Re-Store: A System for Compressing, Browsing, and Searching Large Documents

(Extended Abstract of Invited Presentation)

Alistair Moffat Raymond Wan

Department of Computer Science and Software Engineering
The University of Melbourne
Victoria 3010, Australia

`www.cs.mu.oz.au/~{alistair,rwan}`

We describe a software system for managing text files of up to several hundred megabytes that combines a number of useful facilities: effective text compression; phrase browsing; and fast interactive searching.

Mechanisms for compressing text have been studied for many years, and a wide range of effective methods has been developed. But compression of text makes it unwieldy in other ways – it must be decompressed before being viewed, and it is harder to directly search. One way of addressing these concerns is to build an index for the source document, and search via a query interface [Witten et al., 1999]. Then the passages sought by the user can be identified using Boolean or ranked queries, and only those selected passages need be fetched and decompressed. Another alternative is to use a compression mechanism that is amenable to compressed pattern matching, and undertake the equivalent of an exhaustive linear search in the compressed text to locate passages of interest [de Moura et al., 2000]. Again, only small fragments of the source document might be eventually presented to the user.

In this presentation we consider a third approach, and describe a software compression and searching system – dubbed RE-STORE – that supports browsing within the compressed text based on phrases extracted from the text, and fast identification and decompression of the passages in the text containing those phrases. In our system, the user selects one or more terms of interest from a static list that is somewhat akin to a vocabulary derived from the document, and is then free to explore situations in which those terms appear – either as components of longer phrases in which other phrases are joined to the right or the left, or by exploring the constituent parts of any phrase encountered during the browsing session. When a set of "interesting" phrases has been collated, locations in the compressed document at which those phrases occur are determined, and appropriate windows of text displayed to the user. For example, the term "United" is part of all of the phrases "Manchester United", "United Airlines", "United Kingdom", and "United States", and the latter is then in turn a constituent part of the more specific phrases "President of the United States" and "United States Postal Service". Using the browser, we might start with the word "United", but eventually indicate that "United States Postal Service" is the phrase that we seek occurrences of in the source document.

Our system does not involve any radical new technologies, and in some ways is more engineering than research. The source modelling mechanism used is based on the RE-PAIR system of Larsson and Moffat [2000]; the browsing system is similar to the PHIND mechanism described by Nevill-Manning et al. [1997]; and the underlying coder is a simple binary mechanism that could be assigned as an undergraduate project. Nevertheless, we believe that the drawing together of these components provides a useful opportunity, and it is that usefulness we sought to capture in our software.

Figure 1 introduces the major components of the RE-STORE system. The three subsystems – for compression, block merging, and browsing – are RE-PAIR, RE-MERGE, and RE-PHINE. Initially, a text document is compressed using the RE-PAIR compression algorithm [Larsson and Moffat, 2000]. The idea behind RE-PAIR is very simple – frequently appearing pairs of adjacent characters are replaced by a new symbol. For example, if the character pair "th" is the most frequent in the text, it is replaced everywhere by a new symbol, integer 256. The process repeats until there are no more adjacent pairs of characters that appear more than

Proceedings of the *Sixth Australasian Document Computing Symposium,* Coffs Harbour, December 7 2001. The full version of this paper appears in the proceedings of the 2001 *Symposium on String Processing and Information Retrieval,* November 13–15, Laguna de San Rafael, Chile. Figure 1 and the bulk of the text of this extended abstract are taken directly from that paper.

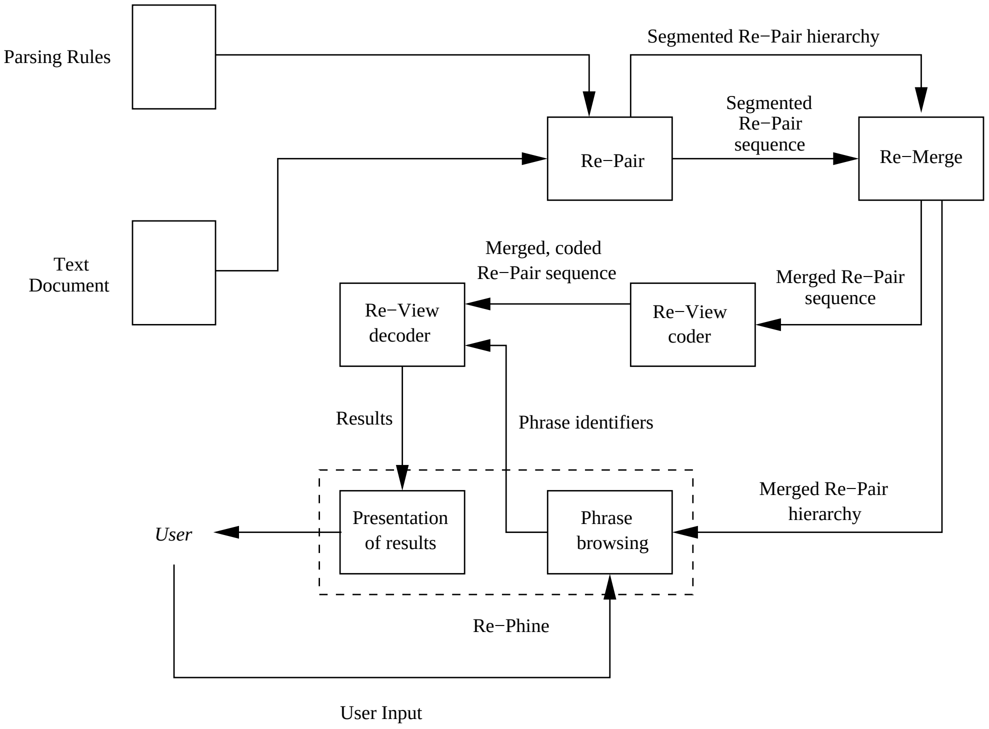

Figure 1. Overview of the RE-STORE system.

once. A consequence of performing the replacements in decreasing order of pair frequency is that common phrases are formed first, and then longer phrases that use them as components are formed. We also introduce a modification to RE-PAIR that forces phrases to be word aligned.

In order for encode-time memory consumption to be bounded, RE-PAIR segments the source document into blocks. From a compression point of view, segmentation has little impact. But from a browsing point of view, we wish to continue to regard the source document as being monolithic, and it is this requirement that gives rise to the need for RE-MERGE, which combines the compressed blocks generated by RE-PAIR back into a single structure. The multi-pass merging process is somewhat similar to the iterative process performed by the RAY system of Cannane and Williams [2001], which also does symbol pair replacements.

The compressed document can then be examined via the phrase browser we call RE-PHINE. Using RE-PHINE, a user can navigate through the structure of phrases which make up the compressed document and determine a set of phrases of interest. The browsing operations allow phrases to be extended to either the right or left by the inclusion of adjacent phrases that appear twice or more in the source document. Extension can be continued until no further phrases are available; or can be paused at any time.

Finally, the locations in the compressed text at which those phrases occur are identified, and a window of text surrounding each occurrence decoded and presented. Searching and fast decoding is made possible by a static byte-aligned coder designed specifically for this task.

The overall effect is that, without use of any explicit index or vocabulary, nor any structures other than the original RE-PAIR-compressed text, files of up to several hundred megabytes can be reduced to around 30% of their original size, and be phrase-browsed and explored without full decompression being required.

References

A. Cannane and H. E. Williams. General-purpose compression for efficient retrieval. *Journal of the American Society for Information Science and Technology*, 52(5):430–437, Mar. 2001.

E. S. de Moura, G. Navarro, N. Ziviani, and R. Baeza-Yates. Fast and flexible word searching on compressed text. *ACM Transactions on Information Systems*, 18(2): 113–139, 2000.

N. J. Larsson and A. Moffat. Offline dictionary-based compression. *Proc. IEEE*, 88(11):1722–1732, Nov. 2000.

C. G. Nevill-Manning, I. H. Witten, and G. W. Paynter. Browsing in digital libraries: A phrase-based approach. In *Proc. ACM Digital Libraries 97*, pages 230–236, July 1997.

I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann, San Francisco, second edition, 1999.