Reorganizing Compressed Text ⋆

Nieves R. Brisaboa[1], Antonio Fariña[1], Susana Ladra[1], and Gonzalo Navarro[2]

[1] Database Laboratory, University of A Coruña, Spain
{brisaboa,fari,sladra}@udc.es
[2] Department of Computer Science, University of Chile, Chile
gnavarro@dcc.uchile.cl

Key words: Word-based compression, searching compressed text, compressed indexing.

Recent research has demonstrated beyond doubts the benefits of compressing natural language texts using word-based statistical semistatic compression. Not only it achieves extremely competitive compression rates, but also direct search on the compressed text can be carried out faster than on the original text; indexing based on inverted lists benefits from compression as well.

Such compression methods assign a variable-length codeword to each different text word. Some coding methods (Plain Huffman and Restricted Prefix Byte Codes) do not clearly mark codeword boundaries, and hence cannot be accessed at random positions nor searched with the fastest text search algorithms. Other coding methods (Tagged Huffman, End-Tagged Dense Code, or (s, c)-Dense Code) do mark codeword boundaries, achieving a self-synchronization property that enables fast search and random access, in exchange for some loss in compression effectiveness.

In this paper, we show that by just performing a simple reordering of the target symbols in the compressed text (more precisely, reorganizing the bytes into a wavelet-tree-like shape) and using little additional space, searching capabilities are greatly improved without a drastic impact in compression and decompression times. With this approach, all the codes achieve synchronism and can be searched fast and accessed at arbitrary points. Moreover, the reordered compressed text becomes an *implicitly indexed* representation of the text, which can be searched for words in time independent of the text length. That is, we achieve not only fast sequential search time, but indexed search time, for almost no extra space cost.

We experiment with three well-known word-based compression techniques with different characteristics (Plain Huffman, End-Tagged Dense Code and Restricted Prefix Byte Codes), and show the searching capabilities achieved by reordering the compressed representation on several corpora. We show that the reordered versions are not only much more efficient than their classical counterparts, but also more efficient than explicit inverted indexes built on the collection, when using the same amount of space.

⋆ In *Proceedings of the 31st annual international ACM SIGIR conference*, pages 139-146, Singapore, July 2008, doi: http://doi.acm.org/10.1145/1390334.1390360

