Segmenting discrete data representing continuous speech input

by R. D. Faulk
F. Goertzel Gustavson

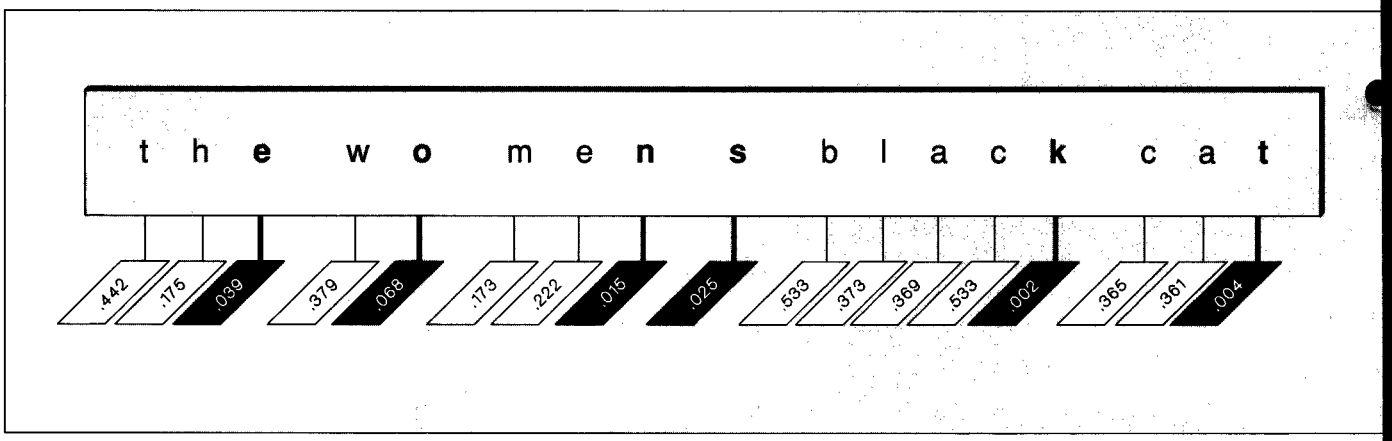
A probabilistic method for segmenting continuous speech into lexical units is described. The algorithm assumes initial conversion of the continuous speech signal to a discrete representation over some suitable alphabet. The problem of determining such alphabets is not considered. Experiments used keyed input in English, French, German, and Russian. We hypothesize that the low error rates obtained in the experiments can also be achieved with data representing actual speech. The paper discusses an area of linguistic science, and outlines a method for investigating it.

The achievement of natural-language communication between humans and computers, involving such capabilities as speech recognition and language translation, must be preceded by some probabilistic understanding of how linguistic skills are acquired in the presence of limited data and without benefit of *a priori* knowledge of structure. Historically, this problem first arose for linguists wishing to construct grammars of unanalyzed languages from audio tapes of native speech. The linguist first had to transcribe the recorded speech data into a continuous stream of phonetic characters. The next step was to divide, or segment, the resulting character strings into meaningful units such as words, roots, stems, and endings to further specify the grammar. It is this second step in the process of grammar construction that is addressed in this paper, partly because of its traditional interest for linguistics, but primarily because we believe it is crucial in the development of natural-language communication with computers.

Informal rules for segmentation were proposed in 1954 by the linguist Zellig Harris[1]—before the advent of modern computer science. At that time it was not yet practical to raise questions about random sampling from natural languages and how much data might be required to obtain reliable results from algorithmic computation. This paper considers these questions and offers some suggestions toward their solution.

Recent research has advanced the theory and application of grammatical formalism in computing science. However, current linguistic theory does not explain, nor even consider relevant, such phenomena as the intuitive ability of linguists to segment speech data and that of illiterate bilinguals to translate without explicit knowledge of grammar. The investigation of such phenomena is an appropriate and challenging objective of linguistic research. The fundamental problem of such research is to elucidate in probabilistic terms the properties of language data that make grammar construction possible. This knowledge can then serve as the basis for automatic construction and revision of local grammars based on the increasing amounts of ambient data available in modern information systems. Such a capability

Figure 1 Computed variety indices with threshold .086



would have numerous practical applications, including natural-language communication with computers. One of the authors has discussed a probabilistic method of language translation motivated by this approach.[2,3]

Specifically, this paper describes an algorithm for segmenting continuous speech utterances into lexical units. The algorithm assumes the initial conversion of the continuous speech signal to a discrete representation over some suitable alphabet, but is independent of the choice of alphabet. The determination of optimal alphabets for this purpose is a task for the type of research described above. Thus, the words alphabet and character are used here for convenience, realizing that in practice, input data elements might correspond to units of speech such as phones, allophones, phonemes, etc. The method requires initial training data, but no dictionary or other external information about the input, and has yielded single-digit error rates in experiments with keyed input in English, French, German, and Russian from which all delimiting information (i.e., space, punctuation, and capitalization) was removed. We hypothesize that comparable results can be obtained with data representing speech.

The method is essentially a generalization of the informal rules proposed by Harris and may be described as follows:

Given a string of input characters to segment, a *variety index* is computed for each character of the input string, using frequency data obtained from the training set. The resulting computed values are smaller at the last position of lexical segments than at other positions of the input string. Segmentation is accomplished by comparing the variety indices with a threshold.

For example, in one experiment the input data contained the sequence

"... thewomensblackcat ...",

and the computed variety indices were as shown in Figure 1. Applying a threshold of .086 to these values produced the segmentation indicated in the figure.

The principle involved is illustrated in the children's word-spelling game called "ghost." In ghost, a player announces the first letter of some word. For most languages, this may in effect be any letter of the appropriate alphabet. The next player adds a second letter—thus specifying the first two letters of some word—and so on, until one player cannot avoid giving a letter that completes the spelling of a word of four or more letters. That player earns a penalty and begins the next round. A variant of the game allows letters to be added to the left or to the right of the current string. A player may bluff, but scores a double penalty if challenged and unable to specify a word beginning with (or containing) the proposed sequence. The progressive reduction in the number of available alternatives (i.e., the *variety* of choices) during a round of ghost illustrates the process, described by Harris,[1] which is the conceptual basis for computing variety indices.

Variety indices are thus probabilistic estimators of the number of possible characters at successive positions of a natural-language input stream. The underlying assumption is that lexical segment boundaries are characterized by low values of this quantity.

Variety indices are probabilistic estimators of the number of possible characters at successive positions.

In the next sections of this paper, we discuss the data used to test the method and the actual calculation and interpretation of variety indices. Later sections describe the experimental data and methodological considerations, and some experiments and their results. Finally, some conclusions are offered.

Training data

The term *training data* as used here refers to a corpus of lexically complete speech utterances represented in terms of some alphabet. Utterances may be words, phrases, or sentences. It is not assumed, however, that the utterances conform to any syntactic rules of formation, but only that they are composed of recurring lexical elements. For computational purposes the utterances in a corpus (C) are concatenated to form a single string (S) of length L, containing $L(L+1)/2$ substrings. For example, C might consist of the concatenation of the product of a list of spoken words. Such a corpus might then be used to segment any utterance composed of words in the given vocabulary as in Experiments 4 and 5, described later.

The computation of variety indices is defined in terms of the frequencies in S of the substrings of S. For this reason, it is useful to construct a table of substring frequencies. The table need only contain entries for substrings of S with frequencies greater than 1 and for those of minimum length with frequency equal to 1. For example, if the substrings *ab* and *abc* occur only once in S, then only *ab* need be entered in the table. Every substring in S occurs with a frequency equal to or greater than 1, with shorter substrings generally having higher frequencies.

A table can be constructed by generating, sorting, and counting the substrings of S. In practice, it is not necessary to generate all of the substrings of S, but only those of length less than or equal to some specified maximum. Once generated, the information in the table is conducive to the use of rapid lookup strategies.

Calculation and interpretation of variety indices

The following discussion assumes the existence of a substring frequency table T derived from some given corpus of utterances C. A new utterance or character string (I) is to be segmented. I is not necessarily contained in C but presumably is drawn from the same language. The term *bi-string* refers to any ordered pair $<s_1,s_2>$ of substrings of I such that both s_1 and s_2 either begin or end at the same position in I and differ in length by exactly one character, where s_1 denotes the shorter member of the pair. A bi-string is left-aligned or right-aligned, depending on whether s_1 and s_2 begin or end in the same position of I. We refer to left-aligned and right-aligned bi-strings simply as left and right bi-strings, respectively.

A bi-string ratio r is defined as a function of T and an arbitrary bi-string of I:

$$r = f(T, <s_1,s_2>) = f_2/f_1, \text{ (string } s_1 \text{ found in T)},$$

or

$$r = 1 \qquad (s_1 \text{ not found in T}),$$

where f_1 and f_2 denote the frequencies in S of s_1 and s_2.

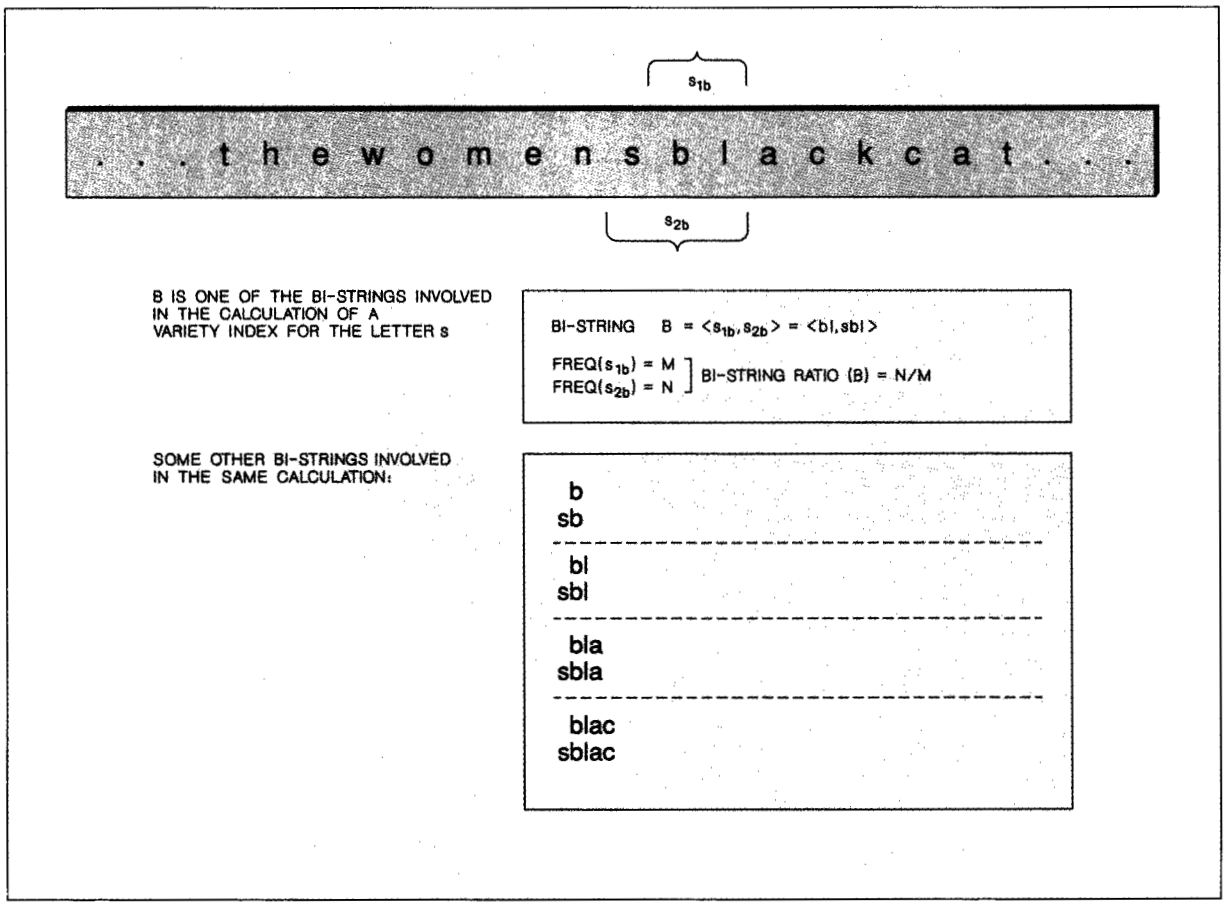
Values of r range over the unit interval. Thus

$$0 < r \leq 1,$$

since the frequency of s_2, the longer member of $<s_1,s_2>$, can never exceed that of s_1, the shorter member. The value of r determined by $<s_1,s_2>$ is referred to as a bi-string ratio, and is said to be proper if and only if $f_1 > 1$. See Figure 2 for an illustration of the above terminology.

Let n=1, ... , N, where N denotes the length of I, and let I be treated as circular, i.e., the last character of I

Figure 2 Bi-strings and bi-string ratios



is followed by the first. Values of n correspond to characters in I. Then for each value of n, n determines sets of left bi-strings and sets of right bi-strings, which in turn determine sets of proper left and right bi-string ratios, respectively. (See Figure 2.)

A variety index v associated with character position n of I is defined as the product of the mean of the proper left bi-string ratios and the mean of the proper right bi-string ratios determined by n and the selected sets of bi-strings. Trial runs were made using different sets of left and right bi-strings as the basis for computation. In some cases, comparable good results were obtained. This paper presents the best results obtained using a fixed set of bi-strings.

We offer the number e as an interpretation of what is measured by the calculated variety indices. If K is the size of the alphabet representing the utterances in C, then e is defined as

$$e = \text{int}[(v)(K-1)] + 1.$$

Since $0 < v \leq 1$,

$$1 \leq e \leq K.$$

In other words, e is the estimated number of characters that might have occurred with variety index v at position n of the input string. As stated in the introduction, the method assumes that low values of this quantity characterize segment boundaries.

Because calculation of the values of v for the N positions of I can each proceed independently, the proposed method has strong potential for implementation in a parallel processing environment.

Experimental data and methodological considerations

Data for the experiments were obtained by random sampling from grammatical product languages. A grammatical product language (GPL) is a set of sentences defined by one or more arrays of lists such that the sequence obtained by randomly selecting one element from each list in a specified array is always a sentence of the language. Such an array of lists is called a grammatical product definition (GPD), and the set of sentences it defines, a grammatical product (GP). To exhibit any given sentence as a GP member, one need only include one or more of the sentence elements in a list of possible alternatives. The resulting sequence of lists is a GPD. In general, natural languages can be viewed and exhibited in terms of GPDs as unions of disjoint grammatical products.

A GPL may be any desired subset of a natural language, or an arbitrary language of interest for experimental purposes. As a particular case, a GP may be defined as a product of a single list, where the n-place product of a given list is understood to be the set of ordered n-tuples that can be formed from elements of the list. Figure 3 defines the GPLs from which data for the present five experiments were obtained by random sampling. For the first three experiments, sentences were generated by choosing one element randomly from each list in a GPD. For the last two, the language was considered to be any sequence of elements in the given vocabulary. In terms of grammatical products, such languages are unions of disjoint list products. For the purposes of the present experiments, the elements in the sequences obtained by the above method were concatenated to form continuous character strings.

The concept of grammatical products effectively implements the operation of random sampling from languages, as distinguished from random sampling from discursive data such as newspaper text, literary text, or speech. This distinction is fundamental for the purposes of the present type of research. The precise formulation and testing of linguistic hypotheses and the facilitating of the design and testing of programs intended for large-scale linguistic computations are made possible using GPL data. GPL data are easier to obtain than statistically equivalent discursive data and allow exact formulation of desired program behavior and performance evaluation. Examples of hypotheses are (1) the one illustrated in this paper where lexical boundaries are characterized by low values of the variety index, and (2) in the case of language translation, that similar sentences have similar translations.[2,3]

Next discussed is the concept of *sampling units* relative to random samples of sentences drawn from a grammatical product. If r is the number of sentences in a random sample drawn from the GP, then U, the number of sampling units in the sample, is defined as follows:

$$U = r/m$$

where m = the product of the two longest list sizes in GPD.

As defined above, m is the size of the minimal subset or subsets in GP such that every possible pair of list elements that can occur together in a sentence of GP occurs in at least one sentence of the subset. (If GP is defined as the product of a single list, then m is simply the square of the list size.)

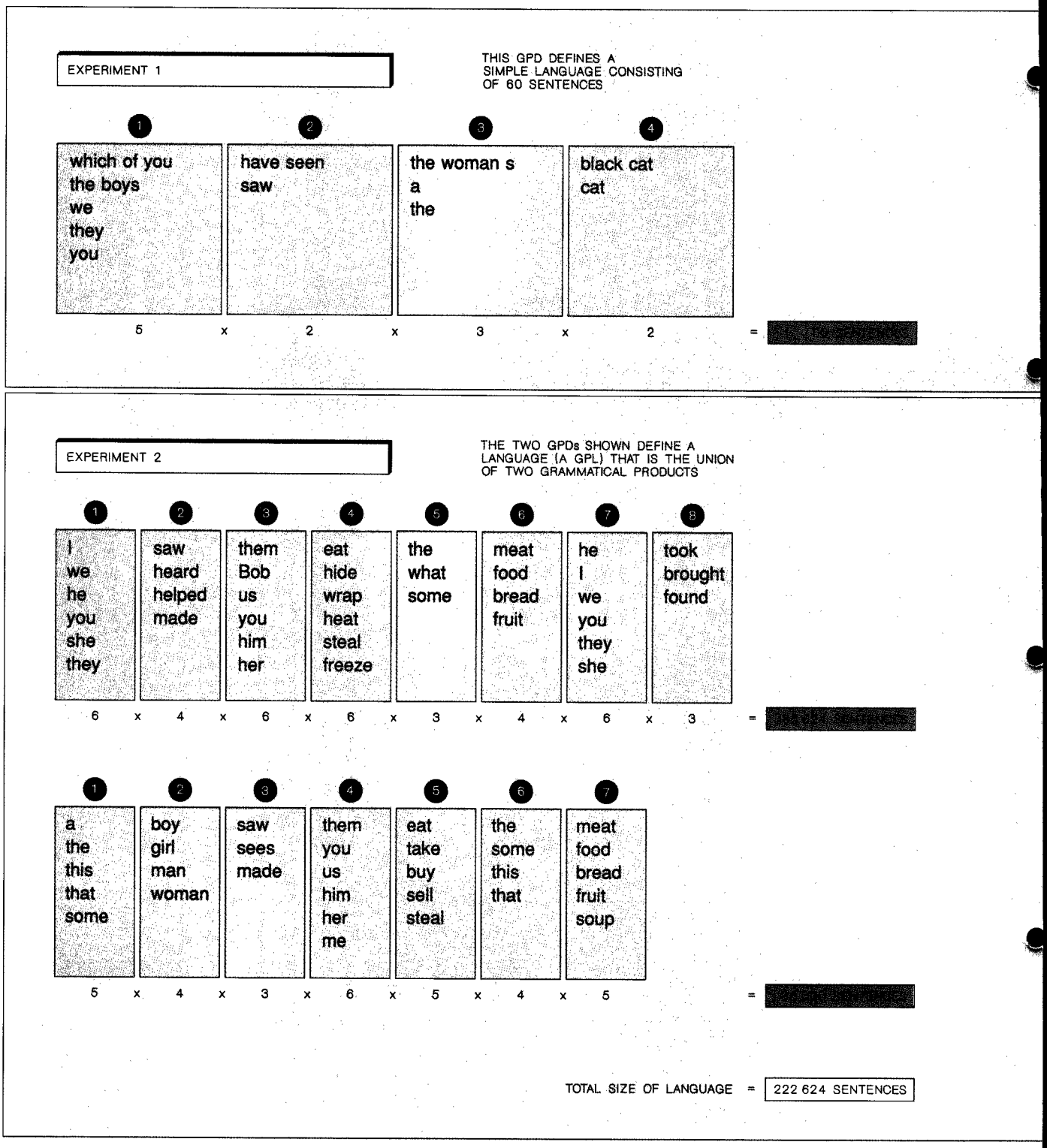
The size of a grammatical product language is the sum of the sizes of the grammatical products included in it. The size (N) of a single grammatical product is the product of the list sizes in the associ-

Table 1 Summary of experiment results

Experiment	Sample Description	Language Size	Size of Training Set	Total Language of Training Set (%)	Sampling Units	Number of Variety Indices Completed	Error rates (%)		
							Omission	Insertion	Total
1	English sentences	60	23	38.3	1.530	517	11.22	3.68	14.90
2	English sentences	222 624	101	.045	1.530	2797	6.61	6.33	12.94
3	Basic English*	4×10^7	462	2.5×10^{-4}	1.100	2837	4.76	4.51	9.27
4	English vocabulary (20 words)	1.3×10^9	801	5.9×10^{-5}	2.003	5331	.15	.09	.24
5	French vocabulary (45 words)	3.8×10^{11}	3873	1.01×10^{-6}	..074	1170	4.02	4.02	8.04

* See Reference 4.

Figure 3 Grammatical product definitions



EXPERIMENT 3

THE VERTICAL WORD LISTS IN THIS GPD WERE ORIGINALLY ARRANGED IN CONCENTRIC CIRCLES ON C. K. OGDEN'S "WORD WHEEL"

1	2	3	4	5	6	7
will	put	good	rule	to	boy	slowly
would	take	brown	book	in	cart	wisely
might	give	first	dog	after	house	and
may	keep	this	land	down	edge	but
	let	simple	cup	against	end	if
	have	clean	bag	among	wall	or
	get	warm	box	across	cat	though
	make	small	copy	by	horse	when
	come	new	key	at	ice	here
	go	cheap	pot	before	fire	now
		strange	knife	with	land	
		some	pin	between	roof	
			hat	off	paper	
			pen	under	seat	
			thing	through	top	
			food	up	glass	
			sand	on	list	
			gold	from	table	
			wax	over	farm	
			oil	about	floor	
					boat	

4 x 10 x 12 x 20 x 20 x 21 x 10 = 40 320 000 SENTENCES

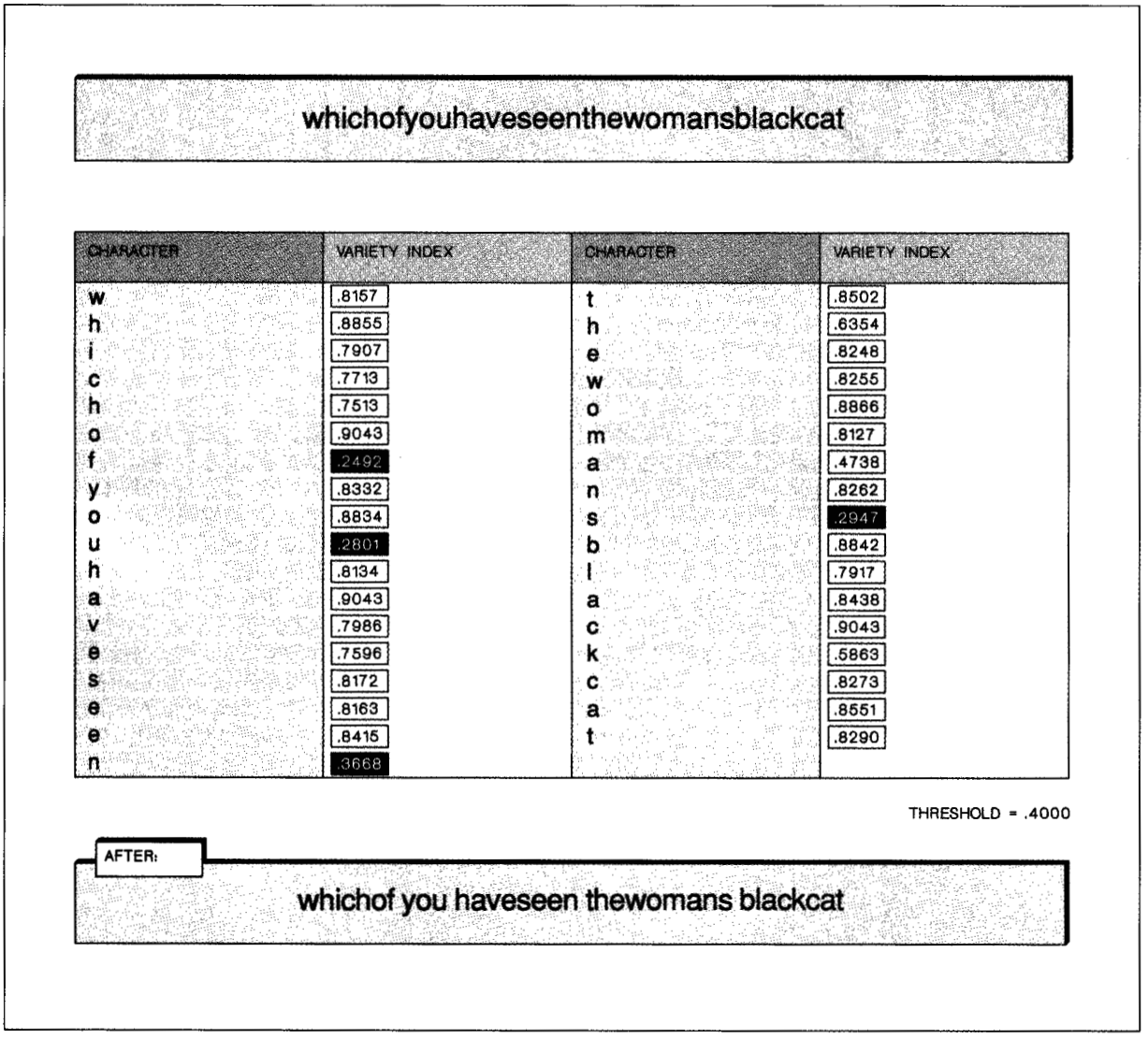
EXPERIMENT 4

a	s
black	saw
boy	see
cat	seen
had	the
has	they
have	which
man	woman
men	women
of	you

EXPERIMENT 5

a	est	l'	qu'
avez	femme	nègre	voit
avont	femmes	nègres	voient
ce	garçon	négresse	vous
c'	garçons	négresses	voyez
chat	homme	noir	vu
de	hommes	noirs	vue
du	il	ont	vus
d'	ils	que	vues
elle	la	quell	un
elles	le	qui	une
	les		

Figure 4 A segmentation example

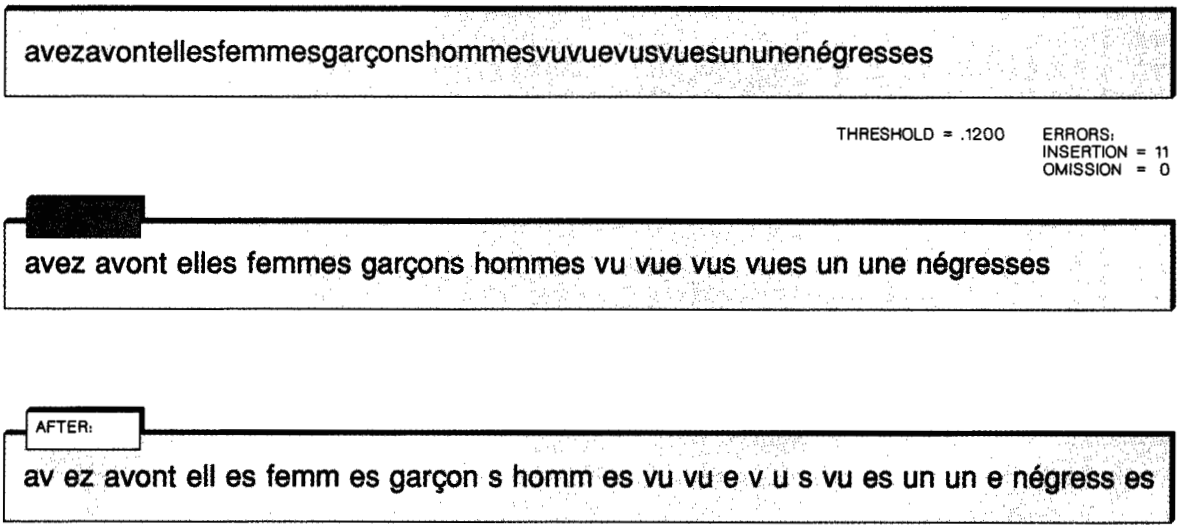


ated GPD. (If GP is defined as the product of a simple list of L elements, then N is simply L^k where k is the length of sentences of GP, and the size of the entire language is the sum of the powers of L from 1 to an arbitrarily specified maximum sentence length.) Table 1 documents the language sizes and the number of sampling units involved in the present experiments. Note that it is the number of sampling units, rather than the language size, that correlates positively with the error rates. The calculation of error rates is explained below.

Experiments

The experiments were performed using an IBM XT personal computer. The programs were written in BASIC and are available upon request, along with a demonstration version of an experiment in the form of an educational exhibit. The segmentation program obtains input strings from the training data or, optionally, the strings may be entered manually by the user. Spaces supplied by the user do not enter into the calculation of variety indices, but if the

Figure 5 Another segmentation example



input string does contain blanks, the program computes a threshold value that minimizes the total number of insertion and omission errors. This threshold is used to measure the success rate. In either case, the experimenter is allowed to enter and see immediately the effect of different threshold values on the output string. Figures 1, 4, and 5 show some typical results of the segmentation process. In each figure, the input string is shown above one or more of the computed variety indices, threshold, and error counts. The variety indices are compared to the threshold and when an index is less than the threshold, a blank is inserted after the corresponding character to designate a segment boundary. *Before* contains the input string as entered by the experimenter, with (optional) spaces indicating the expected locations of segment boundaries. (The error counts are meaningful only when this option is exercised.) The segmented output string appears as *After*.

Figure 3 presents the GPDs for the GPLs in the experiments. Experiment 1 defines a GPL consisting of one 60-sentence GP, presented here primarily for the sake of illustration, though it is worth noting that a considerable variety of structure can be captured even in a small GPD. Experiment 2 defines the GPL of an experiment first performed (but not reported) with comparable experiments in German and Russian in 1976. This GPL is the union of two grammatical products containing 186 624 and 36 000 sentences for a total of 222 624, of which 492 contain the lexically ambiguous sequence "themeatthemeat". Experiment 3 is taken verbatim from the grammatical product definition referred to as a "word wheel" in Reference 4. Experiments 4 and 5 illustrate GPLs defined as unions of list products.

Results

Table 1 summarizes the results of the experiments. We observe that variety indices computed by the proposed method can be as much as an order of magnitude smaller at segment boundaries than at other input positions (see Figure 1). The results also demonstrate the ability of the proposed method to discover possibly unsuspected lexical structure (see Figure 5). The results of Experiments 4 and 5 confirm the expectation of a strong correlation between observed error rates and the number of experimental

sampling units, as defined in the section on experimental data.

Conclusions

While the volume of computations required by the present algorithm is large, it has strong potential for implementation in parallel processing environments because the calculation of each variety index is independent of the others. It is also possible to store the resulting lexical segments and their frequencies for use in the more expeditious processing of subsequent input. This paper identifies an unexplored area of linguistic science, outlines a method for its investigation, and demonstrates the feasibility of the proposed method of speech segmentation as a practical solution.

Acknowledgments

We are grateful to Winfred P. Lehmann of the University of Texas for pointing out the relevance of Harris' work on morpheme recognition, and for his comments on our work over the years; to Kenneth Powell of IBM Corporate Marketing Education (now retired), without whose vision, interest, and encouragement this paper would not have been written; and to Jay Friedman of the *IBM Systems Journal* staff, for his invaluable assistance.

Cited references

1. Z. S. Harris, "From Phoneme to Morpheme," *Language* **31,** No. 2, 190–222 (1954).
2. R. Faulk, *The Phenomenon of Interlingual Correspondence: A Quantitative Formulation of the Translation Problem for Natural Languages*, Research Report RC-1501, IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598 (1965).
3. R. Faulk, "An Inductive Approach to Language Translation," *Communications of the ACM* **7,** No. 11, 647–653 (1964).
4. C. K. Ogden, *Basic English*, Harcourt, Brace and Company, New York, NY (1934), p. 305.

Ramon D. Faulk *16 North Broadway, Apt. 3H, White Plains, New York 10601.* Mr. Faulk obtained his B.A. degree in philosophy and French in 1959 at the University of Texas in Austin. While doing graduate work in linguistics he began developing a probabilistic approach to machine translation. In 1961 he joined the IBM Thomas J. Watson Research Center in Yorktown, New York, to continue that effort, the results of which appeared in 1964 and 1965. In 1969 he joined the IBM Field Engineering Headquarters in White Plains, New York, as a staff programmer, while privately continuing his research in computer linguistics. A preliminary series of experiments in 1976 yielded results leading to those reported here. Mr. Faulk was a recent member of the Translation Project in the Speech Recognition Department of the Research Center in Yorktown.

Fran Goertzel Gustavson *Information Systems Department, Pace University, Bedford Road, Pleasantville, New York 10570.* Dr. Gustavson obtained her A.B. degree in 1963 in mathematics and French at Vassar College, an M.S. in mathematics from New York University in 1967, and a Ph.D. in computer science from Polytechnic Institute of New York in 1979. From 1964 to 1969 she was a research staff member at the IBM Thomas J. Watson Research Center in Yorktown, New York. She joined Pace University in 1971 and is currently Professor of Information Systems and Chairperson of the Information Systems Department.

Reprint Order No. G321-5400.