Distributional Regularity and Phonotactic Constraints are Useful for Segmentation

Michael R. Brent and Timothy A. Cartwright

November 16, 1995

Abstract

In order to acquire a lexicon, young children must segment speech into words, even though most words are unfamiliar to them. This is a non-trivial task because speech lacks any acoustic analog of the blank spaces between printed words. Two sources of information that might be useful for this task are distributional regularity and phonotactic constraints. Informally, *distributional regularity* refers to the intuition that sound sequences that occur frequently and in a variety of contexts are better candidates for the lexicon than those that occur rarely or in few contexts. We express that intuition formally by a class of functions called *DR functions*. We then put forth three hypotheses: First, that children segment using DR functions. Second, that they exploit phonotactic constraints on the possible pronunciations of words in their language. Specifically, they exploit both the requirement that every word must have a vowel and the constraints that languages impose constraints on word-initial and word-final consonant clusters. Third, that children learn which word-boundary clusters are permitted in their language by assuming that all permissible word-boundary clusters will eventually occur at utterance boundaries. Using computational simulation, we investigate the effectiveness of these strategies for segmenting broad phonetic transcripts of child-directed English. The results show that DR functions and phonotactic constraints can be used to significantly improve segmentation. Further, the contributions of DR functions and phonotactic constraints are largely independent, so using both yields better segmentation than using either one alone. Finally, learning the permissible word-boundary clusters from utterance boundaries does not degrade segmentation performance.

Knowing a language implies having a mental lexicon—a memorized set of associations among sound sequences, their meanings, and their syntactic privileges. One of the major difficulties children face in acquiring a lexicon is that the speech stream lacks any acoustic analog of the blank spaces between printed words. Thus, the basic units of linguistic input are not words but entire utterances. The child's task is not merely to identify the meaning and syntax associated with each word, but also to discover the words themselves.[1] This paper presents a partial theory of how children discover words and a set of computer simulations aimed at determining the adequacy of this theory.

We view lexical acquisition as having phonological, semantic, and syntactic components. The phonological component identifies candidate words in an utterance by segmenting it without reference to meaning or syntax. The other components identify the meaning and syntactic privileges of the candidate words, wherever possible. Candidates for which consistent semantics or syntax are never found do not become part of the lexicon itself. Thus, the other components help to filter out any errors in phonological segmentation. Conversely, the phonological component focuses the efforts of the syntactic and semantic components on sound sequences that are likely to be words. This model is consistent with various degrees of phonological focusing. At one extreme, the phonological component might segment perfectly, as assumed in most attempts to explain the syntactic and semantic components of lexical acquisition (e.g., Pinker, 1984; Siskind, this issue). At the other extreme, the phonological component might produce many alternative segmentations of each utterance, shifting the entire burden of word identification to the semantic and syntactic components. However, these components appear to be overburdened already, in that current theories are hard-pressed to explain their function even under the assumption of perfect phonological segmentation. Thus, it seems most likely that the phonological component carries a substantial part of the burden of segmentation. In this paper, we assume that to be the case.

[1] By *utterance*, we mean a complete act of speaking, surrounded by silent pauses and ending in the prosodic characteristics typical of clause boundaries (Fisher & Tokura, 1996; Jusczyk & Nelson, 1996). Utterances are typically sentences, noun-phrases, interjections, or, sometimes, isolated content words. By *word*, we simply mean a sound sequence for which a native speaker of a language, after exposure to appropriate input, forms a memorized association with meaning and/or syntactic privileges. The question of which linguistic entities are words in this sense is considered further in the general discussion.

The paper is organized as follows. The remainder of the introduction surveys possible theories of the phonological component of children's word discovery procedure. The next section presents our own hypotheses, followed by a series of simulation experiments. Finally, the discussion critically assesses the simulations, relates our hypotheses to existing behavioral evidence, and discusses prospects for future research.

Isolated Words versus Segmentation

When the absence of acoustic spaces is pointed out, many people express the intuition that children learn words by attending to single-word utterances. This intuition has never been developed into a coherent theory, but one possible interpretation is this: Children can discriminate single-word utterances from multi-word utterances. They acquire words by entering the single-word utterances into their mental lexicons. This idea has several weaknesses:

1. No adequate method for discriminating between single-word and multi-word utterances has ever been proposed (Christophe, Dupoux, Bertoncini, & Mehler, 1994).

2. Only words that occur in isolation can be learned. Some words—determiners, for instance—are very unlikely to occur in isolation. Even when instructed to teach their 12-month-olds a noun referring to a body part, some mothers use that word in isolation rarely or not at all (Aslin, Woodward, LaMendola, & Bever, 1996).

3. Adults are obviously able to learn new words by segmenting them out of multi-word utterances. The proposal that infants learn words by some completely separate mechanism implies a maturational switch whose mechanism remains unexplained. Further, adults' ability to segment novel words out of multi-word utterances would require a separate explanation.

4. Laboratory evidence shows that children have substantial segmentation capabilities by the onset of lexical acquisition. Jusczyk and Aslin (1995) familiarized 7.5-month-old American infants with two different monosyllabic words and then presented them with passages that either did or did not contain the familiar words in sentences. The infants listened significantly longer to passages with familiar words than to passages with unfamiliar foils. They also listened longer when the familiarization and test stimuli were reversed, so that the test phase required the infants to recognize isolated words with which they had been familiarized in sentential context. Similar results have since been obtained for bisyllabic

> words as well (Newsome & Jusczyk, 1995). These results indicate that, at 7.5 months, infants are able segment speech well enough to identify words in sentential context. It would be surprising if this ability were not used for early lexical acquisition.

The naive isolated-words account has largely been rejected in favor of the view that processing multi-word utterances is critical for word learning. On this view, children must consider as candidate words sound sequences they have never heard in isolation. Two broad categories of segmentation strategies have been proposed to explain how they might accomplish this. *Word form* strategies exploit regularities that distinguish the phonetic or phonological forms of words from those of certain non-word sequences. For example, sequences beginning in two stop consonants, such as “pcat”, cannot be words in English. *Sub-sequence frequency* strategies are based on the frequencies of various sound sequences in the input. The next two sections focus on word form strategies and sub-sequence frequency strategies, respectively.

Word Form Strategies

One class of word form strategies is based on allophony—the fact that there can be systematic differences among the ways a phoneme is pronounced when it occurs word-initially, word-internally, and word-finally. For example, stop consonants tend to be aspirated more strongly when they occur word-initially than when they occur word-internally (Church, 1987). In French, when a vowel followed by a stop consonant spans a word boundary, both the vowel and the stop closure tend be longer than when the same phonemes occur word-internally. Further, French infants can distinguish phonemically identical sound sequences that differ only in whether they were spliced out of sentences in which they spanned a word boundary (Christophe et al., 1994).

Another frequently discussed strategy is based on regularities in the rhythmic forms of words. For example, initial syllables of English content words rarely have reduced vowels (Cutler & Carter, 1987, estimate 10%). Conversely, internal syllables of English content words usually have reduced vowels (Cutler & Carter estimate 75%). Stress in English is highly correlated with full vowels, so this is roughly equivalent to saying that the strong-weak stress pattern is much more common in English content words than the weak-strong pattern. It has been proposed that such regularities play a role in early word discovery (Cutler, 1994, 1996; Cutler, Mehler, Norris, & Segui, 1994). For example, children might prefer to place boundaries before all strong syllables, all other things being equal. Children learning other languages would use rhythmic patterns appropriate to their languages (Cutler et al., 1994). Jusczyk, Cutler, and Redanz (1993) have found that 9-month-old American infants listen longer to words with strong-weak stress (the predominant pattern in English) than to words with weak-strong stress, but 6-month-olds show no preference. This suggests that 9-month-olds may have the language-specific knowledge needed to pursue a stress-based strategy.

In this paper, we focus on a third word form strategy based on *phonotactic constraints*. For present purposes, there are three types of phonotactic constraints:[2]

- **Vowel Constraint:** Every word must contain a vowel.
- **Boundary Clusters Constraint:** Each language specifies a finite set of consonant clusters that can occur at the beginning of a word, before the first vowel. For example, “gdog” is not a possible word of English because the sequence “gd” is not permitted word-initially. Word-final consonant clusters are similarly restricted.
- **Internal Clusters Constraint:** Each language specifies a finite set of consonant sequences that can appear between two vowels within a word. For example, the central cluster in the utterance *whatsthis*—/tsð/—cannot occur within English words (except compounds).

Taken together, the first two constraints allow only two segmentations of the pronunciation of *bigdog* into two words:

Segmentation	Constraints violated
b igdog	Vowel
bi gdog	Boundary Clusters
big dog	
bigd og	
bigdo g	Vowel

We propose that children use the vowel constraint and the boundary clusters constraint in early segmentation. In Experiments 2–4 below we investigate the usefulness of these constraints for segmentation.

Although there are cross-linguistic generalizations, the specific sequences permitted by each of the three constraints vary from language to language.

[2] Although there is a great deal of interesting theory involving phonotactic constraints (e.g., Clements, 1990, and references therein), refining the crude description supplied here would distract from our main point.

Thus, the theory that children exploit phonotactic constraints for early segmentation is tenable only if children know something about the specific constraints of their language at the onset of lexical acquisition. Laboratory evidence suggests that they do. In several experiments, 9-month-olds listened longer to words that are phonotactically permissible in their language than to those that are not, while 6-month-olds showed no preference (Friederici & Wessels, 1993; Jusczyk, Friederici, Wessels, Svenkerud, & Jusczyk, 1993).

Ultimately, the theory that children exploit phonotactic constraints in early word discovery requires an explanation of *how* the constraints are learned before the word discovery problem has been solved. We propose the following explanation for how boundary clusters are learned: Infants assume that the permissible word-initial clusters are exactly those that occur utterance-initially and the permissible word-final clusters are exactly those that occur utterance-finally. Experiment 4 below investigates the effectiveness of this strategy. We have not developed a specific proposal for how children learn which sounds count as vowels for the purposes of the vowel constraint in their language, but Ellison's (1991, in press) algorithm provides a promising lead.

Sub-Sequence Frequency Strategies

Sub-Sequence frequency strategies can be divided into two types: Word-based strategies use the words they have already found to segment new utterances. Boundary-based strategies attempt to identify individual word boundaries in the input, without reference to words per se. They segment out words only as a side effect of finding word boundaries—the words that are segmented out are not used for identifying other boundaries. We first review two boundary-based approaches, then one word-based approach. In the next section, we propose a novel word-based approach.

Boundary-based Strategies Harris (1954, 1955) proposed the first sub-sequence strategy as a method for linguistic analysis, rather than a strategy for child language acquisition. Harris's (1955) method for segmenting phoneme strings into morphemes involves counting the number of different phonemes that can follow each initial sub-sequence of an utterance. One of his examples is based on the utterance *he's quicker*, transcribed as /hɪyzkwɪkər/. He determined the following successor counts:

Init. subseq.	Successors	Init. subseq.	Successors
/h/	9	/hɪyzkw/	6
/hɪ/	14	/hɪyzkwɪ/	10
/hɪy/	29	/hɪyzkwɪk/	28
/hɪyz/	29	/hɪyzkwɪkə/	14
/hɪyzk/	11	/hɪyzkwɪkər/	28

The number for each substring was obtained by checking, for each phoneme of English, whether or not there is a grammatical utterance of English beginning with that substring and continuing with that phoneme. For instance, the count of 29 for /hɪyzk/ was based on examples such as *He's cranky*, *He's quiet*, and so on. In all, 29 different phonemes can follow /hɪyzk/. Harris's method is to posit morpheme boundaries after phonemes whose successor count is no less than that of their immediate neighbors. Given the data tabulated above, his method yields the correct analysis of /hɪyzkwɪkər/ as /hɪy·z·kwɪk·ər/. However, Harris hedges by saying that "Very small rises... are not, in general, a basis for segmentation." This makes it clear that his method is intended as a guideline to linguists rather than an autonomous procedure of the type that would be needed for language acquisition.

Saffran, Newport, and Aslin (in press) propose a related strategy based on estimating the conditional probabilities of a syllable given its predecessor. For example, the conditional probability that the next syllable will be /no/, given that the previous syllable was /ki/, is written Pr(/no/|/ki/) and defined as

$$\mathrm{Pr}(/\mathrm{no}/|/\mathrm{ki}/) = \frac{\mathrm{Pr}(/\mathrm{kino}/)}{\mathrm{Pr}(/\mathrm{ki}/)}$$

The probabilities on the right-hand side can be estimated by their relative frequencies in the input, yielding

$$\mathrm{Pr}(/\mathrm{no}/|/\mathrm{ki}/) \doteq \frac{\mathrm{freq}(/\mathrm{kino}/)}{\mathrm{freq}(/\mathrm{ki}/)}$$

Saffran, et al. suggest that, "A learner, then, might hypothesize word boundaries upon discovering troughs in the transitional probabilities between syllables."

Neither Harris's method nor Saffran, et al.'s approach attempt to find more instances of the words they segment out of utterances. While these ideas could provide hints to the child's segmentation procedure, it seems most likely that the segmentation procedure uses the words it has already found as a source of information about how to segment new utterances.

Word-based Strategies Olivier (1968) proposed a procedure that maintains a lexicon and uses it for segmenting. This procedure treats the input as an unbroken sound sequence, ignoring utterance boundaries. It is based on a class of probabilistic models called *word grammars*. A word grammar consists of a finite set of sound sequences representing words, called the *logical lexicon*, and a probability distribution on those sequences. The model assumes that the input was generated by stringing together words selected at random according to the probability distribution given by the word grammar. For example, consider the following word grammar:

Logical lexicon	Probabilities
a	.1
b	.1
aa	.4
bb	.4

It can produce the string *aabb* in four different ways, with probabilities ranging from $.4 \times .4 = .16$ for *aa bb* to $.1 \times .1 \times .1 \times .1 = .0001$ for *a a b b*.[3] In this case, the most likely segmentation of *aabb* is *aa bb*. If the probabilities are changed to

Logical lexicon	Probabilities
a	.5
b	.2
aa	.1
bb	.2

then *a a bb* becomes the most likely segmentation, with probability $.5 \times .5 \times .2 = .05$.

It is relatively straightforward to construct an algorithm that will find the most likely segmentation of an input, such as *aabb*, given a fixed word grammar. Conversely, the frequencies of words in a segmented input can be used to estimate the probabilities of those words in the word grammar that generated the input. For example, the segmented input "do you see the kitty see the kitty do you like the kitty" suggests a word grammar of the form

[3] We have glossed over some of the mathematical niceties—see Olivier (1968) for details.

Logical lexicon	Estimated probabilities
do	2/13
you	2/13
see	2/13
the	3/13
kitty	3/13
like	1/13

where the probability of each word is estimated by its frequency in the input divided by 13, the total number of word tokens in the input.

Before processing any input, Olivier's algorithm begins with an arbitrary word grammar. It then processes each chunk of input as follows. First, the most likely (maximum-likelihood) segmentation of the current chunk is computed, assuming that the current word grammar is correct. Then, the frequencies of the words in this segmentation are added to their frequencies in the segmentations of all previous chunks. The probabilities for the current grammar are then re-estimated on the basis of the new frequencies. So far, the algorithm is quite elegant and mathematically well-founded.[4] However, it does not solve the word discovery problem. Words that are not in a given word grammar cannot occur in a maximum-likelihood segmentation based on that grammar. Thus, parsing and re-estimating can change the probabilities of words, but can never result in learning new words. This problem is inherent in the maximum-likelihood approach. Each word in a word grammar corresponds to one parameter of the probability model, and the maximum-likelihood approach provides no guidance about how to change the set of parameters in the model. In order to make the algorithm work, Olivier uses an ad hoc mechanism for adding frequency-one words to the logical grammar and removing them periodically if they never occur in segmentations. The resulting algorithm fails to suggest general principles of language learning.

[4] Indeed, algorithms of this type have since become a popular method in statistical estimation for other language applications. See, e.g., Pereira and Schabes (1992), and sources cited therein.

Segmentation as Optimization

We view segmentation as an *optimization* problem. A simple example of optimization is finding the value of x between -1 and 1 for which x^2 is smallest.[5] One way to approach this problem is by *search and evaluation*—evaluate x^2 for various x's between -1 and 1, retaining the one for which x^2 is smallest among all the x's tried. Many different search procedures are possible. For example, values between -1 and 1 could be selected at random with uniform probability. A more systematic procedure would be to try

$$x = \frac{1}{2}, \quad x = -\frac{1}{2}, \qquad x = \frac{1}{4}, \quad x = -\frac{1}{4}, \qquad x = \frac{1}{8}, \quad x = -\frac{1}{8}, \qquad \cdots$$

In general, optimization is the attempt to find the element of some *domain* for which the value of some *objective function* is minimal (or maximal). Objective functions are also known as *cost functions* or *evaluation functions.* In the previous example, the objective function was $f(x) = x^2$, and the domain was the real numbers between -1 and 1. However, the domain need not be numeric. In our case, the domain is the set of all possible segmentations of an input sample. Our objective function yields a number for each segmentation. Given an input sample, the goal of the optimization is to find the segmentation of that sample for which the objective function is minimal.

An Objective Function for Segmentation

Later in this section, one particular objective function for segmentation is defined. Its properties are then used to define a class of related objective functions. First, however, it is worth considering some example values of the function to be defined. Suppose the input consists of the following three utterances:

doyouseethekitty
seethekitty
doyoulikethekitty

Figure 1 shows the value of the objective function for six different segmentations of these utterances. The objective function yields the smallest value on the segmentation that is correct, except for treating *thekitty* as a single word. This mistake is plausible, since *the* and *kitty* always occur in sequence in this small input sample. The objective function is slightly larger for the correct segmentation. The third best segmentation shown is the one in which *you*

[5] This particular optimization problem can be solved analytically, but many others cannot. For the sake of illustration, we ignore the analytic solution.

Segmentation	*f*	Segmentation	*f*
do you see $\boxed{\text{thekitty}}$		doyouseethekitty	
see $\boxed{\text{thekitty}}$	35	seethekitty	50
do you like $\boxed{\text{thekitty}}$		doyoulikethekitty	
do you see the kitty		d o y o u s e e t h e k i t t y	
see the kitty	39	s e e t h e k i t t y	75
do you like the kitty		d o y o u l i k e t h e k i t t y	
do $\boxed{\text{yousee}}$ the kitty		doy ous eeth ekitty	
see the kitty	45	see thekitt y	76
do you like the kitty		do youl ike thek itty	

Figure 1: Example values of an objective function for segmentation. Boxes highlight small differences from the correct analysis.

Segmentation	Representation		Length
	Lexicon	Derivation	(Objective)
do you see $\boxed{\text{thekitty}}$	1 do 2 $\boxed{\text{thekitty}}$ 3 you 4 like 5 see	1 3 5 $\boxed{2}$	25+10=35
see $\boxed{\text{thekitty}}$		5 $\boxed{2}$	
do you like $\boxed{\text{thekitty}}$		1 3 4 $\boxed{2}$	
do you see the kitty	1 do 2 the 3 you 4 like 5 see 6 kitty	1 3 5 2 6	26+13=39
see the kitty		5 2 6	
do you like the kitty		1 3 4 2 6	
do $\boxed{\text{yousee}}$ the kitty	1 do 2 the 3 you	1 $\boxed{7}$ 2 6	33+12=45
see the kitty	4 like 5 see 6 kitty	5 2 6	
do you like the kitty	$\boxed{7}$ $\boxed{\text{yousee}}$	1 3 4 2 6	

Figure 2: Three segmentations (left) with the corresponding representations (center) and their lengths in characters (right). Boxes highlight differences from the correct analysis.

and *see* are treated as a single word whenever they occur in sequence. This is not a very plausible mistake, since both *you* and *see* occur separately in the input sample. This implausibility is reflected in the relatively high value of the objective function. The segmentation in which each utterance is treated as a single word yields a still worse (higher) value. The segmentation in which each letter is treated as a single word yields a much worse value. Finally, a segmentation in which each hypothesized word occurs only once yields the worst value of all those shown. Of course, there are many more possible segmentations of this input—the ones shown here were selected to make particular points.

Having given some examples of an objective function, we now show how to compute that function for any segmentation of any input. Three examples of the computation are shown in Figure 2. The leftmost column of Figure 2 shows three alternative segmentations and the rightmost column shows the corresponding values of the objective function. These values are computed by first constructing a representation of the segmentation and then counting the total number of characters (letters and digits) in that representation. The middle columns of Figure 2 show the representation corresponding to each segmentation. The representation for a given segmentation has two parts, called the *lexicon* and the *derivation*.

1. The *lexicon* for a given segmentation is constructed by (a) listing the word types that occur in that segmentation, (b) assigning each word type an arbitrary, unique index, and (c) recording the index next to the word. For example, the index of the hypothesized word *thekitty* in the top representation is 2.

2. The *derivation* is constructed by replacing each word token in the segmentation with its index. For example, the last word in the top segmentation is *thekitty*, so the last index of its derivation is 2, the index of *thekitty*.

The value of the objective function for a segmentation is the total number of characters (letters and digits) in its lexicon, plus the total number of characters in its derivation. For example, the representation of the first segmentation in Figure 2 has a lexicon consisting of 25 characters and a derivation consisting of 10 characters. Summing the number of characters in the two components yields 35, the value of the objective function applied to this segmentation.

To get an intuitive feel for how this objective function works, it is worth comparing the computation for the first row in Figure 2, where *thekitty* is treated as a single word, to that of the third row, where *yousee* is treated as a single word. The objective function is higher in the *yousee* case primarily because *you* and *see* occur separately in the input, so they must have separate entries in the lexicon *in addition* to the entry for *yousee*. The words *the* and *kitty*, on the other hand, always occur together, so the entry for *thekitty* renders separate entries unnecessary. This illustrates what might be described as the *context variability effect*: The objective function expresses a preference for interpreting sub-sequences that occur in a variety of contexts as independent words. There is also a *frequency effect*: The more frequently two sub-sequences occur adjacent to one another, the more the objective function will reward treating them as a single word when they occur together. This effect can be seen by comparing the derivation portions of the *yousee* and *thekitty* examples. In both cases, fewer indices are needed when two adjacent sub-sequences are treated as a single word—that is, represented by a single index. However, this savings in indices is much greater for *thekitty*, which occurs three times, than for *yousee*, which occurs only once. These effects can be summarized as follows: the more frequently a sequence occurs, and more different contexts those occurrences are in, the more the objective function will reward treating that sequence as a word.

Let us assume for a moment that all indices consist of just one digit. (Index lengths are considered further below.) Then the number of digits in the derivation portion is the same as the number of word tokens in the segmentation. The number of digits needed to list the indices in the lexicon is the same as the number of words in the lexicon—i.e., the number of word types in the segmentation. The number of phonemes (or, in Figure 2, letters) needed to list the words in the lexicon is the sum of the lengths of the word types in the segmentation. Thus,

$$f(S) = |\textsc{tokens}(S)| + |\textsc{types}(S)| + \sum_{w \in \textsc{types}(S)} \ell(w) \tag{1}$$

where S is a segmentation, $\textsc{tokens}(S)$ is the set of word tokens in segmentation S, $\textsc{types}(S)$ is the set of word types in segmentation S, and $\ell(w)$ is the length of word w, measured in phonemes. (Vertical bars around a set indicate the number of items in the set.) This formula makes it possible to compute the objective function for a segmentation without actually constructing its representation.

Minimimum Representation Length Induction

This objective function was obtained by applying the Minimum Representation Length (MRL)[6] method (Rissanen, 1989; Wallace & Boulton, 1968) to the segmentation problem. The MRL method involves constructing a system for representing an input sample in terms of a hypothesis about the system that generated the sample (e.g., the lexicon) and a hypothesis about how that particular sample was generated by the system (e.g., the derivation). The value of the objective function on a given hypothesis is the length of the representation of that hypothesis. MRL is rooted in statistical inference and can be viewed as a mathematical generalization of the Maximum-Likelihood approach Olivier used. MRL has been applied to language (e.g., Ellison, in press; Stolcke & Omohundro, 1994) and to modeling language acquisition (Brent, Gafos, & Cartwright, 1994).

The representational schema illustrated in Figure 2, while simple, is by no means unique. A number of variations are possible. For example, word indices could be represented using only the binary digits 0 and 1 rather than the decimal digits 0 through 9. If the number of word types were between 3 and 9, then some of them would have muli-digit indices in the binary representation, whereas all word types could have single-digit indices in the decimal representation. In general, the smaller the set of digits used to represent indices, the faster index length grows as a function of type count, and the more important indices become in the overall objective function.

[6] MRL is also known as Minimum Description Length (MDL).

Another possible variation involves the assignment of indices to word types. For example, all word types could be assigned indices of equal length. Thus, if the decimal digits are used to represent indices and the lexicon contains between 11 and 100 word types, then all word types would be assigned two-digit indices: 00, 01, 02, . . . , 99. A better alternative, however, is to assign indices to word types in such a way that the most common words get shorter indices than the rarest words.

For each variant of the basic representational schema, there is a variant of the objective function in Equation (1). However, these variants are all quite similar, in that they all increase with:

1. The number of word types in the segmentation

2. The number of word tokens in the segmentation

3. The sum of the lengths (measured in phonemes) of the words in the lexicon

Segmentation algorithms that try to minimize such objective functions will tend to prefer segmentations with fewer types over those with more, segmentations with fewer tokens over those with more, and segmentations with shorter tokens over those with longer ones. If shorter indices are assigned to more common words, then the representation length also depends on the frequency distribution of the words. The most natural frequency-dependent methods of assigning indices yield objective functions that, all other things being equal, increase with:

4. The entropy of the relative frequencies of the words

An alternative terminology for entropy is the *average information*, per word token (Shannon & Weaver, 1949). Segmentation algorithms that try to minimize entropy will tend to prefer segmentations in which a few words account for most of the frequency over those in which the frequency is divided more evenly. We call any function that has these four properties a *distributional regularity (DR) function.*[7]

Three Hypotheses

We are now in a position to state our three hypotheses:

[7] More precisely, a DR function is any function that increases when each of the four factors either increases or remains constant.

1. Children use multiple strategies to determine which sound sequences to make lexical entries for, including a strategy based on a DR function. Specifically, they prefer segmentations on which the DR function has a lower value over those on which it has a higher value.

2. Children prefer segmentations that do not violate the vowel constraint or the boundary clusters constraint, as they apply in the language being acquired.

3. Children learn the phonotactic constraints of their language from unsegmented utterances. Thus, the learning of phonotactic constraints does not depend on the success of early attempts at segmentation. Specifically, they learn the legal boundary clusters of their language by attending to utterance boundaries (Aslin et al., 1996).

The first hypothesis leaves open the question of which DR function children use, or even whether all children use precisely the same one. Rather, it rests on the notion that a wide variety of DR functions are useful for segmentation. Likewise, it does not say anything about the search procedure children use—that is, how they decide which segmentations to evaluate before settling on one. Further, the second hypothesis makes no claims about whether children also use other word form strategies.

The next section explores the degree to which these hypotheses are capable of explaining the phenomenon of word discovery by infants. The method is computer simulation of the proposed strategies, using phonetic transcripts of child directed speech as input. The first experiment investigates the usefulness of DR optimization for segmenting utterances into words, starting from a state in which no words are known. The second experiment investigates the usefulness of the vowel constraint and the boundary clusters constraint for a child who is segmenting by DR optimization. The third experiment investigates the interaction between the segmentation method and the use of phonotactic constraints. The fourth experiment investigates how the hypothesized strategy for learning the permissible boundary clusters affects the performance of a learner using DR optimization.

Experiment 1

In this experiment we attempt to determine whether DR optimization is useful for segmenting utterances into words, starting from a state in which no words are known. To do this, we designed a specific algorithm for optimizing DR functions, implemented it as a computer program, and measured its performance on a model of the child's segmentation task. The model task is segmenting phonetic transcripts of child-directed speech into 'words,' where a word is defined as the string of phonemes corresponding to the letters that would appear between spaces in orthographic form. That is, a word is a printer's word. Although this model task may differ from the one children face, we believe it is similar enough to provide insight into the natural task (see General Discussion).

To determine whether DR optimization is useful, we needed some baseline against which to compare it. For forced-choice tasks, equi-probable random choice is used as a baseline. The closest analogy for segmentation is inserting word boundaries at random positions in the input, with equal probability at each position. The problem, though, is how many word boundaries to insert. The solution we chose is to insert the *correct* number of word boundaries at random. It is important to note that this baseline does not represent pure "chance." It exploits a very valuable piece of knowledge that neither the DR optimization algorithm nor, presumably, infants have access to—the number of words in the input. If the DR algorithm performs as well as the baseline, that will demonstrate that it is as useful as knowing the number of words in the input, which is far from trivial. To emphasize this point, we call the baseline algorithm *word count* rather than *random.* Because the word count algorithm is stochastic, we ran it 1,000 times on each input sample and averaged the scores.

Method

Algorithm The DR optimization hypothesis is compatible with a number of different algorithms, each consisting of a DR function and a search procedure. The DR function used in these experiments is

$$f(S) = 3|\textsc{types}(S)| + (\log_2 P)\left(\sum_{w \in \textsc{types}(S)} \ell(w)\right) + |\textsc{tokens}(S)| \times H(S) \quad (2)$$

where P is the number of phonemes in the input alphabet and $H(S)$ is the entropy of the relative frequencies of the words in the segmentation:

$$H(S) = -\sum_{w \in \textsc{types}(S)} \frac{f(w)}{|\textsc{tokens}(S)|} \log_2 \frac{f(w)}{|\textsc{tokens}(S)|}$$

Appendix A shows how (2) was derived from the number of characters needed to represent S in a particular representational system.

The search portion of the algorithm organizes alternative segmentations according to the number of word boundaries they contain, beyond the utterance boundaries that are fixed by the input. The search begins by applying the objective function f to the input with no added word boundaries. It proceeds by adding boundaries until it reaches the segmentation in which there is a word boundary between each pair of adjacent phonemes (see Figure 3). Either one or two new boundaries is added at a time. We use the term *major step* to describe the process during which the algorithm decides whether to add one new boundary or two, and where to add the new boundary or boundaries. In Figure 3, major steps are separated by horizontal lines. Each major step consists of computing the objective function f on a number of alternative segmentations and selecting the one on which the objective function is smallest. (The segmentations selected at each step are boxed in the figure.) The segmentations that are evaluated at each major step are modifications of the one selected at the previous major step. Specifically, all the segmentations that result from adding either one or two new boundaries to the segmentation selected at the previous major step are evaluated. When no more boundaries can be added, the segmentation on which the objective function has the smallest value—among all those that were evaluated—is the output of the algorithm.

At the first major step in Figure 3, the segmentation *a pet a dog* is chosen, because it contains only three word types whose lengths sum to seven characters. In this example, adding more boundaries to *a pet a dog* increases the number of types and tokens without reducing the sum of the lengths of the types. Thus, in this example the best segmentation found at each subsequent major step is worse than the best segmentation found at the first major step. Eventually, the search reaches the segmentation in which each pair of adjacent phonemes is separated by a word boundary, so no more boundaries can be inserted. At that point, it outputs the segmentation on which f has the smallest value, among all those evaluated in the entire search. In Figure 3, the correct segmentation—*a pet a dog*—would be returned.

Input Both the DR optimization algorithm and the word count algorithm were applied to broad phonetic transcripts of spontaneous child-directed English. Orthographic transcripts made by Bernstein-Ratner (1987) were taken from the CHILDES collection (MacWhinney & Snow, 1985) and transcribed phonetically. The speakers were nine mothers speaking freely to their children, whose ages averaged 18 months (range 13–21). Although these children are a bit beyond the earliest stages of lexical acquisition, very few transcripts of speech to infants are publicly available, and this was the best collection for our purposes.

The philosophy behind our transcription was to preserve all phonemic distinctions while minimizing the number of subjective judgments and the amount of labor required. Accordingly, every word was transcribed the same way every time it occurred, regardless of context. Diphthongs were transcribed as

Input: apet adog						
1	a pet adog	ap et adog	ape t adog	apet a dog	apet ad og	apet ado g
2	a p et adog	ap e t adog	ape t a dog	apet a d og	apet ad o g	
2	a pe t adog	ap et a dog	ape t ad og	apet a do g		
2	$\boxed{\text{a pet a dog}}$	ap et ad og	ape t ad og			
2	a pet a dog	ap et ad og	ape t ad og			
2	a pet ad og	ap et ado g				
2	a pet ado g					
1	$\boxed{\text{a p et a dog}}$	a pe t a dog	a pet a d og	a pet a do g		
2	a p e t a dog	a pe t a d og	a pet a d o g			
2	a p et a d og	a pe t a do g				
2	a p et a do g					
1	$\boxed{\text{a p e t a dog}}$	a p et a d og	a p et a do g			
2	a p e t a d og	a p et a do g				
2	a p e t a do g					
1	$\boxed{\text{a p e t a d og}}$	a p et a do g				
2	a p e t a d o g					
1	$\boxed{\text{a p e t a d o g}}$					

Figure 3: The search algorithm used in these experiments, applied to the input sample *apet adog*. Horizontal lines separate major steps. The best segmentation found at each major step is highlighted with a box. The leftmost column indicates the number of new boundaries inserted into the best segmentation found in the previous major step.

a single character, as were r-colored vowels. Syllabic consonants were distinguished from the corresponding non-syllabic consonants. Thus, the first and last sounds in *label* were represented by different characters. Onomatopoeia (e.g., *bang*) and interjections (e.g., *uh* and *oh*) were eliminated, for three reasons: they occur in isolation much more frequently than ordinary words, and hence would have skewed the scores toward better performance; their frequency is highly variable from speaker to speaker, so they would have increased the variance in the scores; and there is no standard spelling or pronunciation for many of them, so we could not tell from the orthographic transcript what sound was actually uttered.

Finally, all word boundaries were removed, but utterance boundaries marked in the Bernstein-Ratner transcripts were left intact. We did not insert any additional utterance boundaries, even at obvious sentence boundaries. (Apparently, the original transcript contains utterance boundaries only between utterances separated by an audible pause.) Each transcript was truncated to about 1350 segments, not counting those that immediately precede an utterance boundary. Truncation was necessary both because simulations on longer input samples would have required excessive computer time, and because the lengths of original transcripts varied widely. Segments preceding utterance boundaries were not counted in order to balance the number of points where there is a decision to make about the presence of a word boundary. The average number of words per transcript was 527 and the average number of utterances was 170.

Scoring To score the segmentation produced by an algorithm, we lined it up, phoneme-by-phoneme, with the standard orthographic segmentation. This is illustrated in Figure 4, where "|" indicates a word boundary. Each word in the algorithm's segmentation was scored as a true positive (hit) if it lined up exactly with a word in the standard segmentation; otherwise it was scored as a false positive (false alarm). Each word in the standard segmentation that did not line up exactly with a word in the algorithm's segmentation was scored as a false negative (miss). For example, the segmentation in the top half of Figure 4 contains one true positive (*do*), three false positives (*yousee*, *thek*, *itty*), and four false negatives (*you*, *see*, *the*, *kitty*). However, if a boundary is added between *you* and *see*, as in the bottom half of Figure 4, the result is three true positives (*do*, *you*, *see*), two false positives (*thek*, *itty*), and two false negatives (*the*, *kitty*).

These raw scores were then converted to two measures of performance, the percent *accuracy* and percent *completeness* of the algorithm's segmentation,

		TP	FP	FN
Standard	d o|y o u|s e e|t h e|k i t t y			
Algorithm	d o|y o u s e e|t h e k|i t t y	1	3	4
Standard	d o|y o u|s e e|t h e|k i t t y			
Algorithm	d o|y o u|s e e|t h e k|i t t y	3	2	2

Figure 4: The computation of true positives (TP), false positives (FP), and false negatives (FN) for two segmentations that differ by the addition of a single boundary.

defined as follows:

$$\text{accuracy} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

$$\text{completeness} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

For example, the top segmentation of Figure 4 scores 25% for accuracy and 20% for completeness, while the bottom segmentation scores 60% for both.

Results and Discussion

Table 1 shows the accuracy and completeness of the segmentations returned by both the DR and word count algorithms, averaged over the nine mother-child dyads. The DR optimization algorithm produced segmentations that were both significantly more accurate (t(8)=7.4, $p < .0001$)[8] and significantly more complete (t(8)=13.0, $p < .0001$) than those produced by the word count algorithm. This result indicates that this particular DR algorithm is substantially more useful for segmentation than knowing the number of words in the input.

Comparison of Error Types It is also interesting to compare the kinds of errors and omissions the two algorithms make. Two kinds of errors are of particular interest: separation of a stem from its inflection, and failure to separate

[8] All t tests in this paper use matched pairs.

	WC	DR
Accuracy	13.4 (1.9)	41.3 (12.7)
Completeness	13.4 (1.9)	47.3 (9.1)

Table 1: Percent accuracy and completeness for the Word Count (WC) algorithm and the DR optimization algorithm (DR), averaged over nine mother-child dyads. Standard deviations are shown in parentheses.

two or more correct words (concatenation). Isolating stems may be useful to the child. Concatenations, which are often semantically and syntactically coherent units, may be less problematic than other types of errors. About 1.1 percent of the putative words found by the word count algorithm are actually stems, whereas the figure for the DR optimization algorithm is about 3.5 times greater. Thus, if stems are viewed as unproblematic, the difference in accuracy between DR optimization and word count is greater than indicated by Table 1. About 6.7 percent of the putative words found by the word count algorithm are actually concatenations, whereas the figure for the DR optimization algorithm is almost 1.5 times greater. Thus, if concatenations are viewed as "partially correct," once again the difference in accuracy between DR optimization and word count is greater than indicated by Table 1. This analysis suggests that DR optimization not only makes fewer mistakes and fewer omissions than the word count algorithm, but the mistakes and omissions it does make are less problematic.

Other DR Functions We have also tried two other DR functions, together with the same search procedure, and obtained the same pattern of results (data not shown). In fact, the average performance measures were nearly identical for all three DR functions. This suggests that the nature of the input, together with the general structure of DR functions, may be a more significant determinant of performance than which specific DR function is used.

Other Search Algorithms We found that, unlike the choice of DR function, the choice of search algorithm can affect performance substantially. For example, a search algorithm that was restricted to inserting only one boundary at each major step, rather than one or two, did not perform as well as the one described above. We speculate that having the option of inserting two boundaries simultaneously yields better performance primarily because it takes two boundaries to segment out a word from the middle of an utterance. Another search that did not perform as well was starting with word boundaries between every pair of adjacent phonemes and *removing* one or two boundaries at each major step. Presumably, this fails because utterance boundaries are indistinguishable from other positions in the initial state, where there are boundaries at all positions. Thus, utterance boundaries provide no information to guide the initial stages of the search.

Since we have explicitly refrained from speculating about the kinds of search procedures children might use, these observations do not bear directly on our DR hypothesis. Nonetheless, our hypothesis is only viable to the extent that there exists a search procedure that is both effective and psychologically plausible. The existence of an effective search procedure provides grounds for optimism, but the existence of less effective ones suggests that the issue of search procedures is not a trivial one.

Taken together, these results indicate that DR optimization could provide an effective engine to drive the segmentation process. Thus our first hypothesis, if correct, would go a long way toward explaining children's capacity for segmenting utterances, and hence their capcacity for acquiring a lexicon.

Experiment 2

Our second hypothesis is that children prefer segmentations that do not violate the vowel constraint or the boundary clusters constraint for their language. In this simulation experiment we set out to test whether these constraints would be useful to an English-learning child who was using them in conjunction with a DR function. To this end we ran the DR algorithm described in Experiment 1 again, but this time we constrained the search to segmentations in which every word contains at least one vowel and begins and ends with legal boundary clusters. The search terminates when adding another boundary at any position would violate one of the constraints. Figure 5 shows how these constraints affect the search in Figure 3. For this particular input, the entire effect is due to the vowel constraint. Notice that inserting an additional boundary at any position in the boxed segmentation would create a word without a vowel.

Method

Same as Experiment 1, except as described above. A list of boundary clusters permitted in English, which we compiled by hand and provided to the program, can be found in Appendix B.

Input: apet adog				
1	a pet adog	ap et adog	apet a dog	apet ad og
2		ap et a dog		
2	$\boxed{\text{a pet a dog}}$	ap et ad og		
2	a pet ad og			

Figure 5: The search shown in Figure 3 with the added restriction that every word must contain a vowel.

	DR	DR+V	DR+V+BC
Accuracy	41.3 (12.7)	68.2 (5.1)	75.7 (4.4)
Completeness	47.3 (9.1)	60.1 (6.8)	68.3 (6.9)

Table 2: Percent accuracy and completeness of the segmentations returned by the unconstrained DR algorithm (DR), the DR algorithm with the vowel constraint (DR+V), and the DR algorithm with both the vowel and the boundary clusters constraints (DR+V+BC), averaged over the nine mother-child dyads. Standard deviations are provided in parentheses.

Results and Discussion

Table 2 shows the accuracy and completeness of the segmentations returned by the unconstrained DR algorithm (DR), the DR algorithm with the vowel constraint (DR+V), and the DR algorithm with both the vowel and the boundary clusters constraints (DR+V+BC), averaged over the nine mother-child dyads. When all words were required to contain vowels, the DR algorithm produced segmentations that were both significantly more accurate (t(8)=8.5, $p < .0001$) and significantly more complete (t(8)=5.8, $p < .0004$) than when no constraints were imposed. When, in addition, all words were required to begin and end with legal word-boundary clusters, the DR algorithm showed another significant improvement in both accuracy (t(8)=7.5, $p < .0001$) and completeness (t(8)=5.4, $p < .0007$). Both constraints appear to be useful in segmenting child-directed English by DR optimization. Thus, if our first and second hypotheses are correct, they would represent a more complete explanation of

	Accuracy		Completeness	
	WC	DR	WC	DR
Unconstrained	13.4 (1.9)	41.3 (12.7)	13.4 (1.9)	47.3 (9.1)
+V+BC	39.9 (2.5)	75.7 (4.4)	39.9 (2.5)	68.3 (6.9)

Table 3: Percent accuracy and completeness of the segmentations returned by the four algorithms, averaged over the nine mother-child dyads. Standard deviations are provided in parentheses.

children's ability to segment utterances than the first hypothesis alone.

Experiment 3

In this experiment we investigate the interaction between the segmentation method and the use of phonotactic constraints. Logically, it is possible that DR optimization and the phonotactic constraints are partially redundant—i.e., they are picking up on some of the same linguistic regularities. On the other hand, the phonotactic constraints might guide the optimization into the right neighborhood of the search space, thereby giving an even greater boost to the DR optimization algorithm than to the word count algorithm. Finally, it is possible that the improvement in performance due to the DR strategy and that due to the phonotactic constraints are largely independent and additive. Investigating these possibilities required one additional simulation: the word count algorithm together with both phonotactic constraints.

Method

Same as Experiment 2.

Results and Discussion

Table 3 shows the accuracy and completeness of the segmentations returned by the four algorithms, averaged over the nine mother-child dyads. The phonotactic constraints boosted the mean completeness of the DR algorithm by 21.3 percentage points, less than the 26.5 percentage points by which they boosted the mean completeness of the word count algorithm. However, the phonotactic constraints boosted the mean *accuracy* of the DR algorithm by 34.4 percentage points, *more* than the 26.5 percentage points by which they boosted the mean accuracy of the word count algorithm. Taken together, these differences suggest that the phonotactic constraints are about equally valuable to the DR algorithm and the word count algorithm. That is, the contributions of phonotactics and DR optimization to the improvement in overall performance are largely independent.[9]

Experiment 4

So far, the simulations have used an externally provided list of permissible boundary clusters. Our third hypothesis is that children learn which clusters are permissible at word boundaries by assuming that they are the same as the ones that occur at utterance boundaries. This experiment tests the effect of such a strategy on the performance of a DR optimization algorithm using the vowel constraint. It is based on lists of all the clusters that occur at the beginnings and ends of utterances in a corpus of child-directed speech. The performance of a DR algorithm using these as the permissible clusters was compared to the performance the same algorithm using the complete list of permissible boundary clusters in Appendix B.

Method

Broad phonetic transcripts of spontaneous speech by nine mothers to their children were made, as in Experiment 1. There were three sessions with each dyad at intervals of several months, containing a total of 9,790 utterances. (The inputs in all the previous experiments consisted of the first part of the first session with each dyad.) Lists of all the clusters that occur at the beginnings and ends of utterances were compiled. The DR program was then run with the observed utterance-initial clusters as the only permissible word-initial clusters, and the observed utterance-final clusters as the only permissible word-final clusters.

Results and Discussion

Appendix C lists the 18 permissible word-initial clusters that *did not* occur utterance-initially and the 36 permissible word-final clusters that *did not* occur

[9] Because the variances differ greatly from one treatment to another, a simple ANOVA is not appropriate. However, the magnitudes and directions of the apparent differences make the question of statistical significance moot, so we did not pursue corrections or alternative statistics.

	HAND	UB
Accuracy	75.7 (4.4)	76.3 (5.0)
Completeness	68.3 (6.9)	69.2 (8.1)

Table 4: Accuracy and completeness of the DR algorithm using the vowel constraint and the hand-coded list of boundary clusters (HAND) versus the list of clusters gleaned from utterance boundaries (UB). Results are averaged over nine mother-child dyads, with standard deviations shown in parentheses.

utterance-finally. Given the large number of missing clusters one might expect a DR algorithm using the list gleaned from utterance boundaries to perform substantially worse than one using the complete list. Table 4 shows that this is not the case. On average the DR algorithm performed slightly *better* with the clusters gleaned from utterance boundaries than with the complete list, although this improvement did not approach statistical significance (accuracy $t(8) = 1.19, p < .22$; completeness $t(8) = 1.34, p < .27$).

Despite its lack of statistical significance, the reason for the apparent improvement is interesting. It is best illustrated by the voiced inter-dental /ð/, which occurs at the ends of a few, fairly rare English words, including *bathe*, *clothe*, and *seethe*. It also occurs at the beginning of some extremely common words, including *the*, *this*, and *that*. Since it is a permissible word-final cluster, segmentation algorithms can and do put it at the ends of words when it belongs at the beginnings. For example, /si ðə/ (*see the*) is frequently mis-segmented as /sið ə/. Since /ð/ does not occur at the end of any of the 9,790 utterances in our corpus, this error is not possible when the permissible word-boundary clusters are inferred from observed utterance-boundary clusters. The cost of avoiding these mistakes is that the few words that do end in /ð/ cannot be segmented correctly. However, correct segmentations with word-final /ð/ appear to be much rarer than the incorrect ones, so the "mistake" of forbidding it at the ends of words tends to improve performance.

These results suggest that the utterance-boundary strategy for inferring permissible word boundary clusters would provide the full benefit of the boundary clusters constraint, at least for children learning English. Thus our three hypotheses, if correct, would represent a significant step toward an explanation of children's capacity to segment that does not rely on externally provided knowledge of the language being learned.

General Discussion

These simulations were aimed at determining the degree to which the proposed strategies are capable of explaining word discovery by infants. The results show that these strategies, as we have implemented them, are useful for the task on which they were tested—discovering the standard orthographic segmentation of broad phonetic transcripts of child-directed English. This task is an imperfect model of the task infants face. We begin the discussion by focusing on possible differences between the model task and the natural task. Next, we consider the search algorithm used in the simulations and its relation to our hypotheses. Finally, we consider behavioral experiments that bear on our hypotheses.

The Model Task

Input The raw acoustic signal that strikes the infant's ear passes through many stages of processing before the specifically linguistic ones. However, it is not known precisely how the processed signal that serves as the input to lexical acquisition is represented. In our model task, the input is a broad phonetic transcript in which all occurrences of each word are transcribed identically. The representation that serves as input to early lexical acquisition might be similarly segmental in nature, but include more phonetic detail than our transcripts. On the other hand, it might use atomic units larger than the phonetic segment—for example, the syllable or mora (Bertoncini, Bijeljac-Babic, Jusczyk, Kennedy, & Mehler, 1988; Jusczyk, Jusczyk, Kennedy, Schomberg, & Koenig, 1994; Jusczyk & Derrah, 1987; Jusczyk, Kennedy, & Jusczyk, 1995). Or it might be something entirely different. However, DR functions apply to any input represented as a string of symbols, whether the symbols stand for broad phonetic classes, narrow phonetic classes, indivisible syllabic units, or something else. There is no guarantee that DR functions will be equally useful for all input codings, but in the absence of further evidence, our best guess is that the distributional properties of different input codings are not fundamentally different.

Similarly, phonotactic constraints can be reformulated for various input codings. A more detailed phonetic input might make phonotactics even more useful, by providing for more detailed phonotactic and allophonic cues to word boundaries (Christophe et al., 1994; Church, 1987).[10] A syllabic representation suggests reformulating the phonotactic constraints as specifying, for each syllable, whether it can occur word initially, whether it can occur word in-

[10] It would also raise the question of how children discover when two phonetically distinct sound sequences represent the same word—i.e., which phonetic differences are phonemic.

ternally, and whether it can occur word finally. Phonotactic constraints can probably be reformulated for other representations as well, and it seems likely that this would not fundamentally alter their usefulness.

To strengthen these intuitions, it would be worthwhile to develop alternative input codings and replicate the current simulation results using them. Such efforts can never be definitive, though, until the representation infants use is understood in detail.

Output It is not known precisely which linguistic entities people memorize and associate as a whole with syntactic and semantic properties. What is clear is that the units must be small enough to be recombined into all the sentences of the language, but large enough to carry meaning. In addition, some intuitions about the contents of the mental lexicon are fairly clear: It must contain monomorphemic content-words like *dog*, *bike*, and *nice*; and, it probably does not contain morphological fragments such as *bo* (from *bog*), *bigdo* (from *bigdog*), and *nicel* (from *nicely*), nor syntactic fragments like *toydown*, nor long, rare phrases like *difficulttwoandahalfmilehike*. In between the clear cases lie derived words like *interdepartmental*, compounds like *skywriter*, idioms like *kick the bucket*, and inflected forms like *dogs*.[11] It may even be that different speakers of the same language differ in which phrases they memorize as a whole and which they compose on-line from smaller memorized units.

In our model task, the goal was to segment the input into words, defined by the placement of spaces in orthography. Although it is not known which types of morphemes, words, and phrases are to be found in the mental lexicon, there is probably substantial overlap between its contents and the units defined by the placement of spaces in orthographic form. For example, a small majority of the tokens in child directed English are function words, and a larger majority are monomorphemic. Long words, including most derived forms, and long phrases, including most idioms, are rare in child directed speech. To the extent that the units of the mental lexicon differ from orthographic words, scoring the simulations by the orthographic standard does not obviously favor our hypotheses. It seems reasonably likely, then, that scoring against a more cognitively motivated standard, if one were available, would not alter the broad pattern of results reported here.

[11] Although, there is interesting work on the definition of *word* from the linguistic perspective, that work does not appear to bear on the question of which associations between sound, meaning, and syntax are memorized. Indeed DiSciullo and Williams (1987) claim the question of which forms are memorized is of no linguistic interest.

Search Procedures

The search algorithm used in the simulations 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. Thus, although the simulations demonstrate the existence of an effective search procedure, they leave open the question of whether there exists one that is both effective and cognitively plausible. Interestingly, a number of algorithms that model lexical access by adults can segment utterances composed entirely of known words on-line (e.g., McClelland & Elman, 1986; Norris, 1994). DR optimization with the current search method, on the other hand, segments an entire input corpus without any prior lexical knowledge. What is needed is an algorithm that segments individual utterances on-line using as much lexical knowledge as is available, but still functions adequately at the stage where it has little or no lexical knowledge.

Behavioral Experiments

Behavioral experiments in two paradigms bear on our hypotheses indirectly and could, with minor modification, be made to bear more directly. In this discussion we focus primarily on the DR optimization hypothesis, but the same paradigms could be used to investigate the effect of phonotactic constraints on segmentation.

Saffran, et al. (in press) investigated adults' ability to segment an artificial language based on the phonemes of their native language. Their subjects were told that they would hear a "nonsense language" that contained words, but no meaning or grammar. They were then exposed to three seven-minute blocks of continuous synthesized syllables in which the six trisyllabic words of the artificial language were repeated in random order, with no pauses. Afterward, the subjects were required to choose the "word" from pairs consisting of one word of the artificial language and one non-word. The fact that subjects performed significantly above chance suggests that they were responding to the frequency of sub-sequences of the input, since no other cues were present. This is what our DR optimization hypothesis would predict, although other sub-sequence frequency strategies could make the same prediction. In fact, one choice in each test pair did not occur anywhere in the training stimuli, so the subjects may have been responding to this difference rather than pursuing any segmentation strategy per se.

Relevant results are also beginning to emerge from work with infants in the head turn preference paradigm (Nelson et al., 1995). Initially looking for effects of allophonic cues to segmentation, Jusczyk, Hohne, and Bauman familiarized 9-month-olds with either *night rate* or *nitrate*, and then exposed them to passages containing one of these two stimuli in sentential context (Jusczyk, personal communication). They did not find any evidence that the infants discriminated between the passage containing the familiar stimulus and the passage containing the unfamiliar one. They then did another experiment in which the *night rate* passage was replaced by one in which each occurrence of *night* was the first word in a different compound—e.g., *nightcap, nightgown, nightgame*, etc. Otherwise, the new passage was structurally similar to the old one. This time, the subjects did listen significantly longer to the new passage, where *night* was followed by a variety of words. This is precisely what would be expected if the subjects were using DR optimization, since DR functions tend to favor treating sequences that occur in a variety of contexts as separate words. In another set of experiments, Newsome and Jusczyk (1995) familiarized 9-month-olds with bisyllabic words having a weak-strong stress pattern, like *guitar*. Subjects listened significantly longer to passages containing the strong syllable of the familiar word (e.g., *tar*) than to passages containing a foil *when the test passage varied the context of the familiar word.* This effect disappeared when the familiar word was always followed by the same word (e.g., *is*) in the test passage. Subjects familiarized with a nonsense word *taris* did listen longer to the passages in which *guitar* was consistently followed by *is*. Once again, this is what would be expected if the subjects were using DR optimization.

Eventually, we plan to use these paradigms to test predictions that distinguish DR optimization from other possible hypotheses about how sub-sequence frequency could be used for segmentation. The general approach will be to test whether subjects segment in the same way as DR optimization algorithms. However, we believe that the clearest predictions will arise from a processing model in which DR functions are optimized using an incremental search procedure that segments one utterance at a time. Developing such a model is the first step on the path to obtaining behavioral evidence that bears on the DR optimization hypothesis.

Conclusions

These experiments suggest that DR optimization provides a useful source of information that infants could exploit to segment utterances. They also suggest that the combination of DR optimization and phonotactic constraints is more useful than either source of information alone. In addition, we have demonstrated a procedure for learning which clusters may occur at word boundaries in a given language. A procedure for learning which phonemes count as vowels for the purposes of the vowel constraint appears to be feasible (see Ellison, in press). For the first time, then, a system of interlocking strategies that is reasonably effective at discovering words in phonetic representations of natural speech, and that relies on no external, language-specific inputs appears to be on the horizon. In order to progress from reasonably effective to highly effective, it will be necessary to incorporate other information sources and to devise mechanisms for learning their language-specific manifestations.

References

Aslin, R. N., Woodward, J. C., 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* (pp. 117–134). Hillsdale, NJ: Erlbaum.

Bernstein-Ratner, N. (1987). The phonology of parent child speech. In K. Nelson, & A. van Kleeck (Eds.), *Children's language: Vol. 6.* Hillsdale, NJ: Erlbaum.

Bertoncini, J., Bijeljac-Babic, R., Juszcyk, P. W., Kennedy, L. J., & Mehler, J. (1988). An investigation of young infants' perceptual representations of speech sounds. *Journal of Experimental Psychology: General*, *117*, 21–33.

Brent, M. R., Gafos, A., & Cartwright, T. A. (1994). Phonotactics and the lexicon: Beyond bootstrapping. In E. Clark (Ed.), *Proceedings of the 1994 Stanford Child Language Research Forum.* Cambridge: Cambridge University Press.

Christophe, A., Dupoux, E., Bertoncini, J., & Mehler, J. (1994). Do infants perceive word boundaries? An empirical study of the bootstrapping of lexical acquisition. *Journal of the Acoustical Society of America*, *95*, 1570–1580.

Church, K. (1987). Phonological parsing and lexical retrieval. *Cognition*, *25*, 53–69.

Clements, G. N. (1990). The role of the sonority cycle in core syllabification. In J. Kingston, & M. E. Beckman (Eds.), *Papers in laboratory phonology I: Between the grammar and physics of speech.* Cambridge: Cambridge University Press.

Cutler, A. (1994). Segmentation problems, rhythmic solutions. *Lingua*, *92*, 81–104.

Cutler, A. (1996). Prosody and the word boundary problem. In J. L. Morgan, & K. Demuth (Eds.), *Signal to syntax: Bootstrapping from speech to grammar in early acquisition* (pp. 87–100). Hillsdale, NJ: Erlbaum.

Cutler, A., & Carter, D. M. (1987). The predominance of strong initial syllables in the English vocabulary. *Computer Speech and Language*, *2*, 133–142.

Cutler, A., Mehler, J., Norris, D., & Segui, J. (1994). The monolingual nature of speech segmentation by bilinguals. *Cognitive Psychology*, *24*, 381–410.

DiSciullo, A., & Williams, E. (1987). *On the definition of word.* No. 14 in Linguistic Inquiry Monographs. Cambridge, MA: MIT Press.

Ellison, T. M. (1991). Discovering planar segregations. In *AAAI Spring Symposium on Machine Learning of Natural Language and Ontology.* AAAI.

Ellison, T. M. (in press). *The machine learning of phonological structure.* Cambridge: Cambridge University Press.

Fisher, C., & Tokura, H. (1996). Prosody in speech to infants: Direct and indirect acoustic cues to syntactic structure. In J. L. Morgan, & K. Demuth (Eds.), *Signal to syntax: Bootstrapping from speech to grammar in early acquisition* (pp. 343–364). Hillsdale, NJ: Erlbaum.

Friederici, A. D., & Wessels, J. M. I. (1993). Phonotactic knowledge of word boundaries and its use in infant speech perception. *Perception and Psychophysics*, *54*, 287–295.

Harris, Z. S. (1954). Distributional structure. *Word*, *10*, 146–162.

Harris, Z. S. (1955). From phoneme to morpheme. *Language*, *31*, 190–222.

Huffman, D. A. (1952). A method for the construction of minimum-redundancy codes. *Proceedings of the Institute of Radio Engineers*, *40*, 1098–1101.

Jusczyk, P. W., & Aslin, R. N. (1995). Infants' detection of the sound patterns of words in fluent speech. *Cognitive Psychology*, *28*.

Jusczyk, P. W., Cutler, A., & Redanz, N. J. (1993). Infants' preference for the predominant stress patterns of English words. *Child Development*, *64*, 675–687.

Jusczyk, P. W., & Derrah, C. (1987). Representation of speech sounds by young infants. *Developmental Psychology*, *23*, 648–654.

Jusczyk, P. W., Friederici, A. D., Wessels, J., Svenkerud, V. Y., & Jusczyk, A. M. (1993). Infants' sensitivity to the sound patterns of native language words. *Journal of Memory and Language*, *32*, 402–420.

Jusczyk, P. W., Jusczyk, A. M., Kennedy, L. J., Schomberg, T., & Koenig, N. (1995). Young infants' retention of information about bisyllabic utterances. *Journal of Experimental Psychology: Human Perception and Performance*, *21*, 822–836.

Jusczyk, P. W., Kennedy, L. J., & Jusczyk, A. M. (1995). Young infants' retention of information about syllables. *Infant Behavior and Development*, *18*, 27–41.

Jusczyk, P. W., & Nelson, D. G. K. (1996). Syntactic units, prosody, and psychological reality during infancy. In J. L. Morgan, & K. Demuth (Eds.), *Signal to syntax: Bootstrapping from speech to grammar in early acquisition* (pp. 389–408). Hillsdale, NJ: Erlbaum.

MacWhinney, B., & Snow, C. (1985). The child language data exchange system. *Journal of Child Language*, *12*, 271–296.

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

Nelson, D. G. K., Jusczyk, P. W., Mandel, D. R., Myers, J., Turk, A., & Gerken, L. A. (1995). The headturn preference procedure for testing auditory perception. *Infant Behavior and Development*, *18*, 111–116.

Newsome, M. R., & Jusczyk, P. W. (1995). Do infants use stress as a cue in segmenting fluent speech? In D. MacLaughlin, & S.McEwen (Eds.), *Proceedings of the 19th Boston University Conference on Language Development*, Vol. 2 (pp. 415–426). Boston, MA: Cascadilla Press.

Norris, D. (1994). SHORTLIST: A connectionist model of continuous speech recognition. *Cognition*, *52*, 189–234.

Olivier, D. C. (1968). *Stochastic grammars and language acquisition mechanisms*. Unpublished doctoral thesis, Harvard University.

Pereira, F., & Schabes, Y. (1992). Inside-outside reestimation from partially bracketed corpora. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*. ACL.

Pinker, S. (1984). *Language learnability and language development.* Cambridge, MA: Harvard University Press.

Quinlan, J. R., & Rivest, R. L. (1989). Inferring decision trees using the minimum description length principle. *Information and Computation*, *80*, 227–248.

Rissanen, J. (1989). *Stochastic complexity in statistical inquiry.* Singapore: World Scientific Publishing.

Saffran, J. R., Newport, E. L., & Aslin, R. N. (in press). Word segmentation: The role of distributional cues. *Journal of Memory and Language.*

Shannon, C. E., & Weaver, W. (1949). *The mathematical theory of information.* Chicago: The University of Illinois Press.

Siskind, J. M. (in press). A computational study of cross-situation techniques for learning word-to-meaning mappings. *Cognition.*

Stolcke, A., & Omohundro, S. (1994). Inducing probabilistic grammars by Bayesian model merging. ICGI-94.

Wallace, C., & Boulton, D. M. (1968). An information measure for classification. *Computer Journal*, *11*, 185–195.

Appendix A

Following the Minimum Representation Length (MRL) method, the objective function used to evaluate segmentations in the simulations is based on a system for representing segmentations in terms of a lexicon and a derivation. Each segmentation corresponds to a lexicon-derivation pair, as described in the introduction. The value of the objective function for a given segmentation is (approximately) the number of characters in the representations of the corresponding lexicon-derivation pair. This appendix specifies the representational system, and then explains how the objective function approximates the number of characters in a representation.

The primary criterion for a representation is that it should be as neutral as possible—that is, it should not include any arbitrary, unjustified inductive biases. In the objective function illustrated in Figure 2 the use of different character sets to represent the words and the indices is a major source of arbitrary bias. Using a larger character set for the words than for the indices decreases the influence of the word lengths in the objective function, relative to the influence of the type count, token count, and the entropy term. Since we have no basis for choosing the relative importance of these terms, the most neutral course of action would be to use the same character set for both words and indices. Since the indices need not be numeric, they could be represented by strings of alphabetic characters, rather than digits. It is traditional, however, to represent everything in binary. Each alphabetic (or phonemic) character is translated into a binary code—for example, "a" might be 00000, "b" might be 00001, and so on. Binary is used for indices as well. Under this regime blank spaces, carriage returns, and tabular alignment cannot be used to separate different parts of the representation, as they are Figure 2. Often, however, a slight tempering of the neutrality criterion buys a great simplification of the representational system. We simplified the system by using "-" as a special separator character indicating the boundaries between words in the lexicon.

As in Figure 2, the representation of an input sample consists of a lexicon and a derivation. The representation of the lexicon is itself divided into a sequence of words followed by a sequence of indices, where the first index is assigned to the first word in the sequence of words, and so on. The sequence of words is a sequence of binary codes for letters, with a single "-" between each pair of adjacent words. For example, the word sequence of the lexicon

1 do	2 the	3 you
4 like	5 see	6 kitty

would be represented as:

[t][h][e]-[k][i][t][t][y]-[l][i][k][e]-[d][o]-[y][o][u]-[s][e][e]-

where a letter surrounded by square brackets stands for the binary code for that letter—for example, [d] might be 00011. The codes for all the letters have the same number of digits. The number of digits required to assign unique codes to P phonemes is $\lceil \log_2 P \rceil$, the next integer greater than or equal to $\log_2 P$. However, we follow standard practice in using $\log_2 P$ itself, and in using continuous approximations to integers throughout the length computation. The reasoning is that we are interested in measuring information content, and counting bits in an actual representation is just a convenient reification of information content. Thus, the number of bits per phoneme is $\log_2 P$. Multiplying this by the total number of phonemes in the lexicon—i.e., the sum of the lengths of the words in the lexicon—yields:

$$(\log_2 P)\left(\sum_{w \in \text{TYPES}(S)} \ell(w)\right),$$

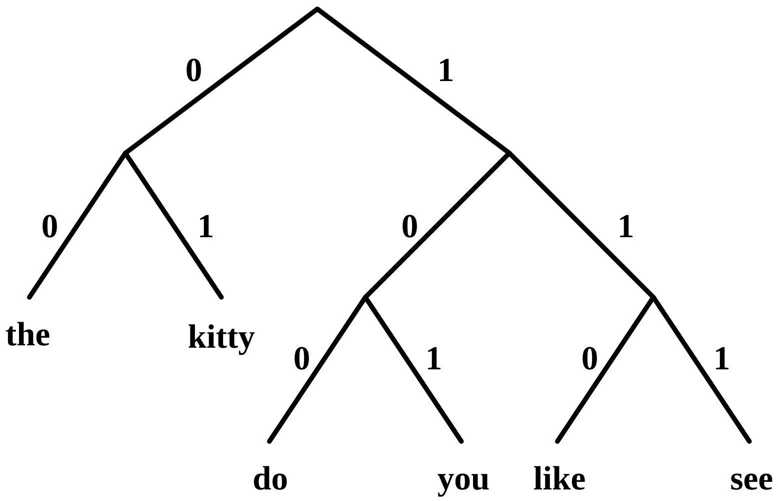

Figure 6: A code tree. The index for each item is the sequence of 1's and 0's on the path from the root to that item.

the last term in Formula 2. In addition, there is one "-" character for each word in the lexicon, so we must add |TYPES| to account for these characters.

The representation of the sequence of indices assigned to the words is the most complex part of the entire representation. Indices are assigned to words on the basis of their relative frequency, using a standard method for data compression called Huffman coding (Huffman, 1952). The procedure for constructing a Huffman code is not important here, since we can compute the necessary lengths without actually constructing the indices. As shown in Figure 6, indices constructed by Huffman coding can be represented by a tree, where the leaves represent the items to be encoded, each left branch is labeled 0 and each right branch is labeled 1, and the index for each item is the sequence of 1's and 0's on the path from the root to that item. For example, the code for *the* in Figure 6 is 00, the code for *kitty* is 01, the code for *do* is 100, and so on. When the code is constructed according to the Huffman procedure, every non-leaf node will be binary branching. In our representation, the leaves do not need to be labeled with words—the words in the word sequence are arranged such that the first word corresponds to the leftmost leaf, the second word corresponds to the next leaf to the right, and so on. However, we do need to represent the shape of the tree—which nodes are leaves and which are branching nodes. This can be done using exactly one bit per node, as described by Quinlan and Rivest (1989). The total number of nodes in a binary-branching tree with n leaves is $2n - 1$, and the number of leaves in our tree is always the same as the number of words in the lexicon. Thus, encoding the index tree requires

$$2|\textsc{types}| - 1$$

bits. Adding this to the length of the words yields a total lexicon length of

$$3|\textsc{types}(S)| - 1 + (\log_2 P)\left(\sum_{w \in \textsc{types}(S)} \ell(w)\right). \qquad (3)$$

Finally, we must consider the length of the derivation portion, which is a sequence of indices assigned according to the Huffman procedure. It is well known that the length of such a sequence, in bits, lies between

$$|\textsc{tokens}(S)| \times H(S)$$

and

$$|\textsc{tokens}(S)| \times (H(S) + 1),$$

where $H(S)$ is the entropy of the relative frequencies of the words:

$$H(S) = -\sum_{w \in \textsc{types}(S)} \frac{f(w)}{|\textsc{tokens}(S)|} \log_2 \frac{f(w)}{|\textsc{tokens}(S)|}$$

(See, e.g., Rissanen, 1986). In practice, however, it lies much closer to the lower bound. Using this as an approximation, and adding it to Formula 3, we obtain a formula for the length, in bits, of the entire representation:

$$3|\textsc{types}(S)| - 1 + (\log_2 P)\left(\sum_{w \in \textsc{types}(S)} \ell(w)\right) + |\textsc{tokens}(S)| \times H(S) \qquad (4)$$

The -1 term is constant for all segmentations, so it does not affect which segmentation yields the smallest value. Dropping the irrelevant -1 yields Formula 2.

Appendix B

The following lists of word-initial and word-final consonant clusters permitted in English were compiled by hand and provided to the program in Expermiments 2 and 3.

Word-initial clusters

ð dʒ ʃ θ θr θw ʍ b bl br by tʃ d dr dw f fl fr fy g gl gr gw gy h hy
k kl kr kw ky l m my n p pl pr py r s sk skr skw sky sl sm sn sp
spl spr spy st str sw t tr tw v vy w y z zy

Word-final clusters

> ð ðd ðz dʒ dʒd ŋ ŋθ ŋd ŋk ŋks ŋkt ŋz ʃ ʃt θ θs θt ʒ b bd bz tʃ tʃt
> d dz f fθfs ft fts g gd gz k ks kst kt l ld ldz lf lft lk lp lps lpt ls lt
> lts lv lvd lvz lz m md mp mps mpt mz n ndʒ ndʒd nθ nθs ntʃ ntʃt
> nd ndz ns nst nt nts nz p ps pst pt s sk sks skt sp sps spt st sts t
> ts v vd vz z zd

Many of the word final clusters occur only in inflected forms. For example, /ðd/ occurs in words like *bathed*.

Appendix C

Among the permissible word-initial clusters listed in Appendix B, the following *did not* occur at the beginning of any of the 10,883 utterances in our corpus of child-directed English (See Experiment 4):

> θw dw fr fy gl gw gy hy my py sk sky spl spr spy tw vy zy

Among the permissible word-final clusters listed in Appendix B, the following *did not* occur at the end of any of the utterances in our corpus:

> ð ðd dʒd ŋθ ŋd ŋks θs θt ʒ bd fθ fs fts gd ldz lft lps lpt lts lv lvd
> lvz md mpt nθs ntʃt nst pst sk sks skt sp sps spt sts vd

[11] This work has benefitted greatly from discussions with Anne Cutler, Adamantios Gafos, Lila Gleitman, Peter Jusczyk, Brian MacWhinney, Gary Marcus, Michael McCloskey, Jacques Mehler, Brenda Rapp, Jenny Saffran, Jeff Siskind, and Paul Smolensky. We also wish to thank the anonymous reviewers, whose comments helped improve the focus and clarity of this paper, and led to the inclusion of Experiment 4. Special thanks are due to Sandor Brent for his careful reading of several drafts and thoughtful comments on them.