9800 293-4
IPO56 copy

TEXT INPUT DEVICE AND METHOD WITH KEY AMBIGUITIES

Field of the Invention

This invention relates generally to a data entry method and apparatus, and more particularly to a method and apparatus for inputting text, for example text messages, from an input device such as a keypad, especially a numeric keypad, where key ambiguities exist and need to be resolved.

Background of the Invention

With the increasing desire to carry smaller and smaller personal communicating devices with optimum functionality such as pagers, cellular phones, and other personal communicator devices, and the desire to automate homes through combinations of telecommunications and cable where smart rooms will have one or more small devices that operate a variety of appliances or control a variety of applications; the need and desire to enter alphanumeric text through non-alpha or numeric keypad is ever increasing.

In the United States of America it is standard for every telephone keypad to have letters as well as numerals displayed on the keys. Thus, the key representing the numeral "2" has the letters A, B and C displayed and the key representing the numeral "3" has the letters D, E and F etc. It is common to represent a telephone number by the letters which correspond to the digits on the keypad. Thus, for example, a railway company with the name "Amtrack" can acquire the telephone number 268 7225 and can advertise to customers that the company's telephone number can be dialed by activating the keys with the letters A-M-T-R-A-C-K. Dialing these letters will result in only one digital sequence, but one digital sequence can be represented by many letter sequences.

In the telecommunications field, methods of name selection are known which use a numeric keypad. For example, it is known in a voice mail system to select a user's voice mailbox by keying in from a

remote telephone the first few letters of a user's name, as a result of which a short digit stream is delivered to the voice mail system. The voice mail system compares the input digit stream with a limited set of options in a look-up table. Sufficient digits must be entered to uniquely identify the desired user. Similarly, automated customer service systems can use this form of text entry. For example an airline information system can request the first three letters of a destination airport and, notwithstanding ambiguities in the letters to be represented by the digits delivered, it can uniquely identify an airport from only three digits because it has a limited "dictionary" of airport names from which to seek a match. The same approach can be used in a telephone speed-dial look up arrangement. A name can be selected from a limited list of names stored in a telephone terminal and a number can be dialed associated with the name selected.

Various other approaches for entering alphanumeric input through a non-alpha, numeric only, or other type of keypad having fewer keys or buttons than the corresponding alphabet have been developed or are known. Many of these methods take advantage of the letters displayed on the standard US telephone keypad. However, often the methods associated with the alphanumeric text through the keypad are cumbersome and difficult to edit. In addition, typically the entry must be restricted or entered in a specific manner such that there is only one option per sequence of input. More specifically, there is a direct correlation for a specific input to a specific letter of the alphabet or a symbol. For example, if a user wants to input the word "call" through a non-alpha keyboard, the user has to enter a specific combination of numbers and/or symbols to indicate the word call. By way of example for a user to enter the word call the user must enter the following: 2***; 2*; 5***; 5***; #. Such a method of inputting is very restrictive, time consuming, and editing is difficult and slow.

The above described methods of data entry do not readily apply themselves to the Chinese language. There are several thousand characters in the Chinese language. For example, there are nearly 7000 Chinese characters supported by the GB coding standard used in China,

and about 13000 Chinese characters supported by the Big5 coding standard used in Taiwan and Hong Kong. It is a major challenge to build a device that can select characters from such a large dictionary using a small number of keys, for example just 12 keys of an electronic keypad.

Therefore a need exists to provide a user with a simple straightforward method and apparatus for inputting text through a keypad having fewer keys than the desired alphabet; and the need exists to provide such a system without resort to massive computational power or memory demand.

Various preferred embodiments of the invention are now described, by way of example only, with reference to the accompanying drawings.

Brief Description of the Drawings

FIG. 1 is a front view of an illustrative text entry device suitable to receive input in accordance with the present invention.

FIG. 2 is a block diagram detailing internal circuitry of the device of FIG. 1.

FIG. 3 is a data structure diagram illustrating a first table stored in the memory of the device of FIG. 2 in which address information is stored.

FIG. 4 is a data structure diagram illustrating a second table stored in the memory of the device of FIG. 2 in which character information is stored.

FIG. 5 is a data structure diagram illustrating a third table stored in the memory of the device of FIG. 2, in which bi-gram information is stored.

FIG. 6 is a flow diagram illustrating the operation of the processor of FIG. 2.

FIG. 7 is a flow diagram detailing the subroutine 110 of FIG. 6 for finding the digit stream ID.

FIG. 8 is a flow diagram detailing the subroutine 400 of FIG. 6 for selecting a character.

FIG. 9 is a flow diagram for finding Chinese characters that are common to a uni-gram list and a bi-gram list.

Detailed Description of the Drawings

Generally, a method is now described comprising the steps of: maintaining in memory a first table of digit streams or sequences representative of characters in a desired language and character addresses corresponding to the digit streams; activating keys of a data entry device (e.g. a keypad, but alternatively a keyboard or a hunt-an-pick digitizer and stylus), where a key represents a plurality of characters to provide an input digit stream; searching the first table in the memory for a digit stream corresponding to the input digit stream and obtaining a corresponding character address. Here, "character" refers generally to letters or to pictographic or grapheme characters.

Since the mid 1950s, a standard phonetic representation of Chinese characters has been adopted known as "pinyin". The pinyin for each Chinese character, like each English word, is made up from combinations of the 26 letters of the Roman alphabet a-z. These 26 letters can be mapped to a telephone or other keypad of only, say, 12 keys, with 3-4 letters displayed on each key. Such a keypad can be used to enter Chinese text using the pinyin system of writing, provided that appropriate means are provided to resolve ambiguities. Such means are now described.

Referring now to the figures, FIG. 1 illustrates a general and preferred embodiment of apparatus in accordance with the present invention. An example of a cellular telephone (or other data entry device such as a wireline telephone, pager or personal digital assistant or telecommunications device) having a keypad is illustrated as generally depicted by the reference numeral 10. The cellular telephone 10 depicted comprises a housing 12, a display 14 and a keypad 16 through which input is received. The display 14 has a text display area

15 and an area 17 for displaying character alternatives. The keypad 16 has twelve keys with digits 0-9 displayed thereon in a standard layout. Also displayed in a standard layout are pinyin letters a-z. These letters can be arranged in other arrangements, for example for greater ergonomic efficiency, but the illustrated layout is preferred on account of established use. The key bearing the digit "1" has the punctuation marks ",", ".", "?" and "!" displayed thereon. The lower left hand key 18 has the symbols "<<" (meaning "backspace" or "scroll left" as is explained below) and * (referred to as "star"). The lower right hand key 19 has the symbols ">>" (meaning "enter" or "scroll right" as is explained below) and "#" (referred to as "pound").

As an alternative keypad layout, the digits 7 through 0 can represent <p r s>, <t u v>, <w x y>, <q z> respectively or digits 7 through 9 and digit 1 can represent <p r s>, <t u v>, <w x y>, <q z> respectively.

In the preferred embodiment illustrated, the pinyin alphabet is used. Alternatively, various other alphabets may be used in accordance with the preferred embodiments of the present invention, such as another Chinese phonetic alphabet (Zhuyin) or a stroke-based alphabet or English, Russian or Asian alphabets. In the preferred embodiment of the present invention illustrated in the figures, the input device is a keypad having fewer individual keys than the number of letters in the alphabet. Therefore, the alphabet is separated into various sequences where a single numeral and a series of letters or symbols are represented on a single key, or block of input, of the keypad or other input device. Pressing of a key delivers a digit, but the digit can represent one of a plurality of letters or characters.

Referring to FIG. 2, the cellular telephone 10 is illustrated as having a microprocessor 30 coupled to the input pad 16 and to the display 14 using standard input and output drivers as are known in the art. Also coupled to the microprocessor 30 are a first memory 31 which is preferably electrically-erasable read-only memory (EEPROM) or flash memory and a second memory 32 which is preferably random access memory (RAM). In the EEPROM memory is stored at least three tables

- Table 1 (33), Table 2 (34) and Table 3 (35). The tables 33-35 are described in greater detail with reference to FIGs. 3, 4 and 5 respectively. An optional loudspeaker 38 is shown coupled to the microprocessor.

Referring to FIG. 3, the data of Table 1 of FIG. 2 is shown. The table comprises a first column 101 storing digit strings that correspond to all the possible pinyins of all the possible Chinese characters. There are 219 possible digit streams (sequences) corresponding to a 408 possible pinyins. (One digit string may correspond to several pinyins because three or four Roman alphabets are mapped to one telephone digit). The various possible pinyins correspond to the many thousands of characters. The first column is 4 bytes in width. To the left of the first column 101 in the figure is an identification number (ID) of each digit string. This number is not stored in the memory. The digit strings are stored in the order of their values, that is, the smaller its value, the smaller its index. The second column 102 is 2 bytes in width and stores the beginning address (in Table 2) of the Chinese characters having the same digit string. (The total number of Chinese characters that have the same digit string can be found from the difference between the adjacent two elements of the second array; for example the difference (7-1=6) between the first and second elements means that there are 6 Chinese characters having the same digit string 2.) To the right of the second column 102 (not stored in the memory) is shown the pinyins that are mapped to the corresponding digit strings.

Table 1 is used, as will be described in greater detail below, for locating a start (and end) address in Table 2 for all the actual Chinese characters that possibly correspond to a given digit stream

Table 2 is shown in FIG. 4 and comprises four columns 110-113. In the first column is shown the hex GB coding representation of a character (2 bytes for each item). There is a row in the table for every Chinese character in the dictionary. To the left of the table in the figure is shown the actual pictographic Chinese character as it would be written or as it appears on the text display area 15 and the lower display area 17 of the display 14. This form of representation need not be stored in Table 2 (but could be stored in Table 2). The Chinese characters

having the same digit string are stored sequentially in the dictionary and then sorted by the hex GB coding values, that is, the smaller its value, the smaller its index. Shown on the left hand side of the figure (but not stored in the memory) are the indexes representing the relative addresses of the Chinese characters in each group of Chinese characters having the same digit stream. The second column 111 stores the uni-gram sorted relative address of each character that has the same digit string (1 byte for each item). It is sorted by the frequency (uni-gram) of its corresponding Chinese character. That is, the higher the frequency, the smaller its index. The frequency is shown to the right of FIG. 4 but this is not stored in the memory. The third column 112 (3 bytes for each item) stores the beginning address in Table 3 of the bi-grams that have the same first character. The fourth array stores the number of bi-grams in Table 3 that have the same first character.

By way of illustration, lines 1 to 6 of the left hand side of FIG. 4 stand for the relative addresses of the Chinese characters having the same digit string "2", and Lines 7 to 9 of the same column stand for the relative addresses of the Chinese characters having the same digit string "3".

Column 111 stands for the sorted relative addresses of the Chinese characters in each group in an order of their frequencies. For example, in the first group (Lines 1 to 6), the frequency in the second line is the highest one (frequency = 11574), so its corresponding Chinese character relative address "2" is stored first (see top line, column 111). The third and fourth columns 112 and 113 stand for the beginning address in Table 3 and the number of bi-grams in Table 3 having the same first character respectively.

Referring now to FIG. 5, Table 3 is shown in which Chinese character bi-gram data is stored. A bi-gram is a combination of two characters with which is identified a frequency of occurrence of that combination in the language in question. For example the bi-gram "qu" is quite common in the English language whereas the bi-gram "qa" is very rare. This information is used to identify likely and

unlikely character combinations, which in turn is used to resolve ambiguities in data input.

In FIG. 5, the first and second columns show the list of the Chinese character pairs of the bi-grams (in Chinese and in "hexadecimal" GB coding form and separated by ":"). The third column shows for the co-occurrence frequency of these two Chinese characters. The fourth column shows the digit string of the second Chinese character in the Chinese character pair. These four columns are not stored in the memory. The fifth and sixth columns 160 and 161 of FIG. 5 are stored as table 35 in memory.

Column 160 contains the corresponding digit string ID showing the location of the digit stream in Table 1. For example, on the third line, the ID of the digit string "22" is "4". The column 161 contains the second Chinese character's relative address in the Chinese character dictionary in Table 2.

Taking the group having the same preceding character "b0a2" as an example, column 160 is sorted by its element values (digit string IDs). If the successive elements in the column 160 have the same digit string ID, then their corresponding elements in the column 161 are sorted in the order of their co-occurrence frequencies in column 160. From this it can be seen that the digit stream ID - relative address pair can uniquely locate the Chinese character stored in the Chinese character dictionary in Table 2.

In each group having the same preceding character of the bi-gram, both of them are sorted first by digit ID, second by bi-gram probabilities.

In Tables 2 and 3 there is no frequency data stored. Instead the order of storage of the uni-grams represented by the sorted relative addresses of column 111 in Table 2 and the bi-grams represented by the sorted digit stream ID - relative address pairs of columns 160 and 161 in Table 3 indicate the frequencies. This is an efficient form of storing of this extensive data.

The structure of the data entry device has been explained and including the structure of the tables in memory. The operation of the device is now described.

In standard data entry mode, a user enters the pinyin of each Chinese character he wants to input by using the telephone keys marked a,b,c,d etc., in the same manner as described in US patent application no. 08/754,453, filed on 11/21/96, which is assigned to the assignee of the present application and is incorporated herein by reference.

A user enters pinyin letters into the keypad 16 and these are passed to the microprocessor 30. The microprocessor has the task of identifying a set of possible characters represented by the pinyin entries and identifying the most likely candidate characters from a set of possible characters. This process is illustrated in FIG. 6.

Starting from step 200, a user enters a digit string in step 210 and the digit string entered is first searched in step 210 through the first table 33 (Table 1 shown in FIG. 3) by using a fast binary search method. Since the digit strings are sorted in the order of their element values, and there are only 219 elements for the pinyin based method, it only requires 8 comparisons. This is a fast and efficient method of identifying the entered digit stream. Details of step 210 are described below with reference to FIG. 7.

Continuing to refer to FIG. 6, upon entry of a new digit stream, the process decides in step 215 whether this new digit stream entry is indicative that a previously entered digit stream resulted in successful user selection of a desired character and, if so, the process concludes that adaptation of the uni-gram and bi-gram models is appropriate as a consequence of this user selection. Such adaptation is performed in steps 211 and 212 and is described later below.

If there is to be no adaptation of the language models, the process skips to step 311. This is the case where, for example, a previously entered digit stream was entered incorrectly and has been deleted and should not be used for adaptation. For simplicity of explanation, this situation is described now.

Using the digit string ID as the index, the microprocessor 30 finds the beginning address and the number of the Chinese characters that have the same digit string from the second table 34 (Table 2 shown in FIG. 4). In step 311, the initial list of the candidate Chinese characters is read from Table 2. By using the address and the number obtained in step 210, all the candidate Chinese characters that have the same digit string can be found from the Chinese character dictionary array described in Table 2. They form the initial list of the candidate Chinese characters.

Following step 311, step 312 causes a list of bi-grams to be read from Table 3 of FIG. 5. Using the previous character that has already been selected by the user, the processor 30 finds the beginning address and the number of the bi-grams that have the same first character from columns 112 and 113 of Table 2. Then the processor locates all the bi-grams that have the same first character from the bi-gram arrays described in Table 3.

Since the bi-grams are sorted in an order of digit string IDs in each group having the same preceding character, the processor can use the fast Binary Search method in step 313 to find a particular digit ID corresponding to the digit string entered by the user. Since the bi-grams with the same digit ID are also stored one by one in an order of bi-gram probabilities, it is easy to find the other bi-grams with the same digit ID by searching the adjacent items in the first array of bi-grams. As soon as all the items with the same digit ID are found, processor 30 also has the relative addresses of the characters that have the same digit ID from the second array of the bi-grams.

In step 314 the final list of the candidate Chinese characters is created. By using the relative addresses (which are stored by bi-gram probabilities) obtained in step 313, the processor 30 gets the characters from the candidate Chinese characters' list obtained in step 311 and puts them into the top of a final candidate Chinese characters' list. Since Table 2 stores the relative addresses of the characters sorted by frequencies (uni-grams), it can be used to get the rest of the Chinese characters from the initial candidate list and put them into the final

one. The processor then displays in display area 17 of display 14 the Chinese characters in an order of the final candidate list.

So far, there has been described a method of text entry into a device comprising the steps of: maintaining in memory a first table of digit streams representative of characters in a desired language and character addresses corresponding to the digit streams; maintaining in memory a second table of characters located at addresses corresponding to the character addresses in the first table; activating keys of a data entry device, where a key represents a plurality of characters to provide an input digit stream; searching the first table in the memory for a digit stream corresponding to the input digit stream and obtaining a corresponding character address; and reading from the second table at least one character located at the character address. It has been described how the step of searching comprises performing a binary search and the step of reading comprises locating a plurality of rank ordered characters from the second memory located at addresses in predetermined locations relative to the character address. Also it has been described how ambiguities between the characters of the plurality of characters are resolved using language model data (e.g. bi-gram and uni-gram data).

There now follows a description of software and a method of selecting a character and reordering the relative addresses of the characters in the memory dependent on the character selected. It will be described how character identifications which correspond to a common digit stream are stored at a plurality of consecutive addresses and relative addresses of characters rank ordered in preference are stored at the consecutive addresses, independent of the consecutive addresses at which the relative addresses are stored.

The adaptations of the language model are fast on-line adaptations. Step 211 of FIG. 6 performs adaptation of uni-gram modeling by operation on the data stored in column 111 of Table 2 and step 212 performs adaptation of bi-gram modeling by operation on the data stored in column 161 of Table 3.

Referring to the first of these adaptation steps, it has been described above that the second column 111 of Table 2 stores the relative address of each character that has the same digit string. It is sorted by the frequency (uni-gram) of each Chinese character. Step 211 operates by placing in the uppermost position 114 in the column 111 the relative address of the character last selected and removing that relative address from its previous position in the column. Relative addresses (in positions 115) of characters having the same bit stream are shifted downwards in column 111 as far as the address removed.

This process is a process of always placing the last selected character at the top of the list of alternatives. The order of storage of the characters in Table 2 remains constant but the order of storage of the relative addresses is changed. As such, true frequency information is lost, but the utility of the arrangement is enhanced because information is retained as to the characters most recently used by this user (whereas true frequency data indicates historical frequency of use by all users).

For example, assume that there are three Chinese characters A, B, and C corresponding to one digit string. Their values and probabilities have the following relations: B < A < C and Prob{A} > Prob{B} > Prob{C}. These three Chinese characters are stored in the dictionary in the order of their values:

1 B 2
2 A 1
3 C 3

In this table, the first column stands for their relative addresses (not stored in the memory), the second column (which is column 110 of Table 2) stands for these three Chinese characters stored in the dictionary, and the third column is column 111 of table 2, which stores the relative addresses of these three character in an order of their probabilities. When the user enters the digit string corresponding to these three characters, the device will display them in an order of

A B C

because the relative address of character A is stored in font of B, and the relative address of character B is stored in font of C. If the user chooses character C as the one he wants to input, only the third column needs to be updated

1 B 3
2 A 2
3 C 1

So next time, when the user enters the same digit string, the system will display these characters to the user in an order of

C A B

because now the relative address of character C is stored in front of A, and the relative address of character A is stored in font of B.

Instead of keeping the order of storage of the characters in Table 2 constant and changing the order of storage of the relative addresses, the order of storage of the characters can be rearranged, provided also that the corresponding data in columns 112 and 113 is rearranged and provided also that relative address data is retained for cross-referencing, because Table 2 is a complete dictionary of characters and information must be retained for locating the actual characters.

The process of Adapting Bi-grams in step 212 is similar to that of the uni-gram in step 211. Again, the example can be considered where there are three Chinese characters A, B, C in the bi-grams corresponding to one digit string and having the same preceding character D. Their bi-gram probabilities have the following relations:

Prob{D,A} > Prob{D,B} > Prob{D,C}.

They are stored in the bi-grams in the order of

4 2 A
4 1 B
4 3 C

Lines 1 to 3 of the first column (which is column 160 of Table 3) stand for the digit string IDs of Chinese characters A, B, and C, respectively, where we assume that the Chinese characters A, B, and C have the same digit string ID "4". The second column (which is column 161 of Table 3) stands for their relative addresses, respectively, in the order of

THIS SHEET HAS BEEN REPLACED

their bi-gram probabilities. The elements of the third column are the second Chinese characters in the Chinese bi-gram character pairs which are represented by their corresponding digit ID-relative address pairs. When the user enters the digit string "22" with its digit string ID being "4", the system will display these characters to the user in an order of

A B C

because the relative address of character A is stored in font of B, and the relative address of character B is stored in font of C. If the user chooses character C as the one he wants to input, only the second column needs to be updated

4	3	C
4	2	A
4	1	B

So next time, when the user enters the same digit string, the system will display these characters to the user in an order of

C A B

because now the relative address of character C is stored in font of A, and the relative address of character A is stored in font of B.

In this adaptation of the language models, the bi-gram has higher priority over the uni-gram. If the Chinese character selected by the user can be found in its bi-grams, then only its bi-grams are updated. Otherwise, its uni-grams are updated.

By way of completeness, details are now described for steps 210 and 400 of FIG. 6.

Referring to FIG. 7, a flow diagram is shown describing in greater detail the process of finding the digit stream ID performed in step 210. The process starts with digit input in step 700. When the digits of a complete character have been entered, the user presses key 19 (FIG. 1) and step 701 recognizes entry of the key "#". When a digit is to be deleted the user presses key 18 and step 702 recognizes entry of the key "*" and deletes the last digit in step 703. With each digit entry, the process returns around loop 704, until key 19 is pressed.

This Sheet Has Been Replaced



Upon pressing key 19, the process proceeds to step 710 which signifies the end of pinyin entry for a single character and commences character identification. In step 711 the fast binary search is performed in Table 1 to locate the character entered. If the search reveals no match to the digit stream entered, step 712 causes the process to proceed to step 720 and a "beep" or other error signal is given through loudspeaker 38 (FIG. 2) or by other indicator. If step 711 successfully finds a match, step 715 causes the process to proceed to step 215 of FIG. 6.

Referring to FIG. 8, the subroutine of step 400 (FIG. 6) begins by displaying in display area 17 of the display 14 the top 10 (or fewer) candidate characters which match the digit stream entered. The user presses a key in step 801.

If the user does not see the character he intended to enter in this list, he can press key 19 or key 18 and steps 802 and 804 respectively will cause steps 803 and 805 respectively to display the next 10 or the previous 10 character respectively.

If neither of keys 19 and 18 is pressed (implying that a digit was pressed in step 801), step 810 causes the character corresponding to the digit entered to be selected. For example if the character desired is displayed in position number 1 in the list and the user enters "1", this character is selected. The selected character is the final output of the device. The selected character is displayed in the text display area 15 of the display 14 and is passed to any application program in operation, for example to a word processing program. At step 812 the program returns to step 200 of FIG. 6. The program is now ready to use the selected character to update the uni-gram and bi-gram models in steps 211 and 212 as described above.

As an alternative method for searching the bi-grams the following is presented. To store the bi-grams requires a large amount of memory. The bi-grams and the bi-gram searching methods described above are specifically designed for one particular input method, i.e., the pinyin based method which is used in mainland China. However, if other input methods, such as a Chinese character stroke based method, are to share the bi-grams with the pinyin based method, more general

bi-grams and bi-gram searching methods need to be provided for both of them at the expense of increasing the computation power.

The alternative bi-gram memory only stores the second characters of the bi-gram (2 bytes for each item). The characters having the same preceding character are stored successively and then sorted by their values as done for the Chinese characters stored in column 110 of Table 2. Since there are less than 8000 Chinese characters required to be modeled by the bi-grams, only 13 bits are required to store one Chinese character. The other 3 bits can be used to order the second characters of the bi-gram based on their bi-gram probabilities.

Again, assume that there are three Chinese characters A, B, C in the bi-grams corresponding to one digit string and having the same preceding character D. Their values and bi-gram probabilities have the following relations:

B > A > C and

Prob{D,A} > Prob{D,B} > Prob{D,C}.

They are stored in the bi-grams in the order of

C 5

A 7

B 6

The first column stands for the Chinese characters stored in the first 13 bits of the 2 bytes' memory. The second column stands for their bi-gram probability indices stored in the last 3 bits of the 2 bytes' memory. So to store one bi-gram requires only 2 bytes' memory (16bits).

The process of getting the initial list of the candidate Chinese characters and the list of bi-grams just described is the same as that discussed above. Since both lists are sorted in the order of their element values, they can be compared with each other to find the Chinese characters common to both lists.

FIG. 9 is a flow diagram for finding these Chinese characters that are common to both lists. In this figure: u_list[M] and b_list [N] are the arrays for storing the Chinese characters in the initial and bi-gram lists, respectively; M and N are the numbers of the elements in these two

lists, respectively; and chin_lists[N] is the array for storing the Chinese characters common to both lists.

In FIG. 9, step 900 is an initialization step. Index counters K1 and K and n are set to 1, 1 and 0 respectively. In step 901, a comparison is performed to determine whether the element of the uni-gram array u_list[K1] is the same as the element of the bi-gram array b_list[K2]. If the elements are the same, the process proceed to branch 910 where, following increment step 911, step 912 sets the first (or next) element of the array chin_list[N] to be equal to the element of the array b_list[K2]. Then in step 913, that element of the uni-gram array u_list[K1] is set to zero, in effect deleting the element from the uni-gram array. In step 914 the index counters K1 and K2 are incremented.

If, following step 901 the array elements are not equal, steps 920, 921 and 922 cause the index counter for the lesser of the two elements to be incremented. In this manner, step 901 is able to perform a comparison of all the elements in increasing value order until in step 930 the last element of both lists has undergone comparison. When this is achieved, step 931 gives a result in the form of a list (chin_list[N]) of the Chinese characters common to both the uni-gram and the bi-gram lists.

The maximum number of the comparisons required is M+N-1, where M and N are the numbers of the elements in these two lists, respectively. The Chinese characters in the resulting list chin_list[N] are then re-ordered based on their bi-gram probability indices and put into the top of the final candidate Chinese characters' list.

Take the same example as discussed before, these three Chinese characters would be listed in an order of A, B, C, because the indices of A (=7) is bigger than that of B (=6), and the indices of B (=6) is bigger than that of C (=5).

Before adaptation, the bi-grams are listed as follows:

C 5

A 7

B 6

If the user chooses character C as the one he wants to input, only the second column needs to be updated

C	7
A	6
B	5

After adaptation, the index of C is changed to 7, and the index of A and B are changed to 6 and 5, respectively.

A new method is now described for generating character bi-gram probabilities.

The conventional method for generating Chinese character bi-gram probabilities just blindly counts the number of the co-occurrence of two Chinese characters in a large Chinese text corpus. The bi-grams are then sorted in the order of co-occurrence frequencies. Only the top N bi-grams with the highest co-occurrence frequencies are selected by the system due to the limitation of memory locations used to store them.

In the new method, the process of generating Chinese character bi-grams is divided into three steps: generation of Chinese word uni-grams and bi-grams; generating the in-word and cross word bi-grams of Chinese characters; and generating the final bi-grams of Chinese characters by merging the in-word and the cross word bi-grams of Chinese characters.

A Chinese word is made of 1 to 6 Chinese characters. However, most of the Chinese words are made of two Chinese characters. For example, the Chinese word "people" contains two Chinese characters, "ren" and "min". A Chinese word dictionary is provided which contains 60,000 Chinese words. The Chinese sentences in the Chinese text corpus used for generating language models are first segmented into Chinese word sequences by using the Chinese word dictionary. The word uni-grams (the number of the word occurrences) and word bi-grams (the number of two word co-occurrences) are then calculated.

The Chinese word uni-grams are first segmented by the Chinese characters in the Chinese character dictionary. The numbers of two Chinese character co-occurrences are then counted and accumulated to

form the in-word bi-grams of Chinese characters. For a 60k word dictionary, there are approximately 50k in-word bi-grams of the Chinese characters. Obviously, the in-word bi-grams contain the most important information about all the Chinese words in the Chinese word dictionary.

The cross word bi-grams of Chinese characters are generated by only counting the co-occurrences of the last character in the first word and the first character in the second word. Experiments conducted show that the number of cross word bi-grams of Chinese characters generated is 1340k. However, the list is truncated after rank-ordering, so that only the 50k cross word bi-grams with the highest frequencies are selected to form the final list of bi-grams of Chinese characters. The cross word bi-grams contain the information about some new words that are not included in the Chinese word dictionary and the usages of some frequently used function words which usually contains only one Chinese character.

There are some overlaps between the in-word and cross word bi-grams of the Chinese characters. So the final bi-grams of the Chinese characters is generated by merging the in-word and the cross word bi-grams of the Chinese characters.

A method of input of data has been described which allows fast input of data from a limited set of keys or other input devices, such as the keys of a 12-key telephone keypad, and which is intuitive for the user and ergonomically efficient. Ambiguities in key input are resolved by language modeling. Uni-gram and bi-gram language models are described, but other models can be used. For example tri-gram and n-gram models can be used if memory is available or if greater efficiency is found to be achievable using such models. The method is efficient in processing terms and has an efficient data structure. These features make it fast in operation and low in cost, so that it can be used in simple cheap consumer devices. Other modifications of detail can be made by one skilled in the art without departing from the scope and spirit of the invention.

* C L M S *

What is claimed is:

THIS SHEET HAS BEEN REPLACED

Claims

1. A method of input of text into a device having key ambiguities comprising the steps of:

maintaining in memory a first table of digit streams representative of characters in a desired language and character addresses corresponding to the digit streams;

activating keys of a data entry device, where a key represents a plurality of characters, to provide an input digit stream; and

searching the first table in the memory for a digit stream corresponding to the input digit stream and obtaining a corresponding character address.

2. The method of claim 1 wherein the step of searching comprises performing a binary search.

3. A method of input of text into a device comprising the steps of:

maintaining in memory a first table of digit streams representative of characters in a desired language and character addresses corresponding to the digit streams;

maintaining in memory a second table of characters located at addresses corresponding to the character addresses in the first table;

activating keys of a keypad, where a key represents a plurality of characters to provide an input digit stream;

searching the first table in the memory for a digit stream corresponding to the input digit stream and obtaining a corresponding character address; and

reading from the second table at least one character located at the character address.

4. The method of claim 3 wherein the step of searching comprises performing a binary search.

This Sheet Has Been Replaced

* C L M S *

THIS SHEET HAS BEEN REPLACED

5. The method of claim 3 wherein the step of reading comprises locating a plurality of rank ordered characters from the second table located at addresses in predetermined locations relative to the character address.

6. The method of claim 5, further comprising resolving ambiguities between the characters of the plurality of characters using language model data.

7. A method of character identification comprising the steps of:

maintaining in memory a table of characters and rank ordered relative addresses of the characters;

selecting a character; and

reordering the relative addresses of the characters in the memory dependent on the character selected.

8. The method of claim 7 wherein the step of maintaining comprises:

storing at a plurality of consecutive addresses character identifications which correspond to a common digit stream when entered from a data entry device having entry ambiguities; and

storing at the consecutive addresses relative addresses of characters rank ordered in preference independent of the consecutive addresses at which the relative addresses are stored.

This Sheet Has Been Replaced

* C L M S *

THIS SHEET HAS BEEN REPLACED

9. A text entry device comprising: a keypad having a set of keys, each key representing a plurality of characters; a memory having stored therein a first table of digit streams representative of characters in a desired language and character addresses corresponding to the digit streams and a second table of characters located at addresses corresponding to the character addresses in the first table; an output device; and a processor coupled to the keypad, memory and output device,

the processor being programmed to receive a digit stream input corresponding to keys activated, to search the first table in the memory for a digit stream corresponding to the input digit stream, to obtain a corresponding character address, to read from the second table in the memory a character identification located at the character address and to provide to the output device a character corresponding to the character identification.

10. A text entry device comprising: a keypad having a set of keys, each key representing a plurality of characters; a memory having stored therein a table of characters and rank ordered relative addresses of the characters; an output device; and a processor coupled to the keypad, memory and output device,

the processor being programmed to receive a digit stream input corresponding to keys activated, to select from a plurality of characters from the table in the memory dependent on the digit stream input; to receive from the keypad a character selection input and to reorder the relative addresses of the characters in the memory dependent on the character selection input.

This Sheet Has Been Replaced

* C L M S *

11. A computer-readable memory having instructions and data stored therein for:

maintaining a first table of character addresses and digit sequences representative of characters in a desired language;

receiving an input digit sequence representing a plurality of characters;

searching the first table for a digit sequence corresponding to the input digit sequence; and

obtaining a corresponding character address.

12. A computer-readable memory having instructions and data stored therein for:

maintaining a table of characters and rank ordered relative addresses of the characters;

receiving a character selection input; and

reordering the relative addresses of the characters dependent on the character selection input.

THIS SHEET HAS BEEN REPLACED

This Sheet Has Been Replaced