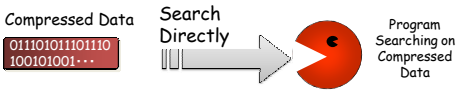
On Performance of VF Codes

Satoshi Yoshida* and Takuya Kida*

*Graduate School of Information Science and Technology, Hokkaido University

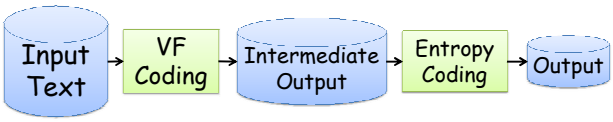
Introduction

We have addressed speeding up of pattern matching on texts by designing a compression method that is suitable for compressed pattern matching. A Variable-to-Fixed-length codes (VF codes for short) are suitable for compressed pattern matching because all codeword boundaries are obvious. VF code is a source coding that assigns fixed length codewords to variable length substrings in an input text. Since existent VF codes have poor compression ratios, they are paid less attentions in spite of having preferable engineering aspect that all codewords are the same length.



		Input Text	
		Fixed	Variable
Compressed Text	Fixed	FF Code	VF Code
	Variable	FV Code	VV Code

In this paper, we have investigated on changes in performance of VF codes with entropy encodings; we experimented combinations of VF codes and entropy codings and measured their compression ratios and compressing times.

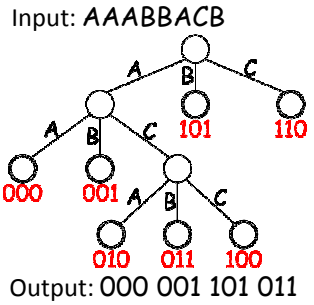


VF codes

A VF code is a source coding that parses an input string into a consecutive sequence of variable-length substrings and then assigns a fixed length codeword to each substring. Typical VF codes use tree structures as dictionaries, which are called parse trees. Each node of them corresponds to a string and each leaf of them has codeword. The encoding is done by traversing the parse tree by input text, and then output the codeword assigned to the node if we can not traverse any more.

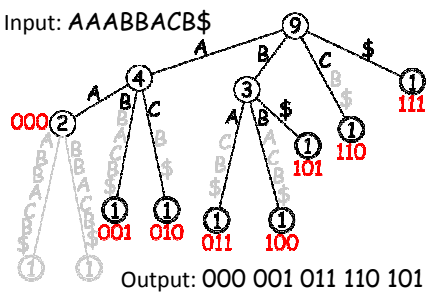
Tunstall Code

Tunstall code [Tunstall 1967] is an optimal VF code for a memory-less information source. It uses a parse tree called Tunstall tree, which is an ordered complete $|\Sigma|$-ary tree that each edge is labeled with a different symbol in the alphabet Σ. For example, given the text T=AAABBACB and the Tunstall tree of right figure, the encoded sequence becomes 000/001/101/011.



STVF Code

Suffix Tree based VF code (STVF code for short) [Kida 2009] is a coding that constructs a suitable parse tree for the input text by using a suffix tree. It is, namely, an off-line compression scheme that encodes after gathering the statistical information of the whole input text beforehand. Since the suffix tree for the input text includes the text itself, we can not use the whole tree as a parse tree. We have to prune it with some frequency-base heuristics to make a compact and efficient parse tree. That is, we select the most frequent leaf in the current parse tree to add its all children in the suffix tree while the number of leaves is less than 2^l where l is the length of codewords. If the added child is a leaf in the suffix tree, we delete the label on the edge from the child to its parent except for the first character of the label.



Experiments

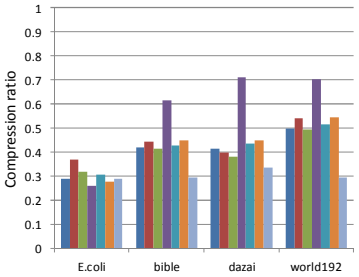
We have implemented Tunstall code [Tunstall 1967], STVF code [Kida 2009], Huffman code, and Range coder [Martin 1979]. We abbreviate them as Tunstall, STVF, Huf and RC respectively. We denote the combination of two compression method as concatenation of their abbreviations with "+" between them, such as STVF+Huf, Tunstall+RC, and so on.

We used E.coli, bible.txt, and world192.txt as test corpora, which are selected from "the Canterbury corpus[1]." We also used dazai.txt, the collection of Japanese novel texts written by Osamu Dazai, from Japanese corpus "J-TEXTS[2]." The file dazai.txt is encoded with UTF-8.

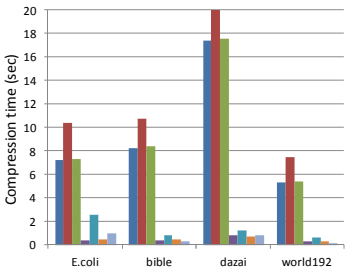
We have compared the compression ratios and compression times of STVF, STVF+RC, STVF+Huf, Tunstall, Tunstall+RC, and Tunstall+Huf, setting the codeword length 16. We have added the results of gzip for reference. The compression ratios of STVF are usually better than those of Tunstall. However, compression ratios of Tunstall+Huf and Tunstall+RC are almost the same level.

In STVF, there is a few or no improvement by applying Huf or RC. Applying Huffman code after VF codes depreciates compression times, whereas Range coder does not make those compression slower.

- STVF Code
- STVF Code + Huffman Code (STVF + Huf)
- STVF Code + Range Coder (STVF + RC)
- Tunstall Code
- Tunstall Code + Huffman Code (Tunstall + Huf)
- Tunstall Code + Range Coder (Tunstall + RC)
- gzip



Compression ratios
(compressed file size / original file size)



Compression times

Conclusion

In this paper, we demonstrated the performance of combinations of VF codes and entropy codings by experiments. Doing entropy codings to VF codes not only improves compression ratios, but makes almost no loss in compression; unexpectedly, there are situations that entropy coding can reduce them. The reason is considered to be that the total I/O time decrease by the reduction in the amount of data stored into a hard disk. From the lack of time, we could not compare our methods to the modern compression methods, such as Dense codings [BFNE 2003], [BINP 2003], BPEX [MTST 2008], and so on. It is also our future work.

References

[BFNE 2003] N. R. Brisaboa, A. Fariña, G. Navarro, and M. F. Esteller, "(S, C)-dense coding: An optimized compression code for natural language text databases," In Proc. of 10th International Symp. on String Processing and Information Retrieval (SPIRE 2003), LNCS 2857, 2003, pp. 122--136.

[BINP 2003] N. R. Brisaboa, E. L. Iglesias, G. Navarro, and J. R. Paramà, "An efficient compression code for text databases," In Proc. of the 25th European conference on IR research (ECIR 2003), 2003, pp. 468--481.

[Kida 2009] T. Kida, "Suffix tree based VF-coding for compressed pattern matching," In Proc. of Data Compression Conference 2009 (DCC 2009), Mar. 2009, p. 449.

[Martin 1979] G. N. N. Martin, "Range encoding: An algorithm for removing redundancy from a digitised message," In Proc. of Video and Data Recording Conference, 1979, pp. 24--27.

[MTST 2008] S. Maruyama, Y. Tanaka, H. Sakamoto, and M. Takeda, "Context sensitive grammar transform: Compression and pattern matching," In Proc. of 15th International Symposium on String Processing and Information Retrieval (SPIRE 2008), LNCS 5280, Nov. 2008, pp. 27--38.

[Tunstall 1967] B. P. Tunstall, "Synthesis of noiseless compression codes," Ph.D. dissertation, Georgia Inst. Technol., Atlanta, GA, 1967.

1 http://corpus.canterbury.ac.nz/descriptions/
2 http://www.j-texts.com/