Learning Parse Trees for Efficient Variable to Fixed Length Coding

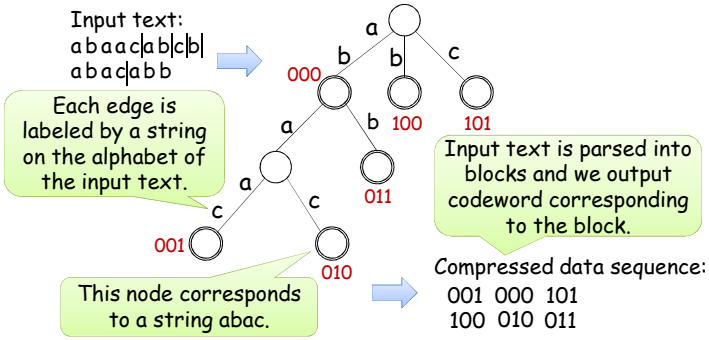
Satoshi Yoshida* and Takuya Kida*

*Graduate School of Information Science and Technology, Hokkaido University.

We address the Variable-to-Fixed length codes (VF codes for short). VF codes are the compression schemes that assign fixed length codewords to variable length substrings of the input text. VF codes enable fast data processing. However, VF codes are rarely used because of the low compression ratio. In this paper, we consider algorithms that train the dictionary by scanning the input text repeatedly.

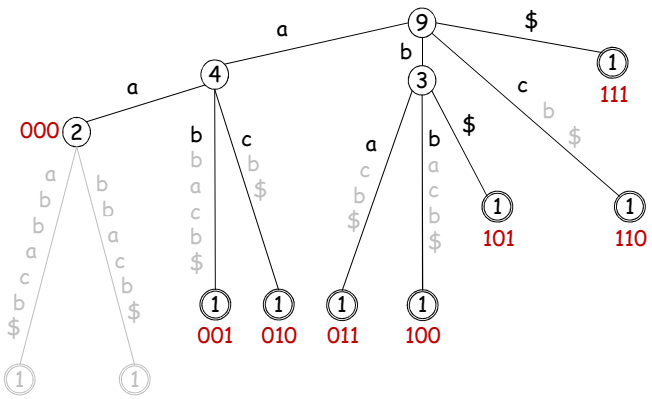
VF codes

VF code is a source coding that parses the input string into a consecutive sequence of variable length substrings (blocks) and then assigns a fixed length codeword to each substring. Since all codeword of a VF code have the same length, we can handle the compressed data easily. Therefore, VF codes enable fast data processing such as information retrieval and data mining.



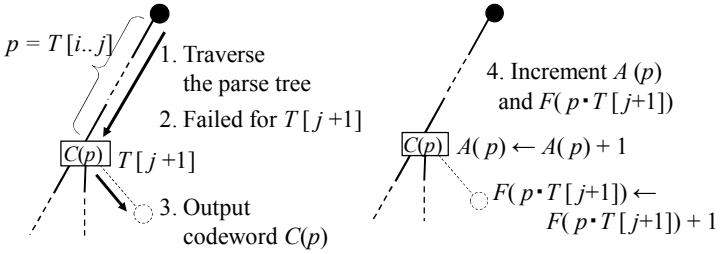
STVF codes

Suffix Tree based VF code (STVF code for short) is a VF code that uses pruned suffix tree as a dictionary. It achieves higher compression ratio than the basic VF codes.



Training by block exchanging

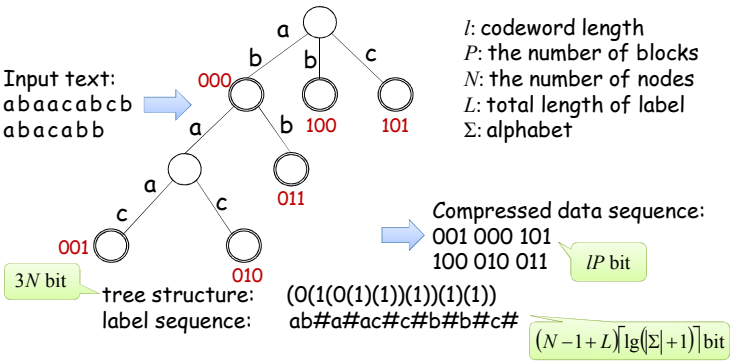
We parse the input text to count frequency of block p (denoted by $a(p)$) and the concatenation of it and the next character t (denoted by $f(p \cdot t)$). If two strings satisfies $f(s)$ is larger than $a(t)$, we exclude t from the dictionary and include s into it.



It takes too much time!

Training by MDL principle

We construct a dictionary that minimizes the sum of the size of dictionary and that of compressed data.

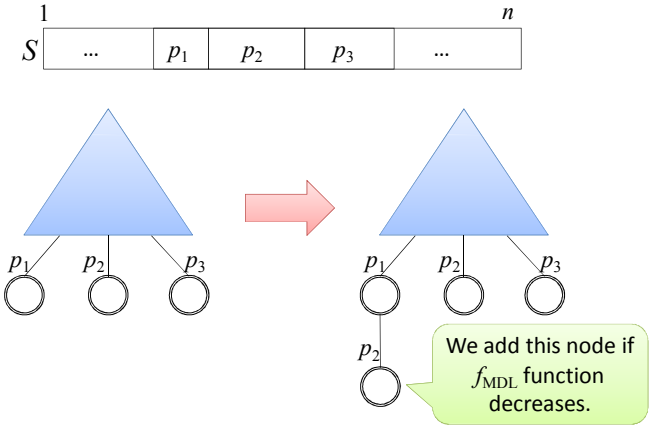


$$f_{\mathrm{MDL}}(S,T) = lP + 3N + (N-1+L)\lceil \lg(|\Sigma|+1) \rceil$$

→minimize

NP-hard

We proposed an algorithm that transforms the dictionary greedily so that f_{MDL} function decreases.



Conclusion

In this paper, we considered two algorithms that train the dictionary by scanning the input text repeatedly. One is that which exchanges useless strings in the current dictionary as a result for the other strings that are expected to be frequently used. The other is that which transforms the dictionary greedily so that the sum of the size of dictionary and the compressed data sequence. Our future work is to conduct experiments and to use some ingenuity to improve the compression ratio and speed.