

Record Preprocessing for Data Compression

Jürgen Abel

University Duisburg-Essen, Department "Communications Systems",
Faculty of Engineering Sciences, Bismarckstrasse 81, 47057 Duisburg, Germany.
Phone: +49 - 2137 - 999333, E-mail: juergen.abel@acm.org.

A new preprocessing scheme for lossless data compression is presented, which exploits the structure of record-based files. Record based files can be database files but also image files, for which a row of the image represents a record, or sequential data files like the file *geo* of the Calgary Corpus. Symbol repetitions which occur at the same position inside a sequence of records with fixed length are detected and treated by a special transformation. The compression rate of such files can often be enhanced if the file is transposed by the record length before compression and after decompression. The presented approach is able to detect files with such a structure and to determine the corresponding record length. The impact on the compression rate is compared between BWT-, PPM- and LZ based compression algorithms. For some files a compression gain of more than 80 percent (e.g. file *kennedy.x15* from the Canterbury Corpus) file can be reached.

The heart of the scheme is a transposition with the proper record length. Clearly, transposition can not lead to a better compression for every file, e.g. normal text files. Therefore, the scheme needs to decide if a file should be transposed at all and if so, what record length should be used for the transposition. Hereto, the detection of the record length is solved first. Given the proper record length, the statistics of the raw file and the preliminary transposed file are compared in order to determine if the file should be transposed for compression at all.

For the detection of the record length l of a file M , a simple but evident offline approach is used, which counts the frequency of symbol distances. For each symbol $M[i]$, the distance d between the current position i and the last occurrence of $M[i]$ is calculated. If $M[i]$ is equal to the prior symbol $M[i-1]$, i.e. $M[i]$ is within a run of repeated symbols, the distance d is omitted, in the other case the frequency counter $f_c(d)$ for that distance is increased and i is saved as the last position of $M[i]$. The distance d , for which $f_c(d)$ reaches its maximum value, represents the proper record length for the transposition.

After the detection of the record length, the file is preliminary transposed. If the variance of trigrams and the average sum of the trigram frequency counts are each ten times bigger for the preliminary transposed file than for the raw file, the file is deemed suitable for transposition.

For a practical implementation, the trigram frequencies counts should be stored inside a hash table, which has much smaller space and time requirements than a simple 24 bit integer table. An acceleration of this scheme at the expense of reliability can be achieved if the trigram statistic is replaced by a bigram statistic instead. The presented approach can be used for all standard compression algorithms, though context based algorithms tend to exploit the transposed structure better than dictionary based schemes.