Direct Access to Variable-to-Fixed Length Codes with a Succinct Index

Satoshi Yoshida, Hirohito Sasakawa, Kei Sekine, and Takuya Kida
Graduate School of Information Science and Technology, Hokkaido University.

1. Background

Variable-to-Fixed Length Code (VF Code)

Compression method that splits the input text into variable length substrings and then converts them into fixed length codewords.

Strong Point

Easy to handle the compressed data.

➪ Enables extract arbitrary codeword in constant time.

Problem

Hard to map position on the original data to that on the compressed data.

➪ Substring Extraction Problem

Goal

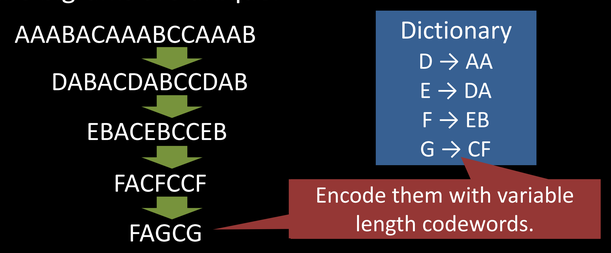
To develop a fast method for the substring extraction problem.

We combine VF codes and succinct index to solve the problem fast.

2. Related Works

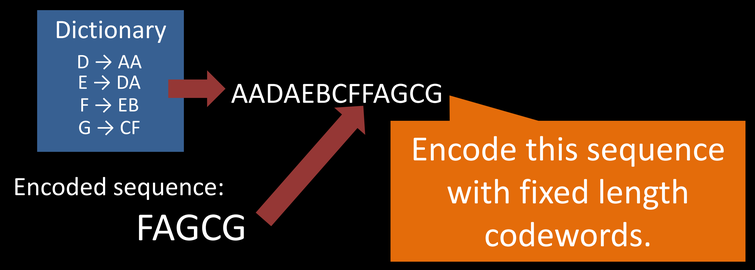
Re-Pair [Larsson & Moffat, Proc. IEEE 2000]

Substitute the most frequent bigram into a new symbol until all the bigrams are unique.



Re-Pair-VF [Yoshida & Kida, DCC 2013]

Encode dictionary and encoded sequence with fixed length codewords.



3. Rank/Select Dictionary

Data structure that answers rank and select queries for bit sequence efficiently.

Rank: the number of 1's appearing from the beginning to the specified position.

Select: the leftmost position that the number of 1's appearing from the beginning to the position is the specified number.

4. Proposed Method

Additional Data Structure for Substring Extraction

We generate a bit sequence B with length $|T|$ where

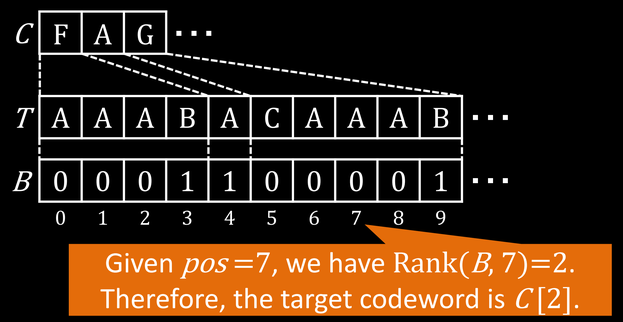
$$B[i] = \begin{cases} 1 & \text{if } T[i] \text{ is the last character of a phrase,} \\ 0 & \text{otherwise,} \end{cases}$$

and embed its rank/select dictionary into the compressed data of Re-Pair-VF.

Substring Extraction from VF Code

1. Given the input position pos on the original text, the position of the target codeword determined by rank(pos).
2. Load the codeword from the disk.
3. Decompress using the dictionary, and then, original text from $T[pos]$ is obtained.

VF code enables skipping to the target codeword in constant time.

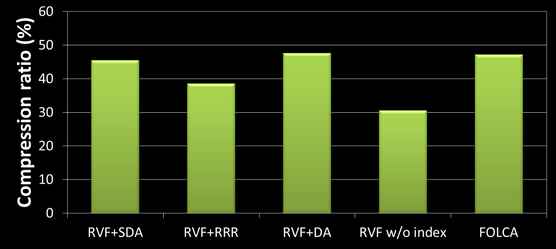


5. Experiments

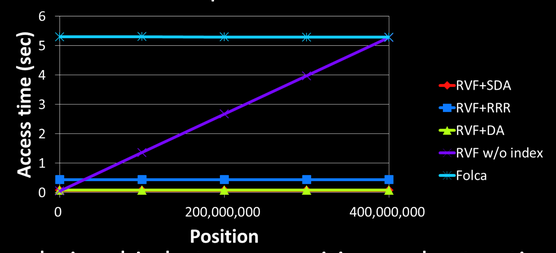
We compared compression ratio and substring extraction speed on an English text[1].

Methods

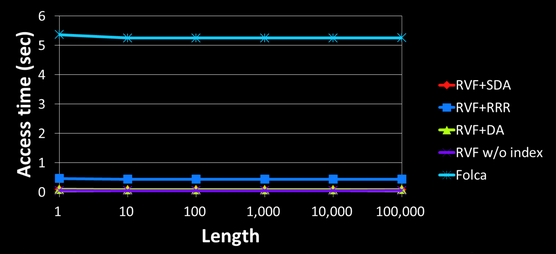
- Re-Pair-VF without index (RVF w/o index)
- FOLCA [Maruyama et al., SPIRE 2013]
- Re-Pair-VF with DARRAY[2] [Okanohara & Sadakane, ALENEX 2007] (RVF + DA)
- Re-Pair-VF with SDARRAY[2] [Okanohara and Sadakane, ALENEX 2007] (RVF + SDA)
- Re-Pair-VF with RRR[2] [Raman, Raman, and Rao, SODA 2002] (RVF + RRR)



Compression ratios.



The relationship between position and extraction time.



The relationship between substring length and extraction time.

Results

- Compression ratio of our methods are comprable with FOLCA.
- Our methods extracts substrings 10—50 times faster than FOLCA.

1. The first 500MB of english from pizza&chili corpus (http://pizzachili.dcc.uchile.cl/index.html).
2. We used Claude's implementation, which is available from https://code.google.com/p/libcds/

6. Conclusion

We proposed a method that solves the substring extraction problem fast in practice by adding an index structure in Re-Pair-VF code. In the experiments the proposed method showed better performance than FOLCA.