Proceedings seventh BENELEARN conference, 109-118, 1997

When small disjuncts abound, try lazy learning: A case study

Antal van den Bosch*, Ton Weijters*, H. Jaap van den Herik*, Walter Daelemans**

* Department of Computer Science
Universiteit Maastricht
P.O. Box 616
NL-6200 MD Maastricht
The Netherlands
{antal,weijters,herik}@cs.unimaas.nl

** ILK, Computational Linguistics
Tilburg University
P.O. Box 90153
NL-5000 LE Tilburg
The Netherlands
Walter.Daelemans@kub.nl

Abstract

Machine learning is becoming recognised as a source of generic and powerful tools for tasks studied and implemented in language technology. Lazy learning with information-theoretic similarity matching has appeared a salient approach, demonstrated to be superior over other machine-learning approaches in various comparative studies. It is asserted both in theoretical machine learning and in reports on applications of machine learning to natural language that the success of lazy learning may be due to the fact that language data contains *small disjuncts*, i.e., small clusters of identically-classified instances. We propose three measures to discover small disjuncts in our data: (i) we count and analyse indexed clusters of instances in induced decision trees; (ii) we count clusters of *friendly* (identically-classified) instances immediately surrounding instances by using similarity metrics from lazy learning; (iii) we compare average sizes of friendly-instance clusters using different similarity metrics. The measures are illustrated by a sample language task, viz. word pronunciation. Two conclusions are arrived at: (i) our data indeed contains large amounts of small disjuncts of about three to a hundred instances, and (ii) there are important differences in feature relevance in the data, exploited appropriately when lazy learning is augmented with information-theoretic similarity matching. We claim that the measures introduced in this paper are useful for predicting the suitedness of lazy learning in general.

1 Lazy learning of language tasks

In language technology, machine learning is becoming recognised as a source of powerful generic tools for learning complex language tasks (Daelemans, Van den Bosch, and Weijters, 1997b). *Lazy learning* (Aha, Kibler, and Albert, 1991; Aha 1997) has been demonstrated to be an especially salient approach to learning various language tasks such as grapheme-phoneme conversion, stress assignment, morphological segmentation, and part-of-speech tagging; for an overview, we refer to Daelemans *et al*. (1997). In many comparative studies, lazy learning is shown to outperform traditional linguistics-based models and other machine-learning approaches such as decision-tree learning and connectionist learning, when applied to various language tasks (Daelemans *et al*., 1997; Van den Bosch, 1997). In these comparative studies, lazy learning augmented

with an information-theoretic weighted similarity function, IB1-IG consistently offers the best generalisation performances (Daelemans and Van den Bosch, 1992; Daelemans, Van den Bosch, and Weijters, 1997a).

Lazy (or *instance based*) learning is the common term for a class of learning algorithms descending from the k-nearest neighbour (k-NN) algorithm (Cover and Hart, 1967; Devijverand Kittler, 1982; Aha *et al*., 1991). Lazy learning is based on the hypothesis that performance in cognitive tasks (e.g., language tasks) is based on computing the similarity between new instances of the task and stored representations of instances encountered earlier, rather than on the application of mental rules abstracted from earlier experiences (Aha *et al*., 1991; Aha and Goldstone, 1992; Daelemans, 1995). Instance-based learning is commonly referred to as *lazy* due to the minimal effort put in the learning process.

This paper offers concrete estimates relating to two assumptions on the suitability of applying lazy learning to language tasks:

1. When classes are spread disjunctively in an instance space over small clusters of instances (*small disjuncts*) in a data set, lazy learning is generally assumed to be the preferred learning approach yielding the best performance (Holte, Acker, and Porter, 1989).
2. Two inherent data characteristics of language data are assumed to cause
   (a) the high performance of lazy learning, viz. that instances occur in *pockets of exceptions* (Daelemans, 1995) (which may be equivalent to the small disjuncts mentioned in (1)), and
   (b) the consistent superiority of information-theory-weighted lazy learning in particular, viz. that instances of language tasks display substantial differences in feature relevance (Daelemans, 1995; Van den Bosch, 1997).

We provide characterisations of the two assumptions in section 2. The remainder of the paper is devoted to proposing and testing concrete measures for testing the assumption. The tests of the proposed measures are performed on data representing a sample language task, viz. learning word pronunciation. Section 3 briefly introduces the sample task and provides the empirical results of a comparative study with learning algorithms applied to the task. In section 4 we investigate the following:

1. We count and analyse the clusters of instances represented at end nodes in the decision trees generated by IGTREE, applied to the word-pronunciation task.
2. We measure the clusteredness of instances with IB1 applied to the word-pronunciation task.
3. We measure the clusteredness of instances with IB1-IG, and compare it with those measured with IB1 applied to the word-pronunciation task, and two distorted variations on the task.

In section 5, we formulate the conclusions to be drawn from the present study.

2 Small disjuncts and feature relevance

All language tasks can be defined as classification tasks (Daelemans, 1995). Given a corpus of examples of the task, data bases of these examples can be built to which machine-learning algorithms can readily be applied. In such data bases, *pockets of exceptions* tend to occur; "Exceptions tend to come in 'families' " (Daelemans, 1996b, p. 5). Keeping members of such families in memory, enables accurate classification of members of the same family occurring in new, unseen data. It can therefore be expected that lazy learning, which keeps all training material in memory, will be able to deal better with reoccurring family members of stored families of exceptions in test material, than decision-tree learning algorithms or supervised connectionist algorithms, which show the tendency to ignore small groups of exceptions considered to be noise. The observation of 'families of exceptions' relates to an observation in machine-learning research that "IB1 performs well for highly disjunctive target concepts" (Aha, 1992, p. 6). A target concept, i.e., a class, is highly disjunctive when the instances of that class are very dissimilar globally, and similar locally. In data sets containing highly disjunctive target concepts, similar instances of the same class are only found in small clusters (Daelemans' (1996) families). The expectation is that IB1 will perform generally better than eager-learning algorithms on such data since it retains all information concerning disjuncts, no matter how small, while decision trees, notably those implementing pruning (Quinlan, 1993), tend to overgeneralise and miss out on disambiguating small disjuncts (Holte *et al*., 1989; Ali. 1996). Our working assumption here is, by abduction, that language data contains many small disjuncts.

Apart from 'families of exceptions', many language tasks display outspoken differences in feature relevance; language data is for a considerable part redundant (Zipf, 1935), and contains fairly localised *hot spots* of information. Therefore, using information-theoretic functions for similarity matching in lazy learning, e.g., as in IB1-IG (Daelemans, Van den Bosch, and Weijters, 1997a), may be advantageous in learning language tasks. With information-theoretic weighted similarity matching, the multi-dimensional feature-value space is rescaled in such a way that instances matching on important features are regarded as more similar to each other than instances matching on an irrelevant feature. It is unclear, however, what the exact influence is on the functioning of lazy learning with information-theoretic weighted similarity matching, as compared to flat weighting in IB1. In sum, our working assumption is that with IB1-IG, clustering of families of instances in the instance base improves.

3 Word pronunciation: the GS task

Converting written words to stressed phonemic transcription, i.e., word pronunciation, is a well-known benchmark task in machine learning (Stanfill and Waltz, 1986; Sejnowski and Rosenberg, 1987; Dietterich, Hild, and Bakiri, 1990). Here, we provide an overview of experiments performed on the word-pronunciation task, i.e., the conversion of fixed-sized instances representing parts of words (windows), to a class representing the phoneme and the stress marker of the instance's middle letter. The task, henceforth referred to as GS (Grapheme-phoneme conversion and Stress assignment) is similar to the NETTALK task presented by Sejnowski and Rosenberg (1986), but is performed on a larger corpus of 77,565 English word-pronunciation pairs (Van den Bosch, 1997). Converted into fixed-sized instance, the full instance base representing the GS task contains 675,745 instances. The task features 159 classes. When neither of these classes would be disjunctively

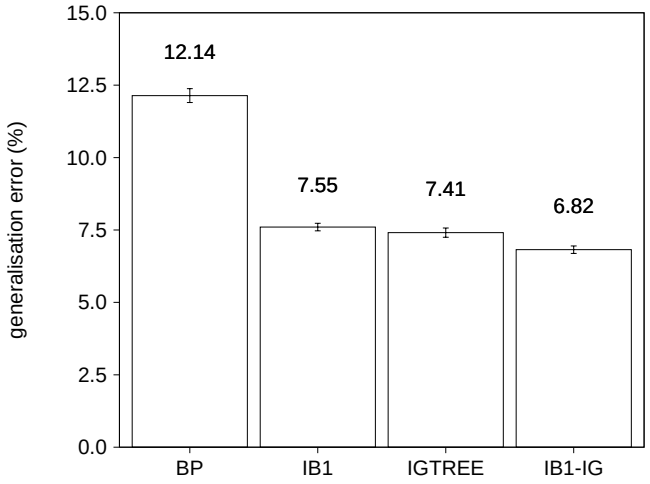

Figure 1: Generalisation performance results on the GS task in terms of percentage incorrectly classified test instances of four learning algorithms.

clustered, one would expect an average cluster size of $675,745/159 = 4249$ instances.

Van den Bosch (1997) reports on 10-fold cross validation experiments performed with back-propagation (Rumelhart, Hinton, and Williams, 1986), the decision-tree algorithm IGTREE (Daelemans *et al*., 1997a), and the two lazy-learning algorithms IB1 (Aha *et al*., 1991; Daelemans *et al*, 1997a) and IB1-IG (Daelemans and Van den Bosch, 1992; Daelemans *et al*., 1997a). Figure 1 displays all generalisation errors in terms of incorrectly classified test instances. A test instance is classified incorrectly when the phoneme part is misclassified, or the stress-marker part, or both.

The results displayed in Figure 1 indicate that IB1-IG performs best on test instances. The differences between IB1-IG and the other algorithms are significant, the smallest difference being between IB1-IG and IGTREE ($t(19) = 9.05$, $p < 0.001$). The results thus display a superior performance of IB1-IG.

4 Measuring small disjuncts

4.1 Counting small disjuncts in decision trees

Small disjuncts can be discovered by counting disambiguated instances at different levels in trees constructed by IGTREE. During the construction of a tree, IGTREE is assumed to disambiguate as large numbers of instances as possible and as early in the tree as possible. The strategy of IGTREE is to detect clusters of instances and to represent them by paths, ending at leaf nodes. When IGTREE is able, for example, to construct a path ending in a leaf node at level three, representing the disambiguated classification of 100 instances, it has discovered three feature-value tests indexing a cluster of instances of size 100 of the same class. The assumption underlying the clustering of instances at leaf nodes in IGTREE is that information gain, computed over the full instance base, provides an adequate approximation of the ordering of features to be investigated for maximising

level	average # instances per leaf		
1	5	6.91	± 2.73
2	172	15.56	± 3.72
3	3413	16.02	± 2.87
4	18842	8.08	± 0.98
5	32017	5.46	± 0.74
6	22565	3.85	± 0.18
7	24208	6.36	± 0.95

Table 1: Numbers of leafs and average numbers of instances represented by these leafs (with standard deviations), for each of the seven levels in trees constructed by IGTREE on the GS instance base.

the numbers of disambiguated instances as high as possible in the tree [1] (Daelemans *et al.* 1997a).

Table 1 lists the numbers of leafs, and average numbers of instances represented by those leafs, per level, produced by IGTREE on the full GS instance base. Two attributes of the tree appear salient: first, at levels 2 and 3 of the tree, IGTREE is able to form clusters of (on average) 16 instances of the same class. Thus, by deciding on two or three feature-value tests, approximately 3,500 clusters of size 16 (on average) can be identified. Second, at deeper levels in the GS tree, clusters size decreases, though on average it remains at three instances or more. For example, on level 4, clusters contain on average approximately 5 instances (the small standard deviations in Table 1 indicate that the average number of instances per cluster per level is quite stable).

We conclude from these results that, under the assumption that IGTREE performs an adequate clustering of the data (which is biased by the specific choice of information-gain feature ordering), the instances in the GS instance base are clustered in small disjuncts of size three to sixteen, on average. It is likely that given enough training instances, one of the minimal two or three instances of a disjunct are stored in memory – each of these single instances can then serve as *mini-prototypes* for test instances belonging to the same disjunct. Thus, storing all training instances of the GS task is expected to favour IB1 and IB1-IG, and disfavour approaches which stop storing information below a certain utility threshold, ignoring small disjuncts, such as decision-tree learning with pruning (Quinlan, 1993; Holte *et al.*, 1989).

4.2 Counting friendly-neighbour clusters with IB1

An alternative method of detecting clusters less biased than the information-gain ordered cluster detection by IGTREE, can be implemented using IB1. We performed leaving-one-out experiments (Weiss and Kulikowski, 1991) in which we computed for each instance in the GS data set a ranking of the 100 nearest neighbours in terms of their distance to the left-out instance. Within this ranked list, we count the ranking of the nearest neighbour of a different class than the left-out instance. This rank number minus one is then taken as the cluster size surrounding the left-out instance. If, for example, a left-out instance is surrounded by three instances of the same class at distance 1 (i.e., one mismatching feature-value), followed by a fourth nearest-neighbour instance of a different

[1] It should be noted that C4.5 (Quinlan, 1993) arguably provides a better approximation than IGTREE, since it adjusts the information-gain ordering of features at every non-ending node in order to maximise the numbers of instances that can be disambiguated as high in the tree as possible.

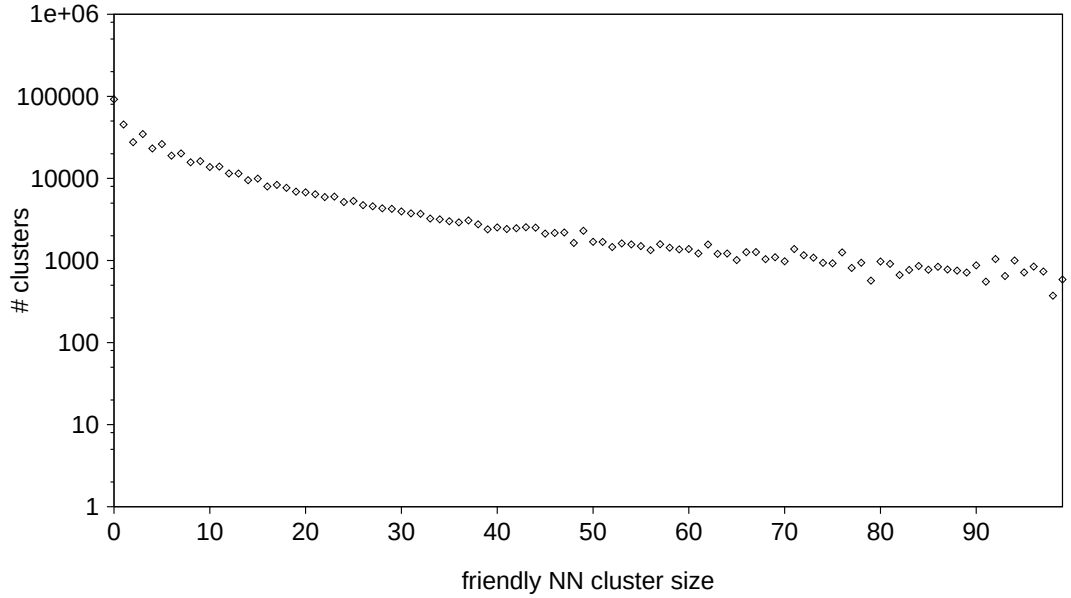

Figure 2: Scatter plot of numbers of friendly-neighbour clusters of sizes 0 to 99, as found by IB1 on the GS data set.

class at distance 2, the left-out instance is said to be in a friendly-neighbour cluster of size three. The results of the leaving-one-out experiment are displayed graphically in Figure 2.

The x-axis of Figure 2 denotes the numbers of friendly neighbours found surrounding instances; the y-axis denotes (in logarithmic scale) the occurrences of friendly-neighbour clusters of particular sizes. IB1 is able to find thousands to ten thousands of clusters of sizes 1 to 60, and still hundreds of clusters containing more than 60 friendly neighbours. Combining these results obtained with IB1 with those obtained with IGTREE, we can conclude that the GS data contains many tens of thousands of small disjunct clusters containing about three to about a hundred instances each.

4.3 Counting friendly-neighbour clusters with IB1-IG

Having collected broad indications for the degree of disjunctive clusteredness of the data, which generally favours lazy learning over greedy approaches (Holte *et al.*, 1989), we now turn to searching a relevant characteristic of our data favouring IB1-IG over IB1. To compare IB1 and IB1-IG, we performed the same leaving-one-out experiment with IB1-IG as described above to compute the numbers and sizes of friendly-neighbour clusters.

We extend the comparison as performed on IB1 in section 4.2 on IB1-IG by introducing two additional data sets, which are derived from the GS data set by systematic distortions:

1. In the first data set, called GS-PERM, we permute randomly for each word all letters along with their corresponding stressed phonemes; i.e, for each word, all letter-phoneme correspondences are shuffled. GS-PERM distorts the context around focus letters, but leaves the correspondences between the focus letters and phonemes intact. This means that the information-theoretic similarity function of IB1-IG is able to detect that the middle letter in an instance is highly relevant for classification, and that all remaining context letters are

data set	average # friendly neighbours IB1	IB1-IG
GS	15.01	25.58
GS-PERM	0.50	3.41
GS-RAND	0.11	0.11

Table 2: Average numbers of friendly (identically-classified) nearest neighbours of GS instances, measured with IB1 and IB1-IG.

irrelevant. IB1 is not taking feature relevance into account, and will miss out on the relevance of the middle letter.

2. The second data set, called GS-RAND, randomises for each word all letters. While the stressed-phonemic transcription is maintained, all letters of the word are randomly picked from the letter alphabet. This distorts all letter-phoneme correspondences, and makes the relation between spelling and pronunciation fully arbitrary (not unlike ideographic writing systems). The information-theoretic similarity function of IB1-IG will not detect any (significant) differences in feature relevance, which will reflect chance and thus be low.

Table 2 displays the average size of friendly-neighbour clusters found by IB1 and IB1-IG in the GS, GS-PERM, and GS-RAND data sets, averaged over all instances. It provides three relevant indications. First, comparing IB1 and IB1-IG on the GS task, it can be seen that larger clusters of friendly neighbours are found with IB1-IG than with IB1. Thus, by weighting the distance function with information gain, a metric from information theory, IB1-IG is able to rescale the instance space in such a way that instances of the same class become surrounded with more instances of the same class as compared to when the similarity function of IB1 is used (Daelemans *et al*, 1997a). Second, the results in Table 2 on the GS-PERM data set indicate that with IB1, the average friendly-neighbour cluster size is smaller than 1, i.e., most nearest neighbours are of a different class. With IB1-IG, however, the average friendly-neighbour cluster size is 3.41: on average, each instance is surrounded by over 3 instances of the same class. Perturbing the context around the focus letter causes the flat-weighted distance function of IB1 to lose the ability to discern between instances of the same class and instances of different classes. In contrast, because of the information-gain-weighted distance function of IB1-IG still recognising that the focus letter is highly relevant to classification, IB1-IG is still able to rescale the instance space making instances of the same class (on average) more similar to each other than instances of different classes. Third, when the correspondence between focus letters and phonemes is lost and the information-theoretic similarity function is not able to detect any relevance differences between features, which is the case in the GS-RAND data set, both IB1 and IB1-IG fail to measure differences in average distance between instances of the same class and between instances of different classes.

In sum, the feature values of word-pronunciation instances, computed on the full data set, display outspoken relevance differences (the middle letter of a window being by far the most relevant for classification); by adapting the distance function in IB1 to this intrinsic global characteristic of the data, IB1-IG is able to employ an empirically appropriate estimate of distance between instances.

5 Conclusions

We conclude that the sample data representing the word-pronunciation task is indeed abundant in small disjuncts. Combining the results obtained with IGTREE (Table 1) and IB1 (Figure 2), we conclude that these small disjuncts contain about three to a hundred instances. Second, we conclude that there are important differences in feature relevance exploited appropriately when lazy learning is augmented with information-theoretic similarity matching in IB1-IG. Thus, the latter algorithm combines two strategies that match the data favourably, in the sense that they relate directly to the abundance of small disjuncts in the data:

1. IB1-IG store all instances in memory. Decision-tree and connectionist approaches tend to miss out on small disjuncts because they ignore certain instances when they fall below a certain utility threshold (e.g., when they occur infrequently, such as instances in very small disjuncts). Lazy learning does not ignore any instances; with small disjuncts it is at an advantage because instances in small disjuncts may reoccur in test material, and lazy learning is always able to match such instances with its stored family members.

2. While the first advantage applies to lazy learning in general, IB1-IG is also able to detect salient differences in feature relevance. Language data, such as the sample data presented here, often displays such differences (Daelemans *et al*., 1997b; Van den Bosch, 1997). With information-theoretic weighted similarity matching, the multi-dimensional feature-value space is rescaled in such a way that instances matching on important features are regarded as more similar to each other than instances matching on an irrelevant feature. Our concrete measurements indicate that, on average, the information-theoretic similarity function causes clusters of friendly instances to contain about 25 instances on average, rather than 15 with the flat-weighted similarity function IB1. The larger the clusters, the more accurate IB1-IG can be expected to classify; generalisation performance results in comparative studies (e.g., Figure 1) corroborate this expectation.

Future work should investigate the transfer of our conclusions to other (non-linguistic) tasks with similar characteristics. We claim that the methods of estimating disjunct clusters in data can be employed to analyse the applicability of lazy learning to many real-world domains. Comparative studies should be performed on benchmark tasks and real-world tasks (e.g., medical diagnosis tasks, visual object recognition tasks). Furthermore, studies with artificial data sets should be performed in which data characteristics are systematically varied and tested (Aha, 1992), to further and refine our understanding of the relations between data characteristics and lazy learning.

Acknowledgements

We thank Eric Postma, David Aha, and Jakub Zavrel for fruitful discussions and comments. Part of this research was done in the context of the “Induction of Linguistic Knowledge” research programme, partially supported by the Foundation for Language Speech and Logic (TSL), which is funded by the Netherlands Organization for Scientific Research (NWO).

References

Aha, D. W. (1992). Generalizing from case studies: a case study. In *Proceedings of the Ninth International Conference on Machine Learning*, pages 1–10. San Mateo, CA: Morgan Kaufmann.

Aha, D. W. (1997). Lazy learning: Special issue editorial. *Artificial Intelligence Review*, 11:7–10.

Aha, D. W. and Goldstone, R. L. (1992). Concept learning and flexible weighting. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*, pages 534–539. Bloomington, IN: Lawrence Erlbaum.

Aha, D. W., Kibler, D., and Albert, M. (1991). Instance-based learning algorithms. *Machine Learning*, 7:37–66.

Ali, K. (1996). *Learning probabilistic relational concept descriptions*. PhD thesis, Department of Information and Computer Science, University of California at Irvine.

Cover, T. M. and Hart, P. E. (1967). Nearest neighbor pattern classification. *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, 13:21–27.

Daelemans, W. (1995). Memory-based lexical acquisition and processing. In Steffens, P., editor, *Machine translation and the lexicon*, number 898 in Springer Lecture Notes in Artificial Intelligence, pages 87–98. Berlin: Springer-Verlag.

Daelemans, W. and Van den Bosch, A. (1992). Generalisation performance of backpropagation learning on a syllabification task. In Drossaers, M. F. J. and Nijholt, A., editors, *TWLT3: Connectionism and Natural Language Processing*, pages 27–37, Enschede. Twente University.

Daelemans, W., Van den Bosch, A., and Weijters, A. (1997a). IGTree: using trees for classification in lazy learning algorithms. *Artificial Intelligence Review*, 11:407–423.

Daelemans, W., Weijters, A., and Van den Bosch, A., editors (1997b). *Workshop Notes of the ECML/MLnet familiarisation workshop on Empirical learning of natural language processing tasks*, Prague, Czech Republic. University of Economics.

Devijver, P. A. and Kittler, J. (1982). *Pattern recognition. A statistical approach*. Prentice-Hall, London, UK.

Dietterich, T. G., Hild, H., and Bakiri, G. (1990). A comparison of ID3 and backpropagation for English text-to-speech mapping. Technical Report 90–20–4, Oregon State University.

Holte, R. C., Acker, L. E., and Porter, B. W. (1989). Concept learning and the problem of small disjuncts. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 813–818. San Mateo, CA: Morgan Kaufmann.

Quinlan, J. R. (1993). C4.5*: Programs for machine learning*. San Mateo, CA: Morgan Kaufmann.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. In Rumelhart, D. E. and McClelland, J. L., editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1: Foundations, pages 318–362. Cambride, MA: The MIT Press.

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

Stanfill, C. and Waltz, D. (1986). Toward memory-based reasoning. *Communications of the* ACM, 29(12):1213–1228.

Van den Bosch, A. (1997). *Learning to pronounce written words: A study in inductive language learning*. PhD thesis, Universiteit Maastricht. forthcoming.

Weiss, S. and Kulikowski, C. (1991). *Computer systems that learn*. San Mateo, CA: Morgan Kaufmann.

Zipf, G. K. (1935). *The psycho-biology of language*. Cambride, MA: The MIT Press. Second paperback edition, 1968.