SYSTEM SUPPORT IN CHINESE DATA ENTRY

Joseph E. Grimes
Cornell University, Ithaca NY, USA

Summary

Our aim is a viable software system to support language data processing involving non-alphabetic symbols, specifically the Chinese characters. The key to the system is the exploitation of certain linguistic relationships between pairs of those characters in sequence.

In the development vehicle for this system, independent modules of data are linked by pointers at two critical interfaces. The first links an input recognizer to a process that recognizes significant pairings. The second links both recognizers to a character generator.

Chinese typists are readily trained to end the input sequence that identifies the first character of a pair with a special delimiter for pairs rather than the usual delimiter. The pairs delimiter alerts the system to look up the pairing potential of the first character. Then it matches the second character of the pair against that potential. The result is automatic contextual disambiguation performed on input codes that otherwise might not identify characters uniquely.

Overview

Up to now, devices for typing Chinese characters or for entering them into computers have not been widely successful. The problem is not one of storing character shapes in a computer or of reproducing them once they are stored; it is rather a problem of designating quickly which of a large number of accessible shapes should be reproduced.

The companion paper by Paul King on human factors and linguistic considerations shows that a satisfactory solution can be worked out by paying attention not just to the graphic shapes involved, but to the characteristics of the Chinese language that stand behind those graphic shapes and their combinations.

The solution has three components. First, it is possible to use nonunique strings of key strokes to identify the shape of a character; that is, many of the identifiers in King's Cornell Code identify two or more characters. Second, since often in Chinese it is a two-character sequence that is significant rather than the individual characters that make it up, access to information about such pairings makes it possible to eliminate, or at least reduce, the ambiguity inherent in the use of nonunique identifiers. Third, in the residual cases where it is still not clear which of several characters or character pairs is intended, it has proved adequate to display the possibilities and interrupt the operator to ask her to indicate which one she wants by typing the number of that one on the screen before her.

The results of this approach to Chinese data entry are encouraging. Speakers of Chinese with a middle school education or better learn the keyboard with about half an hour's instruction. After a two week training session or its equivalent, the median speed is around 40 characters per minute and the best speeds are above 50. The error rate by that time approaches zero, and residual errors are correctible by means of a cursor editor that is built into the system. Operators can type for several hours at a time without fatigue.

The software system that makes this behavior possible is not particularly complex. It derives its power from the amount of information about the Chinese language that it holds in compact form in its internal store: specifically, information about the most likely character pairings in Chinese.

Requirements

To create this system we began with two overall requirements. The first was that it be modular, so that any component of it could be worked on without disturbing the other components, and so that it could be implemented on a variety of physical configurations. Each of the stored data structures is manipulated not only by the main data entry program, but also by utility programs that make it possible to add new characters to the repertoire (for example, to put together a specialized vocabulary for a particular application), or to augment the number of pairings that the system recognizes. Modularity also makes it possible to change output character fonts as desired.

The second requirement was, of course, that all functions of the system be accomplished at a speed that would permit typists to achieve their best performance without any limit being imposed on them. In the disk oriented prototype, this requirement involved paying special attention to minimizing disk accesses when searching through chains of pointers.

The prototype embodies a third requirement that production versions will not have to meet: each typist's performance needs to be logged in a way that slows nothing down. An internal clock puts times and character codes into a buffer that is written to a log file from time to time. A utility program later compares the log with a stored version of what was to be typed. From that comparison it determines the error rate. From the timing information it determines the typing rate. The results of this utility are made available to another that plots the performance of all typists being tested as a function of time.

Stored information

There are two kinds of information that the system uses in deciding which character the typist wants: a file of identifiers, and a file of pairings. The identifier file is used to find a direct match to strings typed in from the keyboard. A successful match against the identifier file yields one or more Chinese telegraph codes (four-digit numbers) that represent all the characters that that identifier could stand for. The file of pairings, on the other hand, tells what other characters could follow a given character in a close-knit relationship to it when it comes first in a pair.

The identifier file is organized as a B*-tree with variable length entries. This type of structure keeps the number of disk accesses needed to match an identifier near the theoretical minimum, with the result that it can be traversed very rapidly. It has the additional advantage that it can be implemented in such a way that the disk blocks nearest the root are kept in a core buffer area, thereby eliminating outright some of the disk accesses that would be needed in a full search. Since the tree is formed at the time when new identifiers are being introduced into the system by a utility program, the complex steps needed to keep the B*-tree in balance are performed only at a time when speed is not a factor.

The result of a match between an identifier and the B*-tree is a string of one or more telegraph code numbers. These are the same four-digit numbers that have been used for years to transmit Chinese characters over telegraph lines. They are defined by a standard code book. The string of one or more telegraph codes that comes from the identifier file represents all the Chinese characters whose shape matches the identifier string. This string of codes is held in an internal buffer, where later stages of the process work on it.

The file of pairings is the heart of the system and the focal point of the patent that has been filed on it. It consists of two parts: index and contents. The index is derived by applying an arithmetic function to the telegraph code that is desired, yielding a fixed offset from the beginning of the file. At that offset is stored an internal pointer to where the contents begin. This makes possible a rapid second access to a nearby location on the disk. The contents, stored at the second location, are variable in length; they are a string of telegraph codes that identify all the characters that are known to pair with the character whose telegraph code forms the original search argument; that is, all the characters whose sequential relationship to the first character is significant in Chinese.

The system also stores graphic information that defines character shapes for display on the screen and for printing. The shape information is indexed just as it is in the file of pairings, using the telegraph code of each character as a pointer. The same algorithm that is used for the file of pairings converts the telegraph code to a fixed disk offset, and at the position on the disk that is so indicated, the information is found that tells where the actual graphic information begins later in the same file.

The form of the graphic information or its display is of no direct concern to the selection logic; it can be treated as a cluster of a data structure that is pointed to, together with the processes necessary to handle it. We have implemented both vector and raster displays, and within each mode any available font can be called up simply by naming the appropriate character file, since all are accessed through the same pointer structure.

Processes

Three major processes select the Chinese character the typist wants. The first recognizes the identifier string that is typed in, using the B*-tree structure of the identifier file. The second process is invoked whenever a character pair is typed in. It recognizes all pairings that match both identifiers in the pair in order. The third process is invoked only if more than one possible result remains after the first two processes have finished; it gets the typist's attention and asks her to make a decision.

The process that recognizes an identifier finds an exact match in a B*-tree of known identifiers. If no match is possible it gets the operator's attention: either the identifier was typed wrong, or it is not yet in the system. In either case something else needs to be typed.

When a pair is typed in, the string up to the special delimiter for pairs is taken as the first identifier and matched separately from the string for the second identifier, which comes between the special delimiter and the final delimiter. For each identifier, the identifier file yields a string of four-digit telegraph codes, one for each of the Chinese characters that that identifier can represent. In about one case out of nine, there is only one telegraph code for an identifier. Frequently, however, there are two; and the number sometimes goes as high as fifteen telegraph codes for one identifier.

When a pair of characters is typed in using the special delimiter to separate them, the presence of that delimiter activates the second process that recognizes known pairings. This process goes through the string of telegraph codes that correspond to the first identifier. For each telegraph code in that string, it looks in the file of possible pairings to see what other codes might form possible pairings with the first one. The process then goes through the telegraph codes that correspond to the second identifier to see if any of them actually does form a pairing with the first. If one does, that pair of telegraph codes is copied into a special array which the third process uses for its final selection.

If on the other hand the pairing the typist reacts to is not yet in the file of pairings, the third process is automatically applied to . the first character of the pair so that it can be disambiguated manually, then to the second member of the pair separately for the same process.

If the process that recognizes identifiers finds only one telegraph code for an identifier, and there is no pairing, then that telegraph code is accepted as the correct representation for the character the typist intended. If there is a pairing, but after the second process is over only one pair of telegraph codes has been found, those two can be taken as the characters the typist intended.

Sometimes, however, an unpaired character identifier corresponds to two or more telegraph codes, and the decision as to which code is intended has to be made by the typist. Much more rarely, two paired identifiers match more than one pair of telegraph codes in the second process, and the decision as to which pair of characters is intended has to be made by the typist. It is also possible for a legitimate pairing to not yet be in the file of pairings, so that each of the two characters the typist typed has to be presented separately for a decision.

In all these cases the third process interrupts the typist by emitting an audible signal to indicate that she needs to divert her attention from what she is typing to the question posed on the screen in front of her. The screen displays the entire list of possibilities, either single characters or pairs of characters. She in return types a number to tell the process which one of them she wants: "2" for the second one displayed, "5" for the fifth, and so forth.

All the telegraph codes that are found by the three processes of matching identifiers, resolving pairs, and deciding among alternatives go into an output buffer in the form of character strings that represent the four-digit telegraph codes. The contents of this buffer are available for any one of several subsequent processes.

The first process that operates on the output string of telegraph codes is an editor that allows a cursor to be moved through the string to the place where some change is to be made. The editor then allows deletions and insertions to be made wherever the cursor is located. All display on the screen, of course, is in the form of the

Chinese characters that correspond to the stored telegraph codes; the codes themselves are never seen by the typist.

The edited string can be stored on a magnetic medium, sent over a communications line, or formatted vertically or horizontally to be sent to a printing device. All these processes are conventional. They give the Chinese data entry system the potential of being used as a computer terminal, a communications terminal, an off line data entry device, or even a simple office typewriter.

The work on which this paper was based received support from the NCR Corporation.