Adaptive Compression of Graph Structured Text

John Gilbert and David M Abrahamson
Department of Computer Science, Trinity College Dublin
{gilberj, david.abrahamson}@cs.tcd.ie

In this paper we introduce an adaptive technique for compressing small quantities of text which are organized as a rooted directed graph. We impose a constraint on the technique such that data encountered during a traversal of any valid path through the graph must be recoverable without requiring the expansion of data that is not on the path in question. While compression can be applied independently to the text at each node using well known techniques, we propose exploiting inter-node context to improve results when using adaptive dictionary based compression methods. The technique we present (Graph LZW) determines the set of nodes which are guaranteed to be encountered before reaching node $x$ while traversing any valid path in the graph [2], and uses them as a basis for conditioning an LZW dictionary [3] for the compression/expansion of the data in $x$.

To the best of our knowledge we are the first to propose compressing the content of nodes in a graph by employing an adaptive model which has been conditioned using inter-node context. Our previous work [1] presented an LZW-based technique which identified context for compressing instruction cache lines for a computer program’s object code. While the underlying source of context was the same as that presented in this paper, the method by which it was exploited to condition and maintain the LZW dictionary during the compression/expansion of a program was dramatically different.

We applied our technique to a collection of six PDA websites. Initial results show that our improved LZW technique reduces the compressed text size for these sites by approximately 20% more than regular LZW. When compared with regular LZW our new algorithm trades additional analysis undertaken at compression time for improved compression ratios. Despite our superior compression, the modifications to a standard LZW decompressor to support our Graph LZW algorithm are minor and do not contribute any significant time overhead to the decoding routine.

References

