TWO SIMPLE PREDICTION ALGORITHMS TO FACILITATE TEXT PRODUCTION

Lois Boggess
P.O. Drawer CS
Mississippi State University
Mississippi State, MS 39762

ABSTRACT

Several simple prediction schemes are presented for systems intended to facilitate text production for handicapped individuals. The schemes are based on single-subject language models, where the system is self-adapting to the past language use of the subject. Sentence position, the immediately preceding one or two words, and initial letters of the desired word are cues which may be used by the systems.

INTRODUCTION

For some years we have been investigating the use of a sizeable sample of a particular individual's language habits in predicting future language use for that individual. The research has taken two directions.

One of these, the HWYE (Hear What You Expect) system, builds a large language model of the past language history of the individual, with special emphasis on the most frequent words of that person, and the result is used in speech recognition. In studying the language model developed by the HWYE system, several simple predictive schemes were noted which are capable of anticipating, during the generation of a sentence, a small set of words from which the next desired word can be selected. The two schemes described here are used for text generation (not speech recognition) in a format that could be of use to a physically handicapped person; hence the schemes have no right context available. One of the schemes does use left context, and the other uses only sentence position as "context". Both are implemented on IBM-PC systems with minimal memory requirements.

MOTIVATION

One hundred English words account for 47 per cent of the Brown corpus (about one million words of American English text taken from a wide range of sources). It seems reasonable to suppose that a single individual might in fact require fewer words to account for a large proportion of generated text. From our work on the HWYE system it was known that 75 words accounted for half of all the text of Vanity Fair, a 300,000 word Victorian English novel by Thackeray (which incorporated a fairly involved syntax, much embedded quotation, and passages in dialect and in French) [English and Boggess, 1986]. We further found that 50 words accounted for half of all the verbiage in a 20,000 word set of sentences provided by an individual who collaborated with us. This latter corpus, called the Sherri data, is a set of texts provided by a speech-handicapped individual who uses a typewriter to communicate, even with her family; it is conversational in nature, as can be seen in Figure 1. Most of the work reported in this paper gives special attention to the set of words required to account for half of all the verbiage of a given individual. We refer to this set as the set of high-frequency words.

You said something about a magazine that <name1> had
about computers that I might like to borrow.
I would some time.
I think we have to pick up the children while <name2>
is in the hospital.
I want to visit her in the hospital.
But you have to lift me up to the window for me to see
the baby.
Well, it's May first now. Help!
I thought it would not be so busy but it looks like it
might be now.

Figure 1. Sample set of contiguous sentences in Sherri data

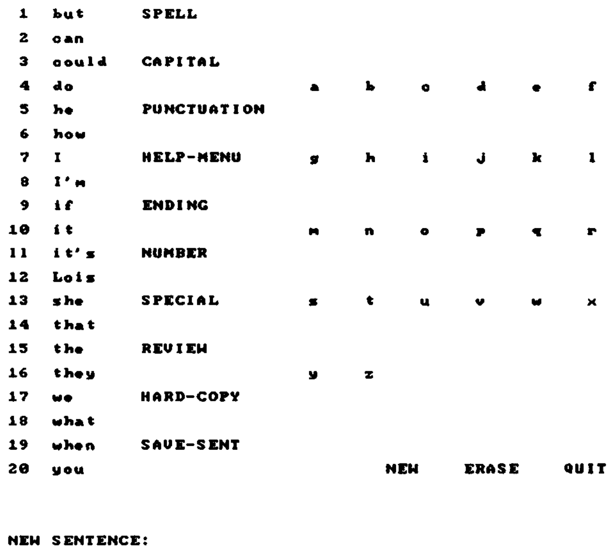
It seems reasonable to suppose that for conversational English, approximately 50 words may account for half of the verbiage of most English users. From the standpoint of human factors, an argument could be made that one should simply put the 50 words up on the screen with the alphabet and thus be assured that half of all the words desired by the user were instantly available, in known locations that the user would quickly become accustomed to. Constantly changing menus introduce an element of user fatigue [Gibler and Childress, 1982]. That argument may especially make sense as larger screens with more lines per screen and more characters per line become more common.

If we limit ourselves to the top 20 most frequent words as a constant menu, only about 30 per cent of the user's verbiage is accounted for. However, it was observed, while working with the HWYE system, that if one looked at the top 20 words for any given sentence position, one did not see the same set of words occurring. Clearly the high frequency words (the set that comprise half of word use) are mildly sensitive to ″context″ even when ″context″ is so broadly defined as sentence position. Different subsets of the 50 member set of high frequency words appear in the set of 20 most frequent words for a given sentence position. Moreover, after processing approximately 2000 sentences from the user, it was still the case that some of the top 20 words for a given position were not members of the high frequency set at all. For example, the word ″they″, a member of the menu for the first sentence position (see Figure 2) and hence one of the 20 most frequent words to start a sentence, is not a member of the global high frequency set.

A preliminary analysis by English suggested that, whereas a constant ″prediction″ of the top 20 most frequent words would yield a success rate of 30 per cent, predicting the top 20 most frequent words per position in sentence would yield a success rate of 40 per cent.

″CONTEXT″ AS SENTENCE POSITION

The simplest scheme, which has been built as a prototype on an IBM PC with two floppy disk drives, presents the user with the top 20 most frequent words that the user has employed at whatever position in a sentence is current. For example, Figure 2 shows the screen presented to the user at the beginning of production of a sentence. On the left is a list of the 20 words which that particular user is known to have used most often to begin sentences. On the right is the alphabet, which is normally available to the user; and in other places on the screen are special functions. (Selection of words, letters



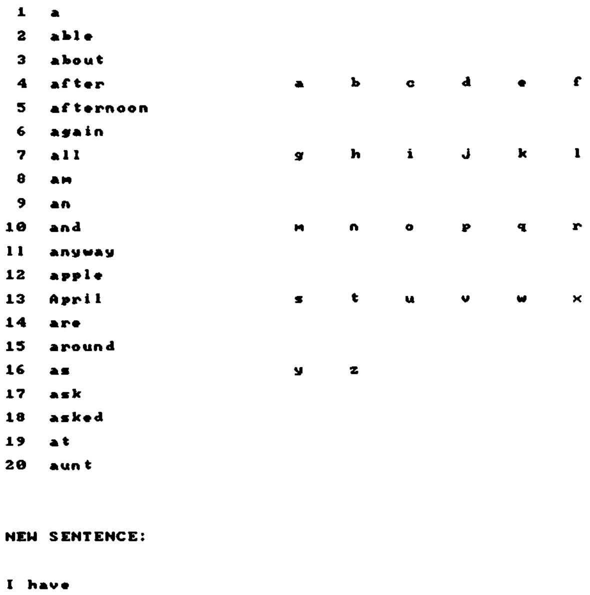
```
1   but     SPELL
2   oan
3   oould   CAPITAL
4   do                      a    b    o    d    e    f
5   he      PUNCTUATION
6   how
7   I       HELP-MENU       g    h    i    j    k    l
8   I'm
9   if      ENDING
10  it                      m    n    o    p    q    r
11  it's    NUMBER
12  Lois
13  she     SPECIAL         s    t    u    v    w    x
14  that
15  the     REVIEW
16  they                    y    z
17  we      HARD-COPY
18  what
19  when    SAVE-SENT
20  you                               NEW    ERASE    QUIT

NEW SENTENCE:
```

Figure 2. Initial Screen

and functions is made by mouse, though the actual selection mechanism is separated from the bulk of the code so that replacement with another selection mechanism should be relatively easy to implement.) The sentence is built at the bottom of the screen. If the user selects a word from the menu at the left, it is placed in first position in the sentence, and a second menu, consisting of the 20 most frequent words that the user has used in second place in a sentence, appears in the left portion of the screen. After a second word has been produced and added to the sentence, a third menu, consisting of the 20 most frequent words for that user in third place in a sentence, is offered, and so on.

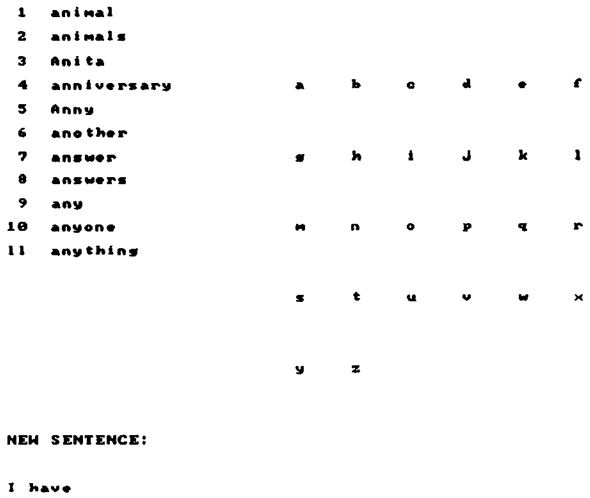
At any time the user may reject the lefthand menu by selecting a letter of the alphabet. Figure 3 shows the screen after the user has produced two words of a sentence and has begun to spell a third word by selecting the letter "a". At this point, the top 20 most frequently used words beginning with "a" have been offered at the left. If the desired word is not in the list, the user continues by selecting the second letter of the desired word (in this case, "n"). The left-hand menu becomes the 20 most frequently used words beginning with the pair of letters given so far. As is shown in Figure 4, there are times when fewer than 20 words of a given two-letter starting combination have been encountered from the user's past history, in which case this algorithm offers a shortened list.

In the case illustrated, the desired word was on the list. If it were not, the user would have had to spell out the entire word, and it would have been entered into the sentence. In either case, the system subsequently returns to offering the menu of most-frequently-used words for the fourth position, and continues in similar fashion to the end of the sentence.



```
1   a
2   able
3   about
4   after              a    b    c    d    e    f
5   afternoon
6   again
7   all                g    h    i    j    k    l
8   am
9   an
10  and                m    n    o    p    q    r
11  anyway
12  apple
13  April              s    t    u    v    w    x
14  are
15  around
16  as                 y    z
17  ask
18  asked
19  at
20  aunt

NEW SENTENCE:

I have
```

Figure 3: User has selected "a"



```
1   animal
2   animals
3   Anita
4   anniversary        a    b    c    d    e    f
5   Anny
6   another
7   answer             g    h    i    j    k    l
8   answers
9   any
10  anyone             m    n    o    p    q    r
11  anything

                       s    t    u    v    w    x


                       y    z

NEW SENTENCE:

I have
```

Figure 4: User has selected "a-n"

The system keeps up with how often a word has been used and with how many times it has occurred in each position in a sentence, so that from time to time a word is promoted to one of the top 20 alphabetic or top 20 position-related sets of words. For details on the file organization scheme that allows this to be done in real time, see Wei [1987]. Details on the mouse-based implementation for IBM PC's are available in Chow [1986].

A SECOND ALGORITHM

An alternative predictive algorithm has been implemented which replaces the sentence-position-based first menu. It pays special attention to the 50 most frequently used words in the individual's vocabulary (the high-frequency words) and to the words most likely to follow them. By virtue of their frequency, these are precisely the words about which the most is known, with the greatest confidence, after a relatively small body of input such as a few thousand sentences.

For each of the 50 high-frequency words, a list is kept of the top 20 most frequent words to follow that word. Let us call these the first order followers. For each of the first order followers, there is a list of second-order followers: words known to have followed the two word sequence consisting of the high-frequency word and its first order follower.

For example, the word "I" is a high-frequency word. The first order followers for "I" include the word "wonld". The second-order followers for "I would" include the word "like". (See Figure 5.) The second-order followers for "I would" also include many one-time-only followers, as well, so the system maintains a threshold for the number of occurrances below which a word is not included in the list of second-order followers. The reasoning is that a word's having occurred only once in an environment that by definition occurs frequently may be taken as counter-evidence that the word should be predicted.

Rather than predict a word with low reliability, one of two alternatives are taken. If the first-order follower is itself a high-frequency word, then low-reliability second-order followers may be replaced with the first-order follower's own followers. ("Would" is a first-order follower of "I" and is itself a high-frequency word. There are relatively few reliable second-order followers to "would" in the left context of "I", so the list is augmented with first-order followers of "would" to round out a list of 20 words.) The other alternative, taken when the first-order follower is not a high-frequency word, is to fill out any short list of second-order words with the high-frequency words themselves.

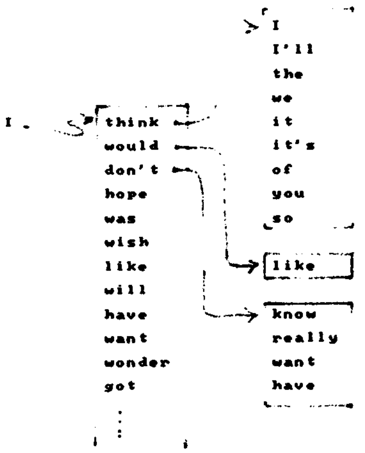


Figure 5. First- and second- followers for "I"

This algorithm is related to, but takes less memory and is less powerful than a full-blown second order Markov model. Each state in a second-order (trigram) Markov model is uniquely determined by the previous two inputs. For an input vocabulary of 2000 words, the number of mathematically possible states in a trigram Markov model is 4,000,000, with more than 8 billion arcs interconnecting the states. Fortunately, in the real world most of these mathematically possible states and arcs do not actually occur, but a trigram model for the real world possibilities is still quite large.

We experimented with abstracting the input vocabulary by restricting it to the 50 highest-frequency words plus the pseudo-input OTHER onto which all other words were mapped. When we did so, the number of states and arcs in the various order Markov models was still fairly large for the real world data [English and Boggess, 1986]. As Figure 6 shows, for example, the rate of growth for a fourth-order abstract Markov model (just the 50 highest-frequency words plus OTHER plus end-of-sentence) is in the neighborhood of 250 new states and 450 new arcs per 1000

	Sherri data		Thackeray data	
words	new states	new arcs	new states	new arcs
1000	527	677	639	830
2000	469	620	624	818
3000	471	636	476	705
4000	399	562	467	716
5000	397	566	463	714
6000	391	579	437	668
7000	337	507	389	642
8000	311	476	370	628
9000	323	500	361	612
10000	285	486	384	629
11000	329	518	348	601
12000	278	448	331	588
13000	276	445	310	543
14000	240	408	291	530
15000	248	425	287	529
16000	244	420	290	533
17000	243	414	269	497
18000	259	446	234	468

Figure 6. Growth of abstracted fourth-order Markov models

new words of text, after 17000 words of input. This was true for both the Sherri data (conversational English) and the more formal Thackeray data. Moreover, the fourth-order Markov model for the abstracted Thackeray data continued to grow. After 100,000 words of input, with a model of approximately 22,000 states and approximately 45,000 arcs, the rate of growth was still more than 1,000 states and 3,000 arcs per 10,000 words of input.

For this particular implementation, however, neither a full-blown Markov model using total vocabulary nor an abstract model using the 50-word vocabulary seemed appropriate. On the one hand, models of the entire vocabulary confirmed that many multiple word sequences did occur regularly. Nevertheless, for any but the simplest order Markov models (orders zero and one), the vast bulk of the networks were taken by word combinations that occurred only once. On the other hand, restricting the predictive mechanism to only the high-frequency words obviously left out some of the regularly occurring word combinations. Our first- and second-follower algorithm described on the previous pages allows lower frequency words to be predicted when they occur regularly in combination with high-frequency words.

PREDICTIVE CAPABILITIES

The data used to test the predictive capabilities of the system were typescripts provided by the user, who was utilizing a manual typewriter; it follows that the results were not biased by the user's favoring sentence patterns that the system itself provided. The system had been given 1750 prior sentences produced by the user and the data collected were for the performance of the system over the next 97 sentences. The 1750 sentences were 14,669 words in length with a vocabulary of 1512 words. Twelve sentences of the 1750 were a single word in length (e.g. "yeah", "no" and "gesundheit") and 51 were of length 20 or greater. Average length of sentence for the initial body was 8.4 words per sentence. The first 200 sentences included transcriptions of oral sentences, which were much shorter on average, since the user is speech handicapped. If the first 200 sentences are omitted, the average sentence length is 8.6 for the following 1550 sentences.

Of the next 97 sentences generated, the shortest sentence was "Thanks again." The longest was "You said something about a magazine that Jenni had about computers that I might like to borrow." The 97 sentences consisted of 884 words (six of which were numbers in digital form), for an average length of 9.1 words per sentence.

Of the 884 words, 350 were presented on the first menu, 373 were presented on the second menu (after one letter had been spelled), 109 were presented on the third menu (after two letters had been spelled), 2 were presented on the fourth menu (after three letters had been spelled, 43 were spelled out in their entirety, and 7 were numbers in digital form, produced using the number screen of the system.

From the above, it is obvious that the device of predicting the 20 most frequent words by sentence position is successful 39.6 per cent of the time; 42.2 per cent of the time, the desired word is among the 20 most frequent words of a given initial letter but not in the 20 most frequent words by position; combining these two facts, we see that 81.8 per cent of the time, this simple prediction scheme presents the desired word on a first or second selection. The desired word is offered in the first, second, or third menu 94.1 per cent of the time, and most of the rest of the time (5.7 per cent of total), the desired word is unknown to the system and is "spelled out", where "spelling" includes producing numbers.

Although the fourth menu, consisting of words with a three-letter initial sequence, presently has a low success rate, it is precisely this category that we expect to see improve as more of the user's words become known to the system through spelling. That is, as time passes, we expect the user to have to resort to complete spelling less and less because the known vocabulary will include more and more of the actual vocabulary of the user. Many of the new words will be low frequency words that we would expect to find on the menu for three-letter combinations after they are known.

The second algorithm, using first- and second-followers of the high-frequency words, was run on 100 sentences, the shortest of which was "Help!" (94 of the 97 test sentences for the first algorithm were represented in the test set for the second.) There were 895 words in the sample, of which 448 were presented on the first menu, 280 were presented on the second (after one letter had been spelled out, 83 on the third (after two letters were spelled), 1 on the fourth, and 83 were spelled out in their entirety (this category included numbers).

Running the second test gave us a very quick appreciation for the value of adding new words to the system as they are encountered, since this implementation of the second algorithm did not. One especially striking example was a word beginning with "w-o" which had never been used before, but which occurred five times in the 100 test sentences and had to be spelled out each time. This was especially irritating since the "w-o" menu (third menu) had fewer than 20 entries and would have accommodated the new word. A comparison of the two columns of Figure 7 suggests that for the text held in common by the two tests, approximately 30 words had to be spelled out by the second algorithm, which were selected by menu in the first algorithm because it added new words to its data sets as they were encountered.

PROPOSED EXTENSIONS

We have several plans for the future, most of them involving the second algorithm. Our first task is to increase the number of sentences in the Sherri data to 3000 and determine how much (if at all) an enlarged base of experience improves the ability of the algorithm to predict

Sentence position algorithm
number sentences: 97
number of words: 884

	words	%	total
first menu:	350	39.6%	39.6%
second menu:	373	42.2%	81.8%
third menu:	109	12.3%	94.1%
fourth menu:	2	0.2%	94.3%
spelled:	43	4.8%	99.2%
numbers:	7	0.8%	100%

frequent word/left context algorithm
number sentences: 100
number of words: 895

	words	%	total
first menu:	448	50%	50%
second menu:	280	31.3%	81.3%
third menu:	83	9.3%	90.6%
fourth menu:	1	0.1%	90.7%
"spelled":	83	9.3%	100%

Figure 7. Comparison of the predictive capabilities.

the desired word on the first try.

In its present form, the system is reliable in its predictions after several hundred sentences by the user have been processed. We intend to take something like the Brown corpus for American English and from it create a vanilla-flavored predictor as a start-up version for a new user, with facilities built in to have the user's own language patterns gradually outweigh the Brown corpus initialization as they are input. Eventually the Brown corpus would have essentially no effect, or at least no effect overriding the user's individual use of language (it might serve as a basic dictionary for text vocabulary not yet seen from the user).

We intend to investigate what effect generating sentences while using the system has on our collaborator. To date, she has obligingly been willing to continue to use a typewriter to generate text, but she does own a personal computer and is able to use a mouse. Our own experience in entering her sentences on the system has made it clear that in many instances she would have expressed the same ideas more rapidly on the system with a slight change in wording. Since the proferred words and patterns are derived by the system from her own language history, they should feel normal and natural to her and could influence her to modify her intentions in generating a sentence. On the other hand, a different handicapped individual (a quadriplegic) has informed us that ease of mechanical production of a sentence has little or no effect on his choice of words, and that would appear to be the case for our collaborator while she uses the typewriter.

Finally, we wish to make use of the much larger amounts of memory available on personal computers by taking account of the followers for many of the moderate-frequency words. For example, in the sentence "would you be able..." the word "able" is not high frequency. Nevertheless, the system could easily deduce what following word to expect, since every known occurrence of "able" is followed by "to". As it happens, "to" is one of the top 20 most frequent words and hence fortuitously is on the default menu after the non-high-frequency word "able", but there are many other examples where the system is not so lucky. For instance, "pick" is usually followed by "up" in the Sherri data, but "pick" is low frequency and "up" is not on the default first menu. Similarly, "think" is a high-frequency word and has a well developed set of followers. "Thinks" and "thought" are not high-frequency and hence are followed by the default first menu. Yet virtually every follower for "thinks" and "thought" in the Sherri data happens to belong to the set of followers for "think". We believe that by storing information on moderate frequency words with strongly associated followers and on clusters of verb forms we may significantly improve the success of the first menu.

RELATED WORK

That a small number of words account for a large proportion of the total verbiage in conversation has been known for some time [Kucera and Francis, 1967].

The idea of using the first several letters typed by a handicapped individual to anticipate the next desired word has been used in numerous systems (e.g., [Gibler and Childress, 1982], [Pickering et al., 1984]). The Gibler and Childress system is typical in that it uses a few-thousand-word vocabulary drawn from the general public, plus a few hundred words specific to the user of the system. The user must type the first two letters before the system provides a menu of words beginning with the letter pair. If the desired word was not on the menu, the user had to spell the word out. It was felt that one letter was not informative enough to warrant a menu. Furthermore, Gilbler and Childress showed that increasing the system vocabulary degraded the performance of their system and they recommended limitation of the vocabulary for human factors reasons.

By contrast, our system costs the user no more effort in terms of selecting the first two letters - if indeed they have needed to go that far; 80 per cent of the time, they haven't needed to provide two letters. Further, there is no question that for our system, allowing the vocabulary to grow is of benefit both to system performance and to user satisfaction.

Galliers [1987] describes a different approach for physically handicapped persons conversant in the Bliss communications system. Communication with Bliss involves a high degree of interpretation by the "listener", and Galliers reports an impressive 75 per cent success rate in automating such interpretation. The Galliers system is single-subject, as ours is, and it does use past history to facilitate interpretation. It was, however, limited to a very small domain for the experiment described.

One statistic cited by this last paper was that the same text produced from the Bliss communication, had it been produced by typing into a word processing system, would have required three times as many key-press operations. Our own ratio of key-press operations to characters produced was 45 per cent for the sentence position algorithm. That is, on average it took 45 presses of a mouse button to produce 100 characters. Part of the reason for such a high ratio has to do with punctuation, capitalization, and special screens such as the number screen, which requires not only the same number of presses of the button as there are digits, for example, but additional presses of the button to summon the screen and quit the menu. But primarily the ratio seems to derive from the fact that many of the words in any text are short - "a", "to", "the", "of", "in", and "on" being examples from this very paragraph. If the first menu does not contain a desired two-letter word, one has to spell the first letter and then make a selection from the second menu - requiring two presses of a button. By contrast, Bliss users commonly use a telegraphic style of communication and omit function words altogether.

CONCLUSION

In summary, evidence exists that for a system built around a single user's language, a prediction scheme that simply anticipated fifty or so words would on average be correct about half the time. Limiting such a system to only the top 20 most frequent words would give a success rate of about 30 per cent. However, not all of the high frequency words are distributed evenly by sentence position. A system that offers the top 20 most frequently occurring words for each position of a sentence was successful about 40 per cent of the time on the next 97 sentences. Allowing a user to reject the first set of words by giving the first letter of the desired word and offering the 20 most frequent words beginning with that letter resulted in success for the combined first and second menus 82 per cent of the time.

After a training body of 1750 sentences (14,669 words), with a vocabulary of 1512 words, it was still the case that about six per cent of the desired words were unknown to the system.

An alternative algorithm for the first offering of 20 words, based primarily on the right hand contexts of the high frequency words, is successful on the first guess 50 per cent of the time.

REFERENCES

Boggess, Lois and Thomas M. English, The HWYE speech recognition system: a user-specific model for expectation-based recognition, in Proceedings of the 25th Southeast Regional Conference of the ACM, Birmingham, 1987.

Chow, C. L. A mouse-driven menu-based text prosthesis for the speech handicapped, M.C.S. project report, Mississippi State University, 1986.

English, T. M. and Lois Boggess, A grammatical approach to reducing the statistical sparsity of language models in natural domains, Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, Tokyo, 1986.

Galliers, Julia, AI for special needs - an "intelligent" communication aid for Bliss users, Applied Artificial Intelligence, 1(1):77-86, 1987.

Gibler, D. C. and D. S. Childress, Language anticipation with a computer based scanning aid, Proceedings of the IEEE Computer Workshop on Computers to Aid the Handicapped, 1982.

Kucera, H. and W. N. Francis, Computational analysis of present-day American English. Brown University Press, 1967.

Pickering, J., J. L. Arnott, J. G. Wolff, and A. L. Swiffin, Prediction and adaptation in a communication aid for the disabled, Proceedings of the IFIP Conference on Human-Computer Interaction, London, 1984.

Wei, Jan-Soong, File organization of Sherri System, M.C.S. project report, Mississippi State University, 1987.