

3. インフォメーション・リトリバーの展望

関根 智明， 昆野 誠司（慶応大学）

1. 概 説

1.1 インフォメーション・リトリバー

リトリバー(Retrieval)という言葉は元来、「取返すこと」とか「回復すること」といつて意味を持つた言葉である。インフォメーション・リトリバー(Information Retrieval)がわが国では一般に「情報検索」と訳しており、また単に略してIRとも呼んでいる。一口で言えば「インフォメーション・リトリバー(情報検索)とは蓄積してある情報の中から必要な情報を探し出すこと」となるであろう。

この情報検索という仕事とそれに関連する仕事をまとめて一般的に述べると次のようになる。

(i) 情報の収集および評価

情報を集め、その情報が正しいものであるかどうか、また貯えておく必要があるかどうかを判断し、取捨選択を行う。

(ii) 探し出すときの便宜を考えて、個々の情報に適当な“見出し”をつける。

(iii) 探し出す手間(時間)ができるだけ少なくなるような方法で、それらの情報を見出しに従って整理し、貯えておく。

(iv) 要求に応じて特定の情報を探し出す。すなわち、質問(問合せ)を分析し、見出しに変換して、これと蓄積してある情報を見出しとを照合する。

(v) 探し出した情報を見やすい形にして、必要としている人に提供する。

1.2 ドキュメント・リトリバーとファクト・リトリバー

今ある人が「情報検索機械」について、知りたがっているとす。その人が必要な情報を得るまでの過程を考えると次の二通りに分けられる。

(i) 「検索機械」についての専門家の所へ行き、それについて詳しく説明してもらい、また実物を観察して必要な情報を得る。

(ii) 「検索機械」について述べている文献をいくつか専門家から教えてもらい、それらの文献を自分で読むことにより必要な情報を得る。情報検索システムをこれに対応させて2つに大別することができる。

(i) 質問で要求している情報そのものを質問者に提供するシステム。

(ii) 質問で要求している情報を含んでいると思われる文献の名前とか所在を示す情報を質

問者に提供するシステム。

Y. Bar-Hillel[1] と W. S. Cooper[2] はそれぞれ自分の論文でこれらの2つのアプローチを次のように呼んでいる。

Y. Bar-Hillel	W. S. Cooper
(イ) Information Retrieval	Fact Retrieval
(ロ) Literature Search	Document Retrieval

このような区別の仕方は特に意味があるわけではない。特定の情報そのものを探し出すのか、文献を探し出すのかによつて。

- (i) 質問の出し方
- (ii) 見出しのつけ方
- (iii) 情報のファイルの構成方法
- (iv) 回答の出し方

などが異つてくるので、システムの説明上便宜的に行つた区別である。(簡単なファクト・リトリバーブルのためのシステムはドキュメント・リトリバーブルのシステムとあまり変わらないが、複雑なファクト・リトリバーブルのシステムは著しく相異なるものになるだろう。)

1.3 一次情報と二次情報

この情報の区別も前節(1.2)で述べた区別と同じように便宜的な分け方であり、特にドキュメント・リトリバーブル(文献検索)を説明する場合に便利であるので、情報管理便覧[17]で述べている定義をここに引用しておくことにする。

「一次情報とは厳密に言えば、少なくとも情報が伝達されるときに既存情報の中にそれと類似なものがないものを一次情報という」。すなわちオリジナリティのあるものをさしているので、単行本、論文、報告書、特許公報文書といったものに含まれている情報を一次情報と呼んでいる。

これに対して、「二次情報とは、一次情報の組織された集合であるといえる」。目録とか索引に含まれている、一次情報の「見出し」とか「所在を示すアドレス」といったものを二次情報と呼んでいる。

(原論文を一次情報、抄録などを二次情報、見出しを三次情報という人もいる)

1.4 検索作業の機械化と自動化

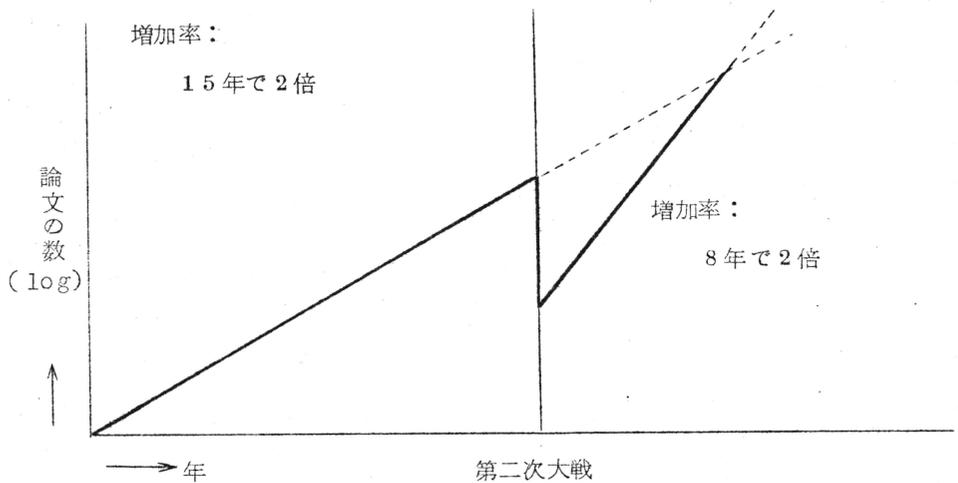
検索作業を手作業で行うものとして、ホール・ソート・カード、エッジ・ノッチド・カード、Peek-A-Boo(はいいいないばあ)カードといったカードを利用したシステムがある。これらのシステムでは原始情報(一次情報)を適当に変換してカードに刻み目をつけておき、このカードのファイルにピンを通したり、光をあてたりして目的のものを見つけようとするものである。これらは一般に次に述べるパンチドカード・システムよりは安価で操作も簡単であるが、当然、貯えられる情報の量にも一回の操作により探索できる量にも制限がある。

探索の作業をもつと能率よく行うために、パンチドカード・システムの機械を検索機械として利用したシステムがある。このシステムでは分類機と照合機が中心になり、また製表機を利用し、カードにパンチしてある二次情報、あるいは一次情報をプリントすることもできる。この種の機械化はすでに1950年代にPCGSの発達と共に文献の探索とか化学に関する情報の検索といった方面で行われていた。

一方では情報検索のために設計された検索専用機がいくつかあり、例えば、マイクロフィルムに情報を貯え、要求に応じてフィルム・デックをソートして必要な情報を要求者に提供するという“FILMOREX システム”や“WALNUT システム”などが開発されている。

情報検索そのものは現在の電子計算機と必然的な結びつきはもっていないと思われるが、次に述べるような理由から最近では電子計算機を用いた情報検索システムがいろいろと開発されている。

(1) 扱う情報の量が遼大になつてきたこと。はじめて計算機にこの情報検索の仕事を行わせたものは、技術情報の利用者と配布者たちであつたが、最近の科学技術の分野で発表される論文の数をグラフに示すと第1.4.1図のようになる。



第1.4.1図

概算であるが、戦前は15年で2倍の割合で増加していた論文の数が、戦後には8年で2倍の割合で増加し、例えば日本科学技術情報センターで昭和38年に扱った外国雑誌の論文数は約20万件(医学と農学を除く)であり、それが39年度には約24万件と推定されている。

(2) 著積してある情報を多目的に使用するようになり、多元的な検索が必要となつてきたこと。すなわち、貯えられているいくつかの情報を要求に応じて適当に組み合わせて提供してくれるようなシステムが必要になつてきたのである。

(3) 人手を省き、検索時間をできるだけ短縮し、作業の自動化が望まれるようになったこと。

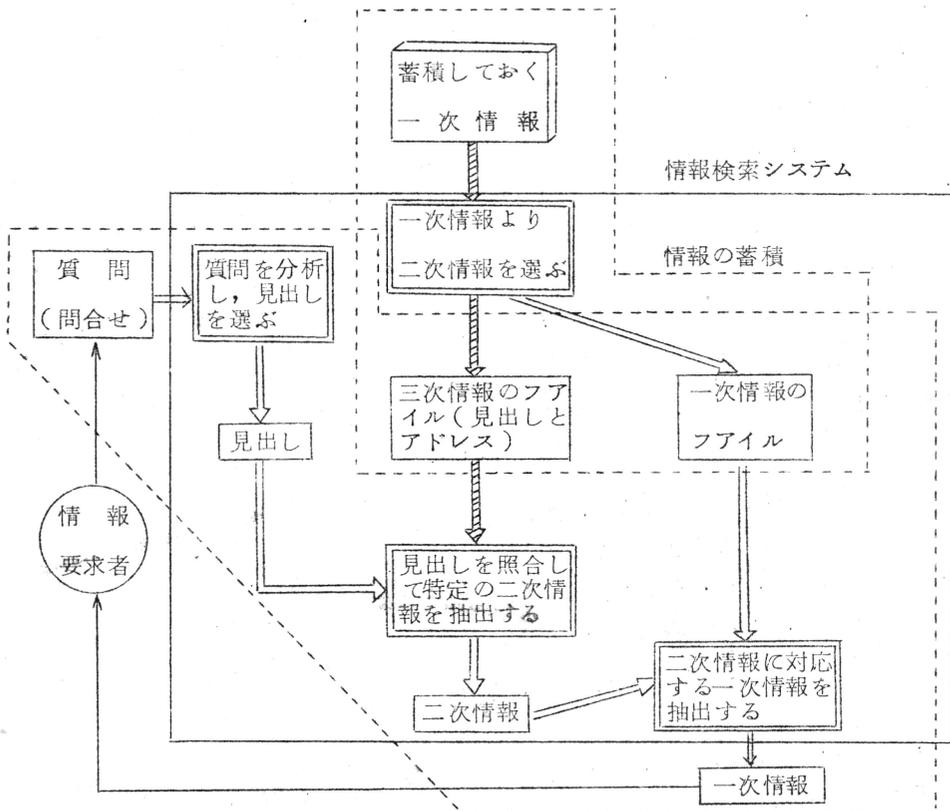
などの理由から、

- (i) 大容量の記憶装置（磁気コア、磁気テープ、磁気ドラム、磁気ディスクなど）をもつていて、
 - (ii) 検索作業の手順を記憶し、自動的に実行することができ、
 - (iii) さらに分類とか照合といった作業がPCS よりもはるかに高速度で実行できる。
- 電子計算機がそのまま検索機械として利用されるようになったのである。

1.5 システム設計上考慮すべき問題点

情報検索システムを一般的に図示したものが第1.5.1図である。

まず、蓄積しておく一次情報（原始情報）を適当な方法で分析（これを主題分析という）し、二次情報を作り出し、一次情報のファイルと二次情報のファイルを作る。この二次情報のファイルには一次情報の「見出し」および「アドレス」が含まれている。一方、質問者から出された質問を主題分析し、「見出し」を作る。この質問の「見出し」と二次情報のファイ



第 1.5.1 図

情報の検索

ルの中から質問で要求している情報を抽出し、それを見やすい形に変換して質問者に提供するものである。

理想的な情報検索システムというのは、情報を要求している人の立場とか、また質問の中に暗に含まれているものを判断して、その人にとって最適な情報を提供するシステムであるが、しかし、「いかにして迅速かつ正確に要求どおりの情報を検索するか」ということがシステムに対する当面の課題であると考えられる。

電子計算機を用いて作業の自動化を図っている情報検索システムを評価する場合に、

- (i) 使用する記憶装置の記憶容量
- (ii) 検索時間
- (iii) コスト

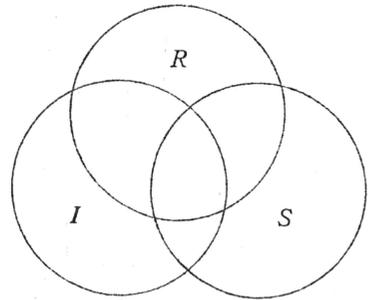
といったものは、実際にシステムを運転しなくても数量的に把握することができる。しかし、要求した情報がどの程度まで検索されているかということは何度か実際にシステムを運転してみないと数量的に把握することができない。文献検索（ドキュメント・リトリバル）システムのこの質的な部分を評価するのに、その分野の専門家が質問を分析して選び出した文献と、システムが検索した文献とを比較するという方法がよく用いられている。

ドキュメント・リトリバル・システムを念頭において、第 1.5.2 図に示すようなドキュメントの 3 つの集合を考えてみることにする。

(1) 蓄積してあるドキュメントの中である特定の質問に関連のある理想的なドキュメントの集合を I とする。

(2) 人間がその質問を分析して蓄積ファイルの中から選び出したドキュメントの集合を R とする。

(3) システム（検索プログラム）が質問を分析し、検索したドキュメントの集を S とする。



第 1.5.2 図

$\bar{I} \cdot R$ の部分は、理想的には不必要な部分ではあるが、質問をするのに用いている言語の不備および各個人の主観の相異により生じた部分であると考えられる。 $\bar{R} \cdot S$ の部分は、情報要求者から見れば不必要な部分であり、 $R \cdot \bar{S}$ の部分はシステムが検索しそこなつた部分である。情報システムを評価する場合、 I と S を比較することが当然のように思われるが、現在のところ、この I を数量的に把握するよい方法がないので、便宜上 R と S を比較している。

この $\bar{R} \cdot S + R \cdot \bar{S}$ の部分、特に $R \cdot \bar{S}$ の部分をいかに少なくすることが検索システムに対する一番重要な問題点であるが、

(i) 自然言語で表現されている一次情報からいかなる二次情報を作り出せばよいか？

(ii) 質問をどのような言語で表現すればよいか？

といつた問題とおきかえて考えるべき問題である。

システムを設計するために考慮すべき点をさらに述べてみると次のようになる。

(iii) 二次情報を複雑にすることにより、検索時間がどのように変わるか。

(iv) 検索時間を短縮するためにはどのように情報のファイルを構成すればよいか。

そこで以下の章でこれらの問題点に触れ、実用化しているいくつかのシステムについて検討してみることにする。

2. 情報の整理

2.1 情報の見出し

2.1.1 「見出し語」および「主題」の定義

蓄積してある情報の中から、必要に応じて特定の情報を探し出すために個々の情報につけるものを第1章では「見出し」と呼んでいた。またこれを人によつては「タグ(Tag. 名札)」とも呼んでおり、この「見出し」に自然言語の語句を用いるか、また特定の人工言語を用いるかによつていろいろな呼び方がある。そこでこれらを整理して次のように定義し、以後は「見出し語」、あるいは「主題」という言葉を用いることにする。

<見出し語> ::= <件名標目> | <ディスクリプタ> | <ユニターム> | <キー・ワード> | <索引語> | <コード> | <階層見出し>

<コネクタ> ::= <接続詞> | <前置詞> | <論理記号> | <role indicator>

<主題> ::= <見出し語> | <見出し語> <コネクタ> <主題>

これらの見出し語を次のように分けることもできる。

- (i) 標準語彙のないもの；キー・ワード
- (ii) 標準語彙のあるもの；件名標目、ディスクリプタ、ユニタームなど。

この区別はシソーラス(用語辞典)があるかないかで分けたものである。

また次のように分けることができる。

- (i) コネクタの使えるもの
- (ii) 階層関係のあるもの
- (iii) 単語を単に並列的に並べたもの

しかしこれらの区別の仕方は絶対的なものではなく、見出し語のそれぞれの特徴を説明するために便宜的に行なつた区別である。そこで次にそれぞれについて簡単に述べることにする。

(1) 件名標目 (Subject Heading)

これは情報内容をあくまでも直接的にそのものズバリに表現する簡単な言葉を用いてい

ることを基本方針にしている。基本的な標目の種類を述べると次のようになる。

- (i) 名詞（普通名詞，固有名詞）
ex. 人間，情報，東京
- (ii) 形容詞のついた標目
ex. 精密な機械
- (iii) 名詞を2つ以上つなぎ合せた標目
ex. 情報検索
- (iv) コネクタで結びつけた標目
ex. 計算の語差，電磁気，情報の分類および探索
- (v) 括弧によつて限定してある標目
これは多義語の場合に用いられる表現方法である。
ex. Plant(Botany)

(2) デイスクリプタ (Descriptor)

情報管理便覧(1092ページ)で「デイスクリプタとは，機械検索に用いられる系統化された用語 (scheduled term) である」と定義しているが，デイスクリプタの種類はほとんど件名標目の場合と同じで，ただ呼び方が違うと考えて差し支えない。

(3) 階層見出し *hierarchical classification*

この「階層見出し」という言葉は一般的に使われているわけではなく，「十進分類コード」とか「ファセット分類コード」のような概念の階層 (class) 関係を示しているコードをさしている。

例えば十進分類コードの一例を示すと

分類コード	主 題
6	応用科学
62	工業技術
621	機械的工業技術
621.8	機械要素，動力伝達
621.88	緊 定 具
621.882	ねじ，ナット，座金，ボルト

(4) ユニターム (uniterm)

キー・ワード (key word)

この2つは単語を原則としていることには変りないが，キー・ワードと呼ばれるものには，シソーラスで定めているような標準語彙がないのが通常である。

情報検索システムでは，新しく入ってきたドキュメントに対しては，まずその主題分析

をして、その中に含まれるいくつかの概念を抽出し、それを「見出し語」に置き換える。一方、質問も主題分析をして概念を抽出し、同じようにそれを「見出し語」に置き換え、両方の見出し語を照合して質問に適合する情報を探し出す。そこで抽出した概念をある約束に従って見出し語に置き換える場合に、当然、「シソーラス（用語辞典）」とか「分類コード表」が必要となってくる。

2.1.2 シソーラス（用語辞典） *Thesaurus*

代表的なシソーラスとしては次のようなものがある。

(1) ASTIA (Armed Service Technical Information Agency) で 1959 年に編集した “Thesaurus of ASTIA Descriptor” このシソーラスは

- { ディスクリプター一覧表 (Schedule of descriptors)
- { 逆引表 (Scope Note Index)

とからできている。

ディスクリプター一覧表には 19 のフィールドがあり、それぞれのフィールドをいくつかのグループに分けて、全体で 292 のグループがある。そして各グループの下にそれを構成している個々のディスクリプタを示しており、すべてアルファベット順にリストしてある。

逆引表には、各ディスクリプタをアルファベット順にリストし、それぞれについてその属するグループ名、“Include”されるもの、および“Also see”されるものも一緒にリストしてある。

(2) アメリカ化学技術者協会 (American Institute of Chemical Engineers) で 1959 年に開発した検索したシステムのシソーラス “Thesaurus covers chemical engineering terms”

これでは見出し語にする単語をアルファベット順にリストし、それぞれについて同義語、密接な関連をもつ語、上位概念および下位概念を示す語などを一緒にリストしてある。

このシステムでは主題分析して抽出した概念を表現するいくつかの単語をシソーラスにより選び、単語と単語の間の “AND” の関係を「リンク (link)」で示し、単語の役割を「role indicator」で示した見出し語を作るようになっている。

2.1.3 role indicator と semantic code

role indicator については前述したが、ここで Western Reserve 大学の role indicator を簡単に紹介する。

KAJ (出発物質)	KQJ (手 段)
KEJ (処理物質)	KOK (状 況)
KUJ (成 分)	KUV (主 題)
KOV (与えられた性質)	KOL (場 所)

KWV (有する性質)	KAL (一般属性)
KAN (処 理)	LOK (関連分野)

概念を表すいくつかの語の役割を role indicator で示し、それを語の前につけて概念をなるべく正確に表現した見出し語を作ろうとするものである。例えば「KAM(研摩), KQJ(研摩材)」は「研摩材による研摩」という概念を表わしている。

Semantic Code というのは Western Reserve 大学の検索システムで用いているコードで、言葉の意味を分析し、それぞれの意味に対してコードをつけ、それらを組合せてその言葉のコードとしたものである。例えば

主 題	コ ー ド				
温度計	装 置	測 定	熱による	温 度	種 類
	MACH	MUSR	RWHT	004	001
電圧計	装 置	測 定	電気による	電 圧	種 類
	MACH	MUSR	LWCT	006	002

2.1.4 分類コードと分類表

情報検索の分野では、この「分類する (Classify)」という言葉をほとんど「ものを特定の階層に分ける」という意味で使っている。

従来、図書の整理とか資料の整理に利用されてきた分類法はいろいろあるが、その代表的なものには

(i) 十進分類法……………国際十進分類表 (UDC)

日本十進分類表 (NDC)

(ii) フアセット分類法……………コロソ分類表

などがある。

このフアセット分類法の特徴は、他の分類法よりも新しい主題を収容しうる能力をもっていることであり、これはフアセットと呼ばれるいくつかのカテゴリ、例えば personality matter, energy, space, time といったものを考え、これらの一定の約束に従って順序づけてコード化してゆくのである。そこで十進分類法のように縦方向にだけ細分できるというのではなく、さらに横方向にも細分できるのである。

時計という主題を「時計」で検索する場合と、「精密機械」という質問でその一部として検索する場合の二通りが考えられる。前者の場合にはなんら概念の分類は必要ないが、後者のような包括的な検索をする場合には概念を分類しておく方が効果的な検索ができるように思われる。

今「精密機械」に「F23」、「時計」に「F231」というコードを割り当てたとする。もし「時計」という主題が階層の一番下位であるとすれば、むしろ「F23 時計」というコードの方が、ものを整理することが目的でなく、検索が目的の場合にはより効率がよくなるよう

に思われる。このように実用的な分類を相対分類と呼び、完全に階層に分けてしまう分類を絶対分類と呼んでいる。

2.1.5 各種見出し語による検索効率の比較

1957年より英国のAslib-Cranfield Research Projectで行なっていた4種の見出し語をそれぞれ用いた索引システムの比較研究の結果について文献[3]、[4]で詳細に述べているので、ここで少しふれておくことにする。

これによると、航空力学関係ののべ18,000件の英語の文献について、次のような4種の見出し語を用いて検索実験を行なっている。

- (i) 国際十進分類法による分類コード
- (ii) 件名標目
- (iii) 特殊なフアセット分類法による分類コード
- (iv) ユニターム

この実験の目的は、文献を主題分析して見出し語を選ぶ場合に起る諸問題の調査と、4種の見出し語による検索効率の比較であつた。そしてこの実験では、見出し語の選択および検索はすべて手作業で行つた。

まずそれぞれの見出し語についての分類表、およびシソーラスとそれらの逆引表を作成した。そして見出し語の選択について異つた経験をもつ3人が各文献について限られた時間内(2,4,8,1,2,1,6分の5種類)に見出し語の選択を行つた。

検索実験は3回行つた。第1回の実験では最初からこの作業に参画して見出し語の選択を行つていた3人のスタッフが400件の質問例について検索を行ない、第2回の実験では検索の経験はもっていないが、文献内容には熟知している5人のスタッフに、簡単に作業規則および記録のとり方の説明をただけで検索を行なさせた。第3回の実験では同じ5人で、検索プログラムを改良して行つた。その結果を第2.1.5表に示す。

	第1回	第2回	第3回
(i)	63.9%	74.2%	77.2%
(ii)	74.7%	80.3	82.5
(iii)	—	74.5	73.7
(iv)	69.9	76.6	86.4

第2.1.5表 成 効 率

この実験の結果から各種見出し語の特徴と問題点を述べてみることにする。

- (1) ユニタームを見出し語に用いた場合が一番成功率が高かつたことは、これが見出し語としてもつとも単純な形式で取扱うことができ、作業に必要とする時間も他に比べて極めて少なくすむからであると考えられる。
- (2) 一口にいつてフアセット分類コードを作成することは非常に手間のかかる仕事で、結

果からみても一番成功率が低い。

この2つの問題点は作業がすべて手作業であつたこと、および見出し語の選択に対し時間の制限があつたことが主な要因であると考えられ、電子計算機を利用したシステムでは必ずしも同じことが言えないと思われる。しかし次に述べる問題点は計算機を用いたシステムでも共通して言えるところであろう。

- (3) 分類表およびシソーラスの逆引表が非常に役に立つた。
- (4) 分類コードを用いた検索システムは包括的な検索に対して有利である。
- (5) 件名標目を見出し語に用いた場合、見出し語の結合に制限があるので、包括的な検索が必ずしも可能でない。
- (6) 件名標目のシソーラスの“see also”参照を用いなかつたために失敗したケースが非常に少なかつた。

ドキュメントを主題分析し、その中からいくつか概念を引き出し、それらを見出し語に置きかえる過程では、しばしば、ドキュメントに含まれる、ある情報内容を落してしまうことがある。またドキュメントの内容を詳しく見出し語で表現しようとする、見出し語が非常に複雑になり、この見出し語への変換という作業が非常に手間のかかる仕事になる。

そこでまず考えられることは、自然言語で表わしてあるドキュメント、あるいはその抄録に使つてある単語の一つ一つを見出し語とみなし、質問が出るたびに貯えてある全文を機械的に読み、質問で指定している語を含むドキュメントを検索するということである。

これについてD. R. Swanson[18]が行つた実験がある。実験結果は1960年10月に発表されているが、それによると過去10年間のPhysical Review誌から選んだ核物理学に関する100件の論文に対して、50個の質問を用意してこの実験を行つている。

- (i) まず数人の核物理学者が各論文を熟読し、各質問に対するその論文の適格度を10点満点で評定した。
- (ii) (i)のグループとは別に、物理学専攻者3人、プログラマ2人、数学専攻者1人、および物理学の知識のある司書1人の計7人のグループは、与えられた50個の質問に対して次の3通りの方法で1000回にわたる検索実験を行つた。
 - (イ) 件名標目を見出し語にする検索法。これでは核物理学に関する件名標目のシソーラスを作成した。
 - (ロ) 全文を磁気テープに記録し、質問でいくつかの単語を指定し、それらの単語を含むドキュメントを電子計算機を用いて検索する方法。
 - (ハ) シソーラスを用いて(ロ)と同じように検索する方法。この実験のために核物理学関件のシソーラスを編集した。

これらの三つの方法で行つた検索実験の結果を次に示すような S_i という値で比較してい

る。

$$S_i = \frac{\text{検索したドキュメントの適格度の和}}{i \text{ という質問に対する全ドキュメントの適格度の和}}$$

S_i を平均すると(イ)の方法では38%, (ロ)68%, (ハ)86%という結果が得られた。

この結果だけを見ると明らかに、人間が手作業によつてドキュメントの主題分析および見出し語の選択を行い、その見出し語をたよりに検索するよりも、貯えてあるドキュメントの全文を検索の対象とする方法がかなりよい結果を得ることになる。しかしここに残されたいくつかの問題点がある。

- (1) (イ)の方法で結果が思わしくなかつたのは、ソース不備、および主題分析を行つた分析者の能力の不足が原因であるとも考えられる。
- (2) 質問のたびに全文を読むことは時間の浪費である。
- (3) 検索効率をさらに上げるためには、ドキュメントを構成する単語のセマンティックスを考える必要がある。

2.2 二次情報作成の作業の自動化

2.2.1 自動抄録

第1章の序論でも述べたように、文献の数が膨大になつてきたこと、抄録を作るという作業はかなり熟練を要し、また個人差があるなどの理由から、電子計算機を用いて、ドキュメントの中で使用される語の頻度を基にして重要と思われる文章を抽出するという統計的な方法や、また構文分析を行なつて文章を抽出するといつたいろいろな方法が考えだされている。

(1) Luhn [5] の方法

Luhnの方法では次のような仮定が前提となつている。

- (i) 文献の主題に関係の深い言葉は、その文献に何回もくり返しあらわれる。
- (ii) そのような言葉を一文中に数多く含むような文章は、その文献の主題をあらわしている可能性が大きい。

この方法では、まず計算機に読み込んだ文献について前置詞、代名詞、冠詞などを取り除いて、各語の使用頻度を数える。頻度がある程度以上の語を重要語とみなし、これをもとに重要因子を計算し、重要因子の値がある程度以上の文章を原文と同じ順序でプリントして抄録とする。この方法ではセマンティックスを全く考慮していない。

(2) 相対頻度を用いる方法

主題に関連の深いことばの出現頻度は高いとしても、逆は成り立たず、出現頻度の高い言葉でも常用語なぞは文献の内容を特徴づけないことが多い。しかしある分野では常用語であつても、他の分野ではそうでない言葉もある。そこで文献中での使用率は高いが、その分野での使用率が低い語が重要語として選ばれることが望ましい。この仮定にもとづき Edmundson[6]らは重要語を表わす尺度として次のような4種を提案している。

$$S_1 = f - r, S_2 = f/r, S_3 = f/f+r, S_4 = \log \frac{f}{r}$$

ただし、 f は文献中での語の使用率、 r はその専門分野でのその語の使用率である。

(3) 構文分析を利用した方法

Climenson[7]らは文章を構文分析して、修飾語句、前置詞句、従属句、従属節といった従属的な要素を文章から消していき、残った部分を集めて抄録とする。

文献には同じ内容の書き換えとか省略がいたるところで見られる。これが原因となり抄録の文章にくり返しが多くなり、語の頻度数をもとにしての統計的な方法の精度が悪くなつたりすると考えられる。この問題を解決する一つの方法は、文の要素を文と文の間で比較し、相互に置き換えが可能な要素を見つけることである。また構造が複雑な長い文は、構文の簡単ないくつかの短い文におきかえて、文と文を比較することもできる。

2.2.2 ドキュメントの自動分類

ドキュメントの文章中に使用されている語の頻度をもとにして、ドキュメントをいくつかのカテゴリーに分類し、自動的にそのドキュメントの見出し語を見つけるという方法である。その1つに factor analysis の手法を利用したものがある [8][9]。

- (i) まず蓄積しておくドキュメントについて、前置詞、冠詞、接続詞といったものを除き、見出し語になりそうなキー・ワードを選び出し、それらの頻度を数える。
- (ii) 使用頻度の多い順にいくつかキー・ワードを選ぶ。(Borko[8],[9]の実験では90コの語を選んだ。)
- (iii) 選り出したキー・ワードの使用頻度について因子分析を行い、それらのキー・ワードをいくつかのカテゴリーに分ける。

$$(iv) P_{ij} = \sum_{k=1}^n L_{kj} \times T_{ki}$$

L_{kj} : キー・ワード K のカテゴリー

j 内での負荷の割合

$j=1, 2, \dots, X$: カテゴリー

$k=1, 2, \dots, n$: キー・ワード

$i=1, 2, \dots, N$: ドキュメント

T_{ki} : ドキュメント i 内のキー・ワード k の頻度

各ドキュメント i について P_{ij} を計算し、これを最大にする j をドキュメント i のクラスとする。

従つてカテゴリー J に分類されたドキュメントの「見出し語」として、そのカテゴリーに含まれるキー・ワードが用いられる。

3. ファイルの構成と探索

3.1 テーブル・ルックアットとテーブル・ルック・アップ

テーブル・ルック・アットの手法というのは、テーブルのある項目を引く場合、あらかじめその項目の位置を知り、他の項目を調べず直ちにその項目の内容を引き出すという方法である。電子計算機で記憶装置内のテーブルをこの手法を用いて引く場合には、

- (i) まずインデックス・レジスタに引数をおき、
- (ii) そのインデックス・レジスタによつて定まつた実効アドレスの内容にオペレーションをほどこす

という2つのオペレーションで実行することになる。

これに対してテーブル・ルック・アップという方法は、テーブルの一つ一つの項目を調べていく方法である。

情報検索では、情報の見出し語のファイルと質問の見出し語との照合という作業は普通このルック・アップ方式がとられており、この作業を計算機に行わせることが非常に時間の節約になるので、これだけのオペレーションを計算機に実行させるシステムもある。計算機によつてはテーブル・ルック・アップのオペレーションを1命令で実行できるものもあるか、いずれにしても時間のかかる作業であるので、部分的にでもルック・アットの方法を用いて時間を節約することが考えられた。しかし記憶装置としてアソシエイティブ・メモリ (associative memory) が使える場合には、もつと能率のよい検索ができると考えられるが、これは現在のところ一般的には実用化していない。

3.2 Tree の応用

ファイルあるいはテーブルの中のある項目を引き出すのに、全部の項目を調べず、その項目に関係のあるものを調べるだけで済ませるためには、それらの項目をグループ分けしてリストしておくことである。「Treeを応用する」ということはそれぞれの項目の間の階層関係を見つけだし、項目を分類することを意味している。

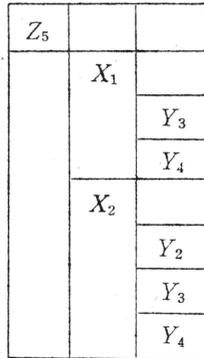
一般に人間がアルファベット順に語を配列した辞書を引いたり、大小の順に数値をリストしたテーブルを引く場合には、無意識のうちに語とか数値を構成している文字、数字の階層関係を見ながら目的のものを捜しているわけである。

電子計算機に記憶装置内の辞書とかテーブルを引かせる場合に次の二通りの方法が考えられる。

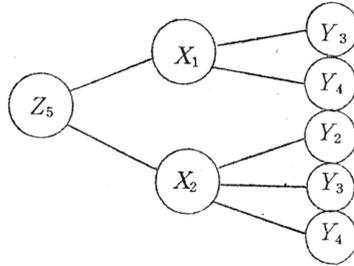
- (1) 語をアルファベット順に配列するとか、数値項目を小さい順に配列するとかして、テーブル・ルック・アップの手法、あるいはバイナリ・サーチの方法を用いて探索する。
- (2) 語を構成する文字を階層に分類し、またテーブルの各項目を階層に分類して、記憶装置内ではマトリックスの形 ("Chain list matrix") で辞書とかテーブルを記憶しておく。

ここで chain list matrix を簡単に例を示しながら説明しておく。

今、 $X_1, X_2, Y_2, Y_3, Y_4, Z_5$ という項目があるとする。これを階層に分類してリストしたのが第 3.2.1 図の(a)であり、その Tree (b) に、Chain list matrix を(c)に示してある。



(a)



(b)

第 3.2.1 図

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
項目	項目番号	データ	レベル	Degree	Source	兄弟関係のリンク	データのリンク
Z_5	1	5	1	2	—	1	—
X_1	2	1	2	2	1	5	—
Y_3	3	3	3	0	2	4	7
Y_4	4	4	3	0	2	3	8
X_2	5	2	2	3	1	2	6
Y_2	6	2	3	0	5	7	5
Y_3	7	3	3	0	5	8	3
Y_4	8	4	3	0	5	6	4

第 3.2.1 図 (c)

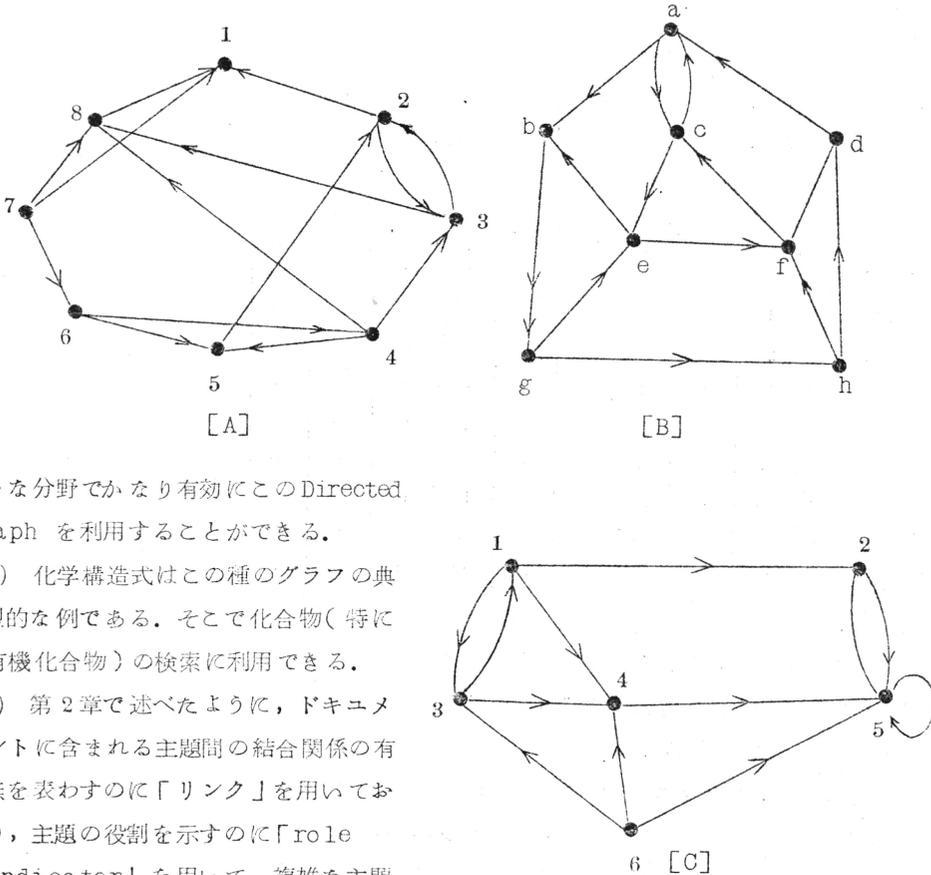
この例の場合、実際に記憶装置に貯えられるのは(2)から(8)までである。データというのはその項目の内容のことであり、この場合はサブスクリプトをデータとした。(4)から(7)までは Tree の階層構造を示したもので、(6)は項目の親子関係を、(7)は兄弟関係を項目番号で示している。(8)は Tree 構造とは無関係に項目の間の cross-reference を項目番号で示したものである。

このように chain list matrix を利用すると、探索時間が節約される反面、多くの場合には占有するメモリーが増加するのである。

3.3 Structure matching

項目と項目間の結合関係を二次元に図示したものを“Directed graph”とか“Association Map”と呼んでいる。グラフ上では、項目を表わすものを“node”と呼び結合の手を矢印をつけた線で表わして“branch”と呼んでいる。Tree というのは各 node に入る branch の数が1か0の Directed graph である。

第 3.3.1 図に示したようなグラフ A がグラフ B と同じであるか、またグラフ C がグラフ B の一部分であるかといったことを機械的に判定することができれば、情報検索の次に述べる



第 3.3.1 図

ような分野でかなり有効にこの Directed graph を利用することができる。

- (1) 化学構造式はこの種のグラフの典型的な例である。そこで化合物(特に有機化合物)の検索に利用できる。
- (2) 第 2 章で述べたように、ドキュメントに含まれる主題間の結合関係の有無を表わすのに「リンク」を用いており、主題の役割を示すのに「role indicator」を用いて、複雑な主題間の関係を極力単純化して一次元のコードで表していた。これに二次元のグラフを利用すれば複雑な関係をかなりよく表現できる。

このグラフの照合の方法はいろいろと考えられるであろうが、ここではその一つの方法として E. H. Sussenguth, Jr[13] の提案した、電子計算機にグラフの照合を行わせる方法を簡単に紹介することにする。

- (i) 蓄積しておくグラフについて、結合関係を表わした「結合マトリックス」と、項目の内容をリストしたテーブルを作り、これらを記憶装置に貯えておく。
- (ii) 質問のグラフについても同じように「結合マトリックス」とテーブルを作る。
- (iii) 両方のグラフについて、それぞれ次のような node の集合を作る。
- (a) 入ってくる branch の数が同じもの
 - (b) 出ていく branch の数が同じもの
 - (c) node の内容(値)が同じもの
 - (d) branch の値が同じもの
 - (e) loop になっている node
- (iv) それぞれの集合について1ずつ照合し、対応するものを見つける。
- (v) 1つの集合に2つ以上の node が残り、対応が見つからなかつたら、それらの node と結合している node の集合を作つて比較していく。

第 3.3.1 図のグラフ A から作つた「結合マトリックス」を第 3.3.2 図に示す。

3.4 分割表を利用したドキュメント

・リストリーバル

ユニターム・コーディネイト・システムと呼ばれているシステムでは、蓄積しておくドキュメントについてそれぞれ主題分析をして単語による見出し語を複数個つける。例えばあるドキュメントに対して A, B, C という単語による3つの見出し語をつけたとする。

	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	1	0
2	1	0	1	0	0	0	0	0
3	0	1	0	0	0	0	0	1
4	0	0	1	0	1	0	0	0
5	0	1	0	0	0	0	0	0
6	0	0	0	1	1	0	0	0
7	0	0	0	0	0	1	0	1
8	1	0	0	1	0	0	0	0

第 3.3.2 図

このドキュメントは質問で“ A ”と指定しても、“ B ”あるいは“ C ”と指定しても検索することができる。また質問で“ AB ”と指定したときは“ A ”および“ B ”の2つの単語を見出し語にもつドキュメントが検索されることになる。

主題分析して見出し語をつけるときに、シソーラスで同義語とか cross-reference の関係にあるような語をある特定の語に統一することもできるが、シソーラスが完備していない場合には質問で指定した語の同義語や cross-reference の関係にある語を見出し語にもつドキュメントを検索しそこなうおそれがある。

そこで H. E. Stiles [14] は、ある語で索引できるドキュメントの数と、もう1つの語で索引できるドキュメントの数が統計学的に関連があるかどうかを 2×2 分割表で判定するときの X^2 の値の対数をとつたものを“ Association Factor ”と定義し、これを用いて質問で指定した語の同義語とか cross-reference の関係にある語を見出し語にもつドキュメントも検索しようという方法を提案している。

$$\text{Association Factor} = \log_{10} \frac{(|fN - AB| - \frac{N}{2})^2 N}{AB(N-A)(N-B)}$$

A : ある語で索引されるドキュメントの数

B : もう1つの語で索引される //

f : この2つの語で索引される //

N : 蓄積してあるドキュメントの総数

検索手順を簡単に説明すると次のようになる。

- (1) 質問で指定した語一つ一つについてそれぞれプロフィール(profile)を作る。このプロフィールには $A, F \geq 1$ となるような見出し語がすべて A, F と共にリストしてある。
- (2) すべてのプロフィールに、あるいは指定した数のプロフィールに現れる見出し語を選びだす。
- (3) 選びだした語を今度は質問の語とみなして(1)と(2)を繰り返す、「質問で指定した語」と「第2ステップで選びだした語」に、このステップで初めて選出された語を加えて拡大した質問の語のリストを作る。質問で指定した語の「同義語」とか「意味上関連の深い語」は第2ステップではまだみつけだされてはいないが、第3ステップで得られるリストの中には含まれているのではないかと styles は考えている。
- (4) それぞれの見出し語の A, F の和をその語の「重み」とする。
- (5) それぞれのドキュメントについて、その見出し語とリストを比較し一致した語の重みを加えてゆき、その和をそのドキュメントの質問に対する関連の深さと見なす。

3.5 綴りの誤りの修正

電子計算機を用いた情報検索システムで、記憶装置内のソースラスとかコード表を引く場合に、インプットした語に綴りの誤りがあると

- (i) 別な語が引き出される場合
- (ii) 見つからない場合

とがある。(i)の場合は語の意味分析を行わないと誤りを防ぐことはできないが、(ii)の場合にはその語の綴りが誤っているものと仮定して、その語に手を加えてもう一度引きなおしてみるという方法がいくつか発表されている。そこで次の2つの方法について簡単に述べてみることにする

(1) C. R. Blair [15]の方法

これは辞書の語も引く方の語も同じ約束に従って略語を作つて辞書を引く方法である。略語は4文字(語の最初と最後の文字、および誤る確率の少ない、すなわち使用される頻度の少ない2文字)で作る。この方法は人間のミスパンチに対しては有効な方法であると思われるが、ランダムに起る入力装置の読み違いに対してはあまり有効な方法とは言えない。

(2) F. J. Damerau [16] の方法

これは辞書を引いて見つからなかつた場合、その語に対して

- (i) 一文字が誤っている。
- (ii) 一文字が足りない。
- (iii) 余計な文字が一文字入っている。
- (iv) 連続する二文字が入れ代っている。

という4つの場合を仮定し、引く方の語を一文字変えたり、辞書の語を一文字変えたりしながら辞書を引きなおすという方法である。ランダムに作った綴りの誤りに対してこの方法を用いて行つた実験では、95%以上の誤りを修正している。

4. 言語の問題

これまでにも述べてきたような、

- (i) 自然言語で表わされたドキュメントおよび質問の主題分析と、見出し語の選択を電子計算機を用いて自動的に行う場合、
- (ii) 自動抄録を行う場合、

などに起つてくる自然言語にまつわる諸問題の一つの解決策として、言語の機械翻訳の技術を情報検索に利用することができる。

- (1) 単語の情報のみでなく、構文の情報も利用する、
- (2) 語尾変化や複合語を処理する、
- (3) 日本語で質問して英語の資料を探する、
- (4) 出力は原文と訳文の双方を印刷する、
- (5) 資料のふるいわけにもちいる、
- (6) 抄録の作成に構文分析を利用する。

といつたときに機械翻訳の技術が利用でき、現在、簡約化した機械翻訳のテクニックを使ってドキュメントのタイトルを翻訳することを試みているグループがある。

一方、簡単なファクト・リトリバブル・システムに対して、質問および検索手順を人工言語を用いて表わし、インプットすることが考えられている。その一例としてRECOL (Retrieval Command Language) という言語がある。これは人事ファイルの検索といつたものに適当な言語であると思われる。

RECOLは次のように5つのセクションにわかれている。

- (1) SELECT …… “AND”, “OR”, “NOT” を用いて、取り出したいレコードを指定する。
- (2) NAME ……パラメータに新しく名前をつけ、レコードを再分類することを指定す

る。

- (3) ASSOC …… 関連のあるレコードの選定を指定する。
- (4) EDIT …… 選びだされたレコードの必要なパラメータをラインプリンタにプリントする形式を指定する。
- (5) SUM …… レコードに対する統計的な処理を指定する。

5. ファクト・リトリバル・システムの一例

ファクト・リトリバル・システムというのは前にも述べたように、質問で要求している情報そのものを貯えてある情報の中から探しだして、要求者に提供するシステムである。複雑なファクト・リトリバルでは質問に対して、貯えてある情報からロジカルな推論を行って回答をださなければならない。自然言語には方言があり、あいまいさがあり、自然言語で表わされた情報からロジカルな推論を行うことが困難であるところから、W. S. Cooper [2] は一つの試みとして、英語の文法にかなり制限を加えたサブ言語 (Sublanguage) を定義し、貯えておく情報を表わす文章も、質問の文章もこのサブ言語で表わし (どちらも英語の平叙文の形式である)、一方システムはこのサブ言語で表わした文章をロジカル言語 (Logical Language) に変換し、質問に対してロジカルな推論を行って、“yes” あるいは “no” というタイプの回答をだすというシステムを提案している。

このシステムのアルゴリズムを簡単に述べると次のようになる。

- (1) 貯えてある文章群の中で問合せの文章を論理的に推論できるような文章が見つかったとき、“TRUE” とプリントして止る。見つからなかつたときはステップ(2)へ行く。
- (2) 貯えてある文章群の中で、その問合せの文章と論理的に矛盾するような文章が見つかったとき、“FALSE” とプリントして止る。見つからなかつたときにはステップ(3)へ行く。
- (3) “UNABLE TO ANSWER” とプリントして止る。

例えば、「酸化マグネシウムは白い、金属の酸化物である」という文章が貯えられているとする。そこで「すべての酸化物は白くない」という問合せに対しては、システムは “FALSE” という答をだし、「酸化マグネシウムは酸化物である」という問合せに対しては “TRUE” という答をだすのである。

6. 電子計算機を用いた索引システムと情報検索システムの実例

6.1 KWIC 索引システム

KWIC (Key word in Context) 索引システムは自動的に索引を作るシステムであり、検索は行わない。

KWICシステムでは次のような手順で各種索引を作成する。

- (1) 一つの文献について、著者名、タイトル、刊行者、文献の出典名、発行年度といった

ALL - COB

KEYWORD INDEX

	ALLOCATION OF STORAGE FOR ARRAYS IN ALGOL 60	CACM611 60
	A STORAGE ALLOCATION SCHEME FOR ALGOL 60	CACM610 441
	A SEMI-AUTOMATIC STORAGE ALLOCATION SYSTEM AT LOADING TIME	CACM610 448
CONSTRAINTS	THE ALPHA VECTOR TRANSFORMATION OF A SYSTEM OF LINEAR	CACM599 33
	AN ALTERNATE FORM OF THE 'UNCOL' DIAGRAM	CACM613 162
	LIBRARY LOADING WITH ALTERNATE ROUTINE SELECTION	CACM616 496
	A METHOD FOR ELIMINATING AMBIGUITY DUE TO SIGNAL COINCIDENCE IN DIGITAL DESIGN	CACM624 211
	ON AMBIGUITY IN PHRASE STRUCTURE LANGUAGES	CACM620 526
	INDEXING OF THE WORD 'AN' HAS BEEN PREVENTED AUTOMATICALLY	
	CODING OF MEDICAL CASE HISTORY DATA FOR COMPUTER ANALYSIS	CACM620 532
	PROGRAMMING-LANGUAGE TRANSLATION THROUGH SYNTACTICAL ANALYSIS	CACM623 145
	SIMULATION AND ANALYSIS OF BIOCHEMICAL SYSTEMS, III, ANALYSIS AND PATTERN RECOGNITION	CACM622 115
	SYNTACTICAL ANALYSIS BY DIGITAL COMPUTER	CACM620 515
	ERRGR ANALYSIS IN FLOATING POINT ARITHMETIC	CACM595 10
	ANALYSIS OF A FILE ADDRESSING METHOD	CACM628 659
CHEMICAL KINETICS	SIMULATION AND ANALYSIS OF BIOCHEMICAL SYSTEMS, I, REPRESENTATION OF	CACM610 509
DIFFERENTIAL EQUATIONS	SIMULATION AND ANALYSIS OF BIOCHEMICAL SYSTEMS, II, SOLUTION OF	CACM621 63
PATTERN RECOGNITION	SIMULATION AND ANALYSIS OF BIOCHEMICAL SYSTEMS, III, ANALYSIS AND	CACM622 115
	A COMPUTER TECHNIQUE FOR HANDLING ANALYSIS OF VARIANCE	CACM628 433
	COMBINING ALGOL STATEMENT ANALYSIS WITH VALIDITY CHECKING	CACM607 418
	ANALYTIC DIFFERENTIATION BY COMPUTER	CACM626 349
	FUNCTION TABLE ALGORITHM FOR ANALYZING LOGICAL STATEMENTS TO PRODUCE A TRUTH	CACM583 4
	INDEXING OF THE WORD 'AND' HAS BEEN PREVENTED AUTOMATICALLY	
	DCIAL DIAGRAMS OF BINARY CONCEPTION AND THEIR APPLICATION TO COMPUTER DESIGN LOGIC	CACM599 28
	AN ENGINEERING APPLICATION OF LOGIC-STRUCTURE TABLES	CACM616 516
	SHIFT-REGISTER CODE FOR INDEXING APPLICATIONS	CACM590 40
MODEL BASIN	THE APPLIED MATHEMATICS LABORATORY OF THE DAVID W. TAYLOR	CACM619 372
	A PREPLANNED APPROACH TO A STORAGE ALLOCATION COMPILER	CACM610 417
	SOLUTION OF EIGENVALUE PROBLEMS WITH APPROXIMATELY KNOWN EIGENVECTORS	CACM627 381
	A FURTHER NOTE ON APPROXIMATING ϵ TO THE x	CACM617 318
	A NOTE ON APPROXIMATING ϵ TO THE x	CACM600 649
FRACTIONS	ON APPROXIMATING TRANSCENDENTAL NUMBERS BY CONTINUED	CACM614 171
DYNAMIC PROGRAMMING	ON THE APPROXIMATION OF CURVES BY LINE SEGMENTS USING	CACM616 284
	TWO SQUARE-ROOT APPROXIMATION TRUNCATION	CACM589 3
	SUCCESSIVE APPROXIMATIONS	CACM588 13
ARY DIFFERENTIAL EQUATIONS	APPROXIMATIONS AND COMPUTER STORAGE PROBLEMS IN ORDIN	CACM615 222
FUNCTIONS	NOTE ON THE CONSTRUCTION OF RATIONAL APPROXIMATIONS FOR THE ERROR FUNCTION AND FOR SIMILAR	CACM618 354
	ON THE COMPUTATION OF RATIONAL APPROXIMATIONS TO CONTINUOUS FUNCTIONS	CACM627 403
LY COMPACT PROCESS FOR THE ADJOINTS OF MATRICES OVER	ARBITRARY INTEGRAL DOMAINS	CACM628 447
	A METHOD FOR EVALUATING THE AREA OF THE NORMAL FUNCTION	CACM615 224
	BESSEL FUNCTIONS OF INTEGRAL ORDER AND COMPLEX ARGUMENT	CACM614 169
	ERROR ANALYSIS IN FLOATING POINT ARITHMETIC	CACM595 10
	A NOTE ON MULTIPLE PRECISION ARITHMETIC	CACM618 353
	MULTIPLE PRECISION ARITHMETIC	CACM600 652
	A BASIC COMPILER FOR ARITHMETIC EXPRESSIONS	CACM611 3
SERIAL COMPILER	BINARY ARITHMETIC FOR DISCRETELY VARIABLE WORD LENGTH IN A	CACM594 13
	FLOATING-POINT ARITHMETIC IN COBOL	CACM625 269
	ON PROGRAMMING SE ARITHMETIC OPERATIONS	CACM588 3
	AN ALGORITHM FOR CODING EFFICIENT ARITHMETIC OPERATIONS	CACM611 42
AUTOMATIC CODING SYSTEM	THE ARITHMETIC TRANSLATOR-COMPILER OF THE IBM FORTRAN	CACM572 9
	ADDRESSING MULTIDIMENSIONAL ARRAYS	CACM624 205
	ALLOCATION OF STORAGE FOR ARRAYS IN ALGOL 60	CACM611 60
	FIXED-WORD-LENGTH ARRAYS IN VARIABLE-WORD-LENGTH COMPUTERS	CACM620 602
	INDEXING OF THE WORD 'AS' HAS BEEN PREVENTED AUTOMATICALLY	
	IN CONSTRUCTING A COMBINED ALGOL AND MACHINE-LIKE ASSEMBLY PROCESSOR	CACM611 36
	A SAP-LIKE ASSEMBLY PROGRAM FOR THE IBM 650	CACM601 7
	A START AT AUTOMATIC STORAGE ASSIGNMENT	CACM605 321
	AN ALGORITHM FOR THE ASSIGNMENT PROBLEM	CACM608 605
	AN ALGORITHM DEFINING ALGOL ASSIGNMENT STATEMENTS	CACM603 170
	A REAL TIME DATA ASSIMILATOR	CACM597 33
	INDEXING OF THE WORD 'AT' HAS BEEN PREVENTED AUTOMATICALLY	
NG STORE	DYNAMIC STORAGE ALLOCATION IN THE ATLAS COMPUTER, INCLUDING AN AUTOMATIC USE OF A BACKI	CACM610 435
	ATLAS, A NEW CONCEPT IN LARGE COMPUTER DESIGN	CACM606 367
	SYNTACTIC AND SEMANTIC AUGMENTS TO ALGOL	CACM604 211
	AUTHOR INDEX, 1958-1961	CACM610 568
	AN EVALUATION OF AUTOCODE READABILITY	CACM623 156
	AUTOMATED WEATHER PREDICTION	CACM613 166
	'AUTOMATIC' HAS BEEN PREVENTED AUTOMATICALLY	
	THE ATLAS COMPUTER, INCLUDING AN AUTOMATIC USE OF A BACKI	CACM610 435
	FOR MECHANICAL LANGUAGES AND THEIR PROCESSORS, A DYNAMIC STORAGE ALLOCATION IN	CACM610 532
	A BAKER'S DOZEN SPECIFICATION LANGUAGES	CACM616 513
	HALLISTIC CAM DESIGN	CACM621 54
	A STRING LANGUAGE FOR SYMBOL MANIPULATION BASED ON ALGOL 60	CACM610 494
	CORE ALLOCATION BASED ON PROBABILITY	CACM611 3
	A BASIC COMPILER FOR ARITHMETIC EXPRESSIONS	CACM625 237
	BASIC ELEMENTS OF COBOL 61	CACM602 85
	THE BASIC SIDE OF TAPE LABELLING	CACM618 336
AND THEIR PROCESSORS	SOME BASIC TERMINOLOGY CONNECTED WITH MECHANICAL LANGUAGES	CACM619 372
MATHEMATICS LABORATORY OF THE DAVID W. TAYLOR MODEL	BASIN THE APPLIED	CACM620 599
	A DECISION MATRIX AS THE BASIS FOR A SIMPLE DATA INPUT ROUTINE	CACM625 278
	COBOL BATCHING PROBLEMS	CACM605 325
	BENDIX G-20 SYSTEM	CACM585 3
	NOTE ON EMPIRICAL BOUNDS FOR GENERATING BESSEL FUNCTIONS	CACM614 169
ARGUMENT	CONVERSION BESSEL FUNCTIONS OF INTEGRAL ORDER AND COMPLEX	CACM606 352
	COMMUNICATION BETWEEN FLOATING-POINT REPRESENTATIONS	CACM627 376
	BETWEEN INDEPENDENTLY TRANSLATED BLOCKS	CACM585 12
COMPUTER WITH AN EXTRACT COMMAND	BINARY AND TRUTH-FUNCTIONAL OPERATIONS ON A DECIMAL	CACM588 6
COMPUTER WITH AN EXTRACT COMMAND	CORRECTION TO BINARY AND TRUTH-FUNCTIONAL OPERATIONS ON A DECIMAL C	CACM594 13
IN A SERIAL COMPUTER	BINARY ARITHMETIC FOR DISCRETELY VARIABLE WORD LENGTH	CACM605 322
	A TECHNIQUE FOR COUNTING ONES IN A BINARY COMPUTER	CACM599 28
DESIGN LOGIC	DCIAL DIAGRAMS OF BINARY CONCEPTION AND THEIR APPLICATION TO COMPUTER	CACM597 27
DECIMAL FRACTION	BINARY CONVERSION, WITH FIXED DECIMAL PRECISION, OF A	CACM581 11
	A PROGRAMMED BINARY CONVERTER FOR THE IBM TYPE 650 CALCULATOR	CACM582 1
	VARIABLE-WIDTH TABLES WITH BINARY-SEARCH FACILITY	

COL - LIP

AUTHOR INDEX

COLLINS JR, GEORGE G. EXPERIENCE IN AUTOMATIC STORAGE ALLOCATION	CALM610 436
COLLINS, GEORGE E. A METHOD FOR OVERLAPPING AND ERASURE OF LISTS	CALM600 696
COLLON JR, PERCY W. SIGNAL CORP. RESEARCH AND DEVELOPMENT ON AUTOMATIC PROGRAMMING OF DIGITAL COMPUTERS	CALM592 22
CONWAY, MELVIN E. PROPOSAL FOR AN UNCOL	CALM580 5
CONWAY, R. W. A QUEUE NETWORK SIMULATOR FOR THE IBM 680 AND BURROUGHS 220	CALM540 22
COOK, J. M. REMARKS ON AN EFFICIENT METHOD FOR GENERATING UNIFORMLY DISTRIBUTED POINTS ON THE SURFACE OF	CALM590 26
CORLEY, HENRY P. T. REQUEST FOR METHODS OR PROGRAMS	CALM584 9
COX, ALBERT G. A NOTE ON MULTIPLE PRECISION ARITHMETIC	CALM618 35
CRESS, H. A. A TECHNIQUE FOR COMPUTING CRITICAL ROTATIONAL SPEEDS OF FLEXIBLE SHAFTS ON AN AUTOMATIC COMP	CALM576 27
CROZIER, J. B. ON THE CONSTRUCTION OF MICROFLOWCHARTS	CALM570 27
CUNNINGHAM, JOSEPH F. WHY COBOL	CALM630 236
CURTZ, T. B. A COMPARISON OF 650 PROGRAMMING METHODS	CALM630 663
DAVIDSON, LEON RETRIEVAL OF MISPELLED NAMES IN AN AIRLINES PASSENGER RECORD SYSTEM	CALM623 163
DAVIS, MARTIN A MACHINE PROGRAM FOR THEOREM-PROVING	CALM627 394
DE WITTE, LEENDERT LEAST SQUARES FITTING OF A GREAT CIRCLE THROUGH POINTS ON A SPHERE	CALM608 611
DES JARDINS, PAUL TWO METHODS FOR WORD INVERSION ON THE IBM 709	CALM600 678
DICKSON, J. C. A DECISION RULE FOR IMPROVED EFFICIENCY IN SOLVING LINEAR PROGRAMMING PROBLEMS WITH THE SI	CALM607 509
DIMSDALE, B. PROGRAMMED ERROR CORRECTION IN PROJECT MERCURY	CALM600 649
DOMALY, W. L. A REPORT WRITER FOR COBOL	CALM625 261
DOM, JAMES PROGRAMMING A-DUPLEX COMPUTER SYSTEM	CALM618 507
EDMUNDSON, H. P. AUTOMATIC ABSTRACTING AND INDEXING, SURVEY AND RECOMMENDATIONS	CALM619 276
ELLENREAGER, KENNETH W. ON PROGRAMMING THE NUMERICAL SOLUTION OF POLYNOMIAL EQUATIONS	CALM600 664
ELLIS, P. V. AN EVALUATION OF AUTOCODE READABILITY	CALM623 126
EMERY, J. C. MODULAR DATA PROCESSING SYSTEMS WRITTEN IN COBOL	CALM625 263
ENGLUND, DONALD THE CLIP TRANSLATOR	CALM611 19
ENCOLI, PAOLO BINARY ARITHMETIC FOR DISCRETELY VARIABLE WORD LENGTH IN A SERIAL COMPUTER	CALM574 13
ERSHOV, A. P. ON PROGRAMMING OF ARITHMETIC OPERATIONS	CALM588 3
EVANS JR, A. THE USE OF THREADED LISTS IN CONSTRUCTING A COMBINED ALGOL AND MACHINE-LIKE ASSEMBLY PROCESS	CALM611 36
FARBMAN, DAVID COPE (CONSOLE OPERATOR PROFICIENCY EXAMINATION)	CALM630 566
FEIGENBAUM, EDWARD SOVIET CYBERNETICS AND COMPUTER SCIENCES, 1960	CALM577 7
FEIN, LOUIS THE ROLE OF THE UNIVERSITY IN COMPUTERS, DATA PROCESSING, AND RELATED FIELDS	CALM629 484
FELDMAN, JULIAN TALL, A LIST PROCESSOR FOR THE PHILCO 2000 COMPUTER	CALM600 648
FERGUSON, DAVID F. FIBONACCIAN SEARCHING	CALM611 65
FEURZEIG, W. COMMENTS ON THE IMPLEMENTATION OF RECURSIVE PROCEDURES AND BLOCKS IN ALGOL 60	CALM619 462
FEURZEIG, WALLACE PUTTING A HEX ON E TO THE X	CALM570 38
FIKE, C. T. ORACLE CURVE PLOTTER	CALM616 279
FISCHER, C. M. COMBAT VEHICLE FIRING STABILITY (ACTIVE SUSPENSION)	CALM603 170
FLOYD, ROBERT W. AN ALGORITHM DEFINING ALGOL ASSIGNMENT STATEMENTS	CALM611 42
FLOYD, ROBERT W. AN ALGORITHM FOR CODING EFFICIENT ARITHMETIC OPERATIONS	CALM620 526
FLOYD, ROBERT W. ON NONEXISTENCE OF A PHRASE STRUCTURE GRAMMAR FOR ALGOL 60	CALM576 21
FLOYD, ROBERT W. REMARKS ON 'ELIMINATION OF SPECIAL FUNCTIONS FROM DIFFERENTIAL EQUATIONS'	CALM617 504
FOATA, DOMINIQUE C. ON A PROGRAM FOR RAY-CHAUDHURI'S ALGORITHM FOR A MINIMUM COVER OF AN ABSTRACT COMPLEX	CALM623 110
FORSYTHE, G. E. VECTORCARDIOGRAPHIC DIAGNOSIS WITH THE AID OF ALGOL	CALM623 165
FOSTER, MALCOLM B. A METHOD OF REPRESENTATION, STORAGE AND RETRIEVAL OF 13 RANDOM CODES IN A 4-DIGIT NUMB	CALM630 435
FOTHERINGHAM, JOHN DYNAMIC STORAGE ALLOCATION IN THE ATLAS COMPUTER, INCLUDING AN AUTOMATIC USE OF A BALK	CALM631 206
FRANK, R. M. A HIGH-SPEED SORTING PROCEDURE	CALM627 401
FRASER, W. ON THE COMPUTATION OF RATIONAL APPROXIMATIONS TO CONTINUOUS FUNCTIONS	CALM630 539
FREDERICK, F. P. A DECISION RULE FOR IMPROVED EFFICIENCY IN SOLVING LINEAR PROGRAMMING PROBLEMS WITH THE	CALM610 430
FREDKIN, EDWARD TRIE MEMORY	CALM620 607
FREED, ROY N. LEGAL IMPLICATIONS OF COMPUTER USE	CALM623 515
FUTRELLE, R. P. SYNTACTIC ANALYSIS BY DIGITAL COMPUTER	CALM611 28
GALLER, B. A. THE INTERNAL ORGANIZATION OF THE MAD TRANSLATOR	CALM617 310
GALLER, BERNARD A. AN ALGORITHM FOR EQUIVALENCE DECLARATIONS	CALM620 570
GALLER, BERNARD A. COMPILING MATRIX OPERATIONS	CALM578 17
GANT, W. T. FLOW OUTLINING, A SUBSTITUTE FOR FLOW CHARTING	CALM627 477
GARFINKEL, DAVID PROGRAMMED METHODS FOR PRINTER GRAPHICAL OUTPUT	CALM627 115
GARFINKEL, DAVID SIMULATION AND ANALYSIS OF BIOCHEMICAL SYSTEMS, I, REPRESENTATION OF CHEMICAL KINETICS	CALM584 10
GARFINKEL, DAVID SIMULATION AND ANALYSIS OF BIOCHEMICAL SYSTEMS, III, ANALYSIS AND PATTERN RECOGNITION	CALM574 20
GIBLIN, JOHN RECURSIVE CURVE FITTING TECHNIQUE	CALM576 22
GILMAN, R. E. A MATHEMATICAL PROCEDURE FOR MACHINE DIVISION	CALM620 441
GLEISSNER, GENE H. NORC HIGH-SPEED PRINTER	CALM610 557
GLUSS, BRIAN FURTHER REMARKS ON LINE SEGMENT CURVE-FITTING USING DYNAMIC PROGRAMMING	CALM603 131
GOFFMAN, W. INEFFICIENCY OF THE USE OF BOOLEAN FUNCTIONS FOR INFORMATION RETRIEVAL SYSTEMS	CALM593 27
GOODE, H. H. SOVIET COMPUTER TECHNOLOGY, 1959	CALM620 576
GORN, S. ON THE CONSTRUCTION OF MICROFLOWCHARTS	CALM619 336
GORN, SAUL MECHANICAL PHRASELOGS, A TIME-MOTION STUDY OF A MINIATURE MECHANICAL LINGUISTIC SYSTEM	CALM610 552
GORN, SAUL SOME BASIC TERMINOLOGY CONNECTED WITH MECHANICAL LANGUAGES AND THEIR PROCESSORS	CALM610 555
GORN, SAUL SPECIFICATION LANGUAGES FOR MECHANICAL LANGUAGES AND THEIR PROCESSORS, A BAKER'S DOZEN	CALM597 24
GRAHAM, J. W. PROCESSING MAGNETIC TAPE FILES WITH VARIABLE BLOCKS	CALM611 28
GRAHAM, R. ON GAT AND THE CONSTRUCTION OF TRANSLATORS	CALM617 310
GRAHAM, R. M. THE INTERNAL ORGANIZATION OF THE MAD TRANSLATOR	CALM623 160
GRAHAM, ROBERT M. AN ALGORITHM FOR EQUIVALENCE DECLARATIONS	CALM627 334
GRAU, A. A. ON A FLOATING-POINT NUMBER REPRESENTATION FOR USE WITH ALGORITHMIC LANGUAGES	CALM611 10
GRAU, A. A. ON TRANSLATION OF BOOLEAN EXPRESSIONS	CALM619 163
GRAU, A. A. RECURSIVE PROCESSES AND ALGOL TRANSLATION	CALM592 6
GRAY, MARION C. BESSEL FUNCTIONS OF INTEGRAL ORDER AND COMPLEX ARGUMENT	CALM577 25
GREEN, J. REPORT ON THE ALGORITHMIC LANGUAGE ALGOL 60	CALM624 213
GREEN, JULIEN POSSIBLE MODIFICATIONS TO THE INTERNATIONAL ALGEBRAIC LANGUAGE	CALM625 342
GREEN, JULIEN REMARKS ON ALGOL AND SYMBOL MANIPULATION	CALM579 21
GREEN, JULIEN SYMBOL MANIPULATION IN XTRAN	CALM617 30
GREENE, I. GUIDES TO TEACHING COBOL	CALM605 323
GREENWALD, IRWIN D. A TECHNIQUE FOR HANDLING MACRO INSTRUCTIONS	CALM627 72
GREES, MANDALAY A CARD FORMAT FOR REFERENCE FILES IN INFORMATION PROCESSING	CALM611 43
GREES, MANDALAY A SURVEY OF LANGUAGES AND SYSTEMS FOR INFORMATION RETRIEVAL	CALM605 323
GREES, MANDALAY ABBREVIATING WORDS SYSTEMATICALLY	CALM629 436
GRUENBERGER, F. J. A ONE-DAY LOOK AT COMPUTING	CALM602 72
GRUENBERGER, FRED A TERMINOLOGY PROPOSAL	CALM606 300
GRUENBERGER, FRED A TURNING POINT IN THE COMPUTER INDUSTRY	CALM583 3
GRUETTE, MURRAY IBM 704 CODE-MUNDOMS	CALM597 33
GSCHEIND, MANS N. A REAL TIME DATA ASSIMILATOR	CALM604 241
GURZT, FRED A HIGH-SPEED MULTIPLICATION PROCESS FOR DIGITAL COMPUTERS	CALM628 447
GUSEMAN JR, L. P. A SIMPLE SEQUENTIALLY COMPACT PROCESS FOR THE ADJOINTS OF MATRICES OF ARBITRARY INTEG	

I-88

INDEX TO THE COMMUNICATIONS OF THE ACM 1958-1962

BIBLIOGRAPHY

- CACM603 168 THE EXECUTE OPERATIONS, A FOURTH MODE OF INSTRUCTION SEQUENCING • P. P. BROOKS
CACM603 170 AN ALGORITHM DEFINING ALGOL ASSIGNMENT STATEMENTS • ROBERT W. FLOYD
CACM603 171 NUMERICAL INVERSION OF LAPLACE TRANSFORMS • LOUIS A. SCHMITTROTH
CACM604 183 ACM CONFERENCE ON SYMBOL MANIPULATION, PROGRAM AND PREPRINTS
CACM604 184 RECURSIVE FUNCTIONS OF SYMBOLIC EXPRESSIONS AND THEIR COMPUTATION BY MACHINE, PART I • JOHN MCCARTHY
CACM604 195 SYMBOL MANIPULATION BY THREADED LISTS • ALAN J. PERLIS, CHARLES THORNTON
CACM604 205 AN INTRODUCTION TO INFORMATION PROCESSING LANGUAGE V • ALLEN NEWELL, F. TONGE
CACM604 211 SYNTACTIC AND SEMANTIC AUGMENTS TO ALGOL • JOSEPH W. SMITH
CACM604 213 SYMBOL MANIPULATION IN ALTRAN • JULIEN GREEN
CACM604 214 MACRO INSTRUCTION EXTENSIONS OF COMPILER LANGUAGES • M. DOUGLAS MCILROY
CACM604 220 PROVING THEOREMS BY PATTERN RECOGNITION, I • HAO WANG
CACM604 235 DECODING COMBINATIONS OF THE FIRST N INTEGERS TAKEN K AT A TIME • RICHARD M. BROWN
CACM604 236 A CONTROL SYSTEM FOR LOGICAL BLOCK DIAGNOSIS WITH DATA LOADING • M. E. SENKO
CACM604 241 A HIGH-SPEED MULTIPLICATION PROCESS FOR DIGITAL COMPUTERS • FRED GURZI
CACM604 245 AN IMAGINARY NUMBER SYSTEM • DONALD E. KNUTH
CACM604 259 OFFICE OF NAVAL RESEARCH DCN VOL 12 NO 2 APR 60
CACM605 299 REPORT ON THE ALGORITHMIC LANGUAGE ALGOL 60 • P. NAUR, J. W. BACKUS, F. L. BAUER, J. GREEN, C. RATZ, J. MCCARTHY, A. J. PERLIS, M. RUTISHAUSER, K. SAMELSON, R. VAUQUOIS, J. H. WEGSTEIN, A. VAN WIJNGAARDEN, M. WOODGER
CACM605 315 WHAT IS A CODE • G. W. PATTERSON
CACM605 319 DIVISIONLESS COMPUTATION OF SQUARE ROOTS THROUGH CONTINUED SQUARING • DIRAN SARAFYAN
CACM605 321 A START AT AUTOMATIC STORAGE ASSIGNMENT • ROBERT L. PATRICK
CACM605 322 A TECHNIQUE FOR COUNTING ONES IN A BINARY COMPUTER • PETER WEGNER
CACM605 323 ABBREVIATING WORDS SYSTEMATICALLY • JUNE A. BARRETT, MANDALAY GREMS
CACM605 325 BENDIX G-20 SYSTEM
CACM606 339 THE FUTURE OF AUTOMATIC DIGITAL COMPUTERS • ANDREW D. BOOTH
CACM606 342 THE DEPARTMENT OF COMPUTER MATHEMATICS AT MOSCOW STATE UNIVERSITY • I. S. BEREZIN
CACM606 345 COMPILING CONNECTIVES • CHARLES J. SWIFT
CACM606 347 MULTIPROGRAM SCHEDULING, PARTS 1 AND 2. INTRODUCTION AND THEORY • E. F. CUDD
CACM606 351 A SHORT METHOD FOR MEASURING ERROR IN A LEAST-SQUARES POWER SERIES • S. M. ROBINSON, G. W. STRUBLE
CACM606 352 CONVERSION BETWEEN FLOATING POINT REPRESENTATIONS • C. PERRY
CACM606 355 THE SOLUTION OF SIMULTANEOUS ORDINARY DIFFERENTIAL EQUATIONS USING A GENERAL PURPOSE DIGITAL COMPUTER • W. H. ANDERSON
CACM606 361 INTERVAL ESTIMATION OF THE TIME IN ONE STATE TO TOTAL TIME RATIO IN A DOUBLE EXPONENTIAL PROCESS • W. R. NEAL
CACM606 367 ATLAS, A NEW CONCEPT IN LARGE COMPUTER DESIGN
CACM606 380 TURNING POINT IN THE COMPUTER INDUSTRY • FRANCIS WAGNER, JEANETTE ORGILL, FRED GRUENBERGER
CACM607 407 DIGITAL COMPUTERS IN UNIVERSITIES
CACM607 408 SOME THOUGHTS ON RECONCILING VARIOUS CHARACTER SET PROPOSALS • EDWARD A. VOORHEES
CACM607 409 THE MULTILINGUAL TERMINOLOGY PROJECT • J. E. HOLMSTROM
CACM607 413 MULTIPROGRAM SCHEDULING, PARTS 3 AND 4. SCHEDULING ALGORITHM AND EXTERNAL CONSTRAINTS • E. F. CUDD
CACM607 418 COMBINING ALGOL STATEMENT ANALYSIS WITH VALIDITY CHECKING • PAUL MCISAAC
CACM607 420 PROGRAMMING COMPATIBILITY IN A FAMILY OF CLOSELY RELATED DIGITAL COMPUTERS • WILLIAM F. LEUBBERT
CACM607 439 OFFICE OF NAVAL RESEARCH DCN VOL 12 NO 3 JUL 60
CACM608 463 MELIAC, A DIALECT OF ALGOL • HARRY D. HUSKEY, M. H. HALSTEAD, R. CARTHUR
CACM608 468 A SHORT STUDY OF NOTATION EFFICIENCY • HOWARD J. SMITH JR
CACM608 476 DIGITAL COMPUTERS IN UNIVERSITIES, II
CACM609 488 AN INTRODUCTORY PROBLEM IN SYMBOL MANIPULATION FOR THE STUDENT • ROBERT F. ROSIN
CACM609 490 TRIE MEMORY • EDWARD FREDKIN
CACM609 500 RAPIDLY CONVERGENT EXPRESSIONS FOR EVALUATING E TO THE X • A. BERIN
CACM609 501 COMMENTS FROM A FORTRAN USER • JOHN M. BLATT
CACM609 509 A DECISION RULE FOR IMPROVED EFFICIENCY IN SOLVING LINEAR PROGRAMMING PROBLEMS WITH THE SIMPLEX ALGORITHM • J. C. DICKSON, F. P. FREDERICK
CACM609 513 DIGITAL COMPUTERS IN UNIVERSITIES, III
CACM600 519 REPORT ON A CONFERENCE OF UNIVERSITY COMPUTING CENTER DIRECTORS
CACM600 522 CONFERENCE REPORT ON THE USE OF COMPUTERS IN ENGINEERING CLASSROOM INSTRUCTION
CACM600 528 AUTOMATIC GRADERS FOR PROGRAMMING CLASSES • JACK HOLLINGSWORTH
CACM600 530 DO IT BY THE NUMBERS, DIGITAL SHORTHAND • R. W. BEMER
CACM600 536 COMMENT ON 'DECODING COMBINATIONS OF THE FIRST N INTEGERS TAKEN K AT A TIME' • JULIUS LIEBLEIN
CACM600 537 A LIST OF COMPUTER SYSTEMS PROGRAMS FOR THE IBM 650, DATATRON 205, AND UNIVAC 55-80
CACM600 538 COMMENTS ON A TECHNIQUE FOR COUNTING ONES • P. R. SHERMAN
CACM600 539 SOME THOUGHTS ON PARALLEL PROCESSING • L. D. YARBROUGH
CACM600 540 CORRIGENDA TO 'SOME THOUGHTS ON RECONCILING VARIOUS CHARACTER SET PROPOSALS'
CACM600 541 EVALUATING NUMBERS EXPRESSED AS STRINGS OF ENGLISH WORDS • CHARLES J. SWIFT
CACM600 542 A NOTE ON THE CALCULATION OF INTEREST • P. Z. INGERMAN
CACM600 544 DIGITAL COMPUTERS IN UNIVERSITIES, IV
CACM600 575 OFFICE OF NAVAL RESEARCH DCN VOL 12 NO 4 OCT 60
CACM600 605 AN ALGORITHM FOR THE ASSIGNMENT PROBLEM • ROLAND SILVER
CACM600 607 COMPILATION FOR TWO COMPUTERS WITH MELIAC • KLEBER S. WASTERSON JR
CACM600 611 LEAST SQUARES FITTING OF A GREAT CIRCLE THROUGH POINTS ON A SPHERE • LEENDERT DE WITTE
CACM600 614 OVER-ALL COMPUTATION CONTROL AND LABELLING • ANATOL HOLT
CACM600 616 A SIMPLE TECHNIQUE FOR CODING DIFFERENTIAL EQUATIONS • P. SEFTON, R. VAILLANCOURT
CACM600 617 NOTE ON EIGENVALUE COMPUTATION • JAN F. ANDRUS
CACM600 618 AN ESTIMATION OF THE RELATIVE EFFICIENCY OF TWO INTERNAL SORTING METHODS • H. MAGLER
CACM600 621 THE SUMADOR CHINO • JAMES L. ROGERS
CACM600 622 CHARACTER SCANNING ON THE IBM 707 • A. E. SPECKHARD
CACM600 632 OPTIMIZERS, THEIR STRUCTURE • R. F. WHEELING
CACM600 638 SURVEY OF PUNCHED CARD CODES • H. J. SMITH JR, F. A. WILLIAMS JR
CACM600 639 SURVEY OF CODED CHARACTER REPRESENTATION • R. W. BEMER
CACM600 644 ON PROGRAMMING THE NUMERICAL SOLUTION OF POLYNOMIAL EQUATIONS • KENNETH W. ELLENBERGER
CACM600 648 FIBONACCIAN SEARCHING • DAVID E. FERGUSON
CACM600 649 A NOTE ON APPROXIMATING E TO THE X • S. LUBKIN
CACM600 649 PROGRAMMED ERROR CORRECTION IN PROJECT MERCURY • B. DIMSDALE, G. M. WEINBERG
CACM600 652 MULTIPLE PRECISION ARITHMETIC • DAVID A. POPE, MARVIN L. STEIN
CACM600 655 A METHOD FOR OVERLAPPING AND ERASURE OF LISTS • GEORGE E. COLLINS
CACM600 658 TWO METHODS FOR WORD INVERSION ON THE IBM 709 • ROBERT A. PRICE, PAUL DES JARDINS
CACM600 659 DIGITAL SIMULATION OF DISCRETE FLOW SYSTEMS • C. J. MOORE, T. S. LEWIS
CACM600 661 COPE (CONSOLE OPERATOR PROFICIENCY EXAMINATION) • DAVID FARMAN, RICHARD KETOVER
CACM600 663 A COMPARISON OF 650 PROGRAMMING METHODS • T. B. CURTZ, J. F. RIORDAN, M. SPOHN
CACM611 3 A BASIC COMPILER FOR ARITHMETIC EXPRESSIONS • W. D. HUSKEY, W. H. WATTENBURG
CACM611 10 RECURSIVE PROCESSES AND ALGOL TRANSLATION • A. A. CRAU
CACM611 15 USE OF MAGNETIC TAPE FOR DATA STORAGE IN THE ORACLE-ALGOL TRANSLATOR • H. BOTTFENBAUCH
CACM611 19 THE CLIP TRANSLATOR • DONALD ENGLUND, ELLEN CLARK

情報を数枚のカードにパンチし、計算機にインプットする。

(2) 計算機は著者名とかタイトルの語の頭文字をいくつかとり出し、それと発行年度の数字を組合せ、自動的にドキュメントのIDコードを作り出す。

(3) タイトルを構成している語のうち、前置詞、冠詞、接続詞、代名詞といった語を除外し、その他の語をキー・ワードとして自動的に選び出す。キー・ワードに従ってタイトルをグループ分けし、キー・ワードがリストの中央にくるようにタイトルを左右にずらし、またキー・ワードがアルファベット順に並ぶように、タイトルとIDコードをプリントし「KWIC索引」を作る(第6.1.1図参照)。

(4) 著者名もアルファベット順にIDコードと共にリストした「著者名索引」をプリントする(第6.1.2図参照)。

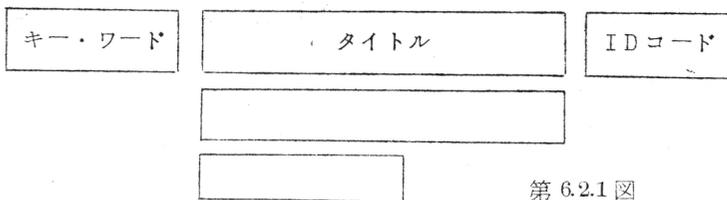
(5) 一つの文献についてインプットした情報を適当にまとめ、IDコードを見出語にした「文献目録(Bibliography)」をプリントする(第6.1.3図参照)。

現在のシステムは、IBM7090、UNIVACIIIなどにできている。

6.2 KWOC索引システム

KWICシステムでは、(i)リストできる表題の長さの制限があり、(ii)表題がローテイトしているために見にくいという欠点がある。そこでこのシステムではタイトルから選びだしたキー・ワードを文脈の外に出し、第6.2.1図に示すようにリストすることになっている。

三菱原子力で開発したユニタム索引システムはこのKWOCシステムの一つであるが、このシステムでは、KWICのインプット情報に加えて新たにサブタイトルもインプットできるようにになっている。



第6.2.1図

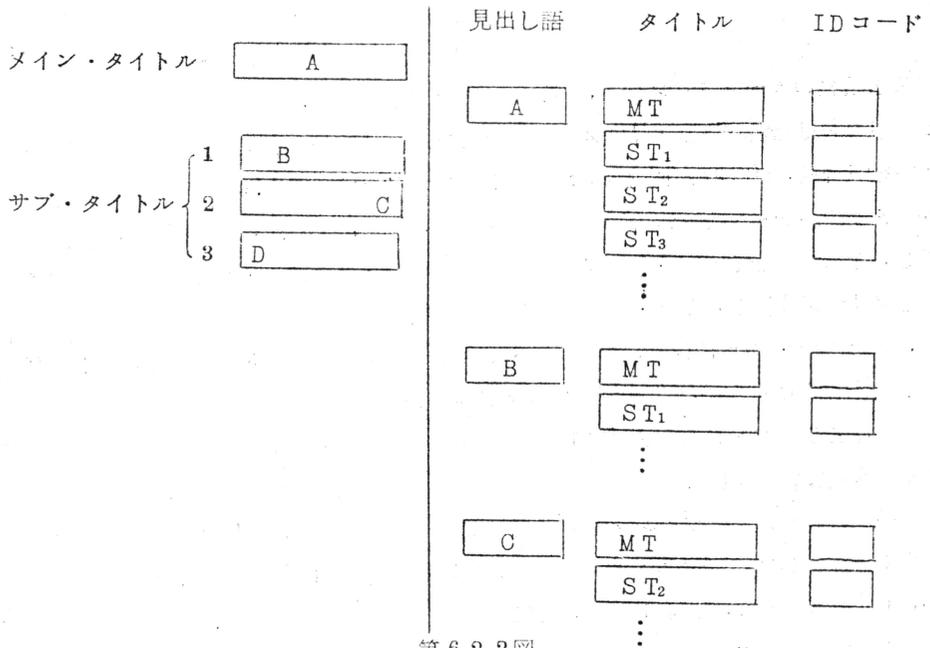
メイン・タイトルとそれに属するいくつかのサブ・タイトルが第6.2.2図の左に示すようなキー・ワードをもつているとすると、このシステムのKWOC索引の一部は右のようになる。

6.3 SDI (Selective Dissemination of Information) システム

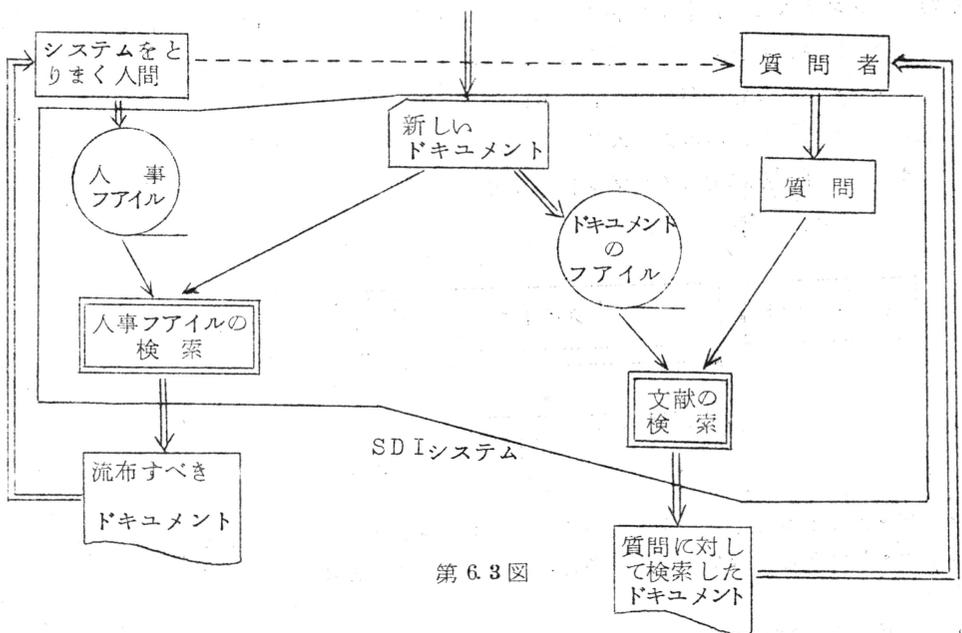
このシステムは情報の「検索」と「流布」という2つの機能をもつている(第6.3図を参照) このシステムは次のような2つのファイルをもつている。

- (i) ドキュメントのファイル
- (ii) 人事ファイル

これは情報の利用者がどんな方面に興味をもち、どんな類のドキュメントを読みたが



第 6.2.2 図



第 6.3 図

ついているかを、ドキュメントにつける見出し語に使用する語で表現したレコードからきている。
 新しいドキュメントがシステムに入ってくると、システムは人事ファイルを検索し、該当する利用者とそのドキュメントを配布する。

利用者からの問合せに対しては、ドキュメントのファイルを検索し、問合せに適合するドキュメントをその利用者に提供する。

6.4 SIR (Selective Information Retrieval) システム

これはRCAのマネジメントサイエンス部門が「人事ファイルの検索」のために開発したシステムである。

(1) 人事ファイル

各レコードに含まれる項目は固定長で、ファイルはSingle realでもMulti-realでもかまわない。

(2) 見出し語の作成

ドキュメント・リトリバルの見出し語の作成に対応する作業として、このシステムで検索する人事ファイルの1レコード内の各項目(職員番号、氏名、年齢、扶養家族の人員数、居住地、特技など)に対して、それぞれ5字以内の名前、およびその項目のファイル内での桁数をDictionary Cardを使つて指定し、記憶装置内にシリーラスを作る。

(3) 質問は、“AND”“OR”、数値の大小関係を示す記号(EQ, GEQといったもの)、およびカットと定義してある項目名とを適当に用いて書き、これをInquiry Cardにパンチして計算機(HITAC3010)にインプットする。

(4) またこのシステムでは回答のアウトプット形式も指定することができる。

このシステムは人事ファイルの検索のために開発されたものであるが、情報のファイルの項目を変更するだけで、カタログの検索、パテントの検索、化学の化合物の検索といった比較的簡単なファクト・リトリバルに応用することができる。

7. あとがき

インフォメーション・リトリバルに関しては、日本でも最近各所で研究しはじめられている。日本電子工業振興協会でも本年6月よりIR委員会を設けて、特に電子計算機を中心とするリトリバル・システムの研究を行つている。本報告はIR委員会の成果の一部をまとめたものである。IR委員会のメンバーは関根智明、昆野誠司、および釜井昌之、黒沢俊雄、水谷静夫、西村恕彦、大駒誠一、高橋達郎、竹内寿、渡辺一郎、山崎利治の諸氏である。

参 考 文 献

- [1] Y. Bar-Hillel
“A Logician reaction to recent theorizing on information search systems”
American Documentation Vol. 8, No. 2, 1957
- [2] W. S. Cooper
“Fact Retrieval and Deductive Question-Answering Information Retrieval Systems”
JACM Vol. 11, No. 2, 1964
- [3] “Report on the Testing and Analysis of an Investigation into the Comparative Efficiency of Indexing Systems”
Cranfield, Oct. 1962
- [4] “分類索引システムの効率比較研究”

- 日本科学技術情報センター, 1963
- [5] H. P. Luhn
"The automatic creation of literature abstracts"
IBM. J. Research & Dev., Vol. 2, No. 2, 1958
- [6] H. P. Edmundson, R. E. Wyllys
"Automatic abstracting and indexing-Survey and recommendations."
CACM. Vol. 4, No. 5, 1961
- [7] W. D. Climenson, N. H. Hardwick
"Automatic syntax analysis in machine indexing and abstracting"
Am. Doc. Vol. 12, No. 3, 1961
- [8] H. Borko and M. Bernik
"Automatic Document Classification"
JACM Vol. 10, No. 2, 1963
- [9] H. Borko and M. Bernik
"Automatic Document Classification Part II Additional Experiment"
JACM Vol. 11, No. 2, 1964
- [10] B. H. Scheff
"A Catalogue Entry Retrieval System"
CACM Vol. 6, No. 7, 1963
- [11] E. H. Sussenguth, Jr
"Use of Tree Structures for Processing Files"
CACM Vol. 6, No. 5, 1963
- [12] H. A. Clam Pett, Jr
"Randomized Binary Searching With Tree Structures"
CACM Vol. 7, No. 3, 1964
- [13] E. H. Sursenguth, Jr
"Automatic Structure-Matching Procedures"
The Computation Laboratory of Harvard University Information
Storage and Retrieval, No. ISR-5, January, 1964
- [14] H. E. Stiles
"The Association Factor in Information Retrieval"
JACM Vol. 8, No. 2, 1961
- [15] C. R. Blair
"A Program for correction Spelling Errors"
Information and Control, 3, 1960
- [16] F. J. Damerau
"A Technique for Computer Detection and Correction of
Spelling Errors"
CACM Vol. 7, No. 3, 1964
- [17] "情報管理便覧"
日刊工業新聞社, 1963
- [18] D. R. Swanson
"Searching natural Language Text by Computer"
Science Vol. 132, Oct., 1960

本 PDF ファイルは 1965 年発行の「第 6 回プログラミング—シンポジウム報告集」をスキャンし、項目ごとに整理して、情報処理学会電子図書館「情報学広場」に掲載するものです。

この出版物は情報処理学会への著作権譲渡がなされていませんが、情報処理学会公式 Web サイトの https://www.ipsj.or.jp/topics/Past_reports.html に下記「過去のプログラミング・シンポジウム報告集の利用許諾について」を掲載して、権利者の検索をおこないました。そのうえで同意をいただいたもの、お申し出のなかったものを掲載しています。

過去のプログラミング・シンポジウム報告集の利用許諾について

情報処理学会発行の出版物著作権は平成 12 年から情報処理学会著作権規程に従い、学会に帰属することになっています。

プログラミング・シンポジウムの報告集は、情報処理学会と設立の事情が異なるため、この改訂がシンポジウム内部で徹底しておらず、情報処理学会の他の出版物が情報学広場 (=情報処理学会電子図書館) で公開されているにも拘らず、古い報告集には公開されていないものが少からずありました。

プログラミング・シンポジウムは昭和 59 年に情報処理学会の一部門になりましたが、それ以前の報告集も含め、この度学会の他の出版物と同様の扱いにしたいと考えます。過去のすべての報告集の論文について、著作権者（論文を執筆された故人の相続人）を探し出して利用許諾に関する同意を頂くことは困難ですので、一定期間の権利者搜索の努力をしたうえで、著作権者が見つからない場合も論文を情報学広場に掲載させていただきたいと思えます。その後、著作権者が発見され、情報学広場への掲載の継続に同意が得られなかった場合には、当該論文については、掲載を停止致します。

この措置にご意見のある方は、プログラミング・シンポジウムの辻尚史運営委員長 (tsuji@math.s.chiba-u.ac.jp) までお申し出ください。

加えて、著作権者について情報をお持ちの方は事務局まで情報をお寄せくださいますようお願い申し上げます。

期間：2020 年 12 月 18 日～2021 年 3 月 19 日

掲載日：2020 年 12 月 18 日

プログラミング・シンポジウム委員会

情報処理学会著作権規程

<https://www.ipsj.or.jp/copyright/ronbun/copyright.html>