大規模なリファレーション DB の開発
―文献同定のための APTS コードの導入―

○浅井　勇夫*

Development of large reference information databases:
Introduction of APTS code for identification of scattered documents.

ASAI Isao*

大規模なリファレーション・データベースの開発には参照文献の入力を必要とする。ここでは，多くの参照文献に点在した文献を同定するために新しいコード APTS を導入する。

APTS コードは参照文献の書誌データにある著者，発行年，標題，そして出典の 4 つの部分から構成する。ここでは，APTS のコード化ルール，コーディング時のエラーの原因，そして，今後の課題について扱う。〔著者抄録〕

The development of large reference information databases needs the input of reference data. This paper introduces a new code "APTS" for the identification of scattered documents. The APTS consists of four search items: author; publication year; title; and source of each of the references from which the bibliographic data are taken. This deals with coding rules of the APTS, causes for the coding error, and several points to be checked in the future.〔Author Abs.〕

＊　大阪府立大学工学部経営工学科（〒 591 堺市学園町1-1） Tel. 0722(52)1161
＊　University of Osaka Prefecture, Dept. of Industrial Engineering (1-1, Gakuencho, Sakai, 591)

1. はじめに

文献末尾にある参照文献をデータベース（DB）化したリファレーションDBは，多くの検索問題を解決し，新しい情報検索の世界を拓く。これは，筆者により，小規模なDBで明らかにしている[1)~3)]。

文献のリファレーションは参照文献，被参照文献，そして文献自体から構成する。リファレーションは文献の一つの属性であり，文献間のリンクを表している。図1は，文献428が9編の文献とリンクしている例を示す。リファレーション属性を持つ文献DBをリファレーションDBと呼ぶ。

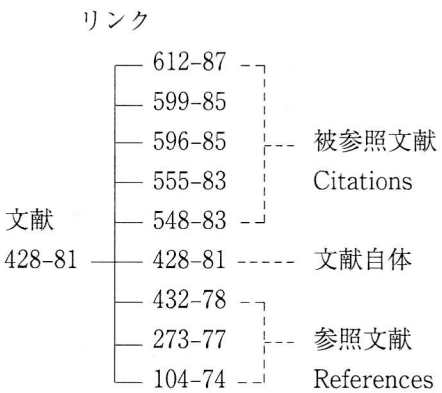

図1　文献428のリファレーション

従来の検索は，検索キーにマッチした文献だけを検索するが，リファレーション検索は文献間のリンクの度合い（関連性）が強いものを検索する。そのため，検索キーにマッチしない文献も検索可能になり，検索結果は関連性の高い順に表示できる。さらに，専門分野の情報構造であるキーワード間，著者間，そして出典間の関係をマッピングすることが可能になる。

被参照文献は参照文献を転置して得られるが，参照文献は入力しなければならない。各所で参照文献データを入力することは重複が生じて不経済なため，大規模なリファレーションDBを開発し，提供することが望まれる。参照文献中の文献書誌事項を全文入力したとしても，文献を自動的に同定するbとは難しい。ここでは，この参照文献の入力問題を解決するために，新しい文献コード（APTS）の開発を試みる。

2. 参照文献の入力

2.1 SCI や SSCI の参照コード

文献属性として参照文献をもつDBとしてISI社のSCI（Science Citation Index）やSSCI（Social Science Citation Index）がある。これは約30年前に開発された非常に貴重なDBの一つである。

SCIやSSCIのコード化の例を示すと，次のとおりである。

SMITH, L.C. (LIBRARY TRENDS, V30, P83, 1981),

LAWANI S.M. (J AM SOC INFORMATION, V34, P59, 1983)。

参照文献のコードは著者と出典に関する書誌項目を一定のルールで作成する。1つの文献を30~50バイトで表し，逐次刊行物だけでなく，単行本やレポートなどもDB化する。

SCIやSSCIを使った被参照分析の研究が多数なされており，情報学では1つの分野を形成している。コード化した参照文献のバイト数が多いため文献を同定し難く，現状のシステムではリファレーションの作成は難しい。

2.2 大規模な DB のコードの活用

参照文献を文献コードで構成すれば，入力は簡単になる。参照文献が既存の文献DB内の文献に該当すれば，その文献コードを付与するだけでよい。例えば，JICSTの科学技術文献ファイルは十数年で800万編以上の文献を蓄積しており，文献の寿命からみて，十分利用可能である。

JICSTの文献コードの例を示せば，次のとおりである。

81A0172918　　B91051798。

この場合，参照文献の入力支援システムを開発する必要がある。大規模なDBから文献の同定が可能な量に短縮したDBを作成しCD-ROM化すれば，パソコンで処理可能である。

しかし，DBはすべての文献を網羅していないため，コード化できない場合がでてくる。単行本

に関するDBを統合すれば，参照文献のコード化率は高くなる。コード化できない文献をいかに扱うかが課題である。

DBの利用は大規模な特許ファイルに対して有効である。網羅性が100%の特許ファイルは参照関係が蓄積され，それから被参照がつくられ，そして，リファレーションが作成できれば，技術情報の有力な分析手法として有用になる。

3. 新しい文献同定コード APTS の開発

3.1 文献の APTS コード

参照文献は文献の基本的な書誌データから構成されている。その書誌データから文献コードを作成できれば，参照文献の入力は簡単になる。誰でも，どこででも，簡単なルールで文献をコード化できるのが望ましい。

1年間に400～500万の文献が生産され，毎年増加している。文献に通し番号を付ける場合，1億の文献で8桁，それに西暦年の4桁を加えて合計12桁あれば十分である。しかし，すべての文献に通し番号を付加することはコストもかかり，実際問題として実現不可能である。

ここでは，新しい文献コードとして，著者，発行年，標題，そして出典をそれぞれ4桁で表し，合計16桁のコードを考える（図2）。そして，このコードを各構成要素の英文の頭文字からAPTSコードと名付ける。

□□□□+□□□□+□□□□+□□□□			
著者	発行年	標題	出典
Author	Publication Year	Title	Source

図2 16 桁の APTS コードの構成

この場合，著者はアルファベットで表し，各桁10通りとして10,000通り，標題はアルファベットで表して10,000通り，出典は数値3桁分とみなして1,000通りの区別ができるものとする。この場合，理論上，毎年1,000億の文献が生産されても重複は生じない。

次に，参照文献が英文の場合のコード化ルールを各項目ごとに説明する。

3.2 著者のコード化

著者のコード化を検討するために658人の著者名を入力したデータファイルを作成した。Last名の平均は6.5字であった。JGAWK言語を用いて，コード化のルールに対応したプログラムをそれぞれ作成し，コードが重複する数と延べ人数を求めた。4桁のコードとして，次の結果を得た。

(1) Last名の最初から4桁
　重複は71で，延べ人数は162名
(2) Last名の最初から3桁とFirst名の頭文字1桁
　重複は16で，延べ人数は34名
(3) Last 名の最初から2桁と最後の1桁，First名の頭文字1桁
　重複は8で，延べ人数は17名

テストしたデータ量は少ないが，明らかに顕著な結果が現れたため，第3の方法を採用した。著者の表示は，「Last名のフルスペル，カンマ，First名の頭文字」で表す場合が多く，コード化しやすい点も考慮した。例えば，

Bradford, S. C. → BRDS

とコード化する。

著者名にも例外的な場合があり，それを列挙すると，

Yu, C. T. → YUUC
MacRae, D. → MAED
Spack Jones, K. → SPSK

などである。この場合，原則として「著者の最初の2文字とカンマをはさんだ2文字」のルールを適用する。

3.3 発行年のコード化

発行年は西暦で4桁の数値で表す場合がほとんどであり，コードはその数値を転記するだけである。発行年の位置は著者のすぐあとにある場合と標題のあとの出典のなかにある場合と半々の割合である。例えば，次のような場合は，

Bradford, S. C.(1934). → 1934

となる。西暦年の下2桁だけで発行年を表し，コードを2桁に短縮することも可能であるが，4桁

の方がコード化のミスは少なくなり2000年以降の年代順のソートは簡単になる。

3.4 標題のコード化

著者の場合と同様に，コード化を検討するために1,180件の標題を入力したデータファイルを作成し，いろいろなパターンのプログラムを作り，検討した。標題は平均62バイトあり，平均8.7単語，また，3字以内の単語を除外した場合は平均5.7単語で構成されている。

標題の中に特殊文字（, . ' " / -）がある場合，例えば，スラッシュやハイフンのある単語は，それを空白に置き換えて，2つの単語に分けるなど，コード化の前に処理しておく。

標題の場合は，平均48字から4字を選択することになり，非常に多くの選択方法が考えられるが，次のような場合のコード化の基準を設定して，コードが重複する数と延べ標題数を求めた。

(1) 標題の最初から4単語の頭文字
重複は87で，延べ数は216標題
(2) 標題から不要語を除いた4単語の頭文字
重複は45で，延べ数は95標題
(3) 3字以内の単語を除いた4単語の頭文字
重複は45で，延べ数は103標題

ここで，(2)の不要語は2字以内の単語とand, for, its, the, withの5単語を使用した。

最初は(2)の方法を採用したが，実際にコード化の作業をしてみると，不要語のルールに慣れるまで時間がかかり，ミスをする。したがって，最終的には簡単なルールの(3)を採用した。

標題の単語が1～3語の場合には例外処理をする。その場合，最後の単語の最後の部分から不足する数の文字を補うことにした。例えば，

3語 Linguistics and information science → LISE
2語 Citation Analysis → CAIS
1語 Bibliometrics → BICS

などである。資料への目の動きから，自然な方法を採用した。

3.5 出典のコード化

書誌データの中で一番扱いにくいのが，この出典データである。ある文献が多数の文献に参照されている場合の記載を比較すると，著者，発行年，標題の記載は大体同じであるが，出典は微妙に異なっている場合が多い。

参照文献は資料別に記載方法が異なるため，そのパターンを把握すれば，資料の種類は判断できる。約5,300編の参照文献の資料別分布を調べた結果，逐次刊行物が59%，単行本が19%，編集物が17%，残り5%がレポートなどであった。ここでは，資料別に3種類のコード化ルールを決めた。

3.5.1 逐次刊行物

どの文献にも含まれる書誌データは，誌名，巻，開始頁である。誌名は省略形の場合があるため，思い切って割愛し，巻の下1桁と開始頁の下3桁でコード化する。例えば，

Journal of Documentation, 28, 11-21 → 8011

となる。

3.5.2 編集物

会議の予稿集や論文集，それに論文を編集した単行本などがここに該当する。標題の後ろに，In ～(Eds.)，In proceedingsなどと記載され，開催地，頁，発行所などが続く。この場合も会議名や本の標題を割愛し，編集物を示す記号「=」と3桁の開始頁でコード化する。例えば，

In proceedings ～ (pp. 248-253). Medford, NJ: Learned Information, Inc. → =248

である。

3.5.3 単行本・レポートなど

単行本，レポート，学位論文など逐次刊行物や編集物に該当しない参照文献を扱う。単行本は著者，発行年，本の標題（イタリック），発行地：発行所のパターンが多い。コードは発行所を表す主要な単語の最初から4字を用いる。例えば，

Academic Press → ACAD
University of Illinois → ILLI

などである。

したがって，出典コードの1桁目が数値，=，英字かで逐次刊行物，編集物，あるいは単行本などかが分かる。

3.6 日本語文献のコード化

日本語文献のコード化は英文の場合に準じる。漢字1字は2バイト必要なので，英字の4字は漢字では2字である。著者コードは姓と名から最初

の1字ずつで構成する。発行年コードは半角4字で表す。

標題コードは最初の2単語の頭文字各2バイト分で構成する。単語の区切りは，ひらがなや特殊記号のあるところと，漢字，英字，カタカナの文字列が他の文字列にかわるところとする。英字やカタカナは半角に変換し，カタカナの濁音や「ー」は除く。例えば，

情報解析の教育の経験　→　情教
SCISEARCH Fileによる低温　→　SC低
文節索引のテーブル表示　→　文ﾃﾌ

などである。出典コードは英文と同様に扱う。

4. 考　察

コード化の評価基準を列挙すると，次のようになる。

1) ミスを防ぐために簡単なルールにする
2) コードの総バイト数を少なくする
3) 数十年分の文献量に対応できる
4) 既存のDBから自動的にコードができる
5) 多くのDBにオープンである
6) 同一文献は同一コードを作成する
7) 重複コードを作成しない

ここでは，コード化を検討するために英文の逐次刊行物4年分の約5,300編の参照文献をコード化した。そして，著者，発行年，標題，出典，の4種のサブコードが全部合致するもの，また3種が同じで1種が異なっているものをリストアップするプログラムを作り，どこでエラーが発生したかを検討した。そして，いろいろ試行錯誤を経て，前章のルールに到達した。

コード化に要する時間は，1時間に約100編の参照データをコード化可能である。1文献当たり約20～30の参照文献があるので，1時間に3～4編の文献をコード化可能である。

エラーが発生したところはおもに2か所ある。1つは書誌データ自体，もう1つはコード化の際に生じる。前者に関しては，同じ刊行物内でも書誌データの記載が微妙に異なっており，ときどき発行年や頁などの数値の記載ミスや記載漏れがあり，特定の文献に集中していることを発見した。

後者に関しては，著者のLast名が複雑な場合，標題の副題や長い標題の後半部分がない場合，発行所や大学名をコード化する場合などに発生する。細かい根気のいる作業であるため，コード化を支援するソフトの開発が不可欠である。

誌名や会議名などを割愛したため，コード化は簡単になり，非常に速くなった。また，バイト数が少ないため，パソコンで処理できた。全般的にいえば，非常に満足のいく結果が得られた。

5. おわりに

21世紀を指向した新しいDBと情報検索，それは大規模な三次情報DBの提供である。大規模なDBは5～10年のデータの蓄積が必要で，その準備を開始する時期にきている。大規模な三次情報DBを新たに開発するには，時間と費用のかからないデータ入力方法を開発しなければならない。

ここで開発した文献のAPTSコードは1つの有力な方法である。現在の二次情報DBがAPTSコードを持ち，互いにリンクするようになれば，その相乗効果は大きい。さらに，全文献を網羅する文献台帳の必要性を痛感した。

参照文献

1) 浅井勇夫．Referationを用いた特定専門分野情報の分析．第22回情報科学技術研究集会発表論文集．日本科学技術情報センター，東京，1985，p.135-142.
2) Asai, I. Referation Database for Document Information Analysis. Proc. 50th Meeting of the American Society for Information Science. Boston, 1987, p.1-5.
3) 浅井勇夫．リファレーション型データベースによる知的検索の可能性．第19回ドクメンテーション・シンポジウム予稿集．東京，1989，p.36-40.

質　疑　応　答

質問　川島健也（武田薬品工業㈱）

一次情報作成者が引用文献を間違っている場合がよくあるので，リファレーションデータベース作成時に誤引用をチェックできるようなシステムを作って，修正機能として使えるとよいのだが。

回答　文献台帳があれば，そのようなことが可能である。現時点ではこのシステムでは著者名，発行年，標題，出典のうち4項目または3項目が合致するもののリストが出力できるようになっているので，それを利用して修正を行うことができる。

質問　辻　義人（海洋科学技術センター）

コード番号を4桁にしたのはなぜか。文献の重複率との関係から6桁の方がより正確なのではないか。

回答　桁数が増えれば，正確になるというものでもない。入力の都合からいくと，本当は3桁ぐらいの方が覚えやすくミスも少ないが，重複率等を考えて4桁とした。

1992 第29回
情報科学技術研究集会
発表論文集

日本科学技術情報センター
The Japan Information Center of Science and Technology