficult it is to find a good compromise in the definition of this function. To define more convincing models one has to step away from algorithms which attempt to achieve a global organization.

Another well-known limitation of the standard SOFMs is their inability to deal with time-sequential behavior. The maps are static and cannot provide information about the time-course of processing. However, Zandhuis [20] has shown how the standard approach can be modified to deal with this problem. Another related problem has to do with incremental learning which is not possible in standard SOFM. Some suggestions for solutions to this problem have also been made [3].

7. Other algorithms

Recently some new algorithms have been described to overcome some of the severe limitations of the standard SOFM method. Ritter [14] developed Random Nets (RNs) with the primary objective of escaping the fixed grid structure and therefore avoiding the fixed dimensionality in a map too. Preliminary experiments show that large groups of semantically related words tend to be represented by clusters of connected neurons. Martinetz and Schulten [9] developed the idea of Neural Gas (NG) mainly to improve upon the non-optimal vector quantization performance of the standard SOFM algorithm. These authors included a separate step in which links are successively constructed between the two closest prototype vectors. With the proper parameters, NG achieves a link structure in which all neurons having adjacent receptive fields are connected (i.e., Delaunay triangulation).

8. Relational SOFM

In the preceding, we have pointed to some limitations in the spatial organization obtained in self-organizing feature maps using the Kohonen algorithm. In this section we explore an alternative approach: rSOFMs [18]. Here, form similarities between words are represented in terms of the strength of the connections or links between the neurons that represent them. In parallel with some suboptimal organization process which need not to be optimal (in the first version we still use Kohonen's algorithm), links are established between the units.

Since the activation of the neurons provides a good measure of similarity when the input and weight vectors are normalized (v",w"), it makes sense to use the Hebb-rule shown below.

$$\Delta s_{ij} = \alpha * a_i * a_j$$

with $p_i = \sum_i v_i'' * w_i''$ and $a_i = f(p_i)$, (f = sigmoid fct.)

$\sum_j s_{ij} = c$ (i = best match neuron)

While the map is organizing itself by adapting the weight vectors stepwise, the strengths of the links are adapted too. In rSOFM1 only the links between the best-match neuron and all the other neurons in the map are adapted. Of course, the strengths of the links have to be balanced or normalized, since the Hebb-rule itself leads to a monotonic increase of the connection strengths. With an appropriate value of c, highly similar neurons (for example, variants of the target word) will facilitate each other. Other less similar word units (for example competitors of the target word) will inhibit each other, since the strengths of their links are negative. Ideally, this algorithm leads to bell-shaped functions as shown in figure 4.

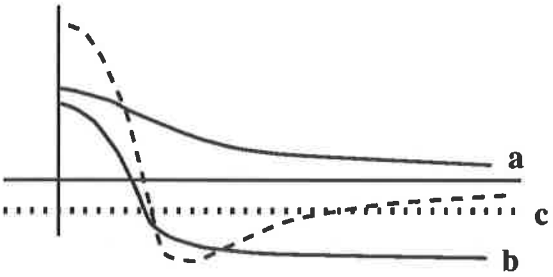

Figure 4: Idealized activation function (a), idealized function of the link strengths (b), and the resulting influence function (dashed line), c is the constant used to normalize the link strengths. The horizontal axis specifies the similarity distance between the best match and competitor units.

The link strengths between neurons decrease in a monotonic fashion with decreasing similarity between the words. Since both functions, the expected activation pattern and the link strengths, are bell-shaped, the resulting influencing function has a Mexican-Hat-like shape.

By combining these links with an appropriate organization algorithm, it is possible to overcome some of the main limitations described above as for example incremental learning, dimensionality considerations, scaling limitations, amount of iterations during training, and especially expressing the similarity relation.

The rSOFM1 algorithm, however, also faces several problems as for example chosing the constant c, limiting the linked words to those which have a certain degree of similarity, and extending the adaptation rule to the links of all neurons proportional to their similarity to the best match neuron.

9. Conclusions

In this paper, we have reported on our attempts to use the standard SOFM algorithm for psycholinguistically motivated modeling of the mental lexicon. After having carefully studied and tested the SOFM-technique we conclude that it is too limited to be of use for our purposes. The standard Kohonen algorithm produces mappings which only partially preserve the topology of the input space. The resulting distances in the SOFMs cannot be interpreted quantitatively. Since our main interest is to model the lateral interactions between neuron clusters as a function of the similarity between the words which they represent, we need an alternative method.

We have briefly presented several new algorithms that constitute improvements over the standard SOFM approach and tried to show that the class of algorithms which we call "relational SOFM" has some attractive properties. By allowing word maps to organize themselves with some algorithm and simultaneously expressing the form similarity between words in terms of the strengths of the links that connect the corresponding neurons, we have a more flexible and attractive mechanism to model relations within a hierarchical layer. The chosen technique can be extended to model bottom-up and top-down interactions too. It still has to be determined whether "relational SOFM" is adequate for handling additional effects documented in the psycholinguistic literature (e.g., phonotactic rules, bottom-up inhibition, top-down effects) in order to construct a psychologically motivated model of word recognition.

Acknowledgments

The authors would like to thank Bob Boelhower, Alex Hogeweg, Sikko Sikkema, and Jan Zandhuis for their contributions at various stages of this research.

REFERENCES

1. Elman, J. L., (1988). Finding the structure in time. TR 880, CRL, University of California, San Diego.

2. Fodor, J. A., & Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: a critical analysis. *Cognition,* 28, 2-71.

3. Fritzke, B. (1991). Self-Organizing Feature Maps with Problem Dependent Cell Structure. In T. Kohonen et al. (Eds). *Artificial Neural Networks.* North Holland.

4. Hanson, S.J. & Burr, D.J. (1990). What connectionist models learn: Learning and representation in connectionst networks. *Brain and Behavioral Science,* 13, 3, 471-518.

5. Hogeweg. H. (1991). Modelling the human mental lexicon with a Kohonen neural network. Technical University, Enschede.

6. Klatt, D. H. (1980). Speech perception: A model of acoustic-phonetic analysis and lexical access. In R. A. Cole (Ed). *Perception and production of fluent speech.* Hillsdale, N.J.: Lawrence Erlbaum Associates, 243-288.

7. Kohonen, T. et al. (1984). Phonotopic Maps - Insightful representation of Phonological Features for Speech Recognition. *Proc. IEEE Seventh Conf. Pattern Recognition* (IEEE Computer Society), 182-185.

8. Marslen-Wilson, W. D. (1987). Parallel processing in spoken word recognition. Functional parallelism in spoken word recognition. In U. H. Frauenfelder & L. K. Tyler (Eds). *Spoken word recognition.* Cambridge, MA: MIT Press.

9. Martinez, T. & Schulten, K. (1991). A "Neural Gas" Network Learns Topologies. In T. Kohonen et al. (Eds). *Artificial Neural Networks.* North Holland.

10. McClelland, J. L., & Elman, J. L. (1986). The TRACE model of speech perception. *Cognitive Psychology,* 18, 1-86.

11. McClelland, J. L., & Kawamoto, A. H. (1986). Mechanisms of sentence processing: assigning roles to constituents. In J. L. McClelland & D.E. Rumelhart (Eds). *Parallel Distributed Processing,* Vol. 2. MIT, Cambridge, MA.

12. Miikkulainen, R. (1990). A Distributed Feature Map Model of the Lexicon. Technical Report UCLA-AI-90-04.

13. Ritter, H., & Kohonen, T. (1989). Self-Organizing Semantic Maps. *Biological Cybernetics,* 61, 241-254.

14. Ritter, H. (1991). Learning with the Self-Organizing Map. In T. Kohonen et al. (Eds). *Artificial Neural Networks.* North Holland.

15. Sharkey, N. E. (1992). Connectionist Representation Techniques, TR 217. University of Exeter.

16. Scholtes, J.C. (1991). Recurrent Kohonen Self-Organization in Natural Language Processing. In T. Kohonen et al. (Eds). *Artificial Neural Networks. North Holland.*

17. Seidenberg, M. S., & McClelland, J. L. (1989). A Distributed, Developmental Model of Word Recognition and Naming. *Psychological Review,* Vol. 96, No. 4, 523-568.

18. Wittenburg, P. (1992). Relational Self-Organizing Feature Maps. Technical Report MPI-NL-TG-3/92.

19. Wittenburg. P., & Couwenberg, R. (1991). Recurrent Neural Networks as Building Blocks for Word Recognition Models. *Proceedings of the Eurospeech Conf.,* 1015-1019, North Holland.

20. Zandhuis, J.A. (1992). Storing Sequential Data in Self-Organizing Feature Maps. Technical Report MPI-NL-TG-4/92.