情報処理学会第37回(昭和63年後期)全国大会

6X-3

書誌情報のファイル格納に関する効率評価

駱 琴, 渡邉 豊英, 吉田 雄二, 稲垣 康善, 柘植 利之
(名古屋大学 工学部) (名古屋大学附属図書館)

1. はじめに

大量の図書目録カードを保有する図書館では、書誌情報をデータベースとしてオンラインで効率良く利用できるようにすることが、今日最も重大な課題となっている。しかし、格納すべきデータ量が膨大であること、新規に購入する書籍に対処しなければならないこと、データベース化するための環境(データ作成のための経費・作業量、計算機システムの処理能力・蓄積能力)が不十分であることなどが原因で、データベース化に対応できないのが実状である。大量の書誌情報をデータベース化するには、データ量が多いことに起因する問題を解決しなければならない。すなわち、データ作成の効率化、データ格納の効率化、データ処理の効率化である。

本稿では、大量の書誌情報を計算機処理可能なデータとしてファイルに格納する場合の問題、すなわちデータ格納の効率化について言及する。格納効率を向上させるには、データを簡潔に表現し、重複性を避け、コンパクトな構成を図る必要がある。このような要求を実現する方法の一つにデータ圧縮技法があり、比較的簡単な処理で適用効果を期待できる。議論の視点は、書誌情報を画像データとコード・データの形態で表現し、適切なデータ圧縮技法をそれらのデータ形態にそれぞれ適用し、データ格納率をどの程度向上させるかを調べ、適用効果を評価することである。

2. 書誌情報の特性概要

大容量のファイル装置が利用可能な現在では、書誌情報を計算機処理可能なデータとしてファイルに格納するには、コード・データと画像データの形態が考えられる。書誌情報をコード・データとして作成すれば、端末などのキーボード入力装置から1記号ずつ入力しなければならず、膨大な経費・作業量が必要となる。一方、画像データでは、スキャナなどの画像入力装置から機械的に一度に入力され、比較的簡単に作成可能である。もちろん、画像データ、コード・データで扱うことの相違はそのデータ量、処理のし易さなどにも影響され、作成効率だけでは優劣を決定できない。少なくとも、格納効率、処理効率の観点から画像データ、コード・データに対する扱いを判断すれば、コード・データの方が現状の計算機システム技術に合致している。

今、データの格納効率を主に検討する立場では、書誌情報をファイルに格納するために、そのデータ量が問題になる。和書・洋書の図書目録カードそれぞれ1000枚に対してデータ量を算定した。コード・データの形態では図書目録カードに記載された書誌項目に従って分類し、洋書は1バイトのEBCDICコードで、また和書は2バイトのJEFコードでデータを作成した。一方、画像データの形態では図書目録カードを原寸大(1カード1168*704画素)でイメージ・スキャナから2値画像として読み込んでデータを作成した。表1がその結果である。コード・データに対して画像データは500倍以上のデータ量である。また、画像データでは単にそれだけを格納しただけでは検索できず、検索の手掛かりとなるコード・データが必要になる。少なくとも、図書目録カードは何百万件と図書館に保管されており、よりコンパクトに表現してファイル容量を減らすことが必要である。

表1 図書目録カード1000件のデータ量

	コード・データ	画像データ	比率
和書	188,410バイト	102,784kバイト	545
洋書	199,754バイト	102,784kバイト	515

「比率=画像データ量/コード・データ量」である。

表2 図書目録カード1000件のコード・データに対する定量的な特性

	和書		洋書	
	文字単位	熟語単位	文字単位	単語単位
総記号数	94,205	29,183	199,754	40,499
異記号数	1,649	9,169	88	10,812
平均記号長	2バイト	6.9バイト	1バイト	4.9バイト
エントロピー	7.5ビット	11.1ビット	5.4ビット	10.7ビット
最大可能な圧縮比	2.12	5.00	1.50	3.69

図書目録カードのコード・データの特性を簡単に述べる。書誌項目は通常の目録規則に沿って、各データが設定されている。表2はデータ量に関する定量的な分析結果である。和書の熟語単位とは、人為的に熟語をデータ列に同定して、データ列中に分ち書き記号を設定した。もちろん、この挿入記号は表2には計上していない。データ量がそれ程多くないにもかかわらず、和書・洋書ともかなり多くの記号列(単語、熟語など)が使用されている。和書の熟語単位では平均記号長が約3.5文字(6.93バイト)とほぼ我々の用いている熟語長と等しく、また洋書の単語単位では平均記号長が約5文字(4.93バイト)で一般的に認められている事実と合致している。最大可能な圧縮比はHuffman法を用いれば、ほぼこれに近くまで圧縮比(=原データ量/圧縮されたデータ量)を達成できることを表し、一文字単位の適用よりも、熟語・単語単位の適用の方がより大きな効果を得ることができる。これらは現在用いられているEBCDICコード、JEFコードの内部表現に伴う冗長度を表していて、エントロピーで示したビット数があれば、

Effects of Data Compression Method for Library Catalog Data
Qin LUO[1], Toyohide WATANABE[1], Yuuji YOSHIDA[1], Yasuyoshi INAGAKI[1] and Toshiyuki TSUGE[2]
1:Faculty of Engineering, Nagoya University, 2:University Library, Nagoya University

最適に記号列を表せることになる。例えば、洋書では１０．６９ビットあれば、完全に個々の単語を区別でき、和書では１１．０７ビットで個々の記号列を区別できる。

3．データ圧縮技法の適用

原データに付帯する冗長な表現、原データを構成する冗長な成分に着目して、その冗長な情報を抑止・縮退させることにより、原データをより簡潔な表現で置き換えて、取り扱うデータ量を縮小させる技法がデータ圧縮技法である（正確には、原データから許容範囲内で情報欠損を図って、データ量を縮小させるエントロピー性データ圧縮技法に対して、この種のデータ圧縮技法を冗長性データ圧縮技法という。本検討では、冗長性データ圧縮技法を考察する。）。計算機システムにおいて、取り扱うデータ量を縮退させることは次の効果を期待できる。

（１）格納に必要なファイル容量を減少できる。

（２）ＣＰＵ（主記憶）とファイル（補助記憶）間のデータ転送時間を短縮できる。

しかし、データ圧縮技法を適用することによって、必要となるデータの圧縮手続き、復元手続きの処理に伴うオーバヘッドがあり、実際には期待する効果を圧縮対象のデータの性質に合わせて検討し、適切なデータ圧縮技法を選定しなければならない。

本実験では、和書・洋書の書誌情報に対してコード・データの場合にＨｕｆｆｍａｎ法、Ｚｉｖ－Ｌｅｍｐｅｌ法を、画像データの場合にＨｕｆｆｍａｎ法を用いた。Ｈｕｆｆｍａｎ法は最適符号化法として有名な方法で、統計的に独立なデータ・リソースに対して適用した場合に最良の結果を示す。しかし、統計的に出現確率が高い語により短い圧縮コードを、そうでない語により長い圧縮コードを割り当てる方法であるために、原データの統計的な性質に強く依存する。統計的な解析結果と異なるデータ・リソースに圧縮コードを適用すると、期待する効果が得られない。実験では、画像データに対してはＣＣＩＴＴに定められたＭＨ法に基づいたハフマン・コードを利用し、コード・データに対しては統計的にデータ・リソースを解析し、圧縮コードを設定した。一方、Ｚｉｖ－Ｌｅｍｐｅｌ法は万能符号化法と呼ばれ、マルコフ連鎖のデータ・リソースに適用可能である。Ｈｕｆｆｍａｎ法と異なって、データ・リソースの統計的な性質に依存せずに圧縮コードを設定できる。Ｚｉｖ－Ｌｅｍｐｅｌ法ではそのデータ・リソースに対する圧縮アルゴリズムの相違から、ｕｎｉｖｅｒｓａｌ法とｉｎｃｒｅ－ｍｅｎｔａｌ法がある。ｕｎｉｖｅｒｓａｌ法は既成成分から複製・生成手続きを基本として、またｉｎｃｒｅｍｅｎｔａｌ法は既成成分の複製手続きを基本として符号化する。Ｈｕｆ－ｆｍａｎ法は統計的に独立性が高い語に対して有効に働き、英文のように各単語が空白で明確に区切られてデータ・リソースに適用可能であるが、和文のように語の単位が明確でないデータ・リソースには適用が難しい。また、Ｚｉｖ－Ｌｅｍｐｅｌ法はマルコフ連鎖のデータ・リソースに適用できるために、和文のように分ち書きされていないデータ・リソースにも容易に適用できる。

4．圧縮実験の結果

コード・データ形態の書誌情報に対して、Ｈｕｆｆｍａｎ法、Ｚｉｖ－Ｌｅｍｐｅｌ法を適用した結果を表３に示す。Ｚｉｖ－Ｌｅｍｐｅｌ法はＨｕｆｆｍａｎ法で一文字単位に圧縮処理を行なった場合に比べて、和書・洋書とも優れた結果を示しているが、Ｈｕｆｆｍａｎ法で熟語・単語単位にデータ圧縮技法を適用した場合に比べるとかなり劣っている。また、Ｈｕｆ－ｆｍａｎ法は一文字単位で圧縮するよりも、熟語・単語単位で圧縮した法が遙かに良い結果となっている。これはＨｕｆ－ｆｍａｎ法が語列の独立性が高いデータ・リソースに対する適用程、良い結果を示すことを明らかにしている。反対に、Ｚｉｖ－Ｌｅｍｐｅｌ法は統計的な語列の独立性を仮定していないために、Ｈｕｆｆｍａｎ法の一文字単位の圧縮結果よりも良い結果となっている。すなわち、和書のように語列の分ち書きが自動的に難しいデータ・リソースには、Ｚｉｖ－Ｌｅｍｐｅｌ法が効果的である。洋書のように語列の分ち書きがあるデータ・リソースの場合には、Ｈｕｆｆｍａｎ法が適していることが実証された。

表３　コード・データの圧縮結果

		和　書	洋　書
Huffman法	文字単位	2.00	1.37
	熟語・単語単位	4.82	3.51
Ziv-Lempel法	universal 法	2.22	2.87
	incremental法	2.54	2.01

次に、画像データの和書に関する書誌情報に対して、Ｈｕｆｆｍａｎ法を適用した結果を表４に示す。コード・データに比べて、かなり圧縮比は良く、画像データの場合には冗長性が高いことが明白である。個々の図書目録カードによっては、２倍以上の差がある。画像データ全体では１３倍以上に圧縮されている。洋書の書誌情報に対する適用結果はまだ調べていないが、圧縮アルゴリズムから推定して、ほぼ同様の結果を得られる。

表４　画像データにおける和書の書誌情報に対する圧縮結果

平均圧縮比	最小の圧縮比	最大の圧縮比
13.17	8.61	17.96

このようにデータ圧縮技法の適用によって、コード・データと画像データの書誌情報を縮小できるが、画像データの場合はコード・データに比べて、まだ約２００倍のデータ量となっている。

5．おわりに

本稿では、書誌情報をファイルに効率良く格納する視点から、データ圧縮技法の適用効果を検討した。その結果、必要なファイル容量はコード・データでは１／３以下に、画像データでは１／１３以下に減らすことができる。また、和書の書誌情報のように分ち書きが自動的に難しいデータ・リソースにはＺｉｖ－Ｌｅｍｐｅｌ法が、洋書の書誌情報のように比較的語列の独立性が高いデータ・リソースにはＨｕｆｆｍａｎ法が有効である。しかし、データ圧縮技法を適用しても、画像データはコード・データに比べ、約２００倍のファイル容量を必要とし、データ作成が容易であるのに反して、データ格納、データ処理では問題が多い。今後、画像データのデータ・リソースの扱いに関して、より適したデータ圧縮技法の開発、画像データからコード・データへの自動抽出技術の研究が課題である。