# 検索効率を考慮したテキストファイル圧縮の検討*

2P－10

大塚真吾†　宮崎収兄‡
千葉工業大学工学部情報工学科§

## 1　はじめに

コンピュータやインターネットの普及により，電子メールやホームページなど電子文書の持つ情報の重要性が高まっている．インターネットから文書ファイルをダウンロードする場合，通信コストやトラフィックの軽減のため何らかの可逆圧縮が行われる．圧縮された文書ファイルに対して検索を行う場合，ファイルの解凍を行ってからする手法や圧縮ファイルを直接検索する手法 [1, 3]，インデキシングを用いた手法 [2] などが考えられる．しかし，圧縮率と検索効率を考えた場合どれも一長一短である．そこで，本稿では高い圧縮率と速い検索が行える二段階圧縮法を提案する．

## 2　圧縮ファイルと検索効率

圧縮ファイルへの検索を考える場合，復号化してから検索を行う手法と圧縮ファイルそのものに対して直接検索を行う手法に大別できる．圧縮率の面から見ると，後者の手法は直接検索を考慮している分性能は下がる．一方，検索効率の面から見ると，前者は検索を行う前に復号化を行うため，その分時間がかかる．

また，ファイルを圧縮する前にあらかじめ検索ファイルや転置ファイルを作成する手法もある．この場合，検索は索引を用いるので格段に速くなる．また，テキストファイルそのものは検索と関係がないので高性能の圧縮をする事ができる．しかし，索引ファイルや転置ファイルがある分圧縮率が落ちてしまう．

このように，圧縮率と検索効率を考える場合，一方の性能を上げるともう一方の性能が下がるといった傾向になる．

*A study on compression of text files for efficient search
†Shingo Otsuka
‡Nobuyoshi Miyazaki
§Department of Computer Science, Chiba Institute of Technology

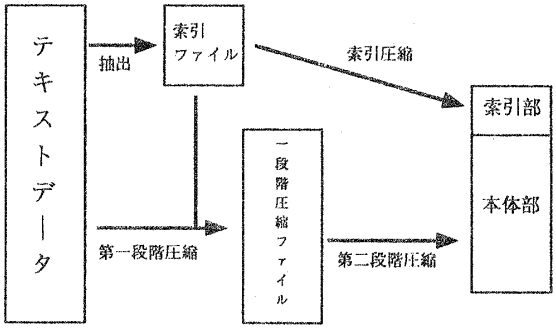


図 1: 圧縮の概要

## 3　二段階圧縮法

二段階圧縮法は図 1 のように索引部と本体部とに分かれる．まず，圧縮対象となるファイルから単語を抽出し索引ファイルを作成する．次に，索引ファイルにある単語の位置情報を用いて圧縮対象となるファイルを符号化する（第一段階圧縮）．更にそのファイルを他の高性能なアルゴリズムで符号化を行う（第二段階圧縮）．一方，索引ファイルは直接検索出来るアルゴリズムで符号化を行う．

復号については符号化手順の逆を行えばよい．検索は索引部を直接検索する事で高速に行うことができる．

## 4　実験と評価

### 4.1　実験環境

文書データとして，英文ファイルの標準的なテストファイルである「Text Compression」の性能評価ファイルの一部を用いて実験を行った．実験は全て PentiumII 350MHz × 2 を搭載した Linux 上で行った．実行時間は UNIX の「time」コマンドで測定した．比較対象として，ハフマン符号化法，LZB，LZW，LHA，gzip を用いた．また，本手法は二段

表 1: 実験結果 1
ファイルA　610856バイト

| 符号化法 | 圧縮率 [%] | 検索時間 [秒] |
|---|---|---|
| ハフマン | 57.2 | 0.49 |
| LZW | 50.8 | 2.07 |
| LZB | 63.9 | 1.23 |
| gzip | 40.8 | 0.13 |
| LHA | 40.7 | 0.18 |
| 本手法 | 39.6 | 0.09 |

ファイルB　610856バイト

| 符号化法 | 圧縮率 [%] | 検索時間 [秒] |
|---|---|---|
| ハフマン | 60.5 | 0.7 |
| LZW | 56.6 | 1.76 |
| LZB | 54.3 | 0.92 |
| gzip | 33.8 | 0.1 |
| LHA | 33.8 | 0.13 |
| 本手法 | 32.4 | 0.06 |

表 2: 実験結果 2
ファイルC　377109バイト

| 符号化法 | 圧縮率 [%] | 検索時間 [秒] |
|---|---|---|
| ハフマン | 65.6 | 0.46 |
| LZW | 61.6 | 1.15 |
| LZB | 60.2 | 0.58 |
| gzip | 38.4 | 0.08 |
| LHA | 38.4 | 0.09 |
| 本手法 | 39.8 | 0.06 |

ファイルD　46526バイト

| 符号化法 | 圧縮率 [%] | 検索時間 [秒] |
|---|---|---|
| ハフマン | 60.8 | 0.06 |
| LZW | 50.2 | 0.13 |
| LZB | 58.6 | 0.07 |
| gzip | 38.9 | 0.04 |
| LHA | 39.1 | 0.01 |
| 本手法 | 49.2 | 0.03 |

階圧縮として LHA を索引の圧縮としてハフマン符号化法を利用した.

検索時間は LZW,LZB,LHA は復号時間と「grep」コマンドの合計時間で表し，ハフマン符号化法と本手法はビット単位の文字列照合が出来る自作のプログラムを用いた．また，gzip はファイルを復号しながら文字列照合を行う「zgrep」コマンドを用いた.

### 4.2　実験結果

実験結果の一部を表 1,2 に示す．本手法と他の手法を比較すると，ファイルサイズの大きいファイル A,B では圧縮率，検索時間ともに本手法が一番良い結果となった．ファイルサイズが少し小さいファイル C では gzip,LHA より圧縮率がやや劣ったが，検索時間は良い結果が得られた．一方，ファイルサイズの小さいファイル D は gzip,LHA より圧縮率が劣り，検索時間は他の手法とあまり変わらない結果となった．他のテストファイルでも，十分にファイルが大きければ圧縮率も検索時間も良くなるという結果が得られた.

## 5　おわりに

本稿では圧縮率と検索効率を考慮した二段階圧縮方式を提案し評価を行った．その結果，ファイルサイズが十分に大きければ，gzip や LHA などの高い圧縮率を実現するアプリケーションと同様な圧縮率で高速な検索を実現する事が可能となった.

今後は本手法が大量の文書ファイルを扱うシステムに対して有効であるか検討していく.

## 参考文献

[1] 松本光崇, 角田達彦, 松本裕治. 圧縮ファイルへの直接検索を可能にする符号化法の提案. 電子情報通信学会論文誌, pp. 41–48, 1996.

[2] 須藤真理, 横尾英俊. 情報検索とデータ圧縮とを統合したシステム mg の日本語化. 情報処理学会情報基礎研究会, Vol. 40, pp. 33–40, 1995.

[3] 多々納勉, 大塚真吾, 宮崎収兄. 圧縮ファイルに直接検索を行う一手法. 情報処理学会第５６回全国大会, Vol. 1, pp. 416–417, 1998.