Def Sci J, Vol 38, No. 2, April 1988, pp 105-117

Computer Recognition Method as Applied to Codification Process for Inventory System of a Large Multidisciplined Laboratory

R. Swaminathan, R.N. Biswas and B.N. Gajbhiye

Defence Research & Development Laboratory, Hyderabad-500 258

ABSTRACT

A variety of algorithms are existing to retrieve the information by means of furnished keys[1]. As suggested by Nakatsu N. *et al*[2]. algorithm for the LCS problem can be used to find string which contain some given words or words similar to them. Some such LCS algorithms with their time responses are discussed[2-6] This paper also describes an effective method of recognizing the information. The method involves two way search giving the weightages to each. Finally the two values are combined to produce a match factor for recognition of information. Its application to codification in inventory system is discussed.

1. INTRODUCTION

In Defence, particularly in a multidiscipline research organization like DRDL, a wide variety of stores are required for the R & D efforts. Because of the largeness of size, the traditional method of ledger system was found to be inadequate to meet the requirement. As a solution, a codified ledger accounting system was started with the ultimate objective of computerising the whole process. In the modified method codification of an item is a key function. Assigning an unambiguous identification, i.e. item code is called codification. The process was being carried out manually for a long time. Manual search of long lists is always a tedious, time consuming and fault prone activity. The problem is further compounded by the different standards of nomenclature and spellings. With a view to improve the situation, a computer aided search and codification method is presented here.

Received 20 July 1987

In the computerised codification process already codified items are stored in a file. The file is known as item-master or item-catalog. The user of a file approaches the item catalog in search of an item. Let us call the item sought by a user a target. When he approaches the item catalog in search of a target, the user has in his possession something which is called a key, which he hopes, will enable him, to recognize the target. The key could be an item code or a nomenclature of the item or some other attribute like consumable/nonconsumable code, unit of measure etc. Assume that the item-catalog is organized with item code or name of an item as a key. Then there is no problem to access the item. But situation is different when item exists in an item-master as *Battery Hydrometer* and by any chance user enters a spelling wrongly, and approaches the catalog with *Battery Hyderometer* or there may be an alteration in the words of the item name as *Hydrometer Battery* or a combination of both stated above *Hyderometer Battery*

In all above situations, search action misfires the target.

To overcome such situations, the method presented uses both character and word scanning. With appropriate weightage scheme to the two scanning processes a mismatch factor is computed. The catalog is sifted on the basis of minimum value of the factor. Thus a target is hit for a null mismatch weight, whereas a minimum mismatch factor gives the nearest match.

2. METHOD OF SOLUTION

A target is said to be hit successfully, if and only if, the number of characters as well as the words of the target sentence matches with those of an existing sentence.

Let T denote a target sentence for which the catalog is to be searched such that

$T = (TW_1, TW_2, TW_3 \ldots\ldots\ldots\ldots TW_2$, wherer $TW_1, TW_2, TW_3 \ldots\ldots\ldots\ldots TW_i$ are the first, second, third $\ldots\ldots\ldots\ldots i^{th}$ words separated by spaces. Again a word consists of a finite string (or sequence) of characters (or symbols).

$$TW_1 = (ABACDB5$$

Similarly, let the string which is to be matched with the target be denoted by

$$E = (EW_1\ EW_2\ EW_3 \ldots\ldots\ldots\ EW_k)$$

where $EW_1, EW_2, EW_3, \ldots\ldots\ldots Ew_k$ are the words of the existing sentence in the file. Words consists of characters like

$$EW_1 = (BACABZ$$

TW_i and EW_k's are composed of characters from the set

$$U = (A, B, \quad\quad Z, 0, 1, 2, \quad\quad 9)$$

2.1 Scan 1 : Character Count Matching

In this approach (Fig. 1) at first the number of occurrences of each character of the target sentence is computed. Thus in TW_1, the character A occurs twice, B twice, C, D and 5 only once each. Similarly, occurrences are counted in an existing string EW_1. In this, the character A occurs twice, B twice, C and Z once each, and D does not appear.

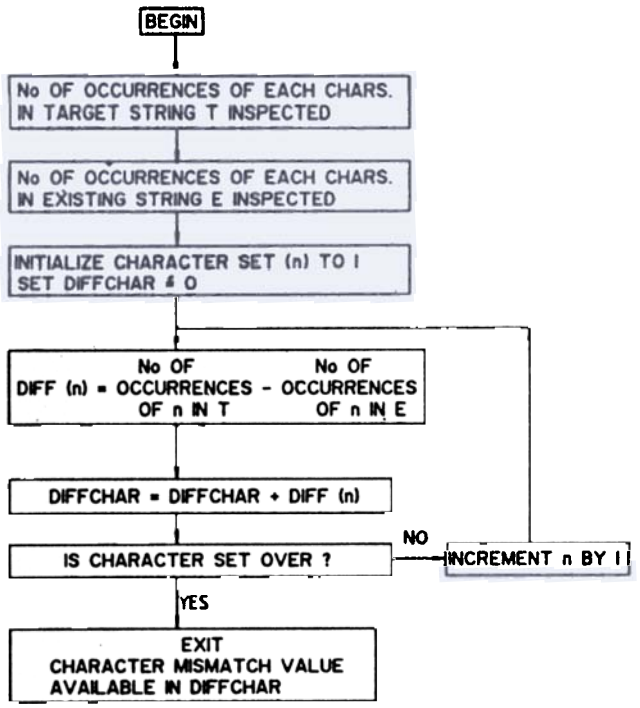

Figure 1. Flow chart of character matching approach

Difference of each type of characters then can be obtained by taking the magnitude of difference of occurrences of the same characters in both the strings. Thus

$$DIFF(A) = \text{modulus of}$$
$$(\text{No. of occurrences of } A \text{ in } T - \text{No. of occurrences of } A \text{ in } E)$$

The magnitude of $DIFF(A)$ gives the difference or let us call mismatch value of character A. Likewise mismatch values for all characters are calculated. Sum of these gives the total mismatch value.

$$DIFFCHAR = (DIFF(A) + DIFF(B) + \ldots\ldots)$$

For null value of DIFFCHAR, both strings may not necessarily match as illustrated in the following example :

Example :

Consider a target string, denoted by T

$$T = POT$$

is to be searched. This consists of characters P, O, T. Occurrences of each of the character is only once each.

Let us assume, the existing string E as

$$E = TOP \quad \text{or}$$

$$E = OPT$$

In above, both the strings consists of characters P, O, and T. Occurrences of these also is only once each in both the strings.

Therefore,

$$DIFF(P) = 0$$

$$DIFF(O) = 0$$

and $$DIFF(T) = 0$$

resulting a character mismatch value

$$DIFFCHAR = 0$$

Therefore, even though the number of occurrences of characters tally the strings do not match. This shows that the order of occurrences of the characters is equally important and cannot be ignored. This fact brings us to the other approach discussed below.

2.2 Scan 2 : Word Matching

In this method (Fig. 2) the target sentence is unstringed into a number of words TW_1, TW_2, TW_3,..... TW_i. These words are stored in the memory for the period of the scan. Similarly the existing sentence is unstringed into a number of words EW_1, EW_2, EW_3,..... EW_k. Then, the first word of the target sentence TW_1 is matched with each word of the existing sentence. For a tallied word, the counter WORD–TALLY is incremented by 1. Thus, all the words of the target string are tallied. The total number of words tallied gives the matching value for both strings. This is complemented with the word count of the target sentence to give another mismatch value.

2.3 Computing Order for Placement in the Table

The total mismatch value of the sentences, called *ORDER*, is given by

$$ORDER = \text{Characters mismatch value} + \text{Words mismatch value}$$

The first approach gives a character mismatch value, while the second approach gives a word match value at first. In order to compute *ORDER*, the word mismatch value is calculated.

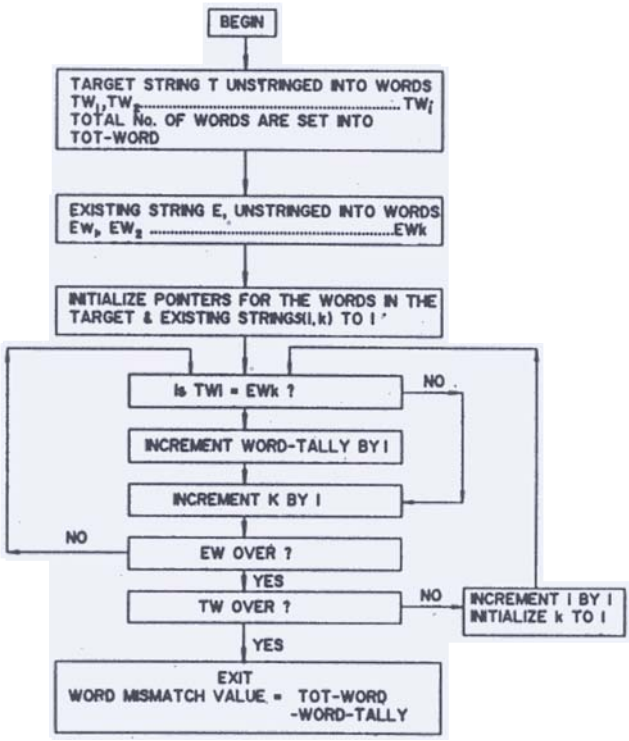

Figure 2. Flow chart for word matching approach.

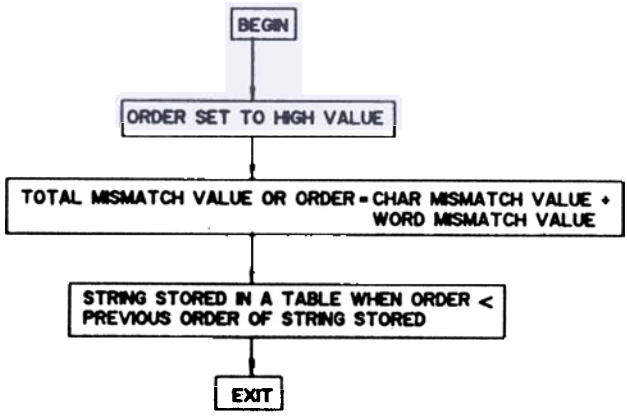

Figure 3. Flow chart for string placement in table.

$$\text{Word mismatch value} = \text{Tot-word} - \text{Word-tally}$$

In our implementation a table of 10 strings is maintained in ascending order (Fig. 3) of their total mismatch value (*ORDER*). The number of strings to be maintained in a particular situation is a choice of the implementor. However, a minimum of two strings should be maintained to cater for the uncertainty and duplication factor.

3. PROGRAM DESCRIPTION

In Defence Research and Development Laboratory (DRDL), Hyderabad, this algorithm has been implemented for the process of assigning an identification code to the store item. In this laboratory a seven digit code policy with 8th character as the check character has been adopted.

In the computerised codification procedure, the item-catalog is arranged in an indexed sequential file with an itemcode as the key.

The process of codification is completed in five stages :

- Selecting series and group code
- Scanning of the catalog
- Display of first ten best match
- Deciding an itemcode
- Insertion of new item code

3.1 Stage 1 : Selecting Series and Group Code

The stores in DRDL are arranged in ten series. The first character of the item code represent a series. At the start of the program the menu of the ten series is displayed on the screen and the user is prompted to select one of them.

Once a user selects a series the menu of the three digits group code for the series is displayed on the screen. Adequate facility like going to the next page, previous page, beginning etc. has been provided for manual selection of a group code. At the end of the selection the user responds to the prompt

3.2 Stage 2 : Scanning of the Catalog

The selection of the group code reduces the size of the scanning area. One would not like, unless it is an exceptional case to read and process the complete length of the file.

In the matching process, the pointer is set to the first end of the group code area, as shown in Fig. 4. Matching begins at the first end and at the second end it terminates.

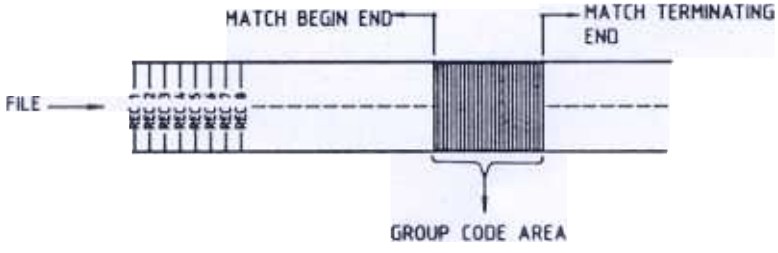

Figure 4. Group code area in the file.

3.3 Stage 3 : Display of First Ten Best Match

With the computation of *ORDER*, a table of ten items is maintained in the ascending order of total mismatch value. The table appears on the screen, when scanning is over. Exact matched item will top the table with zero mismatch value. Item• code of a target nomenclature and details like unit of measure and consumable/nonconsumable also appear.

3.4 Stage 4 : Deciding an Item Code

The table of ten best matched item displayed on the screen complete with other associated information provides enough clue for a decision if the target item exists in the catalog or not.

For the non-existing item also, the table serves as a guide for the purpose of code allotment as the items in the table being the nearest matches, the item code of the target item should be closest to the most appropriate one in the table. The user has to decide which one of the ten is closest to the target item. After a decision a limited search facility is provided for locating a vacant code in the chosen zone.

3.5 Stage 5 : Insertion of New Item Code

The stage 4 above helps to locate a suitable vacant code for the target item when it is not already existing in the catalog. Once such a vacant code is located the computer prompts

Do you want to insert (*Y/N*)?

If response is '*Y*' the item is entered into the catalog with all details available.

4. COMPUTER IMPLEMENTATION

4.1 Program Characteristics

Program discussed has been coded in COBOL 74 language and implemented on HONEYWELL BULL DPS 8/46 system for DRDL inventory system. Some of the characteristics of the system are

- Total No. of the items in the file are 93000
- Indexed sequential file organization

The file management has been preferred in favour of indexed sequential organization for facilitating insertion of new codes in the interactive mode

- Record size 150 bytes
- Restricted permission

The read access to the file is unrestricted but write permission is restricted only to authorised users for an effective control of the codification process

- Set of 36 alphanumeric characters, consisting of 26 alphabets and 10 numerals. Special characters, like /, – % etc. can be included in the set depending on the requirement of an application

4.2 Response Time

Response time depends on the item population in the group code area and system environment at the time of access. It is observed that when system was running on GCOS 3 operating system, average response time was about 2 minutes. However, it has gone up by 3 minutes on GCOS 8 operating system and after connecting RJE's. Table 1 to 4 give the summary of response times against various item populations. For the equal no. of item population, response time varies. This could be attributed to the change in system environment of the time sharing system.

Table 1. Recognition display for target 'battery hydrometer'

Item code	Unit of measure	*C/NC*	Order	Nomenclature
7360903F	No	*C*	0000	Battery Hydrometer
7369903G	No	*C*	0000	Battery Hydrometer
7360100W	No	*C*	0009	Battery Evereday
7369503Y	No	*C*	0012	Battery Leads
7369506S	No	*C*	0012	Grids Battery
7369920G	No	*C*	0012	Battery 3e-rat 6V
7360150F	No	*C*	0013	Battery Dry 1.5V 1R 20
7368305J	No	*C*	0013	Battery Eliminator 9V
7369431X	No	*C*	0013	Battery Thermal Drg No 956-121
7369913D	No	*C*	0013	Battery 6V

Item code ? *<Cr>*

Group code ? 736

Name ? Battery Hyderometer

UM ? No

CNC ? *C*

4.3 Improvement to the System

From the table of response time, it is seen that there is an urgent need for improvement in this area to make the system viable.

In the present algorithm, the character count of each existing item is calculated everytime the item is scanned. Instead, if the counts are maintained in the appropriate record. the calculation time may be saved and response time will improve.

Main contribution to delay being from the environment of the mainframe computer at the time of operation, the effect of environment may be eliminated if

the actual processing is done on a PC which is connected to a central mainframe computer for the purpose of integrity of the master file.

Since the size of the file is very big the access time even in random access mode is considerable. This can be improved by suitably partitioning the master file.

Table 2. Recognition table for target 'battery hyderometer'

Item code	Unit of measure	*C/NC*	Order	Nomenclature
7360903F	No	*C*	0002	Battery Hydrometer
7369903G	No	*C*	0002	Battery Hydrometer
7360100W	No	*C*	0008	Battery Evereday
7360910J	No	*C*	0013	Battery Evereday 266 9V
7369503Y	No	*C*	0013	Battery Leads
7369506S	No	*C*	0013	Grids Battery
7369920G	No	*C*	0013	Battery 3e-rat 6V
7360150F	No	*C*	0014	Battery Dry 1.5V IR 20
7368305J	No	*C*	0014	Battery Eliminator 9V
7369431X	No	*C*	0014	Battery Thermal Drg No 956-121

Item code ? <Cr>
Group code ? 736
Name ? Hyderometer Battery
UM ? No
CNC ? *C*

Table 3. Recognition table for target 'Hyderometer battery'

Item code	Unit of measure	*C/NC*	Order	Nomenclature
7360903F	No	C	0002	Battery Hydrometer
7369903G	No	C	0002	Battery Hydrometer
7360100W	No	C	0008	Battery Evereday
7360910J	No	C	0013	Battery Evereday 266 9V
7369503Y	No	C	0013	Battery Leads
7369506S	No	C	0013	Grids Battery
7369920G	No	C	0013	Battery 3e-rat 6V
7360150F	No	C	0014	Battery Dry 1.5V IR 20
7368305J	No	C	0014	Battery Eliminator 9V
7369431X	No	C	0014	Battery Thermal Drg No 956-121

Item code ? 7360903f
Item code already alloted to
Name . . Battery Hyderometer
UM . . No
C/N . . *C*

Item code ? 7360910
Vacant, Do you want to insert (*Y/N*) ? *Y*
Item code allotted.
Group code ?*

Table 4. Response time against item population

Group code	Item population	Response time Minutes	Seconds
885	43	01	19.91
321	164	02	04.45
511	246	04	13.78
736	320	04	26.54
736	320	02	53.04
736	320	02	16.25
925	372	06	21.57
714	451	10	17.06
944	498	05	09.25
827	656	04	11.80
801	957	06	43.51
801	957	09	46.15
888	1189	15	28.62

5. RUNNING SESSION

Running session illustrates prompt and sample responses during a search process

Computer aided codification

Do you want to know what this system does (*Y/N*)? *N*

Series	Type of stores
	General engineering stores.
	Suspense items
3	Tools and apparatus
4	Raw material and hardware
5	Woods, glassware, optical instruments and furniture items

6	Chemicals, explosions
7	Electrical
8	Electronic
9	General items
A	Machinery and accessories

Enter series ? 7

Group code and key words information Page : 1

Group code	Key words	Group code	Key words
705	Connectors, Probe PCB	708	Switches, Switch Board, Junction Box, Distributed Board
710	Sockets	711	Plugs
712	Fuses, Fuse carriers, Neutral Link, Endcovers	714	Soldering Accessories
716	Sleevings, Insulations, Dielectric materials, Plastic ferrules	718	Wires, copper conductor

Enter page (*N* = Next, *P* = Previous, *B* = Begin, *Q* = Quit) ? *N*

Group code and word information Page : 2

Group code	Key words	Group code	Key words
721	Cables, Cable Glands, Flexible Cards, Choppers	729	Miscellaneous Hardware Microwave component and Adaptors
730	Chokes	731	Transformers, Variacs
733	Relays	734	Lamps, Bulbs
736	Cells, Batteries, Power pack	738	Strain Gauges, Load Indicator
740	Transducers, Accelerometer, Sensitive Calibrating Comparator	742	Servo Components
744	Rheostat	746	Decade Boxes, Bridges, Distance measuring unit

Enter page (*N* = Next, *P* = Previous, *B* = Begin, *Q* = Quit) ? *N*

Group code and key words information Page : 3

Group code	Key words	Group code	Key words
759	Miscellaneous Components (Thermocouple, Cores Crystals, Conformers, Filters, Ring Cores, Toroids Ceramic Rod, Ceramic Commutator Anode Coils, Magnetic Exciter Ferrite Components	760	Meters
762	Testers	763	Motors, Rotor
765	Generators Vibrators, Shakers	767	Cooling Equipments, Air Regulator, Air Circulator
769	Heating Devices	771	Actuators
773	Gyro Instruments	775	Servo Instruments
780	Environmental Testing	799	Miscellaneous Equipments

Enter page (*N* = Next, *P* = Previous, *B* = Begin, *Q* = Quit) ? *Q*

Group code ? 736
Name ? Battery Hyderometer
UM ? No
CNC ? *C*

6. CONCLUSION

We have presented a method which makes possible, for an existing string, finding a correct or nearest match of a given misspelled or altered-word string. We have explained, its application to codification in the inventory system.

Following are the advantages of this method in this application :

a) Revision of coding policy with the help of this algorithm.

b) Managing with a simplified and rugged codification policy.

Method can also be applied to similar areas like decision monitoring system, library information system etc.

ACKNOWLEDGEMENTS

We wish to thank to the Director, Defence Research & Development Laboratory, (DRDL) Hyderabad, for his constant encouragement in the development of this system. Thanks are due to DRDO computer centre, Hyderabad for the computer facilities and also Material management group officers and staff for their valuable co-operation.

BIBLIOGRAPHY

1. Morrison, Donald R., *J. ACM,* **15** (1968), 514-534.
2. Nakatsu, N., Kambayashi, Y. & Yajima, S., *Acta Informatica,* **18** (1982), 171-179.
3. Aho, A.V., Hirschberg, D.S. Ullman, J.D., *J. ACM,* **23** (1976), 1-12.
4. Hunt, J.W. & Szymanski, T.G., *C. ACM,* **20** (1977), 350-353.
5. Hirschberg, D.S., *J. ACM,* **24** (1977), 664-675.
6. Mukhopadhyay, A., *Inform. Sci.,* **20**(1980), 69-82.