Vol. 34 No. 11 情報処理学会論文誌 Nov. 1993

ショートノート

順序保存符号の漢字データへの適用

中津楢男† 片桐友子†

順序保存符号化を人名データに適用した実験結果を示す．順序保存符号はアルファベット順を符号語の上でも保存しているため，この符号を用いてデータ圧縮を行えば，他の圧縮法と違って符号語のままで直接検索，範囲検索，整列などが実現でき，あらかじめ復号化する必要がない．また人名の場合には1つの漢字に複数の読みが存在するが，この問題を，読みの異なる漢字は異なるシンボルとして扱うことで対処した．アルファベット順序としては電話帳式の順序（実験1）と国語辞典式の順序（実験2）の2つを仮定した．その結果，符号化・復号化のための辞書ファイルを含めて，符号化しない場合と比較して，実験1では53%から63%，実験2では67%から75%のサイズに圧縮できた．

Application of an Alphabetic Code to Kanji

NARAO NAKATSU† and TOMOKO KATAGIRI†

A kanji character has some readings in general. To a kanji with k readings, we assign k codewords each of which corresponds to each reading. By this way, a kanji can be stored without its reading. A list of 5,265 names represented by kanjis and their readings is coded by the optimal alphabetic code (apcode) and the alphabetic code for fixed-length records (fapcode). Experiment 1 used the Japanese traditional alphabetic order and experiment 2 used an order suitable for computer processing. The file size reduces to 63% or 53% (experiment 1) and 67% or 75% (experiment 2), by the apcode or the fapcode, respectively (this estimation includes the size of the dictionary used for encoding and decoding). Sorting and retrieval is a usual operation applied to a list of names. These operations can be applied to codewords to get correct results when data is coded by an alphabetic code, otherwise the decoding is required before applying these operations. We conclude the alphabetic code is useful for the compression of kanji.

1. はじめに

通信の場合は符号化して送信されたデータを受信時に1回だけ復号すればよい（しかも符号化された順に復号される）．データベースでは蓄積されたデータが何度もアクセスされ，アクセス順もランダムである．したがってデータ圧縮のために符号化を行えば，その副作用として，データを利用するたびに復号時間が余分に必要になる．符号化の方法によってはデータの区切りが認識できなくて，先頭から順に復号操作を行わなければデータの読みだしすらできない場合もある．しかしながら符号化を工夫すれば，ある操作に対しては復号せずに符号のままで処理できる場合がある．

データベースなどデータ検索の場合には，直接検索のほかにデータ間のアルファベット順を利用して検索される場合が多く，その場合には順序保存符号を利用することで，符号のままで検索が実現できる．著者らは英単語をデータとして，順序保存符号による符号化でデータ量は約6割に圧縮でき（最適符号といわれるハフマン符号と比べてほとんど同程度），検索時間は主記憶にデータを置いた場合，データ圧縮しない場合と同じ（ハフマン符号化した場合の1/5）であることを実験的に示した[3]．

本研究では日本語（特に人名）の符号化を考える．日本語（特に漢字）は文字種が多く，同じ漢字でも読みが幾通りもあるという問題を抱えている．固有名詞，特に人名などは漢字で表現された漢字列だけでは読みのわからない場合もあり，漢字列とその読みがペアで記憶され処理されることが多い．したがって整列（ソート）など実際の処理においては読みが利用され，漢字は出力時に（主として読みやすさという目的で）

† 愛知教育大学総合科学課程情報科学コース
Department of Computer and Information Sciences, Aichi University of Education

利用されるにすぎない．この問題に対しては，同じ漢字でも読みが違えば異なるシンボルと考えることとした．このように符号化すれば，ある漢字コードを持つ漢字の読みは一意に決まるので，漢字列にそれぞれ読みがなをつける必要はなくなる．もちろんシンボル数が何倍にも増えるが，可変長符号化によってデータ圧縮をはかる．

日本語処理の難しさはその独特の整列方式にも見られる．いわゆる国語辞典順[4]では仮名の文字コードを利用した整列ができない．このため文献5)や電話帳のように計算機処理のしやすい順序を採用している場合もある．データベースでは，そのシステムで用いられている順序を意識して利用者が質問を記述するのであれば，なんら問題は生じない．この報告では，漢字1字ごとが読み順に並んでいるという順序（電話帳方式）と国語辞典でいう順序の2つを考える．

実験として本学学生の名前（5,200名あまり）をデータとして符号化を行った．符号化および復号のための辞書を含めて，データ量は電話帳順序では53%から63%に，国語辞典順序では67%から75%に圧縮できた．圧縮率は扱うデータ量の増加にともなってさらに向上が期待できる．

2. 基本事項

本稿では瞬時に復号可能な2進符号だけを考える．瞬時に復号可能な符号で平均符号長が最小になるという意味でハフマン符号[2]がよく知られている．本稿で議論する順序保存符号☆（以下ap符号と呼ぶ）とは，瞬時に復号可能という条件を満たしたまま，ソースシンボルの間に成り立つ全順序が符号語の上でも成り立つような符号をいう．

ap符号の中で平均符号長最小のものを最適ap符号という．

以下では簡単のため，ファイルはレコードの並びであり（いわゆるストリーム），1レコードには1つのデータだけが格納されるものとする．最適ap符号は順アクセスされるファイルの符号化にそのまま利用でき，符号化後のデータ量を最小化する符号である．最適ap符号の計算アルゴリズムは文献1)で与えられている．ファイルの任意の位置から読み始めてもレコードの区切りが識別できるap符号は文献3)で議論されている．

☆ アルファベティック符号[1]と呼ぶべきであるが，文献3)に合わせてこの訳語を用いる．

情報検索ではレコードを固定長で記憶すると便利な場合が多い．符号化後のレコードが固定長になるように符号化する場合には，最適ap符号が適当とは限らない．ファイルを固定長レコードで記憶する場合には，レコード長はデータの最大長とし，レコード長より短いデータは，データの後ろに詰め文字（空白が用いられることが多い）を置く．こうした場合の符号化としては，各レコードを符号化した後の長さの最大値が最小になるような符号化が望ましい．この性質を満たすap符号を固定長レコード用ap符号（以下fap符号と略す）と呼ぶ．fap符号は一般には可変長符号となる．fap符号の計算は文献3)で議論したように非常に難しい．ここでは次のような方法を用いた．

（1） k をデータの最大長にセットする．

（2） 長さ k 以上のデータだけを対象にして，各文字頻度を数え，最適ap符号を求める（この場合頻度0の文字の頻度は1に比べて非常に小さい正の数として計算する）．

（3） このap符号で長さ k 以上のデータを符号化したときの最大長を M とする．

（4） 長さ k 未満のデータを同じap符号で符号化したときの最大長を m とする．

（5） $m \leqq M$ ならば終了．$m > M$ なら，k を1減じて(2)を繰り返す．

終了時のap符号が求める解である．

この方法で得られた解が最良である保証はないが，今回の実験では，データ圧縮に利用しうる程度の解が得られた．

[例1] 集合 E={aabab, acbacbc, baabb, badcc} を考える．E に出現するシンボルの頻度，E に対する最

表1 例1の集合Eに対する各種の符号
Table 1 Codes for the set E in Example 1.

シンボル	頻度	最適ap	fap	ハフマン
デリミタ	4	00	不要	100
a	8	01	001	00
b	8	10	01	01
c	5	110	10	11
d	1	111	11	101

最適ap	010110011000 011101001110101100	
	100101101000 100111111011000	58ビット
fap	0010010100101000 0011001001100110	
	0100100101010000 0100111101000000	64ビット
ハフマン	0000010001100 00110100110111100	
	0100000101100 01001011111100	57ビット

図1 集合Eを表1に従って符号化した結果
Fig. 1 The set E coded by the codes of Table 1.

適 ap 符号，fap 符号，およびハフマン符号を**表1**に示す．

集合 E をこれら3つの符号で符号化した結果を**図1**に示す（わかりやすいようにレコード間に空白をあけてある）．fap 符号では詰め文字としてビット0を使うので，すべて0からなる符号語は使えない点に注意が必要である（詰め文字を誤って符号語と見なすことがあるため）．

3. 人名データを用いた実験結果

3.1 人名データの統計量

符号化に当たって，本学在学生 5,265 人の人名をデータとして用いることにした．1つのレコードは人名（漢字）とその読みがな（半角カタカナ）および1バイトのディレクトリ（漢字の姓の長さを表すのに2ビット，姓の読みがな長に3ビット，名の読みがな長に3ビットを割り当てる）からなるように構成した．このファイルを orgfile と呼ぶ．

人名の場合には「五十嵐(イガラシ)」や「井上（イノウエ)」のように1字ずつ分けられないものや分けると不自然になるものがある．そのようなデータは熟語として1つのシンボルとして扱うことにした．こうした熟語は全部で 98 であった．orgfile に出現する姓の種類は 1,813 種，名前の種類は 2,768 種であった．読みの違う漢字は異なるシンボルと考えるので，orgfile に出現するシンボルは 1,669 種類（漢字 1,610，ひらがな 44，カタカナ 15)，そのうち，名前部分に出現するシンボルは 991 種であった．本学の場合，1つの漢字あたり平均 1.7 の読み方があった．

3.2 符号化の実験結果

前節の頻度をもとに順序保存符号化を行った．電話帳順序を仮定した場合，orgfile を最適 ap 符号によって符号化したファイルを ap_1，fap 符号で符号化したファイルを fap_1 とする．fap_1 の結果は5文字以上の人名に対する最適 ap 符号であり，すべての人名を 54 ビット以内で表現できる．

国語辞典順ではたとえば「いか，いが，いかいよう」の順に単語を並べなければならない[4]．また1つの漢字の読みがなの長さは可変長であるため，漢字列を読み順に並べた場合，同じ読みがなの漢字が連続しない場合が生じる．例えば読み順では「早紀（サキ)，里子（サトコ)，早百合（サユリ)」の順に並ばなくてはならない．

こうした理由から，漢字1字ずつをソースシンボルとする国語辞典順の ap 符号化は困難であることがわかる．姓や名全体を1つのシンボルとして符号化すれば，順序保存符号が得られるが，5,000 人程度では圧縮効果がないことがわかった（姓のうち 64%，名前のうちの 74% が頻度1であった．このため符号化すれば orgfile よりファイルサイズが大きくなった)．ここではそれらの折衷案として，姓は姓全体を1つのソースシンボルとし，名前は1字ずつをそれぞれソースシンボルとして別々に符号化した．異なる姓の種類は 1,813 であったので 11 ビットで符号化できる（姓を固定長で符号化した理由は，姓の大小比較を復号せずに符号語のままで行えるようにするため)．名前部分を最適 ap 符号によって符号化したファイルを ap_2，fap 符号を用いて符号化したファイルを fap_2 とする．レコード同士の比較は前 11 ビットの大小がレコードの順序を表すが，前 11 ビットが等しい場合（同姓の場合）は残りを復号して読みがなを生成し，読みがなの順序比較を行う必要がある．fap_2 の名前部分は長さ3以上の名前を対象にしたもので，$M=29$ ビットであった．

以上のファイル構造とファイルの大きさを**図2**，**表2**に示す（区切りの頻度は 5,265 で，ap_1，ap_2 ともにその符号語は長さ2の 00 であった)．

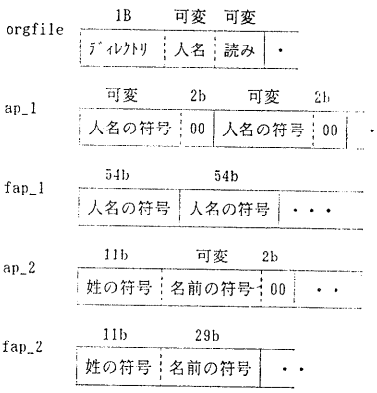


図2 各ファイルの構造
Fig. 2 The structure of each file.

表2 各ファイルの大きさ
Table 2 The size of each file.

ファイル名	ファイルサイズ	補助ファイル
orgfile	85798 B	—
ap_1	26743 B	18835 B
fap_1	35539 B	18835 B
ap_2	19855 B	37977 B
fap_2	26325 B	37977 B

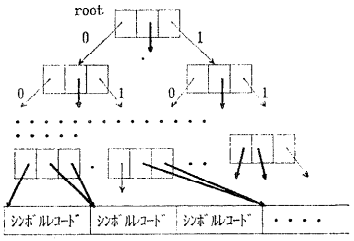


図 3 符号化および復号のためのファイル
Fig. 3 The dictionary file for encoding and decoding.

fap_1 と fap_2 は固定長ファイルなので任意のレコードを直接アクセスできるが ap_1 と ap_2 は先頭から復号しながらレコードを読み出す必要がある.

orgfile 以外は読みがなを省略できるが, 符号化と復号化のための辞書ファイルを別に作成しておく必要がある. 復号と符号化を同時に行うために, ここでは**図 3** に示すデータ構造を準備した (図 3 において, 細線矢印は索引の節点へのポインタ, 太線矢印はシンボルレコードへのポインタを表す). 表 2 には, 図 3 のデータ構造を記憶する容量も示されている. ap_2 と fap_2 では姓用と名前用の 2 種類の辞書を置く. それらのサイズはそれぞれ 26,986, 10,991 バイトであった.

図 3 の索引部は完全 2 分木であり, その節点数はシンボル数−1 である. 1 つの節点は 3 つのポインタからなり, その大きさは 6 バイトとした. 各ポインタは左から順に, 左部分木の根, 右部分木に属する最も小さいシンボルレコード (符号化のために使う), 右部分木の根を指している. シンボルレコードは漢字フィールド長, 読みフィールド長を記憶するディレクトリ 1 バイトの後に, 符号化される漢字列または単漢字とその読みを続けたもので, アルファベット順に並べられている. 図 3 を用いた符号化, 復号化のアルゴリズムは割愛する.

この結果, 符号化・復号化のファイルを加えてもデータ量が 53.1% から 75% ですむことがわかった. 国語辞典順では ap 符号化のメリットを一部犠牲にした折衷案で符号化を行ったが, それでも圧縮率は電話帳順に比べて 12% から 14% 劣った.

データ検索の場合は, 検索のキーとなるデータの入力にはこの補助ファイルを用いた仮名漢字変換を行うことを前提にしている. そのため検索キーの入力時に行われる変換作業の過程で, 入力漢字とその読みがなのペアが確定され, 検索キーに対応したコードを一意に決めることができる.

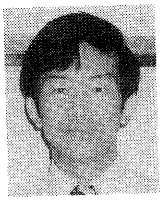
4. おわりに

本稿では漢字データの順序保存符号化法の有効性を実験的に調べた. 符号化は, 入力されるデータの統計量に基づいて行われるが, 統計量が大きく変化したり, 新しいシンボルが出現した場合には, 符号の再計算が必要になる. 最適 ap 符号の動的な計算法はまだ知られていない. たとえそうしたアルゴリズムが得られたとしても, すでに符号化済みのデータの再符号化が必要になりその処理コストは莫大になる. この問題はこうした静的符号化全般が抱える問題点である.

参考文献

1) Hu, T.C. and Tucker, A.C.: Optimal Computer Scarch Trees and Variable-Length Alphabetic Codes, *SIAM J. Appl. Math.*, Vol. 21, No. 4, pp. 514-532 (1971).
2) Lelewer, D.A. and Hirschberg, D.S.: Data Compression, *Comput. Surv.*, Vol. 19, No. 3, pp. 261-296 (1987).
3) 中津: 順序保存符号とその情報検索への応用, 情報処理学会論文誌, Vol. 34, No. 2, pp. 312-319 (1993).
4) 西尾, 岩淵, 水谷編: 岩波国語辞典, 第 4 版, 岩波書店 (1986).
5) 丹羽監修, 日本ユニバック編: 日本の苗字, 表音編, 日本経済新聞社 (1978).

(平成 4 年 12 月 10 日受付)
(平成 5 年 7 月 8 日採録)



中津 楢男 (正会員)

1953 年生. 1977 年京都大学工学部情報工学科卒業. 1979 年同大学院修士課程修了. 同年愛知教育大学教育工学センター助手. 現在, 同大学総合科学課程情報科学コース助教授. 工学博士. データベース, 効率よいアルゴリズムの設計, 記憶と学習に興味を持っている. 電子情報通信学会, ACM 各会員.

片桐 友子

昭和 44 年生. 平成 4 年愛知教育大学教育学部総合科学課程情報科学コース卒業. 同年, 富士通 VLSI(株) 入社. 現在は富士通・三重工場内にある VLSI 研究所にて半導体の品質管理 (主に統計的な品質管理) について研究中.