Can a Computer Really Model Cognition? A Case Study of Six Computational Models of Infant Word Discovery

Eleanor Olds Batchelder (eleanor@roz.hunter.cuny.edu)
70 West 95th Street, Apt. 9H, New York, N.Y. 10025

Abstract

Prelinguistic infants must find a way to isolate meaningful chunks from the continuous streams of speech that they hear. This bootstrapping problem has recently been the focus of several attempts to model the cognitive problem computationally. How can we evaluate whether this kind of simulation is relevant to the cognitive situation, and how can we compare different computational approaches? I discuss my O-B algorithm, a variable-length clustering procedure, and compare it with five other models—three connectionist ones and two statistical programs which use Minimum Description Length as a decision metric. I show that the models differ in their similarity to cognitive processes with respect to: a) the timing of inputs and outputs; b) constraints on the incremental learning process; c) clustering vs. dividing strategy; and d) whether the goal is to find words or to learn word-finding rules.

What Is a Model?

In one sense, any theory implies some model of reality. In a narrower sense, a verbal description is called a model when it describes in detail some complex process, providing a fuller context for empirical observations whose relationship is not transparent. A computer program which simulates the inputs and outputs of some behavior, and possibly the internal processing as well, is also a model. Each kind of model has different strengths and weaknesses. The computer model has greater precision and enforces internal consistency, but the narrative description can often be drawn closer to reality and can include a wider range of relevant phenomena.

Models help increase our understanding by forcing us to look at more aspects of the problem, or more closely, than we might otherwise do, and they can generate new and testable hypotheses. But all models are at bottom a collection of analogies, and both their power and their limitations stem from this fact. Even a precise physical model, such as a model railroad, is limited in realism by being much smaller than the original. In evaluating a model, we must make explicit the various analogies involved, remembering that each analogy is a human construct and all are imperfect.

What Is a Cognitive Computer Model?

Cognitive models depend on analogies with human mental processes. In one sense, a cognitive computer model is by definition unreasonable—computers are very unlike human brains. What's more, the single-purpose computer programs we will discuss here are so far from the complex interactions of human cognition that we will have to stretch our imaginations even to tolerate the comparison. However, we hope that by pushing the analogy as far as we can, we will be rewarded by a heightened awareness and a sharper understanding of the issues involved in cognition.

Cognitive models should be evaluated at least in part on how well they simulate the cognitive process in question. In this paper we will compare models that use different computational algorithms, and show that these involve different degrees of similarity to the infant's mental processes.

In addition, learning models have special features that cognitive process models do not. Realistic computational learning models should be unsupervised models—allowing the computer to "learn" from examples by induction. Supervised models present correctly analyzed instances to be learned, and are useful to discover new ways of organizing the data so as to achieve the output. Unsupervised models present unanalyzed instances and require that the system discover the correct analysis. Models which perform unsupervised learning are also known as "self-organizing" and the resulting structure is called "emergent." The models we will look at are mainly unsupervised, though some have supervised portions.

Cognitive Models vs. Language Engineering

Modeling language in computers is also valued for performing practical work in the world. Imitating cognitive functions is an engineering goal as well as a scientific one, but there are important differences between the two. In scientific modeling, all aspects of the model are evaluated for their closeness to the human condition. In engineering projects, however, only the outputs need to be realistic—the inputs and the processes are not valued for themselves, but can be anything that produces the best output. Therefore, workhorse models will find supervised training just as acceptable as unsupervised training, as long as the training data is readily available and the ultimate output is of high quality.

In both scientific and engineering models, quantitative evaluation is an important criterion, but it is less critical to the cognitive enterprise. A few percentage points on a single performance measure can affect the choice of algorithm for a large production project, but the same comparison is negligible when weighed against significant differences in psychological reality.

In sum, engineering and cognitive goals and methods are distinct, and the two are rarely effectively combined in the same project. While there is some borrowing of techniques back and forth, we should be clear about which is which. In this paper, we have a cognitive goal rather than an engineering one—to show that frequency information can be mobilized in a cognitively realistic fashion.

On the other hand, one criterion of a successful cognitive strategy is its effectiveness. So, we will present some quantitative results for the models, in order to demonstrate that they are roughly equal in their raw segmentation power, and then we will look in greater detail at qualitative criteria.

The Infant Bootstrapping Problem

Infants hear long unbroken streams of speech, which must be separated into chunks that can be attached to meaning:

e.g.:	the man in the moon
not:	them anin them oon

By the end of the first year of life, the infant has started to find words, and by 18 months or so, a vocabulary of about 50 words is achieved and the rudiments of linguistic knowledge are in place. Thus, we hypothesize that this first word-finding process is a prelinguistic and temporary one, soon replaced by more sophisticated techniques.

Linguistic explorations of this bootstrapping process have suggested a number of possible sources of segmentation information, some of which are more available and/or useful than others. Those least likely to be useful are: acoustic cues to word boundaries (not very numerous); phonetic and phonotactic cues (word-based, so require prior knowledge of words); and one-word utterances (too few and not enough variety). More likely to be helpful are: utterance boundaries, which help focus attention on chunks at the edges as word candidates; prosodic cues such as intonation, which helps break up long utterances, and stress, which forms the nucleus of a cluster of sounds; and distributional cues such as cooccurrence frequency, which have been shown to become more useful at the end of the first year (Jusczyk,1997).

None of these has been demonstrated to be adequate by itself, and it is probable that the child uses a combination of several sources. We will consider here how much of the job can be accomplished by distributional cues alone—information about the frequency of recurring chunks.

Computational Models of Word Discovery

The lack of a purely linguistic explanation for this bootstrapping problem has led to a burst of recent research on the influence of frequency on infant segmentation. The six computational models discussed here share an overall approach. In general, a text corpus is chosen and all word boundaries are removed, then a computer program "reads" the corpus and gathers statistical observations to decide where to replace the word boundaries, attempting to duplicate the original text (the "standard").

Of course, as a cognitive analog, this methodology is very flawed. First, the criterion of matching some conventional "word" standard can be justly criticized as not imitative of infant cognition. Children famously use morphemes, words, and phrases as basic units. So orthographic words are only a rough measure of meaning-based segmentation.

Second, all the models use graphemes of some sort, rather than sound waves, to represent the speech signal:

Orthography:	look theres a boy with his hat
Phonemes:	lUk D*z 6 b7 wIT hIz h&t

Although we know very little about how humans represent language, and practically nothing about how human infants do so, certainly it is not as orthography! However, decoding actual sound signals is still beyond the ability of machines, unless considerable linguistic knowledge is supplied.

We now review a number of computational models of the infant's segmentation process. They fall into three major groups: connectionist networks; algorithms using the Minimum Description Length principle; and frequency clustering algorithms. As a benchmark, we add a "dumb default" bigram technique.

First we will briefly describe each project, and give representative quantitative results for them. We can hardly do justice to these projects in such a small space, so the reader is referred to the original reports. Figure 1 shows recall and precision rates in terms of cuts and of words from the respective papers (most algorithms only report one way or the other). The cut metric is more precise, but the word metric is more intuitive and linguistically useful. The rest of the paper will discuss in greater detail their relative merits as cognitive models—whether the strategies and processes they

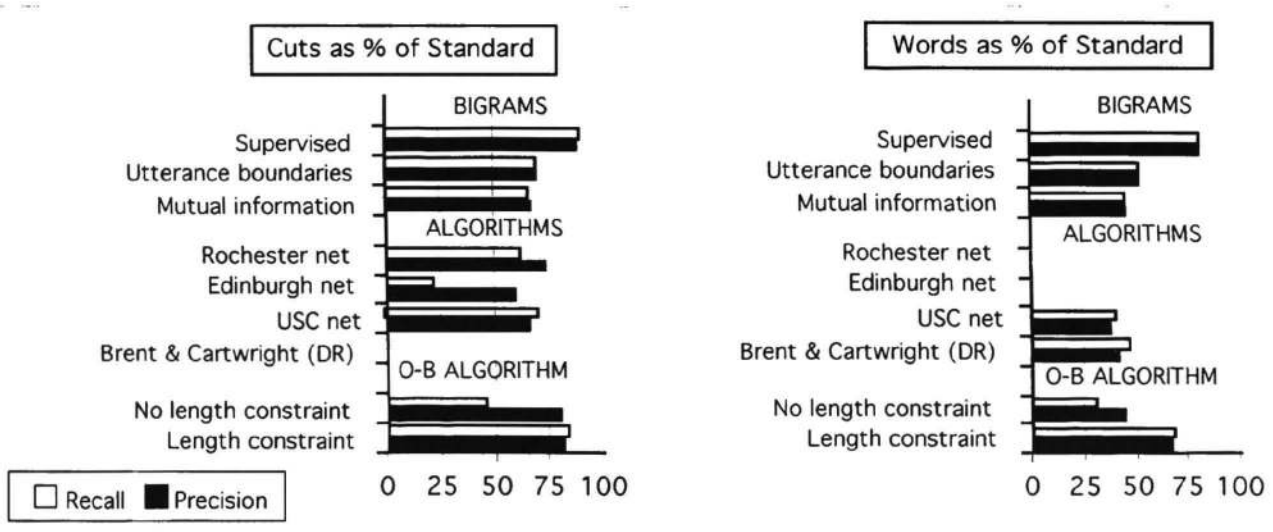

Figure 1: Reported quantitative performance of the various models, measured as number of cuts or words that match the standard. Notes: a) We show only the results due to distribution and utterance boundary information, omitting those from combined cues (frequency statistics plus prosody or phonotactics). b) deMarcken (1997) does not provide comparable performance measures. c) All the results shown here are for English, since only the O-B algorithm reports cross-linguistic results.

use are plausible ones for infants, and whether the timing of events is similar to that which is observed for infants.

Bigrams

The "dumb default," against which more sophisticated techniques can be gauged, is a variety of simple statistical procedures which look only at two-character combinations. A program computes the relevant statistic for each bigram type, ranks the values, and chooses a cutoff point on which to base segmentation decisions. Figure 1 shows results on a 33,000-word phonemic corpus from CHILDES (MacWhinney, 1995) for several statistics, using a cutoff point chosen to create the same number of words as in the standard.

- Word boundaries (supervised): Frequencies of bigrams across standard word boundaries vs. within words are used to segment the same text. This technique outperforms all others, but it uses information not available to the child.
- Utterance boundaries: Frequency of bigram types occurring before and after utterance boundaries.
- Mutual Information statistic for all bigram types (slightly better than the alternatives of raw frequency or transitional probability)

Connectionist Models

Network, or connectionist, models from three laboratories all used feature representations of phonemes (see Table 1 for details). The text was presented as a stream, and the networks were asked to predict the next phoneme, gradually learning to discriminate between phoneme pairs with high vs. low cohesion. In the Edinburgh net, the amount of error in this task was translated into the probability of a word boundary (less error ≈more cohesion and less probability of a boundary). In the other two nets, indicators of utterance boundaries were supplied in the input along with phonemes (supervised training), and one output unit reported the probability of a boundary after each phoneme.

- **University of Rochester**: Aslin, Woodward, LaMendola & Bever (1996) tested whether a connectionist architecture using moving triplets could generalize from utterance boundaries (pauses) to word boundaries.
- **University of Edinburgh**: Cairns, Shillcock, Chater & Levy (1997) ran a very large corpus on a simple recurrent network (SRN) using no boundary information, but did not get very good results. Among other differences between this net and the others, they added "noise" to the input by randomly flipping feature bits.
- **University of Southern California (USC)**: Christiansen, Allen & Seidenberg (in press) used utterance boundaries like Rochester, but with a larger corpus and a SRN, and reported similar results. We show here results without the use of stress cues.

Minimum Description Length (MDL)

MDL is a quantitative metric used to evaluate how compactly a particular lexicon represents a particular text. A computer program trying to construct a "best" lexicon for a corpus uses MDL to select the best one. An MDL-based model must thus try many possible lexicons and evaluate each one, generally a very computation-intensive process.

- **de Marcken (1996)** used an optimizing approach to avoid testing every possible lexicon. He demonstrated his method on several large corpora, using standard orthography, but he segmented hierarchically rather than creating a single series of words. He cited recall rates of over 90%, but he did not give precision rates to balance them against because the number of units created (on several levels) was many times the number of words in the standard.
- **Brent & Cartwright (1996)** tried all possible lexicons exhaustively and were therefore limited to tiny corpora of about 525 words. They used a phonemic representation of the speech of mothers to infants, from the CHILDES data (MacWhinney, 1995). The version reported here is called DR; we omit discussion of later experiments which added several types of phonotactic information.

The O-B Algorithm

Olivier (1968) and Wolff (1977) used similar, but independently developed, algorithms based on identifying as "words" those variable-length clusters which appeared frequently. Batchelder (1997) modified Olivier's algorithm to get the O-B algorithm, which will represent the other two here. Each utterance was parsed as it was received, using the best combination of words according to the current lexicon. "Best" was calculated as the most probable parse in light of experience to date. The results of each parse were then used to revise and extend the lexicon before proceeding to the next utterance. This algorithm resembles the child's hypothesis testing: Words are not discovered abruptly, but gradually become more and more likely as evidence accumulates, or they may fall from consideration if not much additional evidence is encountered.

Table 1. Details of three network experiments (see text). Note: USC inputs and outputs include 1 boundary unit and 2 stress units; they used a local representation for output of phonemes (one bit for each phoneme).

	Rochester	Edinburgh	USC
Corpus of speech directed to	child	adult	child
Size in words	1300	300,000	25,000
Size in segments (characters)	<5000	1,000,000	73,947
Word Token/Type Ratio (TTR)	85	≤25	30
Coding: # segment types	~44	45	36
# binary features (bits)	18	9	11
Training: Iterations	2-3	2	1
Total bits input	≤270,000	18,000,000	813,417
Utterance boundary training	superv.	unsup.	superv.
Net architecture	window of 3	SRN	SRN
Tasks: Predict segment?	yes	yes	yes
Predict boundary?	yes	no	yes
Input units	54+1	9	11+1+2
Hidden units	30	60	80
Context units	n.a.	60	80
Output units	1	27	36+1+2

The O-B algorithm begins with a lexicon of one-character "words." After a few utterances have been seen, two-character words will appear, and so on, with words getting longer and longer as learning proceeds. The average length of word clusters can be constrained by an optimum-length ("optlen") parameter, which downgrades parses with overly long words.

The O-B algorithm results shown here are from a corpus of 75,000 words in standard orthography, both with and without the optlen constraint.

Cognitive Comparison of Models

Table 2 provides a graphic summary of this discussion.

Timing of Inputs and Outputs

One measure of the cognitive reality of a model is how closely it matches the timing of the original. Children receive input continuously and, after a period of no apparent results, begin to produce and understand words, one by one. Discovery of a few primitive words is followed by more words and more complex words, in a steady progression. Their learning process is incremental and continuous, with new inputs modifying the results of learning so far. How do the computer algorithms measure up as incremental and continuous processors?

- The **MDL algorithm** receives the worst score as an incremental process. Both MDL models accept all input as a single event and then process it repeatedly, finally producing a lexicon and a segmented version of the input. Processing consists of placing a trial set of word boundaries in the text and evaluating the result in terms of the length and frequency of the word types, then trying a new set of boundaries, and so forth. Though the computation proceeds by stages, neither the input nor the output is continuous or incremental:

> The search algorithm... operates in batch mode, reading in the entire input before segmenting any part of it. Clearly, children do not work this way. Rather, they add to their lexicons incrementally as new input becomes available. (Brent & Cartwright, 1996:117)

It has been argued that, since the final result is the same, the inefficiencies involved in incremental or staged outputs are not justified (Ling, 1996) but we would like to be reassured that, however inefficient, an incremental version is at least possible. Brent has outlined an incremental version (Brent, 1997), but so far has not presented a working model.

The nature of the MDL algorithm itself would seem to make this unlikely. MDL sets itself to find the best representation of a particular and finite set of data. It assumes a close relationship between a particular text and its "best" lexicon. If the set of input data changes slightly, then a whole new set of computations must be begun. MDL was not designed for the human problem of keeping up with a continuous stream of data, and it seems to be impossible in principle for it to do this.

- The **O-B algorithm** is the most cognitively realistic model, with both input and output occurring incrementally and synchronously. It processes each utterance as it is encountered and produces a segmented version as it goes along. The lexicon is modified for each utterance, adding possible words and continuing to gather evidence about their likelihood.

O-B also resemblea the child in the gradual growth of the words themselves. Although all word-spaces are removed from the text before feeding it to the algorithm, nevertheless the initial default segmentation is a series of single characters. Thus, even at the beginning of the process, a few small "words" can be extracted, and the words get longer with experience, just as in the child's learning process.

- The **networks** receive a mixed score. The input and processing are continuous, with the internal weights changing very gradually throughout the training process. The input is presented phoneme by phoneme and, although some nets reprocess their input, this is presumably in imitation of "more of the same" and is not in principle necessary to the learning process.

The characteristics of the outputs in this respect is not clearly apparent from the published reports. Recall that all three nets had some graded output that was interpreted as the probability of a boundary following each phoneme. There seem to be two logical possibilities for the change in this output as training proceeds, both of which could be described as continuous but not incremental: (1) The likelihood of each of many specific boundaries increases gradually and in synchrony, so that at some point in the course of training a large number of boundaries become recognizable at the same time; or (2) a few boundary points are clearly identified early in the process, with more and more added as training proceeds. The first case is not incremental by any interpretation. The second case, though boundaries are discovered incrementally, produces no usable "words" until far along in the training process. The first boundaries to be identified are few and widely spaced, yielding unmanageable stretches of language in between that are too complex to be accessible to the infant as objects. Counterintuitively, the "words" get smaller and smaller as experience and knowledge increase.

Constraints

All of the models make segmentation decisions by a quantitative assessment of the relative cohesion between various units or clusters. These cohesion metrics form a continuum with no obvious point indicating a shift from a word-internal to a word-boundary condition. How does each algorithm make that decision, and how does each compare with the child's process? What constrains or guides the learning process in each case?

Two algorithms (Wolff, 1977; de Marcken, 1996) produce a nested segmentation, with the most cohesive units on the inside: [s[h[or]]t][c[ut]]. This procedure avoids deciding which letter groups represent morphs, which words, and which phrases. In one sense, such a representation may be more cognitive than rigid word boundaries. Certainly adults, and probably even children, store complex lexical units, with several hierarchical and coexisting levels.

- In the **O-B algorithm**, the "optlen" parameter sets an upper limit on the built-in tendency for words to grow longer. It does this with an evaluation metric that penalizes the use of longer words when there are shorter suitable candidates. This constraint is the analog of a developmental limitation,

Table 2: Summary of cognitive evaluation as discussed.

	O-B	MDL	Nets	Bigrams
Continuous input?	yes	no	yes	no
Continuous output?	yes	no	yes?	no
Incremental?	yes	no	no	no
Constraint	optlen	MDL/hier	?	optlen
Strategy	cluster	cluster	divide	divide
Goal	words	words	rules	rules

the combination of the infant's small working memory and the short life of the bootstrapping process.

The particular value of the "optlen" parameter used in these experiments was the one calculated to produce the same number of word tokens as in the standard, which simplified scoring by bringing precision and recall closer together. For the child there would be no such value, of course, but by hypothesis a gradual working up from very small to larger. In fact, the demonstration by the O-B algorithm that such an incremental clustering process needs some constraint can be seen as a result that confirms the "less is more" hypothesis of Newport (1990) and Elman (1993).

- In the **networks**, it is not clear what constrains the learning process. As mentioned in the previous section, we do not know how the outputs changed over the training process, and the authors did not report how they decided to terminate training. In two cases, since the precision and recall percentages were fairly close together, we can see that the final state of the network produced roughly the same number of word divisions as in the standard (Rochester 86% of the standard, USC 108%). The Edinburgh results, however, showed that the number of divisions made by the network was only 35% of the standard, a severe shortfall, and only 60% of these were correct (21% of the standard).
- The **MDL algorithms** are constrained by the tension between the length of the word type and its frequency. The MDL principle tries to minimize the combined length of the coded text and the lexicon, resulting in the avoidance of both extremes: a long word which occurs rarely is penalized, as is a very short word which occurs too frequently. In the former case the lexicon entry is too long to result in net savings for only a few occurrences. In the latter case, each of the many occurrences will require an index reference that is almost as long as the original word would have been.

MDL is derived from information theory and defines the most efficient encoding of a text:

> The MDL Principle is a well-motivated and theoretically sound principle for data compression and estimation... As a strategy of statistical estimation, MDL is guaranteed to be near optimal. (Li & Abe 1996:1)

Brent & Cartwright extend this to the cognitive sphere:

> ...[T]he notion that the best segmentation of the input is the one with the shortest representation can be interpreted as a formalization of Occam's Razor—the notion that the best explanation of a set of observations (e.g., linguistic inputs) is the simplest. (Brent & Cartwright, in press: 10f.)

It is not clear, however, that infant minds—or even adult minds—work according to such idealized principles. For the human brain, where computation is slow and storage is plentiful, there seems no justification for a scheme which does a lot of work in order to save storage.

To carry the argument a bit further, we can say that MDL and other compression schemes are an attempt to render natural language—which is "natural" to the human mind—more manageable by machines. The most efficient structure for a computer is the least redundant one, but this is not a characteristic of human language, which by its nature is highly redundant.

Clustering vs. Dividing Strategies

With respect to overall strategy, the algorithms pattern into two groups: The nets divide, while the MDL and O-B algorithms cluster. The cluster approach looks for particularly frequent and thus cohesive groups, and treats these as the objects of interest—words; boundaries between word clusters are a side-effect of this process. The divide approach looks for points of unusually low cohesion and treats these as divisions, with words arising between these boundaries as a side-effect of the process. As pointed out above, clustering promotes the incremental growth of outputs, while dividing frustrates it. An even more fundamental difference, perfectly correlated with the cluster/divide dichotomy in our sample, is discussed in the next section.

Learning Words or Rules?

The cluster/divide difference in strategy is linked to a difference in the goal or endpoint of the process. O-B and MDL are modeling the discovery of particular words, which are then recognized as words and used again, while the nets are modeling the discovery of how to discover words by monitoring relationships between phonemes and/or features. The end product of the clustering process is a lexicon of word types. The end product of the dividing process is a "knowledge" of the statistical regularities of word boundaries as encoded in the hidden units of the networks.

As one indication of this focus on process rather than product, all three networks "trained" on one body of data and then "tested" using a held-out portion. This reflects the designers' view of their model as learning a skill, which is then demonstrated on data distinct from that which was seen in training. O-B and MDL, on the other hand, are engaged in a self-organization of the phonemic stream, creating a lexicon and a parsed version of the input. Their goal is not the ability to find new words in new input, but the knowledge store which results directly from the particular input seen.

Which is cognitively more plausible? When we say that children are "learning to segment language," do we mean that they are discovering meaning units one by one and entering them in the first mental lexicon? Or do we mean that they are learning "rules," or regularities, for segmenting their native language into words? This contrast is sometimes cast as two kinds of knowledge: rote vs. rule, "knowing that" vs. "knowing how."

There is a nice distinction here that is intimately related to the child's role as active learner. We must differentiate between the goal object of learning and the path by which it is

reached. Babies are not trying to learn rules or regularities, but to communicate with those around them. To enter the linguistic system, they use an artful combination of imitation and analysis. Instinctively trying to minimize effort and maximize results, they learn to pay attention to the most productive regularities, the most reliable cues.

So "rules," in this sense of probabilistic regularities, are being learned by the infant as a by-product of successful hypotheses. That is, as more and more words are learned, more and more regularities that led to their hypothesis can be confirmed. This is the opposite of the usual sense of "rule" in linguistics as something which generates linguistic objects, rather than something which is induced from them. Perhaps we can say that the words themselves are consciously "known," while the "rules" or cues that lead to successful learning are the kind of implicit learning that we use without conscious awareness (Cleeremans, 1993).

If so, undoubtedly infants are engaged in both "learning what" and "learning how" simultaneously, but the first bears fruit much sooner than the second. The project of learning mature phonotactics, while certainly ongoing throughout this period, is unlikely to lead to the first lexicon. As a result of discovering the first group of words, the child's linguistic knowledge is greatly increased, and it is probable that the first primitive methods of word discovery are soon discarded, or at least vastly reorganized, on the way to the adult linguistic system.

But is it possible that the connectionist nets, too, are learning actual words? How do we know that they are making phonotactic discriminations, not lexical ones? On the one hand, they create no lexicon, and their use of featural representations increases our perception of them as discriminating on phonotactics. But they do evaluate the word boundaries as output and not the rules which presumably find them. Since "connectionist networks are notoriously hard to analyze" (Cleeremans, 1993:205) and we know of no empirical test which can make the necessary distinction, we will accept the judgments of their designers:

> The main empirical claim behind our approach is that subregularities within a domain can be, and are, exploited to the extent that they make useful predictions. In our case the subregularity is phonotactics, the sublexical distributional regularities of phonology. (Cairns et al., 1997:142)

> The most important outcome, however, was the presence of significantly higher activation for word boundaries than for within-word phoneme triplets. This indicates that, in addition to learning phoneme triplets that preceded a phrase/utterance boundary, the model also learned phoneme triplets that preceded a word boundary. (Aslin et al., 1996:129)

In sum, as the goal of a bootstrapping task, learning words seems more plausible than learning how to find words, so the clustering algorithms have an edge in casting the problem in a more lifelike form, and the bigram and connectionist procedures are in this respect less cognitively realistic.

Conclusion

Various characteristics of the child's cognitive processes during early word segmentation can be paralleled in a cognitive computer model to a greater or lesser degree. The real gains of the modeling exercise, however, are a clearer understanding of the nature of the infant segmentation process, and a greater appreciation for what a cognitive computer algorithm can show us and what it cannot.

References

Aslin, R. N., Woodward, J. Z., LaMendola, N. P., & Bever, T. G. (1996). Models of word segmentation in fluent maternal speech to infants. In J. L. Morgan & K. Demuth, (Eds.), *Signal to syntax: Bootstrapping from speech to grammar in early acquisition.* Hillsdale, NJ: Erlbaum.

Batchelder, E. O. (1997). *Computational evidence for the use of frequency information in discovery of the infant's first lexicon.* Ph.D. dissertation, City University of N.Y.

Brent, M. R. (1997). Toward a unified model of lexical acquisition and lexical access. *Journal of Psycholinguistic Research, 26*(3),363-375.

Brent, M. R., & T. A. Cartwright (1996). Distributional regularity and phonotactic constraints are useful for segmentation. *Cognition 61*,93-125.

Brent, M. R., & T. A. Cartwright (in press). The informational structure of word learning. Ms.

Cairns, P., Shillcock, R., Chater, N., & Levy, J. (1997). Bootstrapping word boundaries: A bottom-up corpus-based approach to speech segmentation. *Cognitive Psychology 33*,111-153.

Christiansen, M. H., Allen, J., & Seidenberg, M.S. (in press). Learning to segment speech using multiple cues: A connectionist model. *Language & Cognitive Processes.*

Cleeremans, A. (1993). *Mechanisms of implicit learning: Connectionist models of sequence processing.* Cambridge, MA: MIT Press.

de Marcken, C. G. (1996). *Unsupervised language acquisition.* Ph.D. dissertation, MIT.

Elman, J. L. (1993). Learning and development in neural networks: The importance of starting small. *Cognition 48*,71-99.

Jusczyk, P. W. (1997). *The discovery of spoken language.* Cambridge, Mass.: MIT Press.

Li, H., & Abe, N. (1996). Clustering words with the MDL principle. *Proceedings of COLING.*

Ling, C. X. (1996). "Can symbolic algorithms model cognitive development?" *Proceedings of the 18th Annual Conference of the Cognitive Science Society.* Hillsdale, NJ: Erlbaum.

MacWhinney, B. (1995). *The CHILDES project: Tools for analyzing talk*, 2nd ed. Hillsdale, NJ: Erlbaum.

Newport, E. L. (1990). Maturational constraints on language learning. *Cognitive Science 14*,11-28.

Olivier, D. C. (1968). *Stochastic grammars and language acquisition mechanisms.* Ph.D. dissertation, Harvard U.

Wolff, J. G. (1977). The discovery of segments in natural language. *British Journal of Psychology 68*,97-106.