Learning to Segment Speech with Self-Organising Maps

James Hammerton

Humanities Computing, University of Groningen

Abstract

In recent years, a number of models of speech segmentation have been developed, including models based on artificial neural networks (ANNs). The latter involved training a recurrent network to predict the next phoneme or utterance boundary, and deriving a means of predicting word boundaries from its behaviour. Here, a different connectionist approach to the task is investigated employing self-organising maps (SOMs) (Kohonen 1990). SOMs differ from other ANNs in that they are unsupervised learners. The aim is to investigate whether the SOM can become sensitive to where word boundaries occur, when trained on phonetically transcribed speech.

1 Introduction

A number of models of speech segmentation have been developed in recent years. For example, Brent's (1999) INCDROP model learns to segment speech whilst simultaneously building a lexicon from its input. It works qualitatively as follows. Segment each utterance so as to optimise the following criteria:

1. Minimise the sum of the lengths of all hypothesised novel words in the segmentation.

2. Minimise the number of the hypothesised words in the segmentation.

3. Maximise the product of the relative frequencies of the words in the segmentation. The relative frequency of a word is the number of times the word has occurred so far as proportion of the total number of times all words have occurred so far.

Thus as INCDROP receives an utterance to segment, it segments it according to the current lexicon and the above criterion, hypothesising new words where necessary. The INCDROP criteria can be derived from a probabilistic generative grammar encoding the prior knowledge that sentences are constructed by selecting words from some finite, but initially unknown lexicon and stringing them together. MBDP-1, an implementation of the INCDROP model achieved a segmentation accuracy (in terms of correctly matched words—i.e. where both the start and end of a word are correctly found without intervening boundaries) ranging from around 62% after processing 500 sentences to almost 80% after processing 9500 sentences from the Bernstein-Ratner corpus (Bernstein-Ratner 1987) from the CHILDES project (MacWhinney 2000). The figures for recall ranged from 45% to just over 80%.

Where INCDROP starts with an utterance and then segments it, Batchelder's (2002) model of segmentation, called Bootlex, involves clustering the string of phonemes into a set of words, and also simultaneously builds up a lexicon. Bootlex operates as follows:

1. Initialisation. Initially, the lexicon simply contains the set of phonemes, each having its own entry and a frequency of 1.
2. The first utterance is parsed into word tokens of one symbol each, based on the initial lexicon.
3. For each word token in the utterance just parsed, the corresponding word type in the lexicon has its frequency increased by 1.
4. Before the next utterance, the lexicon is augmented by adding to it new words consisting of consecutive pairs of words in the utterance just parsed. Each pair that is not already in the lexicon is added to the lexicon with a frequency of 1.
5. The second utterance is parsed into words, using only the words in the lexicon, and a score for each possible parse is computed from its likelihood in light of the experience to date, using the frequency counts recorded in the lexicon. The word tokens which make up the highest scoring parse are used to update the frequency counts in the lexicon (step 3 above) and to make new entries (step 4 above).
6. For all remaining utterances, step 5 is repeated, each time using the lexicon as just modified.

When this algorithm was applied to segmenting the Bernstein-Ratner corpus, it achieved a word precision of 67.2% and a word recall of 68.2%.

Some researchers have also applied artificial neural networks (ANNs) to the task. For example, Christiansen et al. (1998) trained simple recurrent networks (Elman 1990) to predict, given the current phoneme of an utterance as input, the next phoneme in the utterance or, if at the end of the utterance, to activate an utterance boundary unit. Using the Korman (1984) corpus from the CHILDES project (MacWhinney 2000), they trained their SRNs for a single epoch on 8181 utterances and tested it on 927 utterances.

During testing they found that the utterance boundary marker's activation levels were higher for lexical boundaries than for word internal positions. When the input consisted of phonemic features, plus stress and utterance boundaries (the network was not automatically reset between utterances and thus had to learn to reset itself), and the utterance boundary marker in the output layer was used to predict lexical boundaries, they obtained a precision of 70.16% and a recall[1] of 73.71% for the word boundaries, and 42.71% and 44.87% respectively for words (i.e. where both the word initial and the word final boundary are correctly predicted without any boundaries in between).

[1] Christiansen et al. (1998) use the terms accuracy and completeness respectively for precision and recall.

In this paper, a new connectionist approach is investigated, in an exploratory manner, employing self-organising maps to create an unsupervised connectionist model of speech segmentation.

2 The Self-Organising Map for Speech Segmentation

The SOM was chosen because it is both biologically plausible and it is an unsupervised learner. Whilst other neural networks can also be considered to be more biologically plausible than traditional symbolic methods, the SOM's biological plausibility extends both to the training of the SOM and its operation once trained in that both the training process and the operation have direct analogues in processes known to occur in the brain. This is not the case for, e.g. SRNs trained with back-propagation, the main connectionist networks to be applied to speech segmentation so far. Regarding unsupervised learning, the SOM is trained without any error signal, in contrast to the SRNs which require an explicit error signal. It should of course be noted that the non-ANN models of Brent and Batchelder are also unsupervised in the sense that the SOM is, but they lack the biological plausibility of the SOM.

It is thus interesting to consider whether the SOM might become sensitive to the phonotactic regularities in language without an explicit error signal and whether it can then be used for speech segmentation as it would provide a potential model for how the child's brain might do it.

2.1 The Standard SOM

The standard SOM, or Kohonen network (figure 1), operates as follows. The network consists of two layers, an input layer and a map layer, the latter typically organised on a 2-dimensional grid. The units in the map layer each have a set of weights equal in size to the number of inputs. When an input is presented, the map unit whose weights are closest to the input is selected as the winner (hence the need for the number of inputs and number of weights on the map unit to agree). The weights are typically initialised randomly in the range of values the inputs can take.

During training, the winner, plus the units in its neighbourhood are moved towards the input by an amount determined by the learning rate. The following equation is used:

$$W_i(t) = W_i(t-1) + H_{ci}(t)(X(t) - W(t-1))$$

Where $W_i(t)$ is the value of the weight vector, i, at time t and $H_{ci}(t)$ is the neighbourhood function for weight vector i, given winner c. In the work presented here, the neighbourhood function is circular and involves a linear drop-off from the winning unit to the edge of the neighbourhood, such that the winning unit's value for $H_{ci}(t)$ is $L(t)$, the learning rate at time t, units outside the neighbourhood are unchanged and units inside the neighbourhood have a value of $H_{ci}(t)$ that is intermediate between $L(t)$ and zero.

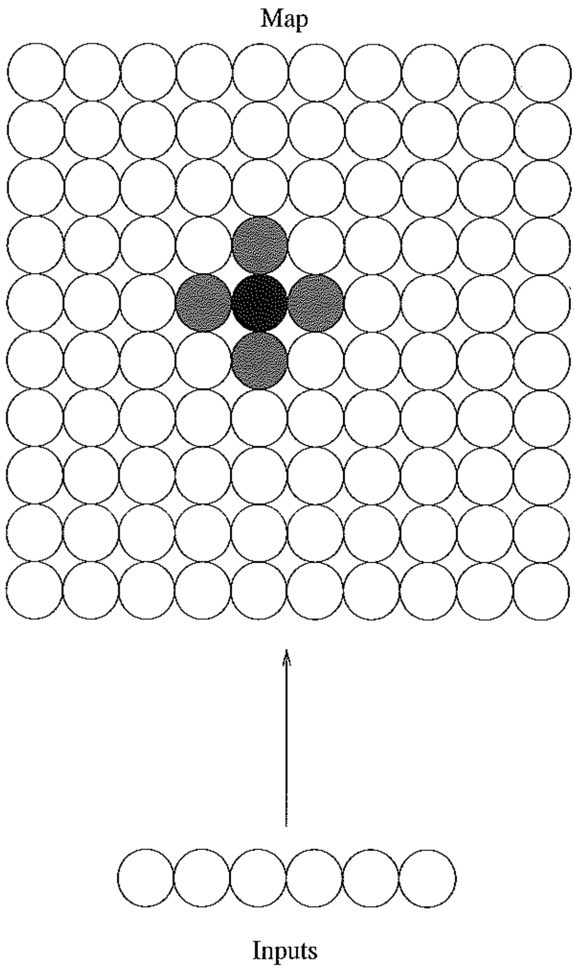

Figure 1: The self-organising map. Each time an input is presented, the distance between the input vector and the weight vector of each unit in the map layer is computed. The winning unit (illustrated here in black) is the unit whose weight vector is closest to the input vector. During training, a neighbourhood (illustrated by grey units) is defined for the winning unit, and the weight vectors of the winning unit and the units in the neighbourhood are pushed towards the input vector. The neighbourhood will start off large and be reduced in size during training.

The formula for the learning rate is:

$$L(t) = A(t) * \frac{1-t}{Max}$$

where $A(t)$ is the initial learning rate and Max is either the number of iterations to be performed (in which case t is incremented after each epoch), or in the single-iteration experiments below, the number of sequences in the training set (in which case t is incremented after each sequence).

At the start of training, the neighbourhood covers the whole map and the learning rate is set high. As training proceeds, the neighbourhood and learning rate are gradually reduced. The trained SOM forms a topological map of the input space, reflecting the distribution of inputs over that space. For example, a densely populated area of input space will be mapped using more units than a sparsely populated area. As such the trained SOM can be used for classification, and can also be used for visualisation of datasets (where the location of the winning unit on the map for each data point can be plotted).

2.2 Adapting the SOM for Speech Segmentation

With the standard SOM, each time a new input is presented, it simply replaces the previous input. Thus the standard SOM has no memory and can only map individual inputs and cannot map sequences. To rectify this, the behaviour of the input layer is modified here.

At the start of a sequence, presenting an input proceeds as normal. However when the next input is presented, the value of the previous input and the pattern representing the current input are added together as follows: $I(t) = P(t) + I(t-1) \times 0.5$, where $I(t)$ is the value of the input units at time t and P(t) is the pattern to input at time t. This proceeds until the end of the sequence. At the start of the next sequence, the memory in the input units is cleared. When using binary patterns as inputs, this ensures that information about the sequence of inputs thus far is available at each step, although the current and most recent inputs will have a high weighting.

Note that by updating the input units in this way, their qualitative behaviour is similar to that of a *leaky integrator*, commonly used as a model of the mean membrane potential of a neuron. However, The decay constant of 0.5 was chosen in an ad hoc manner to balance losing information too quickly against the possibility of potentially irrelevant information (it is assumed the most relevant information for predicting boundaries lies close to the boundaries) being given too much weight.

3 Experimental Methodology

The modelling of speech segmentation is a field in its infancy and there are some important issues about how one goes about evaluating a model that need to be addressed.

One is where do you get the data from? The standard answer in the literature, and the answer adopted here, is to use *child directed* speech, e.g. from interactions between a child and its parents, for which there are several large corpora in use. This seems a reasonable starting point for research because this is speech the child is likely to have paid attention to, it is also practical since it is relatively straightforward to collect the data. However it does miss out the ambient speech going on around a child and which the child might also learn from.

Another issue is in what form are the utterances presented to the model? Ideally the utterances will be presented directly in their spoken/audio form (which is what the child hears), but this would make for some computationally expensive models involving the modelling of acoustic processing as well as the cognitive processing involved in learning language. Again the speech segmentation literature has simplified things and typically presents the model with a phonetic transcription of each utterance, perhaps augmented with information about e.g. stress or prosody, leaving how the child gets from a sequence of noises to a set of phonemes as something to be explained elsewhere. In this work phonetic transcriptions are used. As these are generated from a dictionary (see below), this fails to take into account issues such as the differences in how words are pronounced in child-directed speech from how they are pronounced in adult speech. Nevertheless, most of the models thus far use phonetic transcriptions and only some limited extra information such as stress for their inputs since detailed information about prosody and pronunciation is rarely found in the corpora of child-directed speech, at least so far as textual corpora are concerned. These points must be borne in mind as they are weaknesses not merely of this model but of most of the models developed so far. As the field develops, it is hoped a better match between what the child does and what the models are doing can be developed.

4 Experiments

For all the experiments below, child directed speech was extracted from the Korman corpus (Korman 1984) from the CHILDES database (MacWhinney 2000). This speech came in orthographic form, so phonetic transcriptions were created by translating each word into its phonetic form via the CELEX database (`http://www.kun.nl/celex/`). If a word did not exist in CELEX, then the utterance it occurred in was discarded. At the end of this process, 9644 utterances were extracted. This data set was then split into 2 sets of 4822 utterances one for training, one for testing. Finally, since the sentences were not all distinct sentences, any sentences that were in both the training set and testing set were removed from the testing set. The testing set was thus reduced to 2610 sentences.

4.1 Preliminary Experiments

In this section the results of some preliminary experiments are described (see table 1). Here the SOMs were trained for 400 or 1000 iterations, with an initial learning rate of 0.5 (dropping linearly to zero at the end of training) and an initial circular neighbourhood covering the entire map, dropping to a neighbourhood of radius 1 at the end of training. The neighbourhood function involved a linear drop-off from the centre to the edge of the neighbourhood. The phonemes were represented by orthogonal vectors. The table gives results both for finding word boundaries and for finding words (i.e. where both the word initial boundary and the word final boundary are found with no intervening boundaries).

Finally, a baseline was run that assumed that all utterances consist of a single word and the corresponding precision, recall and fscore computed. This baseline was chosen because it is the lowest level of performance expected if you use indicators of actual utterance boundaries as indicators of potential word boundaries as is done here.

The columns in the table 1 are as follows:

Network: This indicates the size of the map layer in the SOMs. E.g. 10x10 means there are 100 units arranged in a 10x10 grid.

Iterations: This indicates the number of iterations of training used.

Precision: This gives the number of correct boundaries (or words) found as a percentage of the the total number found.

Recall: This gives the number of correct boundaries (or words) found as a percentage of the total number that actually appear in the test set.

Fscore: Where F = fscore, R = recall and P = precision, this is calculated as:

$$F = \frac{2PR}{P+R}$$

As can be seen from table 1, the networks are performing considerably better than the baseline and larger networks involve improved performance. The longer training runs make a small, but insignificant, boost to performance here.

4.2 Single Iteration Training

In this section, the SOMs are trained for only a single iteration, with the learning rate and neighbourhood decreasing during that iteration. The motivation for this is being that this was the training regime used by Christiansen et al. (1998), and they still obtained reasonable performance. Also, one can argue for example that because there are a large number of utterances in the training data, the need to train over repeated iterations is lessened, especially when weight updating occurs after each utterance. As before the initial learning rate is 0.5 and the neighbourhood is circular and initially covers the whole map. The phonemes are also represented by orthogonal vectors. The results are given in table 2. Due to the decreased training time the opportunity was taken to train larger networks.

As can be seen, the use of only a single iteration of training has resulted in a drop in performance of only 1–3 points on the fscore for networks of the same size, and the scores are still well above the baseline figures given in table 1. Also, when increasing network size, performance peaks at 800 units (32x25 units). Thus it appears that whilst there is a small hit in performance, the single iteration training does work.

Table 1: Results of preliminary experiments. All networks start with initial learning rate of 0.5 and initial neighbourhood covering entire map. All results are averaged over 5 runs from different initial weights. Baseline results assume each utterance is a word.

Boundaries

Network	Iterations	Precision(%)	Recall(%)	Fscore
10x10	400	41.55	99.15	58.55
10x10	1000	41.61	99.57	59.09
10x20	400	44.50	98.04	61.21
10x20	1000	44.54	98.45	61.33
20x20	400	49.79	96.03	65.57
baseline	n/a	100	20.24	33.66

Words

Network	Iterations	Precision(%)	Recall(%)	Fscore
10x10	400	10.91	26.04	15.38
10x10	1000	10.97	26.24	15.47
10x20	400	13.79	30.36	18.97
10x20	1000	14.08	31.09	19.39
20x20	400	19.77	38.10	26.02
baseline	n/a	2.53	0.51	0.85

Table 2: Single iteration training. All networks start with initial learning rate of 0.5 and initial neighbourhood covering entire map. All results are averaged over 5 runs from different initial weights.

Boundaries

Network	precision	recall	fscore
10x10	40.77	99.53	57.84
10x20	42.89	98.82	59.81
20x20	46.61	97.24	63.02
32x25	49.22	95.34	64.92
40x40	48.99	94.67	64.56

Words

Network	precision	recall	fscore
10x10	10.18	24.83	14.43
10x20	12.44	28.65	17.35
20x20	16.18	33.73	21.86
32x25	19.05	37.09	25.13
40x40	18.93	36.57	24.94

Table 3: Single iteration training with representation distinguishing vowels and consonants. All networks start with initial learning rate of 0.5 and initial neighbourhood covering entire map. All results are averaged over 5 runs from different initial weights.

Boundaries

Network	precision	recall	fscore
20x20	48.17	94.93	63.91
32x25	51.16	91.22	65.55
40x40	55.63	87.94	68.15
64x50	50.89	91.45	65.38

Words

Network	precision	recall	fscore
20x20	17.47	34.40	23.17
32x25	20.37	36.32	26.10
40x40	24.66	38.97	30.20
64x50	19.79	35.50	25.41

4.3 Changing Input Representation

In this section, the experiments are as in the previous section but now the phonemes are represented by vectors which indicate whether a phoneme is a vowel or consonant, see table 3. Here, the performance has been boosted, and the performance now peaks when the network size is 40x40 (1600 units) rather than 32x25 (800 units) previously.

4.4 Finding Both Starts and Ends of Words

In this section, the model was extended to use a 1 phoneme lookahead and to find both the starts and ends of words. After the SOM was trained the training set was presented again, and if a unit was active at the start of an utterance it was recorded as a unit that when active marks the beginning of a word. If a unit was active on the penultimate phoneme of an utterance it was marked as predicting a word ending after the phoneme in the lookahead buffer and if a unit was active on the final phoneme of an utterance it was marked as predicting a word ending after the current phoneme.

Table 4 gives the results for this experiment. As before, a single iteration of training was used, an initial learning rate of 0.5 and an initial neighbourhood covering the entire map. As can be seen, the overall performance is worse than before, with the fscores being 5-7 points lower than in the previous section for networks of the same size. Indeed the performance here is worse than in any of the previous sections, though still well above baseline.

Table 4: Single iteration training with representation distinguishing vowels and consonants and both the starts and ends being found. All networks start with initial learning rate of 0.5 and initial neighbourhood covering entire map. All results are averaged over 5 runs from different initial weights.

Boundaries

Network	precision	recall	fscore
20x20	38.88	95.20	55.21
32x25	43.59	90.62	58.86
40x40	49.45	87.77	63.25

Words

Network	precision	recall	fscore
20x20	7.42	18.13	10.53
32x25	11.38	23.63	15.36
40x40	17.64	31.29	22.56

5 Discussion

Considering that this is a set of exploratory experiments, the results are encouraging overall. Clearly the SOMs do become sensitive to the phonotactic regularities in the utterances, even when trained on only one presentation of the training data, as indicated by the consistently well above baseline performance across all the experiments. The performances compare reasonably with other connectionist models but do not compare as well with the models of Brent and Batchelder. There is a significant gap in performance between the connectionist models and the more analytic models such as INCDROP and Bootlex.

It should be pointed out however that where the connectionist models process the utterances phoneme by phoneme, the models of Brent and Batchelder process a whole utterance at a time. The latter thus get to act on more information than the former and one would expect them to do better as a result. From a cognitive point of view one might question whether a child does in fact wait until it has heard an entire utterance before segmenting it, and moreover whether the child may create multiple possible parses and then select the most optimal one, though Batchelder (private communication) points out that for child-directed speech which is relatively simple, it may not be that implausible to do so. Of course it is not clear how one would test these points directly and the one undeniable fact about children is that they do learn to do the task very well. Thus the high performance of the models such as Bootlex and INCDROP presents a challenge to the connectionist models, including this one.

One might go further and even question whether children do segmentation at all. However the data are fairly clear on this point. Sentences and utterances are clearly composed of smaller units similar to those which we call words. The very fact that (sufficiently developed) children and adults can create novel combinations of words or use newly learned words with fluency in their native languages is strong evidence for this ability. Thus the question becomes at what age do children develop the awareness of words. For an upper bound on this, the data suggests that by the age of 24 months, children start making their own two word utterances (see Jannedy et al. 1994, 282).

One thing the SRN-based models, INCDROP and Bootlex have in common and which may be a problem for the model presented here, is the fact that the former can (at least in principle) be treated as online learners rather than batch learners. For example, both INCDROP and Bootlex can simply be set to the task of segmenting a corpus of text and they will learn as they do it. In principle it should be possible also to apply the SRN in a similar manner, though one would need analyse the activation of the utterance boundary markers over a window of e.g. the several previous utterances to do the segmentation.

However the SOM training regime draws more heavily on the batch training paradigm. The reduction in size of the neighbourhood and learning rate over time during training do not fill very well with an online learning situation. Moreover, the need to calibrate the map (i.e. work out which units are active on utterance boundaries) also does not fit well with the online learning paradigm either. Nevertheless one could try e.g. having a phase where the learning rate and neighbourhood size decline, and thereafter remain at a small fixed rate, plus re-calibrate the map after each utterance to try and force the SOM into an online learning paradigm. A better approach might be to employ a version of the SOM where the neighbourhood and learning rate effects fall out naturally from its operation.

Returning to the results presented above, there are a number of points to observe:

- The recall is consistently higher than the precision. The model clearly has a bias towards generating boundaries that may be correct. A similar bias is demonstrated in Christiansen et al.'s (1998) work but it is not as pronounced as it is here. For example, when Christiansen et al. train using only phonemes and utterance boundary markers as input, they obtain a precision of 65.86% and a recall of 71.34%. Here the recall is typically above 90% with the precision being in the 40s or 50s.

- Increasing network size does not always improve performance. As noted with the single iteration training, the performance peaked and then declined after a certain network size. Use of a more informative input increased the size of network where performance peaks but does not get rid of the phenomenon. The phenomenon is most probably due to the SOM equivalent of over-training—namely that if the map is too large, it will map the training data so closely that performance on the test data will be harmed.

- Finding both the beginnings and endings of words whilst using a one phoneme lookahead counter-intuitively offered the worst performance here. Why should this be the case? One possibility is that the bias towards generating too many boundaries has been reinforced, however both recall and precision values are low suggesting that overall performance has been harmed.

Future experiments will seek to try and explain this phenomenon.

6 Conclusion and Future Work

This paper has demonstrated that a SOM can become sensitive to phonotactics in child-directed speech and that it can be applied to the problem of speech segmentation with reasonable, though not state-of-the-art, results.

In future work, it is hoped that the limitations of this model can be addressed, its performance improved and the model extended as follows:

- Extracting more data from the Korman corpus. A significant number of sentences were discarded from the Korman corpus due to their containing words that did not occur in the CELEX phonological dictionary. By using a grapheme to phoneme converter it may be possible to incorporate these sentences by passing unknown words to the converter. This should help to improve the performance of the system.

- Incorporating more information. For example, by using phonetic features to describe each phoneme and incorporating stress, it may be possible to improve the performance of the model. Indeed, Christiansen et al. (1998) get significantly better performance when they incorporate such information than when they do not.

- Performing experiments to try and find out why finding the beginnings and ends of words performed so much worse than the finding only the ends of words. E.g. obtaining performance on the training data and separating the finding of starts from the finding of ends.

- More closely matching the training regime used by Christiansen et al. (1998). There are a number of differences between the training regimes used here and in Christiansen et al.'s (1998) work, the latter of which more closely matches what a child is actually doing when learning language.

  - Firstly, Christiansen et al. do not take care to ensure that there are no sentences that are in both the training and the testing sets. From a machine learning point of view this seems invalid. However from the point of view of modelling how children learn language it is valid. By using a single iteration of training and by simply splitting the non-distinct sentences into training and testing set they aimed to match the ecological conditions under which a child learns language more closely than the standard approach manages.

  - Secondly, Christiansen et al. do not reset their network between utterances. Thus their system operates continuously and must learn to reset itself if necessary. However it may not make much difference to performance with the SOM model to do this. Intuitively, as the SOM receives more and more input, the information from earlier inputs decays geometrically and thus the most recent inputs will have the biggest impact on the network's performance. However to a large degree this is also true with SRNs since their memory will also decay quickly over time.

- Looking more closely at the methodology of training/testing the networks. The basic methodology used here was chosen simply to get a feel for how well the networks would perform reasonably quickly. However, even after matching Christiansen et al.'s (1998) methodology more closely, there is still considerable difference between what the child is doing and what the models are doing. Also, even from a pure machine learning point of view, there needs to be a more rigorous comparison of this model to the other models, using the same data sets and splits between training and testing data.

- Incorporating the model into a model of lexical acquisition. Brent and Batchelder's models both incorporate the building up of a lexicon into their model of speech segmentation, a considerable advantage over this and other connectionist models. A challenge therefore is to develop a connectionist model that also develops a lexicon that can be used in the segmentation process. This in fact would be a major challenge for a connectionist model as it requires some form of connectionist representation for the words being found and a memory in which to store those words, neither of which are easily done with connectionist systems.

Acknowledgements

The author would like to thank John Nerbonne for his discussions of this work, Morten Christiansen for answering various questions about his work, and two anonymous reviewers for their very constructive criticisms of this paper.

This work was supported by the Connectionist Language Learning Project, funded by the High Performance Computing and Visualisation programme at the University of Groningen.

References

Batchelder, E. (2002), Bootstrapping the lexicon: A computational model of infant speech segmentation, *Cognition* **83**, 167–206.

Bernstein-Ratner, N. (1987), The phonology of parent child speech, *in* K. Nelson and A. van Kleeck (eds), *Children's Language*, Vol. 6, Lawrence Erlbaum Associates, Hillsdale, pp. 159–174.

Brent, M. (1999), An efficient, probabilistically sound algorithm for segmentation and word discovery, *Machine Learning* **34**, 71–106.

Christiansen, M., Allen, J. and Seidenberg, M. (1998), Learning to segment speech using multiplue cues: A connectionist model, *Language and Cognitive Processes* **13**, 221–268.

Elman, J. (1990), Finding structure in time, *Cognitive Science* **14**, 179–211.

Jannedy, S., Polletto, R. and Weldon, T. (eds) (1994), *Language Files: Materials for an Introduction to Language and Linguistics*, sixth edn, Department of Linguistics, The Ohio State University, Columbus.

Kohonen, T. (1990), The self-organising map, *Proceedings of the IEEE* **78** (9), 1464–1480.

Korman, M. (1984), Adaptive aspects of maternal vocalizations in differing contexts at ten weeks, *First Language* **5**, 44–45.

MacWhinney, B. (2000), *The CHILDES project: Tools for analyzing talk*, third edn, Lawrence Earlbaum Associates, Hillsdale.