WIS-CS-74-212
The University of Wisconsin
Computer Sciences Department
1210 West Dayton Street
Madison, Wisconsin 53706

Received: March 8, 1974

TOUCH-TONE INPUT TECHNIQUES; DATA ENTRY USING A CONSTRAINED KEYBOARD

by

E. J. Desautels and S. B. Soffer

Computer Sciences Technical Report #212

March 1974

TOUCH-TONE INPUT TECHNIQUES:
DATA ENTRY USING A CONSTRAINED KEYBOARD *

by

E. J. Desautels and S. B. Soffer

ABSTRACT

The problems involved in using a smaller alphabet than that naturally called for are investigated, with attention focussed on the use of 12 digit keyboards such as found on Touch-Tone (trademark reg.) telephones. The feasibility of avoiding the use of codebooks or user encodings is examined for some medical information systems, and a technique to minimize redundant inputs is described.

* This work has been partially supported by N.S.F. Grant GJ-36078.

APPLICATIONS

Introduction

This study is directed towards the problems encountered in using a smaller alphabet than that naturally called for in a given application. These problems and attendant solutions are viewed primarily in terms of applications using a 12 digit keyboard, as found on Touch-Tone* telephones, with particular reference to medical applications. How well can a 12 character alphabet be utilized in an environment using up to 128 distinct characters?

Historical Background

The first computer applications were entirely numerically oriented, and the computer hardware developed for these applications was not well adapted to handling non-numeric data. The succeeding generations of computers have been accompanied by input devices capable of generating 64, then 96 or more codes. In the meantime, dial-up access to computers via the switched telephone network has become routine.

In 1967, [15] the Bell Telephone System introduced the Touch-Tone telephone. For the typical customer, it provides a more rapid "dialing" sequence. However it was expected the 12 button keyboard (see figure 1) could be used, after the dialing was completed, for the transmission of information with the receiving party using a suitable decoder. Other firms were also supplying tone encoders and decoders for private-line use, primarily for machine control applications [2]. The human use of this input device presumes that a suitable audio response unit (ARU) provides the appropriate spoken responses (see figure 2).

---

*"Touch-Tone" is a registered trademark of Bell Telephone Co.

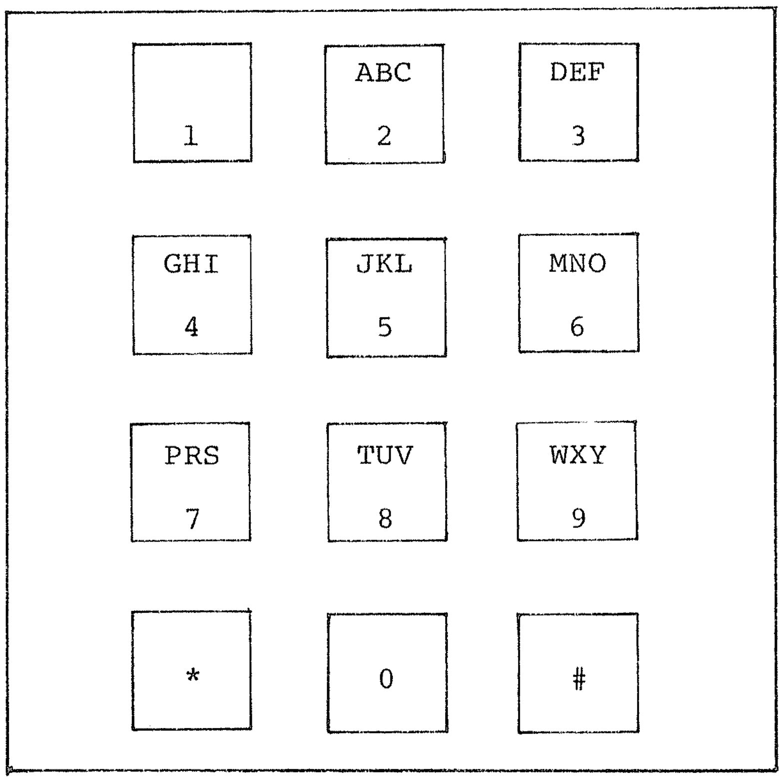


FIGURE 1

Standard Touch-Tone Keyboard

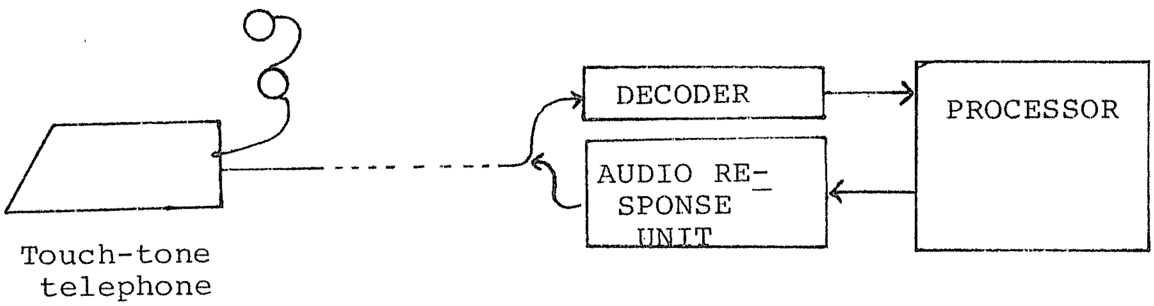


FIGURE 2

Simple Dial-up Access

In those areas of the country in which touch tone service is not yet available, portable acoustically coupled touch-tone pads are available for as little as $66 in unit quantities [8]. The costs for audio response systems have dropped to levels which make them attractive peripheral units [7].

We recognize that in some parts of the country, touch-tone service with a 16 button pad (as in figure 1, with an additional column of 4 buttons for user defined functions) is available for a small extra cost. In this case also, a low cost portable unit may be purchased for use anywhere. However since the 16 button are not likely to be nearly as widely used as the 12 button units, we are restricting our attention to the latter.

Medical Applications

There have been many interesting proposed and many implemented applications. We shall review a few of these briefly.

Medical Applications

In 1969, Allen and Otten [1] described a system with touch-tone input interacting with three programs:

1) drug compatibility information
   input: two drug codes (for 57 drugs)
   output: yes, no, condition unknown
2) computation aid for burn therapy
3) computer aided diagnosis

A medical information system with touch-tone input is described in [11]. A medical audio response telecommunication information network MARTIN with seven uses is described in [4]. This system concentrates on handling the touch-tone input and audio-response, and dynamically calls larger time-sharing systems as the application requires. It supports seven implemented uses:

1) symptom-diagnosis association
2) hematology differential diagnosis
3) pediatric burn therapy calculations
4) intravenous drug compatibility
5) mass spectrometry drug identification
6) hospital medication order and dispensing
7) electrocardiagram analysis

The Youngstown Hospital association is testing a system using touch-tone input to request clinical laboratory tests. When the tests have been completed, the results can be obtained on demand [18].

Plexico discusses several N.I.H. projects using touch-tone, and suggests others worth investigating [12].

University and Educational Applications

The Cornell University TELECUPL system [3] was intended to provide touch-tone preparation, debugging and execution of programs in a subset of the CUPL language (Cornell University program. Language). The University of Michigan supports a general application-independent touch-tone input and audio response service. It appears that in most university settings, the ready availability of full alphanumeric keyboard entry terminals renders the use of touch-tone type devices unattractive.

We note in passing that a simple programming language well adapted for entry by telephone was developed by Strasbourger [16], with the system expecting voice input. The feasibility was demonstrated, but the cost-effectiveness awaits further cost breakthroughs.

Commercial Systems

There are numerous commercial applications; it suffices to describe two of them. The now defunct Seattle banking service [5] offered a calculator service, and the ability to effect some banking transactions and bill paying. The Sears

Roebuck Company is testing touch-tone catalogue ordering in the Toronto area [9], and interim results are encouraging.

Caution

In any high-volume data entry application, touch-tone input with audio response will undoubtedly prove to be a poor way of doing business. Investment in the appropriate keyboard and data display device will quickly pay for itself in terms of employee productivity. But if an application has data entry requirements that include low volume users, with a possibly very mobile customer base, then it makes sense to consider touch-tone input in addition to conventional means.

DATA ENTRY CONVENTIONS

With one exception (Tele-CUPL), all of the above fall into three categories regarding the data entry conventions:

1) all application data and control are numeric, or
2) if non-numeric data or control is required, a code book is used to find the corresponding numeric codes or
3) the user encodes non-numeric data using a given encoding technique.

Category 1 conventions include calculator capabilities (with appropriate conventions for input error rectification). The most widely used encoding technique for category 3 relates directly to figure 1; some combination of * or # with a digit code is used to select the alphabetic or the numeric mode; when in alphabetic mode, a letter is selected by depressing the corresponding key, then depressing key 1, 2 or 3, according to the alphabetic's position (left, middle, right). This is the encoding used in the Michigan system.

The National Institutes of Health developed a touch-tone encoding for ASCII [4], shown in figure 3. One enters the control mode by pressing three or more asterisks. The following digit selects numeric or ASCII mode, or some other function. Once in ASCII mode, a digit pair selects one of 121 ASCII codes. This encoding was used in the MARTIN system.

Allen and Otten had the physician use the 4 digit codes published in Current Medical Technology [6]. The Sears-Roebuck experiment uses the codes in their catalogue, which are essentially numeric (one alphabetic designates which catalogue is involved).

SECOND DIGIT

FIRST DIGIT	1	2	3	4	5	6	7	8	9	*	0
1	SPACE 040	Q 121	Z 132	EOT 004	q 161	z 172	< 074	> 076	DC1 021	(050	1 061
2	A 101	B 102	C 103	a 141	b 142	c 143	[133	] 135	DC2 022	* 052	2 062
3	D 104	E 105	F 106	d 144	e 145	f 146	{ 173	} 175	DC3 023	) 051	3 063
4	G 107	H 110	I 111	g 147	h 150	i 151	~ 176	| 174	DC4 024	CAN 030	4 064
5	J 112	K 113	L 114	j 152	k 153	l 154	^ 136	_ 137	NAK 025	SO 016	5 065
6	M 115	N 116	O 117	m 155	n 156	o 157	` 140	\ 134	SYN 026	' 047	6 066
7	P 120	R 122	S 123	p 160	r 162	s 163	NUL 000	SOH 001	STX 002	ETX 003	7 067
8	T 124	U 125	V 126	t 164	u 165	v 166	ENQ 005	LF 012	ACK 006	BEL 007	8 070
9	W 127	X 130	Y 131	w 167	x 170	y 171	HT 011	VT 013	DEL 177	BS 010	9 071
*	+ 053	= 075	/ 057	! 041	" 042	# 043	$ 044	% 045	& 046		. 056
0	, 054	- 055	SUB 032	? 077	: 072	; 073	BS 010	SI 017	DLE 020	ESC 033	0 060

designates end-of-input
*** designates command

FIGURE 3

Paired Touch Tone to ASCII Code Conversion Table

Ambiguous Data Entry Techniques

It would be much simpler if a user could press the key marked with the desired symbol, without explicitly indicating which of several symbols was involved. Thus entry of "CAT" would require three key-strokes, as would "BAT", and these would not be distinguishable. We will refer to this simple technique as "as-is" keying.

The Cornell Tele-CUPL proposal is based on the use of a 12 key entry device, as in figure 4. However, instead of requiring the client to change input modes, they defined a subset of the application (writing, debugging and running programs in the CU programming language), and introduced a few syntactic changes in CUPL. Thus the use of a variety of input modes or character encodings was avoided, with potential ambiguities resolved by context. It could be proved that any syntactically correct TELECUPL program could be entered unambiguously.

The question arises: which of the preceding applications using encodings or codebooks could be implemented without encodings or codebooks? In a broader context, to which extent is the vocabulary of a given application subject to ambiguous mappings under an "as-is" keying, and what benefits if any can be derived from this approach?

These questions are investigated in terms of a previous medical information retrieval application and a proposed poison information service.

Definitions

The vocabulary for a given application is the set of alphanumeric words or phrases from which any single item is expected to be input by the user, followed by an end-of-item

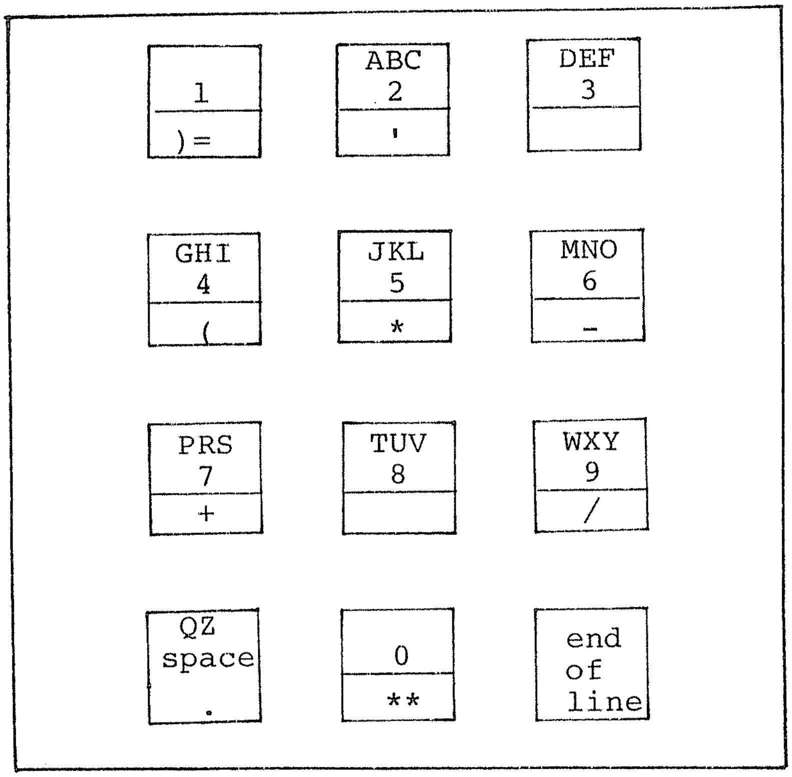


FIGURE 4

Cornell Tele-CUPL Keyboard

indication (perhaps a #). Each such item will be called a string (if the string is a phrase, we will include the embedded blanks, unless otherwise specified). Thus a vocabulary V is specified by the set $V = \{s_1, s_2, \ldots, s_n\}$ with all the s_i unique.

The transformation induced by the "as-is" keying maps V into $V' = \{s_1', s_2', \ldots, s_n'\}$ where in general we may expect duplicates to occur. Let $R = \{r_1, r_2, \ldots, r_m\}$ represent the unique strings in V'. Thus $1 \leq m \leq n$. If $m = n$, then we say the vocabulary V is unambiguous with respect to the "as-is" keying. Generally we will expect some degree of ambiguity, $1 \leq m < n$, and wish to measure this for several applications. We refer to the ratio $\frac{n-m}{n} \times 100$ as the residual ambiguity percentage. The worst would have $m = 1$ with the resulting percentage amost 100% .

Given a vocabulary $V = \{s_1, s_2, \ldots, s_n\}$, we can decompose it into a set of vocabularies $V_1, V_2, \ldots, V_K$, where V_i is the set of all strings of V of length i characters. Letting K denote the maximum string length, we have at most K such sets, since some of them may be empty, and at least one such set (if all the strings of V have the same length).

If we now consider the "as-is" mapping generating V', and discard all but the first digit of each s_i', and then eliminate any duplicate s'_j's, then we denote this set as $R_1 = \{r_{11}, r_{12}, \ldots, r_{1m_1}\}$. We define the number of collisions at length 1 as $n - m_1$. The value is bounded by $0 \leq n - m_1 \leq n - 1$. Discarding V_1, and following the above process, we can construct $R_2 = \{r_{21}, r_{22}, \ldots, r_{2m_2}\}$ whose members are unique 2-digit strings. Then the number of collisions at length 2 is defined as $n - m_2$. We continue in this manner until we have discarded all V_i except V_K. We stop after we have the number of collisions at length K.

By considering the number of collisions at ever increasing lengths (ignoring strings whose length is already exceeded) we can characterize the additional discriminatory information provided by successive incoming characters, as these successive characters reduce the uncertainty in classifing the currently entered item. Note that as the string length l under consideration grows, the uncertainty represented by the number of collisons at length l must reach zero, whether or not the residual ambiguity in the vocabulary is zero.

Recognition algorithms

A simple recognizer for $V = \{v_1, \ldots, v_n\}$ uses a table representing the set $R = \{r_1, \ldots, r_m\}$, with a variable number of entries per row. For $V = \{AA, BA, BAT, CAT, CATS\}$, we would construct

r	v
22	AA,BA
228	BAT,CAT
2287	CATS
246	BIN

The simple recognizer algorithm is:

1. concatenate incoming codes until an end-entry code arrives
2. present the user with the v list corresponding to the r entry which has been constructed. If no such entry exists, so inform the user, and return to step 1.
3. the user responds with a selection code (e.g. 0 for "none-of-these", i = 1,2,... for "accept the ith item")

An anticipating recognizer exploits the structure of the vocabulary, and attempts recognition using as few input codes as possible. The anticipating recognizer can be driven using a table such as

r	k	v
22	3	AA,BA
228	4	BAT,CAT
2287	4	CATS
246	2	BIN

where k indicates recognition requires k codes, or an end-item marker.

The anticipating recognizer is:

1. clear input buffer, set i = 0
2. accept next incoming code and increment i
3. compare the buffer with the initial substrings of length i in the r table. If no match occurs, so inform the user. For each such match, at most one will have i = k . Present its v list for the users selection. If none of the buffer-r table matches have i = k , proceed to step 2.

For both algorithms, the implementations can be optimized by ordering the tables on the r key for the first algorithm, and on the k key, then on the r key for the second. Then superior search algorithms may be used.

USING "AS-IS" KEYING

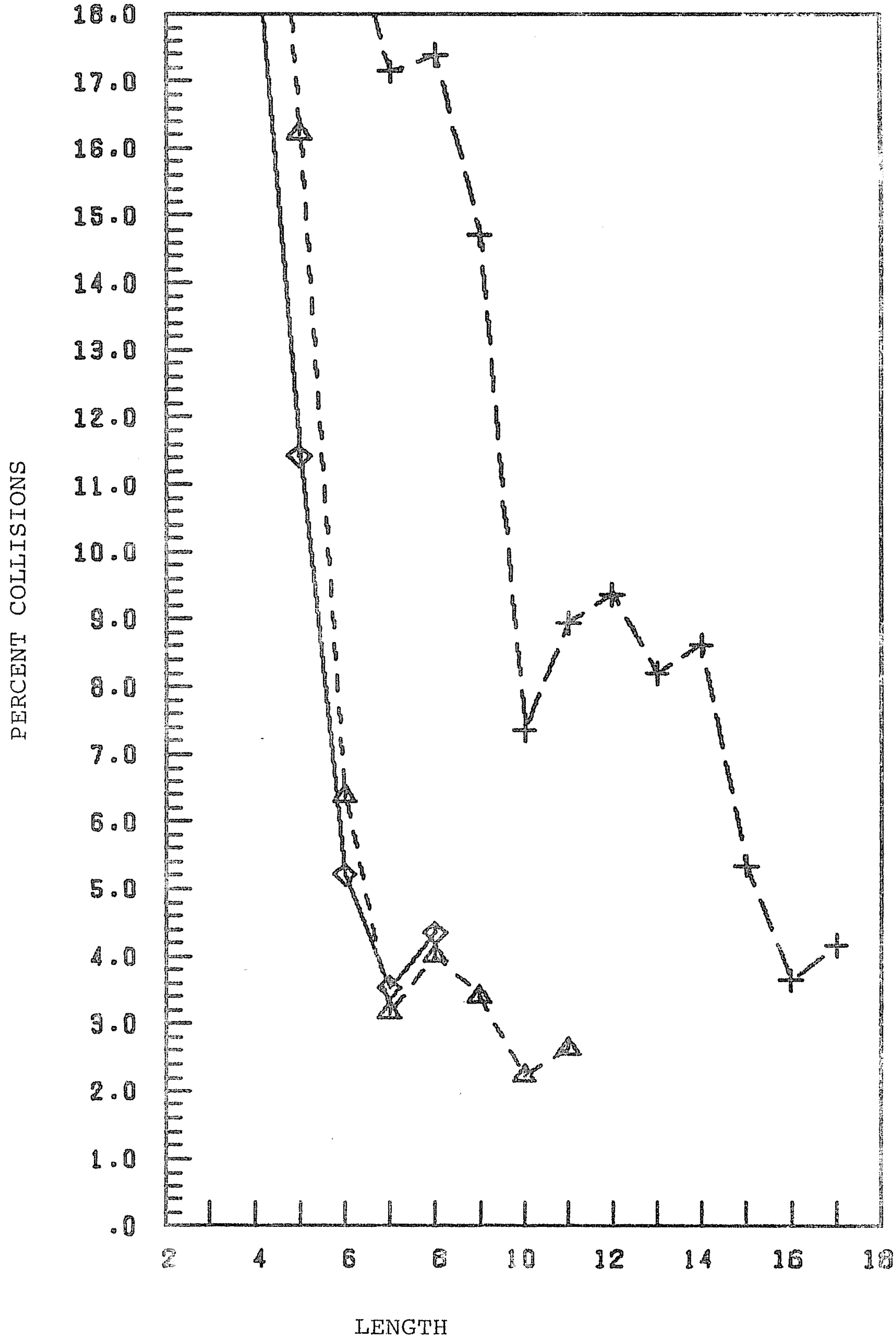
Current Medical Technology Vocabulary

The medical information system in [1] has enquiries based on the use of 4 digit codes assigned to the 2,709 prefered names for diseases in [6]. In effect this system uses a codebook. As implemented, only 72 disease names were included.

Using the "as-is" keying technique (assigning Q and Z to key 9), we found that none of the disease names collided; the vocabulary is unambiguous (whether or not a word-separating mark is used). However the average name phrase length is 19.42 characters (ignoring word separation), the shortest phrase having 6 characters and the longest 30 (actually 37, but truncation at 30 characters had no ill effect), making a 4 digit code more attractive to use.

However, we observe that one can anticipate or predict expected inputs (subject to a user overrule), and graph 1 illustrates this behaviour. The points on the phrase curve indicate the collision percentage for that vocabulary with phrases whose lengths equal or exceed the indicated length. For a specified vocabulary, a collision at a specified string length (always beginning with the leftmost character) indicates that at least two vocabulary items have identical initial substrings. The number of collisions is then one less than the total number of items involved in a collision. Thus at length 1, the 72 phrases have 65 collisions. At length 2, the number of collisions drops to 44, and so on.

The curve labelled words is based on the vocabulary derived from the 72 phrase CMT, consisting only of non-



Graph 1. Comparison of phrases(+), words(Δ), and prefixes(◇).

duplicated words from the original 72 phrases. This word vocabulary has 117 items, ranging in size from 2 to 24 characters, with an average of 9.36 characters per word.

Finally the curve labelled prefixes is based on the vocabulary derived from the 117 word vocabulary, separating nine prefixes (HYPO,HYPER, etc.) and eliminating duplicate entries. This prefix vocabulary has 116 items, with minimum, maximum and average item lengths being 2, 19 and 8.15 respectively. This decrease in size results from many names differing only by frequently used prefixes.

All three CMT based vocabularies are unambiguous. We would expect the full CMT vocabulary of 2,709 entries to be moderately ambiguous.

In applying the anticipating recognition technique, we find that the phrase, word and prefix vocabularies, with average string lengths of 19.42, 9.36 and 8.15 characters can in fact anticipate enquiries with average input lengths of 6.93, 4.39 and 4.15 character respectively.

Poison Information System Proposal

Many communities provide poison information services, usually based in hospitals. The caller specifies which poison, drug, household product or other potion has been swallowed, and the attendant at the information service provides emergency treatment information (e.g. "induce vomitting", "do not induce vomitting",...). The question arises: to what extent could such a service be automated? Then one could assess it whether in fact it should be automated. In exploring this question, we restrict our attention to the enquiry or data-entry portion.

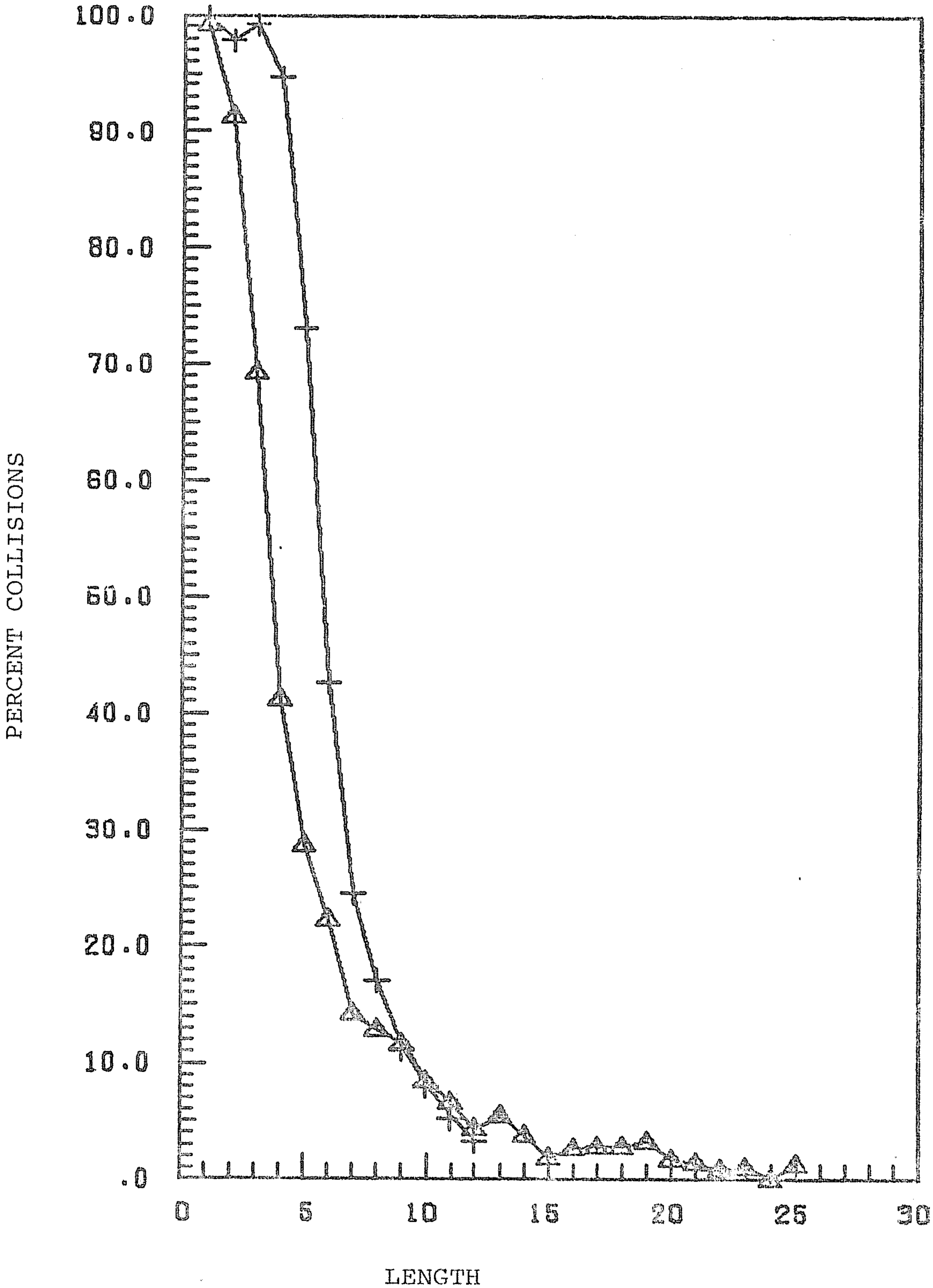
Reference [13] lists 626 words, brand names and phrases for common household poisons. This list will be called the Toxic-1 vocabulary. The abbreviated names for these poisons (e.g. DDT) constitute a 117 item list called Toxic-2, and the chemical names (words and phrase) from the 422 item list Toxic-3. The residual ambiguities are 4, 4 and 8 respectively (0.6%, 3.4%, 1.9%), when these are treated as separate vocabularies. When the three vocabularies are merged and duplicate entries eliminated, the residual ambiguity percentage is 0.53% (6 collisions for 1,133 items). This illustrates the fact that as larger and larger vocabularies are used, even though the number of ambiguities may rise, the percentage of ambiguities may in fact decline.

Dictionary and Index

In order to see whether medical vocabularies were especially predisposed to "as-is" keying, two test vocabularies were selected. One of these, consists of 146 entries from the index of [10], a mathematics textbook. The other, identified as Dictionary, consists of approximately the first 333 entries from the A,B, and C listings of the Webster's dictionary [17], totalling 1,000 items (proper names were excluded). This latter vocabulary can well serve as a "worst case"; it has 49 residual collision, for the highest rate encountered in this study, 4.9%. Ironically, 26 of the 49 collisions were triggered by four letter words. The mathematics vocabulary had a 4.8% ambiguity percentage.

In effect this suggests that perhaps 5% to 10% of the terms needed for a given application might require exception-handling. All vocabularies examined had similarly shaped collison-length groups.

Graph 2 relates the rate of change of uncertainty for the combined Toxic vocabulary with the Webster's dictionary items. The sharp drop in both curves would suggest suggest the use of an anticipating recognizer.



Graph 2. Comparison of toxic group(Δ) with dictionary(+)

Implications

It would appear to be possible to implement a poison information center using "as-is" keying, with few ambiguities resulting. As the vocabulary would normally expand with time, new ambiguities which result can be identified, and since it is an absolute requirement of limited data entry systems that immediate feedback be provided, the user would immediately have an opportunity to specify which, if any, of the pre-stored phrases was acceptable. If none were, then reversion to explicit unambiguous alphanumeric encoding must be used.

Similarly, the results for the Current Medical Terminology vocabularies show that "as-is" keying leads to few ambiguities (none for the subset considered).

In all of the vocabularies, an anticipatory recognizer would reduce redundant inputs, generally by about 50%.

In the CMT-1,2 and 3 vocabularies, the average string lengths are 19.4, 9.4 and 8.2 characters respectively. Unambiguous predictions can be made after an average of 6.9, 4.4 and 4.1 characters respectively. As a practical matter, if a codebook is readily available, then since a 4 digit code suffices in all cases, then "as-is" keying is clearly less efficient. However, this measure of efficiency has really little merit in those instances when a single, or very few enquiries are required. What this suggests are alternate access modes to an information system, whereby the new user accesses it in the "as-is" mode, and progresses to a code-book mode, if and when his level of use profits by it.

A further advantage of "as-is" keying in a system which is infrequently used by a large number of people is its quick adaptability to accept new terms, without the possibility of substantial delays incurred in updating many extant codebooks.

The Bell System is investigating the use of "as-is" keying. In its first phase of investigation, the customer would call the automated directory information service and key in the name of the desired party. Should this be insufficient to uniquely identify the desired party (Mr. Bat and Mr. Cat are indistinguishable, and Mr. Smith is insufficient), the automated service will request additional keying as required. When a unique determination is obtained, the audio response unit will read the desired number. The next phase of development will lead to a call-by-name service; instead of merely finding and reading the desired number, the automated directory assistance service may proceed to do the dialing itself and complete the call. The investigation is being done using the facilities of the Unix time sharing system [14].

Other Considerations

The question of user errors and the sensitivity of any application to these errors has to be investigated. Simple keying errors such as omitting a letter, introducing an extra letter or interchanging two consecutive letters can lead to rejection or faulty recognition. Fortunately because a well designed system provides immediate verification of input with a retry capability, these user errors will not be fatal. If in a given application, misspelling is a frequent occurrence, a "best-match" criterion might be substituted for the "exact match" criterion we have. One can also sanction the most common misspellings by incorporating them in the vocabulary.

In an application such as a poison information service, provision must be made for unanticipated user enquiries, such as might be precipitated by the marketing of a new bleach or floor wax product in an attractive bottle. Therefore the need for unambiguous data entry techniques persists, and further investigation should be pursued in this area.

One obstacle encountered in introducing new applications using new devices in the context of the larger computing systems is the tendency for these to have a single "standard software interface" or "device support routine" for a given device, based on the need to make that device look like other devices on the system. The immediate and major benefit is to make most, if not all, system services available via the new device. Unfortunately the user may find his application insulated from the special characteristics which might be exploited. This is a classic example of the "generality versus optimization problem." Its solution lies in providing both a standard interface and an escape mode for user-defined purposes.

We would expect that the ambiguity levels reported here could be further decreased by using an alphabetic-to-digit mapping differing from that shown in fig. 1. This warrants investigation for an "in-house" system, but would somewhat defeat the attraction of using "as-is" keying on an "as-is" telephone. The need for push-button overlays is one step removed from requiring a code book.

CONCLUSIONS

Some applications in the medical area may use touch-entry on an "as-is" keying basis, minimizing the need for user encodings or codebook lookups, with relatively few (5-10%) ambiguities, if any. Any such ambiguities may further be satisfactorily resolved on a selection basis.

The anticipatory recognition technique for decoding the inputs may reduce the keying requirements by 50% or more.

Also, the touch-tone phone is not a general purpose device, but would have its forte in restricted applications, and should be considered as a secondary input medium for more general applications.

REFERENCES

1. Allen, Scott I., and Otten, Michael., The Telephone as a Computer Input-Output Terminal for Medical Information. JAMA 208, 4, April 28, 1969, pp. 673-679.

2. Bramco Controls Division, Ledex, Inc. 550 S. College St., Piqua, Ohio, 45356.

3. Conway, R. L., and Morgan, H. L., Tele-CUPL: A Telephone Timesharing System., CACM 10, (Sept.) 67, pp. 538-542.

4. DeLeo, James M., White, William., and Songco, David, Medical Audio Response Telecommunications Information Network, Bio-Medical Computing 3, 1972, pp. 293-305.

5. Glass, Robert L., "In-Touch Out of Touch". Computerworld, Jan. 9, 1974, p. 1.

6. Gordon, B. L. (Ed.), Current Medical Terminology, American Medical Association, Chicago, Illinois, 1964.

7. Interface Systems, Inc., 5 Research Drive, Ann Arbor, Michigan, 48103.

8. Interface Technology, Inc., 10500 Kahlmeyer Drive, St. Louis, Missouri, 63132.

9. McIntyre, Hugh C., "Sears Testing Touch-Tone Shopping", Computerworld, Nov. 7, 1973, p. 2.

10. Meyer, P. L., Introduction to Probability and Statistical Applications. Addison-Wesley, 1972.

11. Otten, Michael, Allen, Scott I., Plexico, Perry, and White, William C., An Audio Input-Output Computer System for Medical Information, Proceedings of the 24th National Conference of ACM, 1969, pp. 477-484.

12. Plexico, Perry S., Computer Driven Voice Response in Medical Applications, Division of Computer Research and Technology, National Institutes of Health, 1972.

13. Press, Edward, Accidental Poisoning in Children, Charles C. Thomas, Publ., Springfield, Illinois, 1956, pp. 52-117.

14. Ritchie, D. M., and Thompson, K., The UNIX Timesharing System, (to appear in Communications of the ACM).

15. Soderberg, J. H., Campbell, R. R., and Bates, F. E., The Touch-Tone Transmission of Digital Information. Proceedings of International Convention of IEEE. New York City, March 1967.

16. Strasbourger, E., Recognizing a Continuous Spoken Language Using Analytic Upstream Encoding, Ph.d. Thesis. University of Washington, Seattle, 1972.

17. Websters Seventh New Collegiate Dictionary, G. & C. Merriam Company, Springfield, Massachusetts, 1969.

18. Youngstown Hospital Association, Data Input Voice Output Telephone System, Computer (Nov. 1973) p. 45.

BIBLIOGRAPHIC DATA SHEET	1. Report No. WIS-CS-74-212	2.	3. Recipient's Accession No.

4. Title and Subtitle
Touch-tone Input Techniques; Data Entry Using a Constrained Keyboard

5. Report Date
March 1974

6.

7. Author(s)
E. J. Desautels and S. B. Soffer

8. Performing Organization Rept. No. 212

9. Performing Organization Name and Address
The University of Wisconsin, Computer Sciences Department, 1210 W. Dayton Street, Madison, Wisconsin 53706

10. Project/Task/Work Unit No.

11. Contract/Grant No.
GJ-36078

12. Sponsoring Organization Name and Address
National Science Foundation
Washington, D. C. 20500

13. Type of Report & Period Covered

14.

15. Supplementary Notes

16. Abstracts
The problems involved in using a smaller alphabet than that naturally called for are investigated, with attention focussed on the use of 12 digit keyboards such as found on Touch-Tone (trademark reg.) telephones. The feasibility of avoiding the use of codebooks or user encodings is examined for some medical information systems, and a technique to minimize redundant inputs is described.

17. Key Words and Document Analysis. 17a. Descriptors
Telephone, touch-tone, data entry, medical applications, information, retrieval, audio response, terminals, remote access, portable devices

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report)	21. No. of Pages
Unlimited	UNCLASSIFIED	18
	20. Security Class (This Page) UNCLASSIFIED	22. Price

FORM NTIS-35 (10-70)   USCOMM-DC 40329-P71