Data Recovery Based on Intelligent Pattern Matching

JunKai Yi, Shuo Tang, and Hui Li

College of Information Science and Technology,
Beijing University of Chemical Technology, China
yijk@mail.buct.edu.cn, tangshuo2005@126.com,
leehuui@163.com

Abstract. To solve the problem of data recovery on free disk sectors, an approach of data recovering based on intelligent pattern matching is proposed in this paper. Different from the methods based on the file directory, this approach utilizes the consistency among the data on the disk. A feature pattern library is established based on different types of files according to the internal constructions of text. Data on sectors will be classified automatically by data clustering and evaluating. When the conflict happens on data classification, the digestion will be initiated by adopting context pattern. Based on this approach, the paper achieved the data recovery system aiming at pattern matching of txt, word and pdf files. Raw and formatting recovery tests proved that the system works well.

Keywords: Data recovery, Fuzzy matching, Bayesian.

1 Introduction

Computer data missing often occurs by personal mistakes or incidental reasons. Sometimes the data are too precious to be evaluated by money. So data recovery is very important. Currently, there are many kinds of good and useful data recovery software, most of which are developed based on file directory [1] which could not make full use of data on free sectors. So the data could be missed. This disadvantage is also utilized by criminals to make anti-restore data [2] so that neither data could be collected as evidence nor valuable clues could be found.

By taking advantage of the data on free sectors, this paper is proposing a data recovery method based on intelligent pattern matching, aiming to restore text files, such as *txt*, *doc* and *pdf* files. Firstly, a binary feature pattern library is established for different file categories by analyzing their internal format [3]. Secondly, in order to determine which kind of file they may belong to, data on sectors are classified by clustering [4] and evaluating automatically and the types of the files are identified. Here, each sector is a unit. When conflict of sector data classification happens, it will digest with the reference of the context of the sector and the encoding pattern [5]. Finally, data are organized into different files and recovered according to the data feature pattern library.

2 Data Recovery

Data recovery is to restore data which are lost or damaged by hardware-disable, incorrect operation and/or other reasons, in other words, is to restore them back to its original state. For most of cases, it is able to be restored as long as the data isn't covered. If the sector can be read and written normally, data recovery can be divided into three classes. They are respectively base on file directory, file data character and incomplete data. Functionally, data recovery can be classified into deletion recovery, format recovery and Raw recovery. Deletion recovery means to find and recover deleted file; Format recovery means to recover files on formatted disk; Raw recovery means to restore files ignoring any file information system.

3 Specific File Structure and Feature Pattern Library

3.1 Specific File Structure

Each specific file has its own format. File format is a special encoding pattern of information used for the computer to store and identify information [6]. For instance, it can be used to store pictures, procedures, and text messages. Meanwhile, each type of information can be stored in the computer by one or more file formats. Each file format usually has one or more extension names for identification or no extension name in some cases. File structures are defined as follows:

<*file*>|<*code*>{<*header*> : <*body*> : <*trailer*>}

For instance : <*txt*> <*Unicode* | *UTF*> {<*0xFFFE*> : <*body*> : <>}

file: file type; *code*: encoding pattern; *header*: file head; *body*: file content; *trailer*: file tail.

(1) Word Document Structure

Word file's structure is more complicated than the *txt* file's. It is made up of several virtual streams including Header, Data, Fat Sectors, MiniFat Sectors and DIF Sectors. Word pattern is as follows:

<*word*><*Unicode*>{<*header*><*stream1*, *stream2*…><*trailer*>}

(2) PDF Structure

Generally speaking, a *PDF* file can be divided into four parts. The first is file header, in the first line of the PDF file, specifies the version number of a *PDF* specification that the file obeys. The second is file body, the main part of *PDF* files, which is formed by a series of objects. The third is cross-reference table, an address index table of indirect object, which is used to realize the random access to indirect objects. The last is file tail, which declares the address of the cross-reference table, points out the file catalog, so that the location of each object body in *PDF* file can be found and random access can be achieved. It also stores encryption and other security information of the *PDF* file. *PDF* pattern is as follows:

<pdf>{<Header><Body><xref table><trailer>}

3.2 The Definition of Feature Pattern Library

Feature pattern is seen as an ordered sequence composed of items and each item corresponds to a set of binary sequences [7]. During the pattern matching, items can be divided into three types according to the role they play. They are feature item P, data item D, optional item H.

1) Feature items P: To identify common features of different files, such as the feature item of A *Word* file always begins with 0xD0CF11E0.
2) Data items D: To show the body of the file.
3) Optional items H: the data used to fulfill the integrity of file.

3.3 Pattern Library Generation

The processes of pattern library generation are listed as follow steps. Firstly, compare different files with the same type and generate candidate pattern set; Secondly, apply it to the procedure of training data recovery; Thirdly, compare the recovery result with the original file in order to evaluate the candidate patterns and then screen out patterns which meet the requirements; Lastly, the pattern library of this type of file is achieved. There are three files provided, they are 1.doc, 2.doc and 3.doc. Three patterns can be obtained after binary comparison with each other. The three patterns are E_1, E_2, *and E3*.

$$E_1 = P_1 H_1 P_2 D_1 H_2 \ \ ....D_n P_n E_n$$
$$E_2 = P_1 H_1 P_2 D_1 H_2 \ \ ....D_n P_n E_n$$
$$E_3 = P_1 H_1 P_2 D_1 H_2 \ \ ....D_n P_n E_n$$

3.4 Cluster Analysis of Pattern

(1) Pattern Similarity Calculation

The pattern generated by existing files is an ordered sequence of items which are made up of binary sequences. With pattern E_1, E_2, the definition of similarity is

$Sim\left(E_i, E_j\right)$: $Sim\left(E_i, E_j\right) = \max(Score(Comm(E_i, E_j)))$. In the definition, $Comm(E_i, E_j)$ is the common subsequence of E_i and E_j . $Score(Comm(E_i, E_j))$ is the score of the common subsequence.

(2) The Definition of the Common Subsequence Score

There are two given sequences $A=\{a_1, a_2, a_3.....a_n\}$ and $B=\{b_1, b_2, b_3......b_n\}$. If there exist two monotone increasing sequences of integers $i_1< i_2< i_3......< i_n$ and $j_1< j_2<$

j_3......< j_n satisfying $a_{ik}=b_{jk}=c_k$ (k=1,2.......), then we can call $C=\{c_1, c_2, c_3......c_n\}$ is the common subsequence of A and B, and C can be denoted by symbol $Comm(A,B)$.

The expression of common subsequence score defines as follow:

$$Score(Comm(E_i, E_j)) = \frac{\sum_{i,j=1}^{n} Num(Comm(E_i, E_j))}{|E_i| + |E_j| - Num(Comm(E_i, E_j))}$$

Among them, $Num(Comm(E_i, E_j))$ denotes the account of items contained in the $Comm(E_i, E_j)$.

(3) Set the Similarity Threshold

According to threshold, patterns are classified and form the pattern library of corresponding documentation. For instance, *txt* pattern library: $\{E_1, E_2\}$, $E_1= D_1$ (When *txt* is stored in *ASCII*, store the file body directly. P= { }); $E_2= P_1\ D_1$(When txt is stored in UNICODE or UTF8, the file begins with 0xFFFE, that is, P_1 = 0xFFFE);

(4) The Classification of Sector Data

The current file system is distributed by clusters and a cluster as the smallest units; moreover the cluster is composed by many sectors. Typically, the size of a sector is 512 bytes. However, in order to exclude the affection of file systems, this system recovers files in sectors. Furthermore, since most files are not stored continuously, it is necessary to match the data on sector with the feature pattern library one by one in order to determine the file type stored inside.

If the data A of a sector is given, to determine which type of document the sector data belongs to and make $P(A/S)$ maximum, then $\hat{S} = \arg\max_s P(A/S)$

According to Bayesian formula

$\widehat{S} = \arg\max_S \frac{P(S)P(A/S)}{P(A)}$, $P(A)$ is constant when A is given, therefore,

$\widehat{S} = \arg\max_s P(A/S)P(S)$.

According to the result, the ones with the maximum probability can be classified into certain kind of file. However, this method does not include the match of data items D, because data items are abstracted from file body, while the body of the document is uncertain, so it will not be able to measure the matching degree. Consequently, with regard to the process of data item, the main idea is to determine its property determine its properties by checking its encoding mode and the context of its neighbor sectors. So, $\widehat{S}_n = \arg\max\{P(S_{n-1}), P(S_n), P(S_{n+1})\}$.

(5) Pattern Evaluation

After comparing the result of data recovery with standard document, we can divide files into successfully restored ones and unsuccessfully restored ones according to the result matched with pattern E. So, we can calculate the credibility of selected pattern E [8]:

$$R(E) = \frac{Corr(E)}{Corr(E) + Err(E)}$$

Among them, $R(E)$ is the credibility of the selected pattern *E*, $Corr(E)$ is the number of files which are successful recovered by selected pattern *E*, and $Err(E)$ is the number of files which are unsuccessful recovered. According to the result, patterns are sequenced and the one with higher credibility will get priority.

4 Recovery Process

4.1 Recovery Process

By analyzing the internal structure of documents, a data recovery method based on pattern matching is proposed. It combines feature pattern of files with data association. (Figure 1)

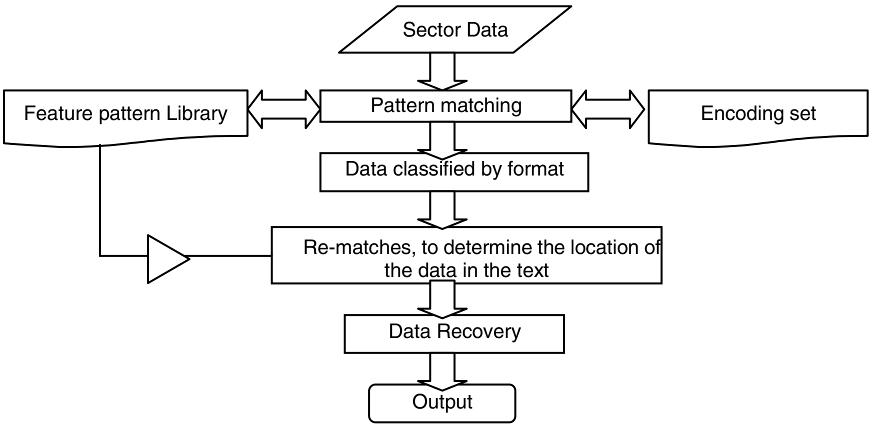
Sector Data
Feature pattern Library
Pattern matching
Encoding set
Data classified by format
Re-matches, to determine the location of the data in the text
Data Recovery
Output

Fig. 1. Data recovery flow chart

4.2 Solving Data Conflict

Data conflicts are mainly caused by the fact that there is more than one file with the same type on hard disk. The conflicts have two kinds. One is the data which has almost the same similarity with pattern matching and the other is the data which cannot match with pattern matching.

For data conflicts, an approach based on context pattern is adopted [8]. The context pattern is seen as an ordered sequence which is composed of neighbor sectors where the data is stored, i.e., $W_{-n}W_{-n-1}...W_{-2}\ W_{-1}$<*PN*> $W_1\ W_2...W_n$.

<*PN*> represents data conflict, W refers to context data of *PN*, n represents the index of the sector.

Algorithm processes are as follows.

Set the similarity threshold l=0.5, n=1, n<8

1. Expand data PN into $W_{-n}W_{-n-1}\ldots W_{-2}\ W_{-1}$<$PN$> $W_1\ W_2\ldots W_n$;
2. Match with the feature pattern library, if l <0.5, then n ++, and return 1.
3. If $n \neq 8$, $W_{-n}W_{-n-1}\ldots W_{-2}\ W_{-1}$<$PN$> $W_1\ W_2\ldots W_n$ will be classified, the position in the pattern is recorded; If $n = 8$, it means that there is no appropriate place, then the data on this sector will be treated as useless data and been abandoned.

5 Experimental Result and Analysis

Feature pattern library is generated by 3 *txt* documents, 6 *word* documents and 6 *pdf* documents, and the pattern similarity threshold S = 0.4. After making internal testing on generated patterns, six of them are selected to form feature pattern library. It includes 2 *txt* patterns, 2 *word* patterns and 2 *pdf* patterns, respectively named E_1, E_2, E_3, E_4, E_5 and E_6.

A hard disk and both a new and an old USB flash disks are selected to make Raw and formatted recovery respectively. Each disk has 10 files on it. The results are showed in Table 1, Table 2:

Table 1. Result of Raw Data Recovery Based on pattern Matching

Disk	Size	Number of recovery files	successful rate
New USB flash disk	128M	8	80%
USB flash disk	128M	14	50%
USB flash disk	256M	20	30%
Hard disk	5G	31	10%

Table 2. Result of Formatted Recovery Based on pattern Matching

Disk	Size	Number of recovery files	successful rate
USB flash disk	128M	9	60%
USB flash disk	256M	13	35%
Hard disk	5G	25	90%

As we can see from the tables, the recovery result of a new USB flash disk is the best. It is because the majority sectors of a new USB flash disk have not been written yet and most files are stored continuously. This reduces the conflicts on data classification, and it is convenient for pattern matching. While the disk has been used for a long time, the sector data becomes very complicated because of the increasing number of user operations, which will make the matching more complicated.

It is obviously that the effect of file recovery is related to disk capacity and serving time. The larger disk capacity and the more files it stores, the more conflicts would be caused on data classification; the longer serving time, the more complicated the data would become, which results in more difficulties in pattern matching.

6 Conclusion

Making full use of data on free sectors, data recovery based on intelligent pattern matching has a good effect on restoration of text files, provides a new approach to the development of data recovery software in the future, and also improves the efficiency of computer forensics. However, there are lots of works to further improve, including to improve the accuracy of extraction of feature patterns, to expand the scope of the pattern library, to further improve the intelligent processing of related sectors, to extract the central meaning of the text and enhance the matching accuracy. Currently this approach only deals with text files, but it is feasible to expand the scope to other files because other files also have their own file formats and encoding patterns, based on which their characteristic pattern library can be developed. With this data recovery approach, the data utilization ratio of free sectors can be enhanced, the risk of data loss can be reduced and the recovery efficiency can be improved.

References

[1] Riloff, E.: Automatically Constructing a Dictionary for Information Extraction Tasks. In: Proceedings of the Eleventh National Conference on Artificial Intelligence, pp. 811–816. AAAI Press / The MIT Press (1993)

[2] Yangarber, R., Grishman, R., Tapanainen, P.: Unsupervised Discovery of Scenario Level Patterns for Information Extraction. In: Proceedings of Sixth Applied Natural Language Processing Conference (ANLP - 2000), Seattle WA, pp. 282–289 (2000)

[3] Zheng, J.h., Wang, X.y., Li, F.: Research on Automatic Generation of Extraction Patterns. Journal of Chinese Information Processing 18(1), 48–54 (2004)

[4] Qiu, Z.-h., Gong, L.-g.: Improved Text Clustering Using Context. Journal of Chinese Information Processing 21(6), 109–115 (2007)

[5] Liu, Y.-c., Wang, X.-l., Xu, Z.-m., Guan, Y.: A Survey of Document Clustering. Journal of Chinese Information Processing 20(3), 55–62 (2006)

[6] Abdel-Galil, T.K., Hegazy, Y.G., Salama, M.M.A.: Fast match-based vector quantization partial discharge pulse pattern recognition. IEEE Transactions on Instrumentation and Measurement 54(1), 3–9 (2005)

[7] Perruisseau-Carrier, J., Llorens Del Rio, D., Mosig, J.R.: A new integrated match for CPW-FED slot antennas. Microwave and Optical Technology Letters 42(6), 444–448 (2004)

[8] Papadimit riou, C.H.: Latent Semantic Indexing:A Probabilistic Analysis. Journal of Computer and System Sciences 61(2), 217–235 (2000)