44—S 001

ファクトリトリーバルに関する
理論的考察

(遠隔情報処理システムの開発)

昭和45年3月

財団法人 日本情報処理開発センター

序に代えて

社会、経済の発展にともない、各種情報の蓄積、加工、供給を有機的かつ効果的に行なう方法として、特にコンピュータによる情報処理の役割りの重要性が認識されております。

また、最近は情報社会への指向とあいまつて情報処理分野の拡大とともにその高度化の方向が検討されつつあり、大きな発展期を迎えているといえます。

しかし、このような情勢において、情報処理および情報処理産業の前途には、その発展の要件およびこれが他産業に与える影響といつたわが国経済の動向に関連する諸問題を始め情報処理方式、ハードウエアおよびソフトウエアの技術、各種の標準化、情報処理技術者の養成等、解決を要する幾多の課題があります。

当財団は情報処理に関するこれら諸問題解決のため、各種の調査、研究事業を実施しておりますが、この研究報告書はその一環として、情報管理の基礎となる情報検索システムの1技法であるファクトリトリーバルの理論的研究を推進するため、財団法人大阪科学技術センターにその研究を委託した結果をとりまとめたものであります。

なお、この事業は日本自転車振興会の機械工業振興資金による「昭和44年度情報処理に関する調査・研究補助事業」のうち「遠隔情報処理システムの開発」の一部として実施したものであります。

ここに本研究実施にご尽力ならびに御支援を賜わつた関係各位に心より感謝の意を表しますとともに、本報告が各方面に利用され、わが国情報処理産業発展の一助として寄与できますよう念願いたす次第であります。

昭和45年3月

財団法人　日本情報処理開発センター

会　長　　難　波　捷　吾

はじめに

情報工学は、情報の収集・蓄積・加工、ならびにその利用の技術と、加工、利用の方法を規定する基準を設定し、利用から行動・情報発生へと連なる一連のプロセスを解明する科学よりなる。

前者はいわゆる情報処理技術であり、後者は社会工学の要素をもつ。これを一つのシステム工学として見るならば、この両者を融合し、一つの体系として作り上げていくことが自ら要求される。今日の情報処理技術研究に課せられた重要な研究課題の一つがここに見出される。

われわれは、このような背景に立って、情報の収集、蓄積、検索技術の開発を目的とし、昭和40年、大阪科学技術センターに情報検索システム研究会が設置され、さらに昭和44年日本情報処理開発センターとの共同研究態勢が設立されたのを機に、その方面の開発研究に着手してきた。この間、研究者は情報検索システムならびに図面検索システムの構成をはじめ、情報の分類、蓄積のための代数系の作成にたづさわり、その成果を発表してきた

しかし、情報の円滑な利用は、その利用者の環境条件に関係なく、あくまでも自然語による即時的遠隔情報処理システムが可能になってはじめてその門が開かれるものと考えられ、これはまた今日の情報工学発表のための急務であると考えられる。本報告書はこのような観点より行なわれた研究の成果である。この方面への研究のいと口ともなれば幸甚である。

昭和45年3月

大阪科学技術センター情報検索システム研究会
代表幹事　手塚慶一

情報検索システム研究会メンバー

手塚慶一	大阪大学工学部
長谷川利治	京都大学工学部
豊田順一	大阪大学基礎工学部
梶谷浩二	近畿大学工学部
打浪清一	大阪大学工学部
鶴身征雄	大阪大学工学部
篠原健	大阪大学工学部

目　　次

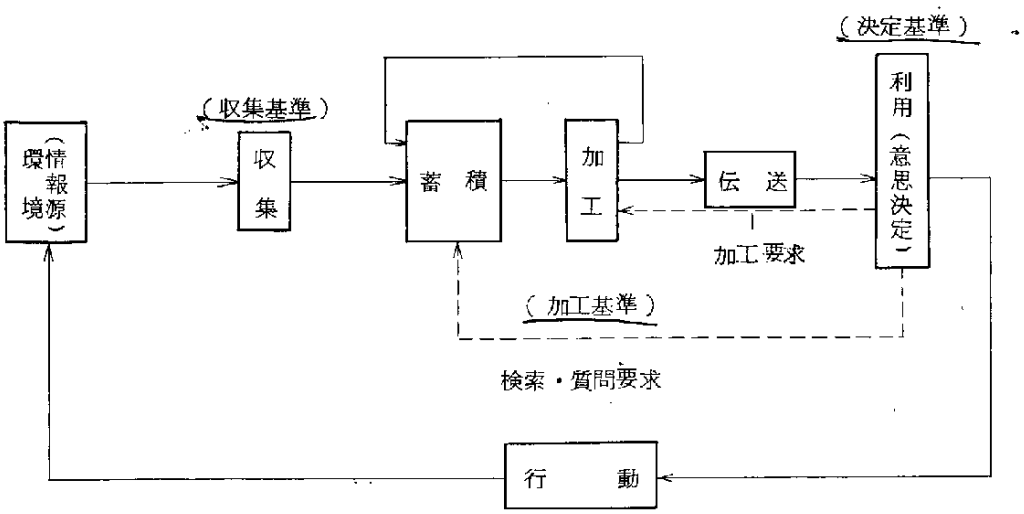
I （総　　論）

1. 緒　　論

人間社会における合理化、能率化の精神は、わが国においては２０年前のオペレーションズ・リサーチの導入に端を発し、種々の角度から論ぜられ、実地への適用と改良を積み重ねながら今日の成長をもたらすに至っている。

しかし、そこには電子計算機の急速な進歩・普及が大きな援助となっていることを見逃すことができず、とくに電子計算機のもつ大容量情報処理能力は、高度の合理化技術を実際に有効ならしめた大きな力となっており、ここに今日の情報化時代を生み出した一因を見出すことができる。人間生活の中にしめる電子計算機の役割のうち、とくに重大な効果をもたらすものに情報検索がある。誰でもが、いつ、いかなるときでも、要求する情報を適確に、直ちに入手しうることが、正しい意思決定をもたらせ、生活を合理的に向上せしめうる上に極めて重要であることは論を俟たない。

一般的な情報システムの構成原理図を図Ⅰ－1に示す。



図Ⅰ－1　情報システム原理図

この図Ⅰ－1より明らかなように情報検索技術は、高信頼度をもつ大容量記憶装置開発のハードウエア技術、情報の収集、適用（意思決定）を合理化するシステム工学的技術ならびに情報の

分類、蓄積・読み出しを高速適確に行なわしめるソフトウエア技術の3部門よりなる。とくに今日のわが国情報処理技術と情報検索におけるソフトウエアの重要性を対比させた場合、これの開発の必要性を改めて認識せざるを得ないのである。われわれの日常用いる自然語により、いかなる遠隔の地点よりも、要求する情報を入手しうるソフトウエアの開発は今日、人間社会の能率を向上させ、また漸くその必要性が認められつつあるナショナル・インフォーメーションのシステムを作り上げる上において不可欠なものであろう。

計算機のバッチ処理による数値計算を第一世代のソフトウエアと呼ぶならば、ＭＡＣ（Machine Aided Cognition）処理による意味処理は第2世代のソフトウエアと呼ぶことができる。

データ伝送回線の一般への開放が行なわれようとしている今日は、まさにこの第2世代が開かれようとする情報工学的改革時点であるとみることができる。今日まで、計算機は主に給料計算、経理計算などの数値計算には大きな効果をもたらしたが、意味的な処理のからんだものには余り役にはたっていない。しかし、今日ではＭＡＣ使用などに見られるように、人間と機械がオンラインで会話を行ない、互いにその短所を補いながら非数値の情報の処理（連想思考、情報検索、機械翻訳、その他発見的・創造的思考）をはじめ、より高次の情報処理を行なうことが強く要請されるようになつて来ている。

数値計算には充分利用できる計算機が前述のように高次の情報処理に用いえない原因は、意味そのものの定義は勿論、意味の解釈と処理、各種思考のアルゴリズムなどが明らかになつていない点にあるとみることができる。これらの諸点の解明なくしては本来の情報検索システムを構成することは不可能である。

われわれはこのような観点から、遠隔情報処理システムの開発研究に当り、そのシステムが所期の目的に叶う動作を行なうような、システム構成の技術の確立を目指した。すなわち、日本語による（自然語で表わされた）応答系をもつシステムの可能性を言語の意味論的な2つの立場（自然語の意味的樹枝状表現と位相的表現）から確かめ、その設計法を明示することができた。

とくにⅢ以後に示した意味の位相性を表現するための意味空間（シソーラス）の導入、ならびに、それを用いた文献情報検索システムの情報表現は、従来この種の研究で用いられていなかつた考え方であり、これにより、情報の構造を意味論的に明らかにし、検索論理システムなら

びにそのプログラムの開発と実用化に指針を与え得たものと信ずる次第である。この点に対し、大方の御批判を抑ぎたい。

2. 遠隔制御処理システムの基本構想

非数値の情報の加工、処理などを行なうシステムは、一度で所期の結果を得ることが少なく、アルゴリズムやパラメータを適当に修正しながら結果を求めていくという発見的思考、試行錯誤法を行なってゆかねばならない。

また特殊な大容量データは、大きなデータバンクにしか蓄積されてないことが多い。

このような情報あるいは計算機の使用を行なうものとして、遠隔制御オンラインシステムの開発が望まれる。このようなシステムでは中央処理装置を遊ばせない為に、当然ＴＳＳのマルチ処理を行なうことになる。

以下にそのシステム基本構想の概要をのべる。

2－1 遠隔制御業務

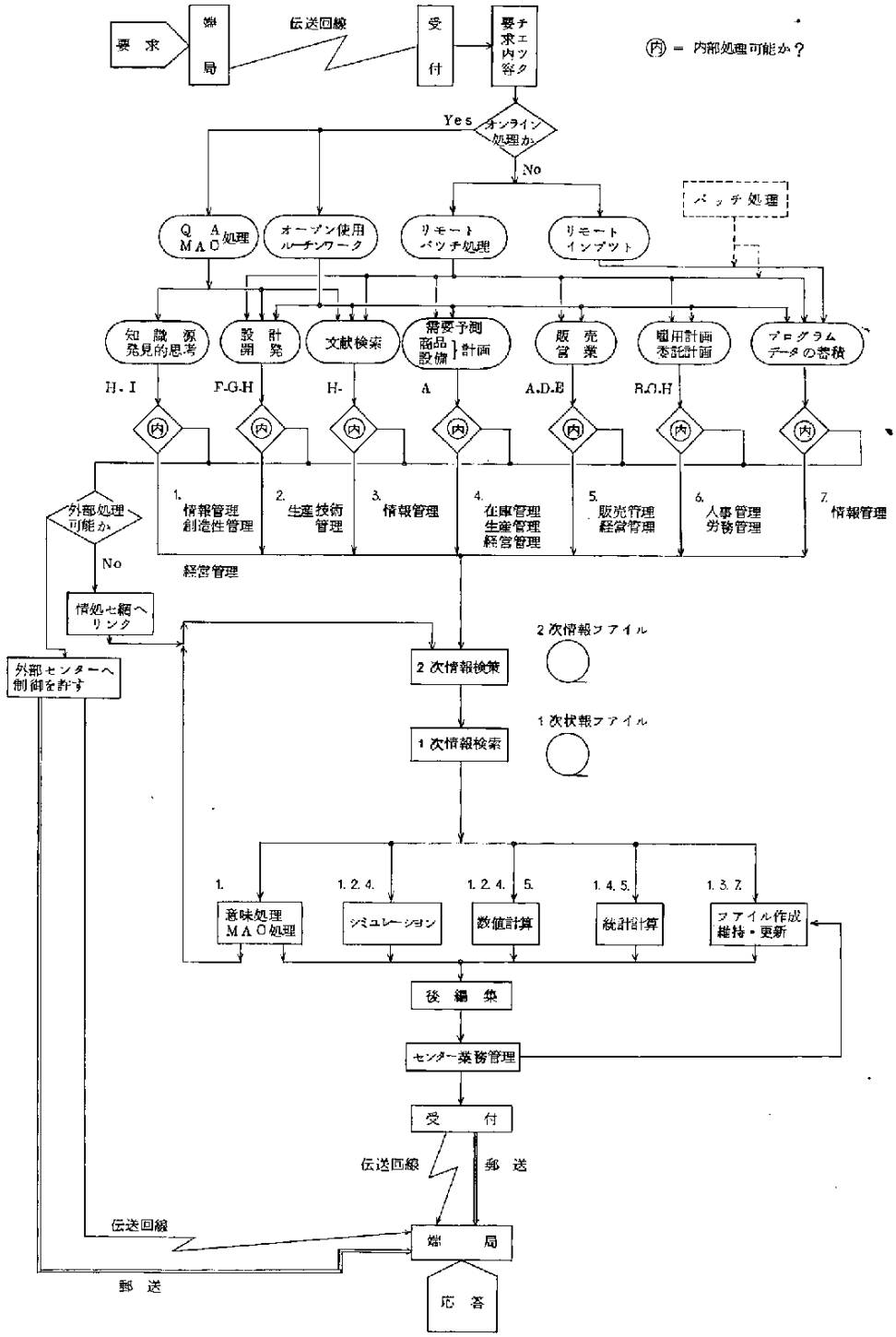
計算センターで、サービスを行なうのは、バッチ（クローズド）と、遠隔オンライン利用とに大別されるが、遠隔制御業務機能図を図Ⅰ－2に示す。これは、私有ファイルだけを用いて行なう処理は省略し、公開情報を併用して処理する場合を挙げており、処理業務には、次のような方法が考えられる。

(i)バッチ処理 (ii)リモートバッチ処理 (iii)オンライン・オープン処理（ルーチン・ワーク）
(iv)オンラインＭＡＣ処理

次に、これを使用プログラム、データの与え方から分類すると、次のように分類できる。

(1) クローズド・タスク（計算に使うプログラム、データをも使用者が揃えて出すもの）

(2) オープン・タスク（計算に使用するプログラムに、データバンクのライブラリを使用したり、使用するデータに、データバンクのものを呼び込んで、自センターＣＰＵを用いて処理を行なうもの）



図Ⅰ－2　計算センター遠隔制御業務機能図

(3) リモート・タスク（オープンの場合は、プログラムライブラリやデータを、端局直結のＣＰＵに伝送回線を通じ他センターから呼びこんだのに対し、これはライブラリやデータのあるデータバンク直結のセンターへプログラムを伝送し、そこのＣＰＵを用いて処理するもの）

(4) ハイブリッド・タスク（システムの能力、経済性から(2)(3)を混用するもの）

(1)の場合は端局と、それに直結する計算センターのシステムだけで話が出来るが、(2)以下の場合は、地区、中央の情報処理センターとリンクしたシステムを考えなくてはならない。

それらの情報処理センターの機能図を図Ⅰ－３に示す。

またこれらセンターの持つべき、データバンクのデータの種類と、それの用途を表Ⅰ－１に示す。

表Ⅰ－１　公開共通情報一覧

	情報種別	情報用途概略	情報の形	処理内容	サービスの型
A	経済動向 国民生活 財務統計	需要予測、設備計画 商品計画 マーケッテイング	統計 調査報告	情報検索 統計計算	2
B	労働動向	雇用計画	統計 ニュース	情報検索 統計計算	3 （２）
C	特殊技能者動向	雇用計画 委託研究、委託業務	統計　技能 一覧表　所在 ニュース　勤務先	情報検索	2 3 4
D	商品現場、物価 荷動き、株価	購売 営業	ニュース	情報検索 統計計算	2 3
E	製品、部品、材料	購売、営業、設計 マーケテイング	カタログ資料 仕様書	情報検索 統計計算	2 3 4
F	生産技術	設計、製造	方法・データ	情報検索 計算	
G	管理システム	経営管理システムの設計	方法・データ	情報検索	
H	文献、論文 研究報告、特許	研究開発 設計	二次情報リスト 本文	情報検索	2 3 4
I	百科辞典 辞書	知識源 発見的思考	辞典、辞書 類語集	情報検索 意味処理	1

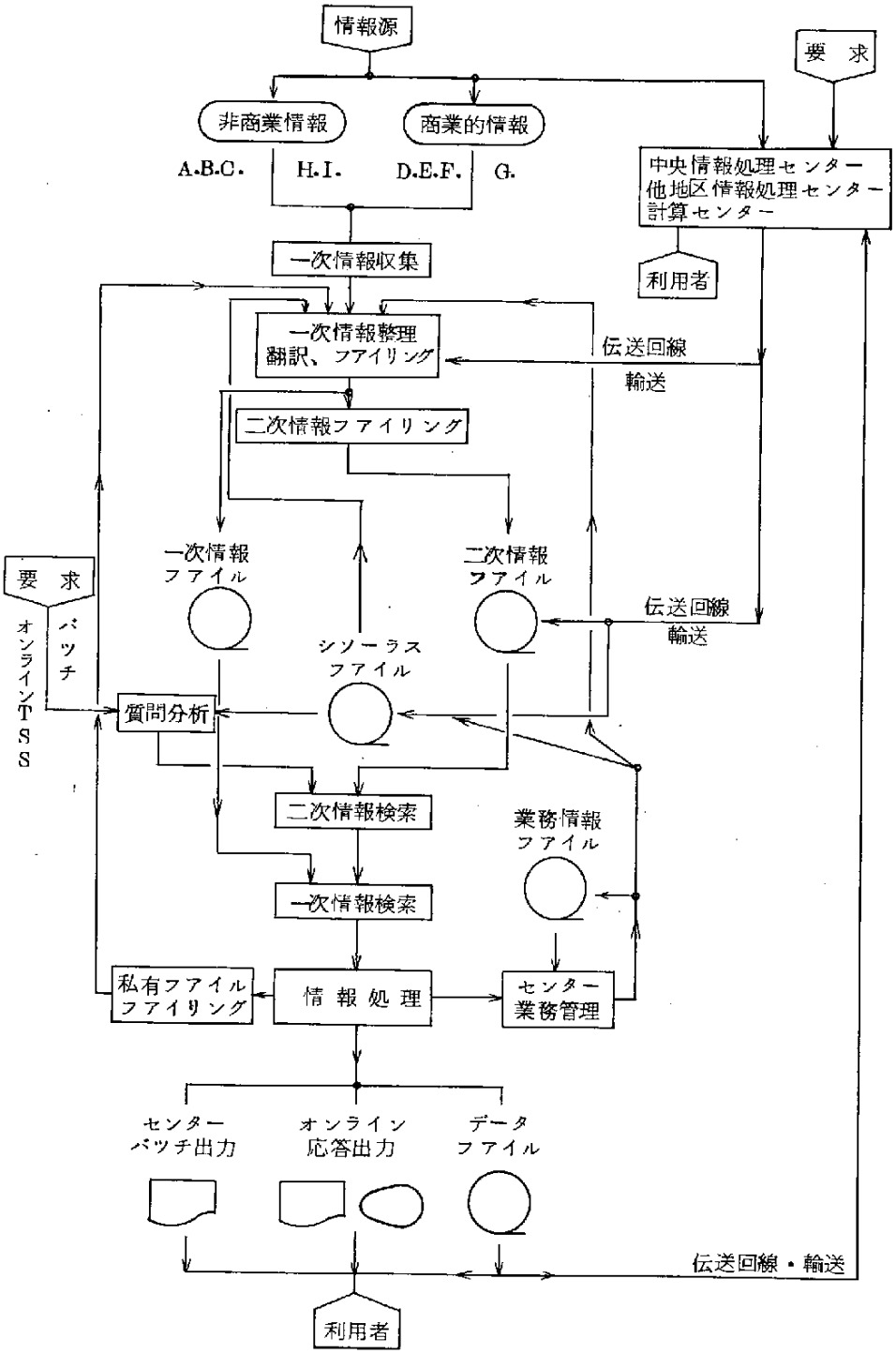


図 1－3　中央／地区 情報処理センター機能図

これらの業務内容のうち、単能のもの（例、切符予約）は技術的てもさほど問題はない。

オンラインのその長所を生かす使い方としては、処理途中で、使い方がわからない部分は計算機から教えてもらい、計算途中でも、途中結果をみながら、プログラムとかパラメータを変更、修正、ファイル化を実時間で行なうところにある。

そのような処理を行なわす為には、意味解釈、情報検索の処理が大きな問題となる。

そこで、本研究ではこの意味解釈、情報検索を主眼に論ずる。

2－2　システム構成

中央にメインデータバンクを持つ中央情報処理センターをおく。地区毎に地区情報処理センターをおき、その地区でよく使用するデータを蓄積したサテライトデータバンクを添える。その下に個別の計算センターをおく。

計算センターの構成を図Ⅰ－4に示す。

各レベルの処理センターは、高速の伝送回線で、センターと端局は低速の伝送回線で結合する。

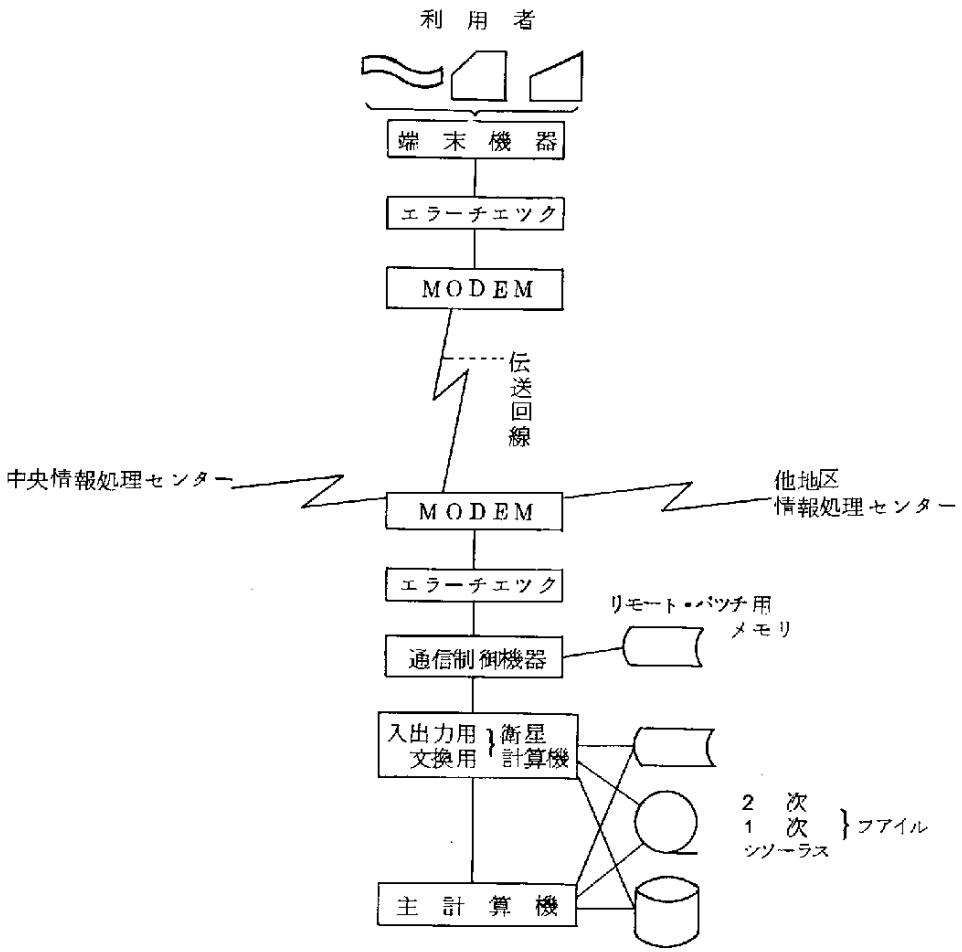
オペレーテイング・システムは、制御プログラム、処理プログラム、支援プログラムから構成されるが、バッチ処理のみのオペレーテイングシステムに比べ、情報ネットとのリンクによる、システム資材の有効な管理を行なう為の入出力管理プログラム、Machine Aided Cognitionを可能にする為のコマンド、会話用言語、共通公開情報、私有情報のファイリング、データのアクセス等が必要となる。

2－3　コ　マ　ン　ド

コマンドの機能としては次のものが必要である。

(1) 登録、終処理　(2) 翻　訳　(3) 実　行　(4) 使用法指導　(5) 情報網へのInterrupt命令　(6) ファイル、ライブラリの保存引用、加工命令

コマンドは、計算機からの応答に対し、リアルタイムで人間が返答しなくてはならない為に、出来るだけ自然語に近く、複雑な変数をとる文法ではなく、使いやすいものでなければならない。

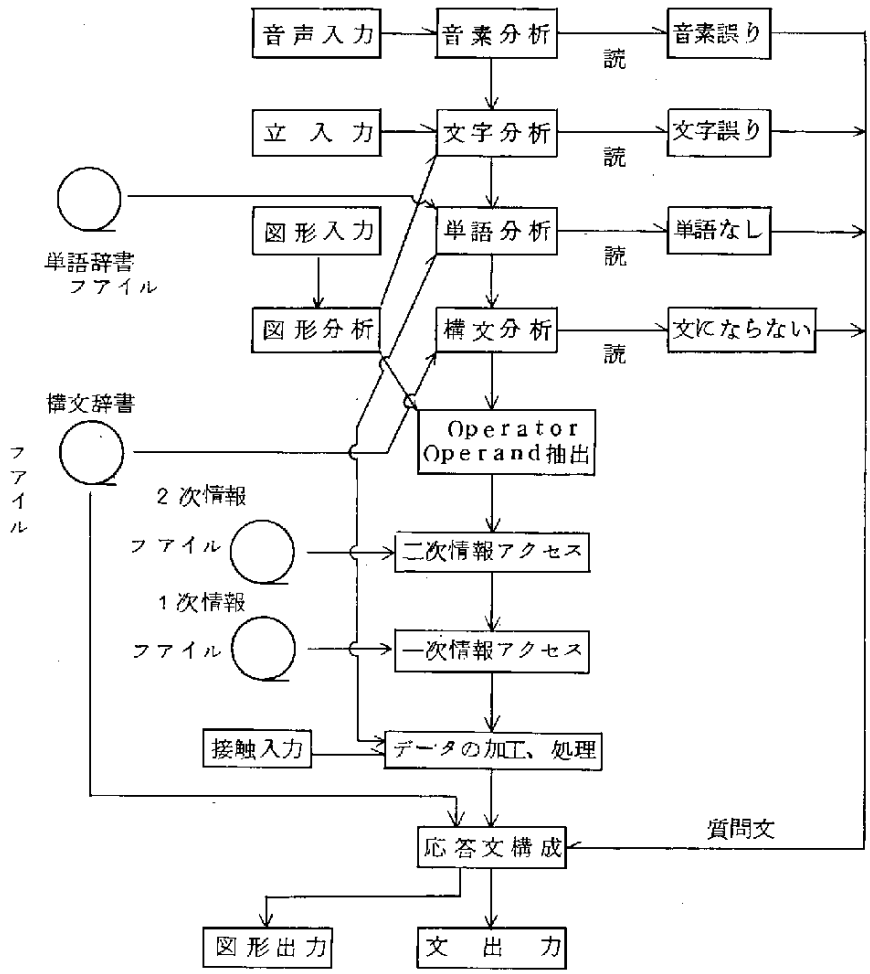


図Ⅰ－4　遠隔制御端末局からの利用フローチャート

2－4　会話用言語（プロセッサ）

heuristicな処理が出来ることが望ましい。コンパイラは、インタプリテイブな動作も行なえ、任意の命令、部分プログラムの実行ができその翻訳時間も人間の思考速度にあつていること、またＩＲもリアルタイムで可能なことなどが必要であり、これらを考慮して、会話用言語を構成しなくてはならない。

また会話、応答過程を詳しく分析したが、その概略を図１－５に示す。



図Ⅰ－5　会話、応答過程

3.　樹枝状データ構造と位相型データ構造

情報検索や会話応答システムを構成するには、そのシステムで使用する言語と、データの構造のとらえ方およびその表現法が問題となる。

自然語もしくは、自然語に近い形式言語を入力とするとき、統辞論的な表現レベルが存在する。自然語または、それに近い形式言語で表現されるレベルを表層レベルとよぶ。

表層レベルの各表現は、統辞論によりその生成構造が解明される。これらが解明され、やはり統辞的なシステム内の言語へ翻訳されるが、この表現レベルを深層レベルとよぶ。例えば、構文分析した結果のPマーカ表現であつたり、逆ポーランド記法であつたりする。このレベルに於ては、データの表現もやはり統辞的な形で行なわれる。例えばアイウエオのリストとか、階層分類とか、類語集などの類である。このような表現を樹枝状データ表現とよぶ。

表層レベルと深層レベルの二層で、検索などの処理を行なうものを、樹枝状応答システムとよぶ。この二レベル処理はChomsky の説に基くものである。

しかし意味的な処理は、この二レベルだけでは十分に表現や処理を行なえないので、もう一段奥に、情報の意味の位相性を表現する意味レベルを考える。意味レベルは一般に近傍空間を台として用い、これを情報空間として使用する。表層、深層の他にこの意味レベルと、計三レベルを考え意味処理を行なわそうというのは筆者の一人、打浪の提唱した方法であるが、この場合の意味レベルにおけるデータ表現を位相型データ構造表現とよぶ。そしてこれらを用いての検索、会話応答系を位相型会話応答システムとよぶ。

樹枝状データ構造では、その取扱い処理が殆ど統辞処理ですむのに対し、位相型データ構造では、統辞処理だけではなく、統計処理を含む意味処理が入つてくる為に複雑にはなるが、よりきめの細い処理が行なえる。

これらの間の関係を図1－6に示す。

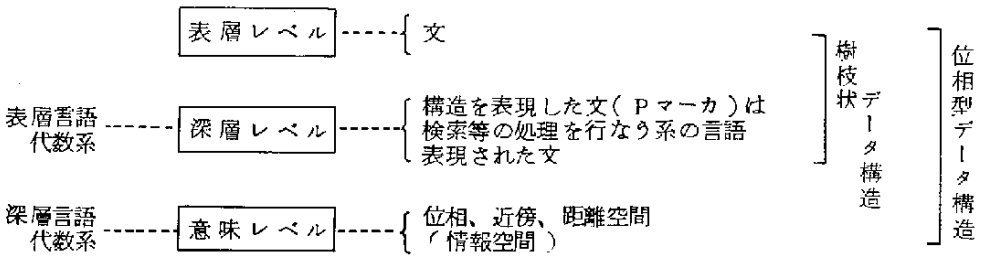


図1－6　言語代数系と樹枝状、位相型データ構造

表層レベルと深層レベルの間の翻訳は、変換器とよぶもので行なえ、これはPush Down Transducer あるいは、Stack Transducerなどでその数学的な性質が論じられる。

深層レベルと意味レベルの間の翻訳は、深層レベルの各単語、単語の組合せ法則などを情報

空間内の領域、領域の合成法とが処理法への写像を行なうことによりなされる。

情報空間とその上での処理算法で構成される代数系を深層代数系とよぶ。

これに対し表層、深層レベルで、語、文とその上での論理代数的、照合的処理をあわせて、表層言語代数系とよぶ。

本研究は、遠隔情報処理システムの構成が目的なので、表層、深層両言語代数系についての数学的な性質については省略するが、その特徴等については、各システムの所で概説する。

4. 位相型データ構造（情報空間）の作成

位相型の場合には、情報空間（意味空間）とよばれるものが、データの構造を表現し、樹枝状の場合のシソーラスに対応する。

シソーラスは、関係の有無で記述する線的なものであるに対し、情報空間は意味の位相関係が総て表現された面的あるいは立体的なものである。

ことばの表わす意味の位相的関係は次のような方法により抽出出来る。

(1) 意味差分法（ＳＤ法）

(2) 非計量要因解析法

(3) 類似度解析法

(4) 各種体系分類、階層分類、類語集などから統計的手法により求める方法

ここでは(3)の方法を用い、情報空間内の元（ことばの表わす概念）の間の距離は、それらの間の類似度と同じ順序関係にあるように、しかも出来るだけ低次元内に歪なしに収める方法を、パラメータを変えたり、式の修正を行ない、意味空間、情報空間の構成を行なった。その構成アルゴリズムを図Ⅰ－7に示す。構成に要した時間は、ＮＥＡＣ２２００－５００で次の表Ⅰ－2、表Ⅰ－3の通りである。

これから判断すると、歪をある許容範囲内におさえた時に収束する次元数元の、データ数に対する割合は、データ数の増加と共に減少する傾向にあり、増加傾向はデータの組合せ数に比例する指数関数的ではなく、むしろ対数関数的であり、データ数がある程度以上になると、増加の割合は

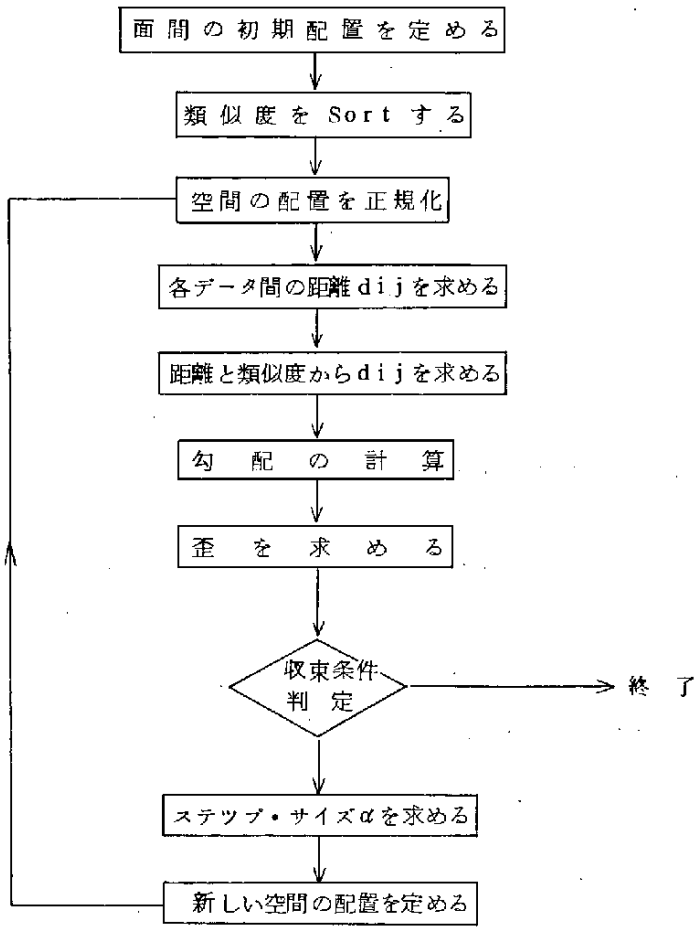


図1－7　意味空間構成アルゴリズム

表1－2　データ数による計算時間

データ数	10	20	30
計算時間	1.2秒	6.6秒	28.2秒

表1－3　データ30個の時の計算時間

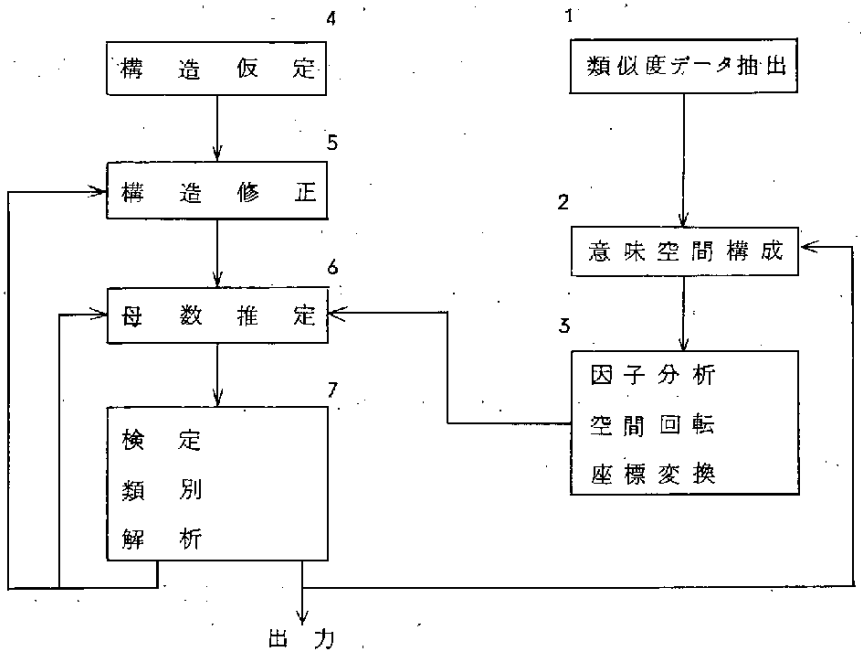
次元数	27	24	21	18	15	12	9
計算時間	58	35	31	28	26	20	15

非常に緩やかになるものと思われる。

この空間構成法を用いるとデータが単語や句のときは意味空間を、データが論文の時には、論文概念空間を構成する。

この情報空間の上での処理法を求める為に連語を一例として選び、単語を組合せて連語を作るのは、意味空間では対応する概念領域をどのように組合せて、合成概念領域を求めればよいのかを知る為に、単語、連語を含む意味空間を構成し、その算法を統計的に抽出する実験を行なつた。

そのアルゴリズムを図Ⅰ－8に示す。



図Ⅰ－8　意味空間に於ける算法抽出の手順

その結果、連語という演算は、情報空間中で

$$X_3 = A X_1 + B X_2$$

$$= B (\lambda X_1 + X_2)$$

の形の構造をもつということが推論出来た。これは、具体的には、連語は連語を構成する二つの単語の概念領域の内分点にその合成概念領域がくることを示す。

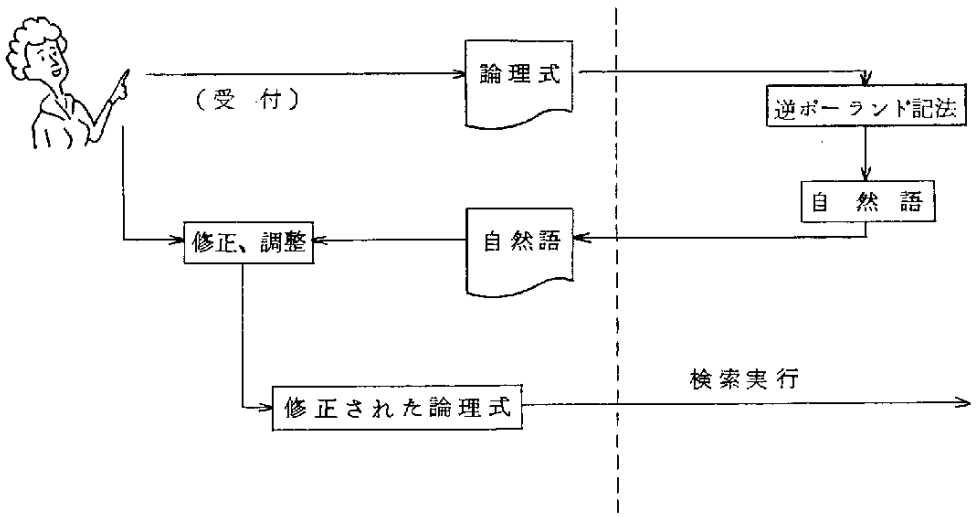
このような算法がすべて解明されたなら、文を構文分析し、それぞれの単語の概念領域を文法で指定される情報空間の算法で合成してゆくことにより意味解釈が行なわれることになる。

5. 樹枝状データ構造表現の自然語事項検索システムへの応用例

自然語による会話応答システムにおいては、質問分析というか、質問を検索系の言葉で表現する方法が、システムの効率に大きく効いてくる。そこで質問分析の二方法について、比較検討を行なつた。

一方は、樹枝状データ構造を順次下層にたどつてゆき、必要な情報を求めようとするもので、上層から、各階層の分岐枝を総て質問者に列挙して、必要な枝（該当カテゴリ）を選択してもらい、次々と下層へたどつて、最終的に求めるデータを得ようとするものである。この方法は、情報ファイルそのものの知識さえあれば、形式言語により情報ファイルの部分を抽出する方法を知らなくても質問分析が行なえるが、システムの占有時間が指数関数的に増大し、また完全な体系分類は不可能な為に、分類、アクセスの出来ないものが存在する。

もう一つの方法は、このように樹枝を直接たどるものではないが、質問を形式言語で記述し、それを検索システムの言語（逆ポーランド記法など）に翻訳し、それを再び逆変換し、自然語に近い形式言語になおし、質問者にフイードバツクし、質問意図とあつているかどうかチエツクし、あつていないときは、修正、調整を行ない、依頼質問そのものへ収束させ検索を行なわせようとするものである。その手順を図Ⅰ－9に示す。



図Ⅰ－9

この方法は、形式言語の文法を熟知したものでないと使いこなせず、一般むきの会話用言語とはいいにくい。

結局質問の表現および分析法としては、出来るだけ自然語に近い形式言語をとり、その言語で許される適当な文型を選び、文中の変数項に必要な語句を挿入し、質問文を作成するいわゆるL一言語の形式を用い、対話形式で検索目的を明確にしてゆく方法がよい。

よつて対話の一形式として、形式言語で記述した質問を、Syntax-Oriented Transducer で、検索系の逆ポーランド論理表現に翻訳し、再びそれを自然語に翻訳しなおし、質問者に照会し、質問の修正、検索語の調整を行なつた後に検索を行なう、オープン情報検索システムを考察した。

翻訳プログラムは、データ記述項目を、否定、論理和、論理積、包含関係を用いて合成した論理式を、逆ポーランド表記に翻訳し、また逆ポーランド表記された論理式を日本語に翻訳を行なう（上記結合子を、オヨビ、マタワ、等となおし、デス等をつける）手順を調べ、フローチャートに示した。

日本語は逆ポーランド表記と、その語順が似ている為容易に翻訳が行なえる。

質問の分析が行なえると、検索が問題となる。そこで、自然語による事項検索システムを、二種構成した。

事項検索システムが備えるべき条件としては次のものがあげられる。

(1) 特定目的の為の人工言語ではなく、自然語での質問受付が出来なければならない。

(2) 入力文にある内容に対し論理的に演繹が可能でなければならない。

これは、自然語で書かれた情報を入力する度に内部表現に翻訳されると共に、代入の法則などの演繹が可能でなければならないことを意味している。

ところが自然語を使用しようとすれば、次のような問題点を克服しなければならない。

(1) 自然語の経年変化

(2) 自然語の値域が広すぎる

(3) 自然語のあいまいさ

(4) 自然語で書かれた内容の正否を判断する内なる判断基準が存在しない

(5) 自然語は理論的に明確な構造をしてない

これらの問題点は、対象を限定することにより実用的な観点から解決を行なわねばならない。

以上の事項を考慮して自然語による事項検索システムを二種構成した。これらは情報の演繹も行なえるシステムである。

その一つは、W.S.Cooper の型の事項検索システムで、この方式は演繹の方向が定まつておらず、すべての組合せをチエツクする方式である。質問文から次のようなアルゴリズムに基ずき、論理的関係を検出し答を出す。

(1) 質問文の論理的帰結になる文章を、格納文章集合中から、ルツクアツプにより検索する。もしみつかれば"ＴＲＵＥ"と打出す。

(2) (1)で失敗した時は、質問文と論理的に矛盾する文章を検索する。みつかれば、"FALSE"と印字する。

(3) (2)も失敗した時は、"ＵＮＡＢＬＥ　ＴＯ　ＡＮＳＷＥＲ"と印字する。

データは、アリストテレスの四つの定言的判断の形で表現され、質問文はこの上へ翻訳され、この上で比較、真偽判定を行なう。

もう一つの事項検索システムは、演繹の方向づけを与えたもので、道路網に関する検索システムで、入力はＣＦＧにより生成される文を使用する。

検索のタイプとしては次のものが取扱える。

(1) ある地点から、ある地点迄の道路網

(2) (1)において経由点を指定したもの

(3) (1)に最適であるとの付加条件を与えたもの

路線の検索としては、各道路の通行のしやすさを〔０，１〕区間の数値で表現した道路行列演算をWarshallの方法で行なう。

以上二つの事項検索システムを具体例として構成した。

6. 位相型データ構造表現の文献情報検索システムへの応用例

遠隔制御情報処理システムにおいては、意味解釈ならびにそれにともなう情報検索が非常に重

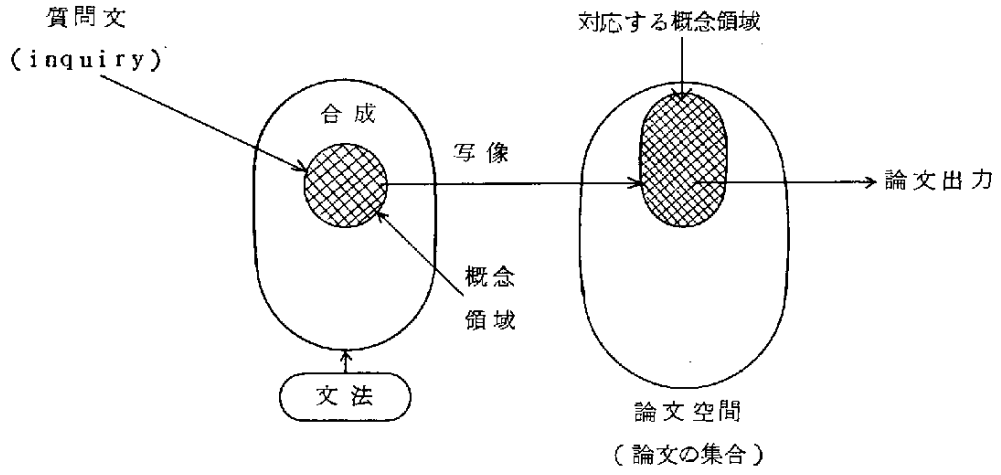
要な役割をもつ。このために、われわれはその処理を行なうシステムを実際に構成し、これらの諸事項がどの程度満足に行なわれるかの実験を行なつた。

システムの持つファイルは、一次情報ファイルと、二次情報ファイルであるが、位相型データ構造の場合は、論文概念空間ファイルを用いて検索する。樹枝状データ構造の場合は、二次情報ファイルの分類を次々と下位にたどつていつて検索する。

実験は位相型データ構造の場合について行なったので、その場合について述べる。

検索方法としては、(i)文章型シソーラスを用いる方法と、(ii)標準点を用いる方法がある。

文章型シソーラスを用いる方法は、文を構文分析し、意味空間の各単語の概念領域を文法で指定される方法で合成してゆき、質問の概念を論文空間中に写像し、それに近い概念領域をもつ論文を、該当論文として検索しようとする方法である。(図Ⅰ－１０)

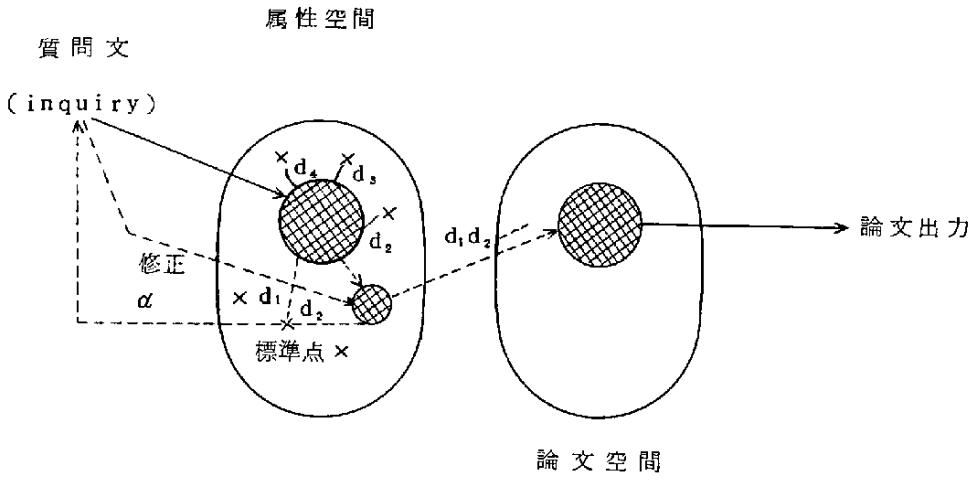


図Ⅰ－１０　文章型シソーラスを用いる検索法

標準点を用いる方法は、質問文を属性空間に写像し、属性空間に予め定めておいた標準点との類似度を求め、それを質問者にフィードバックし、目的のものに近くなるように修正を行ないながら、質問概念の類似度を定め、それを論文概念空間へ写像し、近い論文を検索する。

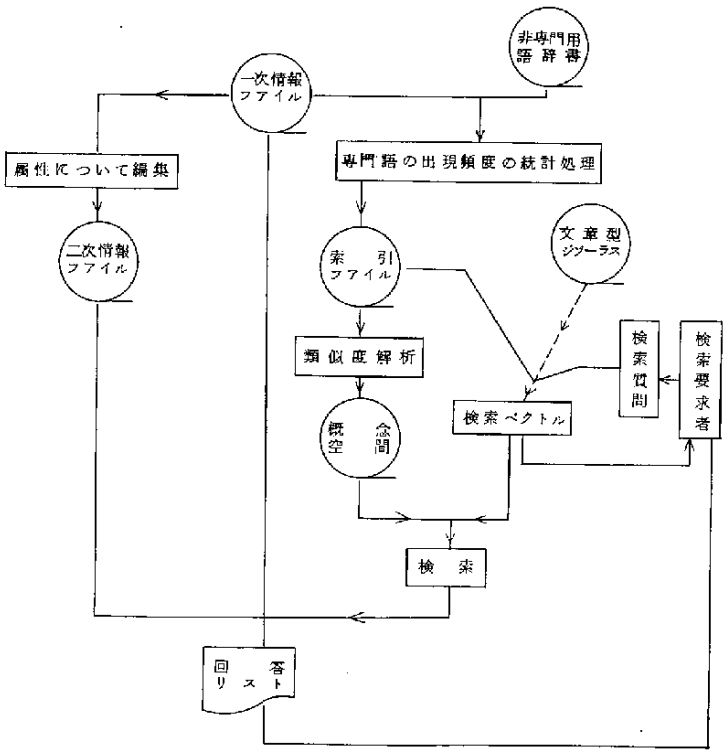
(図Ⅰ－１１)

論文概念は超深面上に分布しているので、質問ベクトルと垂直な超平面を適当な位置にシフトして行なう。実際には質問点と各論文点との内積の大きさが、質問概念への近さを表わす。

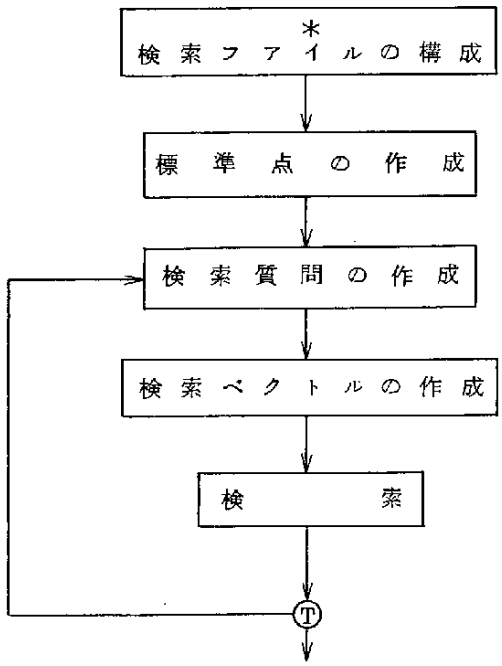


図Ⅰ－１１　標準点を用いる方法

その検索システム・フローを図Ⅰ－１２に示す。また検索アルゴリズムを図Ⅰ－１３に示す。



図Ⅰ－１２　検索システム・フロー



図Ⅰ－１３　検索アルゴリズム

実験は標準点を用いる方法で行なつたが、標準点の選び方としては、次のようなものがある。

(1)　人間が与えられた情報をもとに代表元となるものを選択する。

(2)　論文概念空間において、丁度一様分布する論文どおしを選出し標準点とする。

(3)　論文の類似度をもとに、分布の分散の少ないものを選び出し標準点とする。

実験では検索時間の関係で(3)の方法で標準点を選出した。この標準点の選択は、検索能率に大きく影響してくる。

電子通信学会誌論文（１９６８年度の一部）米国計算機学会誌ＣＡＣＭ（１９５９年度）掲載論文などを対象として行なった検索実験の検索効率を、各パラメータと共に表Ⅰ－４に示す。

また検索に要した時間をパラメータと共に表Ⅰ－５に示す。

これから次のようなことがいえる。

(1)　論文概念空間の歪よりは、標準点の選び方のほうが検索能率には大きく効いてくる。

(2)　標準点の個数は、論文概念空間の次元数以上、論文概念空間の歪は７％以下なら効率の良い検索が期待できる。

表Ⅰ－4　検　索　効　率

データ数	次元数	標準点の数	歪	再現率 α	適合率 β
20	8	8	6.7%	100%	90.9%
20	6	8	19.1	0	0
20	6	15	19.1	100	8.0
30	18	20	2.5	100	100
50	35	15	3.0	78.9	93.7

表Ⅰ－5　検　索　時　間

データ数	20	30	50
検索時間	1.2秒	4.6	8

(3)　検索質問を論文概念領域へ写像する方法が検索能率に一番大きく効いてくる。

(4)　検索時間も、この程度ならリアルタイム処理にも、あまり待たせないですみそうである。

ファイルの維持更新は、追加の場合は、追加論文を質問点と同じ取扱いで論文概念空間へ写像し、その点へファイルすればよく、削除はただ単にその点を除くだけでよい。追加、削除により空間の歪は増すと考えられるが、その追加削除数が全論文数に比して十分小さいときは、標準点の選び方でそれを補うことができる。従つて大量の追加、削除がないなら、論文概念空間の再構成は半年か一年に一度程度で十分である。

結局、歪が7％程度の出来るだけ低次元の空間を選び、標準点の数を論文概念空間の次元数より十分大きくとれば、検索効率、検索時間、空間構成時間および記憶容量の点からいつて有効な検索システムが構成されることがわかつた。

Ⅱ (樹枝状型言語による応答システムの設計)

Fact Retrieval Systemの構成理論

1 緒　　論

本編は、Fact Retrieval Systemと呼ばれる情報検索システムの一範疇について、その問題点のうちいくつかの問題をとり上げ、それらについて論じたものである。

Fact Retrieval Systemに限らず、情報検索システムの構成に際しては、何を対象とし、どのようなサービスをするかということをまず考慮しなければならないことは当然である。対象の設定が終るとこれに拘束されたサービスの内容もある程度規定されてしまうことが多いが、しかし、情報要求の形態等には大きな自由度が残されている。情報要求、つまり検索質問は情報要求者の質問概念が容易に表現できる事が必要である。このためには、我々が日常使用する自然語でもって質問を記述することを許容しようではないかとの考えが、でて来るのは当然である。しかし、4で詳しく述べるように自然語を完全に処理することは不可能であり、また、自然語の取扱い方法にも大きな問題が存在することも事実である。本章の目的の一つはこのような問題点を浮きぼりにすると共にその解決の方向を示そうとするものである。2では、情報検索システムにおける質問形式について述べ、質問形式はいかにあるべきであるかについて論じた。3は、遠隔情報処理システム等にみられる対話型の情報検索システムの質問分析に関連する質問受付の形式について、日本語の特性を利用した逆 syntax-oriented transducer と呼ばれるものについて述べた。4は、情報の演繹のできる Fact Retrieval System の例として、W.S.Cooper のシステムをとり上げ、これについて、Fact Retrieval Systemの規定の仕方、およびその構成の具体例についてその哲学等について述べた。5は、情報の演繹が可能であり、しかも情報の演繹の方向づけの与えられたシステムの一例として、PR(Path Retrieval) System と呼ばれる道路網を対象とする情報検索システムについて述べた。6は、この章に対する結論である。

2 情報検索システムにおける質問形式の分類

情報検索システムの端末において、質問者が質問（情報要求）を持って来た場合、これを明確

化し、抽出する過程には以下に述べるような本質的に異なる二つの過程が考えられる。すなわち、

(1) 最初のタイプは、質問者(情報要求者)がシステムで定められた質問表現用の形式言語で自己の質問を記述し、これを窓口に提示するやり方である。

(2) また、別のタイプとして、システムがシステム内の情報を階層毎に幾つかの分野に分類しておき、質問者には階層毎に二つ以上の分野を提示することにより、この内から質問者の要求に合致するものを選択せしめ、この提示を高い階層から低い階層へと繰返すことにより、質問者の要求を明確にしてゆくというものである。この二つの質問分析のやり方には、大方の予想の通り一長一短があることは当然であり、質問者に要求される能力、システムの作り易さ、あるいは、システムの占有時間等の問題がからみ、どちらを選ぶかは総合的に判断されるべき性質のものであると考えられる。

(1)のタイプの形式言語によって質問を記述するシステムは、使用する語彙、使用する質問表現用の文の文形等について一つの約束を行なうものである。例えば、Bacchus の Normal Form と呼ばれる記法に従って質問文を書くというのがそれである。Bacchus のNormal Form とはつぎのようなものである。ここでは、ALGOLで用いられる単純論理式をブール代数の規則に従って生成する例について述べたものである。

<単純論理式> ::= <包含関係式> | <単純論理式> ≡ <包含関係式>

<包含関係式> ::= <論理項> | <包含関係式> ⊃ <論理項>

<論理項> ::= <論理要素> | <論理項> ∨ <論理要素>

<論理要素> ::= <論理素要素> | <論理要素> ∧ <論理素要素>

<論理素要素> ::= <論理素子> | ¬ <論理素子>

<論理素子> ::= <論理値> | <変数> | <函数記号> | <関係> | (<論理式>)

<関係> ::= <単純計算式> <関係演算記号> <単純計算式>

<関係演算記号> ::= < | ≦ | = | ≧ | > | ≠

このタイプの質問のやり方を図示すると図Ⅱ－1のようになる。質問者は、上述のような Bacchus の Normal Form等で定義された形式言語で情報空間の一点、または部分集合を記述する必要があり、かなりの熟練が必要とされる。例えば、上述のALGOLで処理可能な単純論理式を書くためにはこの生成ルールをよく理解していなければならないし、これにはずれる

ものはungrammaticalであるとして拒否されてシステムは受け付けて呉れない。したがってこのような形式言語を利用する考えを徹底的に押し進めたシステムは全くの素人（たとえ情報空間そのものには知識を有していても）にも利用できるものにはなり得ないことは当然である。

(2)のタイプのシステムは、情報空間を互いに共通集合を持たないいくつかの部分集合に分割し、

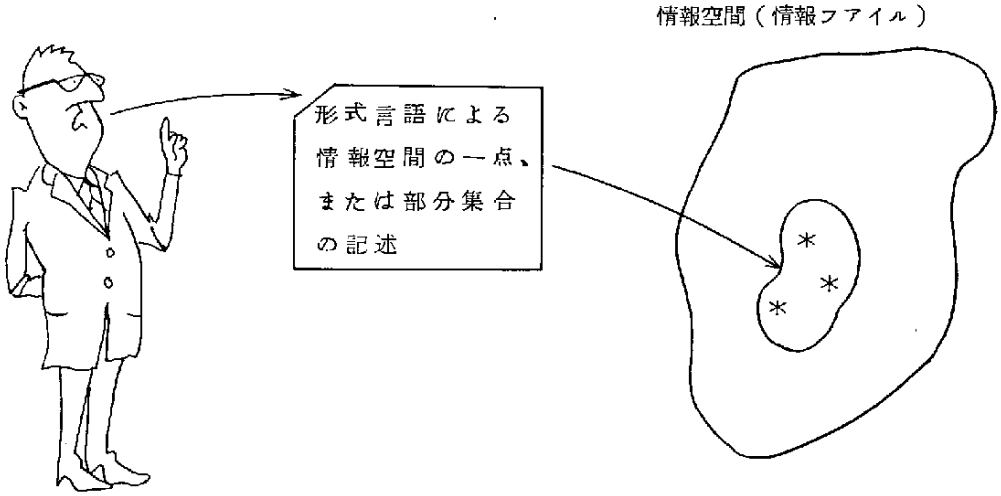


図　Ⅱ－1

この部分集合を順次質問者に提示し、質問者の質問概念の含まれる部分集合を選択せしめ、一つの部分集合が得られたら、更にこの部分集合を細分割したものについて選択を行なわせしめ、目的の部分集合を指定するというやり方である。この方法は図Ⅱ－2に図示される通りである。

したがって、この方法は、情報空間そのものにのみ知識があれば、形式言語等によって部分集合を記述するといった手間はなく、システムが出す質問に例えばYes あるいはNo の答えをすればよいので一般向けである。しかし、システムの占有時間が指数関数的に増大するという点の他に、つぎのような本質的ともいえる欠陥を有する。すなわち、情報空間の分割の仕方がこの方法によるシステムの死活を決めるという点である。境界の明確でない情報は、扱い切れない訳である。例えば色の問題がこれに相当すると考えられる。色の集合は、本来fuzzy集合的であり質問者のイメージに描く色の概念とシステムに予め用意してある概念とはかなりの程度に一致することはあっても全く一致するとは考えられない。したがっていずれの選択過程でも選択の仕様がなく、正しい答えが得られないことがある。

これらの事実を綜合すると、情報検索システムの質問形式は以下のような形が望ましいと考えられる。

(1) 形式言語の能力を極限まで高めた形としての自然語の使用を許す。

(2) 自然語、あるいは自然語に近い形式言語を用いて質問を表現し易くした用紙の使用。これは、質問分析の作業の大部分を予め質問者に行なわしめることによっていままでの欠点を補

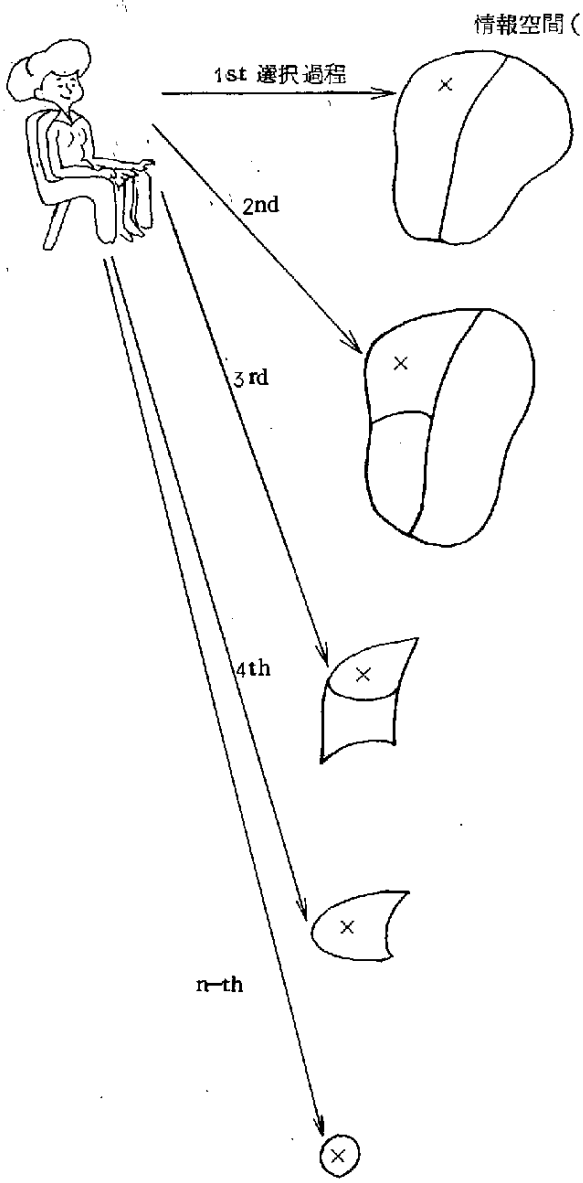


図　Ⅱ－2

おうとするものである。このシステムでは、質問者は適当な文を選び、文中の空白の所に必要な語句を記入する、いわゆるL－languageの形式をとることになるであろう。

(3) 対話形式の積極的利用、特にこれについては日本語による例を挙げて3で詳しく述べる予定である。

この節では、質問の形式について考察を行なうとともに、質問はどのように行なわれるべきであるかについて検討を行なった。

3　情報検索システムの質問分析に関連する質問受付けの一形式について

3．1　はじめに

本節では質問形式を対話型にし、これを更に積極的に押し進めた形での質問の受付けと質問分析のやり方について日本語に例をとり述べることにする。質問分析とは、質問者の持つ質問概念をいわゆる索引言語の上で表現するということである。通常このためには、前節で述べた質問形式のいずれかを通して論理表現等に持ち込もうとする立場が取られる。形式言語による質問の記述を行なうときには、syntax－oriented translatorと呼ばれるものによってこの作業が遂行される。本節ではとにかくこのようにして得られた論理表現を直ちに検索に使用するのではなく、対話の一形式として従来質問者が知り得なかったこの論理表現を再び自然語（日本語）に直して質問者に照会し、質問そのものの修正あるいは索引語の調整を行なった後検索に移るという一種のOpen　情報検索システムにおける初期質問リストを自然語に（日本語に）翻訳するシステムを対象としている。上述のようにこのシステムの目的は、対話モードを積極的にとり入れることにより、情報検索システムの検索効率を最大限に上げようとするものである。

3．2　システムの概要

3．1で述べた概念に基づくシステムの全体は、図Ⅱ－3に示される。質問者が発した質問がいくつかの変換過程を経て論理表現された場合、システムの解釈が質問者の質問概念と異なる事が応々にしてある。このような場合従来のシステムではシステムの解釈を是とすることにより検

索が行なわれてしまったが、図Ⅱ－３の如きシステムでは、質問者に日本語に逆翻訳された質問文を照介することによりこの欠点は除かれる。以下の節では、ブロック図中に示されている論理式を逆ポーランド表記すること、また逆ポーランド表記された論理式を日本語に翻訳すること等を中心に議論を進めることにする。

３．３　論理表現より逆ポーランド記法への変換

３．３．１　は　じ　め　に

日本語の語順は、逆ポーランド記法と極めてよく一致することが知られている。[ ] 逆ポーランド記法で表記されている式は、その記号の配列順を少しも崩さず日本語に変換することが可能である。本節ではまず論理式の構成規則について述べ、つぎに質問者の質問概念を記述した論理式をまず逆ポーランド記法に変換する手法について述べる事にする。

３．３．２　論理式の構成

本節では論理表現中で用いる論理式の構成について定義を行なう。但し、定義式中ξ、ηは被演算項であり、これが通常は索引語に相当するものである。他の記号は慣用的用法に従うものであり特に問題はないと考えられる。

定義式は、Bacchus の Normal Formにしたがって書くことにする。

(1)　下記の手続きのうち少なくとも一つによって作られた表現は、well−formed clause（作りのよい句）である。

$$\gamma ::= \neg\xi \mid \xi \cup \eta \mid \xi \cap \eta$$

$$\Gamma ::= \neg(\gamma) \mid (\gamma) \cup \xi \mid \xi \cup (\gamma) \mid (\gamma) \cap \xi \mid \xi \cap (\gamma) \mid (\gamma) \cup (\gamma) \mid (\gamma) \cap (\gamma)$$

$$\Sigma ::= \xi \supset \eta \mid \Gamma \supset \xi \mid \xi \supset \Gamma \mid \Gamma \supset \Gamma$$

(2)　well−formed clause に「．」を後置した表現は、well−formed sentence（作りのよい文）である。

(1)、(2)の二つの手続きによって論理表現中で使用する論理式が定義できた訳である。図Ⅱ－３中の論理式は必ずこの手続きによって生成されていなければならないし、これ以外のものは拒否されることになる。

3 3 3 逆ポーランド記法について

ポーランドの数学者ヤン・ルカシエーヴイチによって考案された表記法である。通常の記法で論理式、例えばa U bのように被演算項a、bのあいだに演算記号を入れて書くところをa b Uのように被演算項のすぐ後に演算記号を書く方法である。一般にn個の被演算項a_1 a_2 …… a_n にτというn項演算を施すとき、a_1 a_2 …… a_n τ のように被演算項の直後に直接τをつける。τを被演算項に後置するときに特に逆ポーランド記法と呼ぶ。この方法で書くと、通常の記法で書いたときにはカッコを必要とする場合にもカッコを用いないで書くことができる。（parenthesic-free notation）例えば、

a U (b∩c)

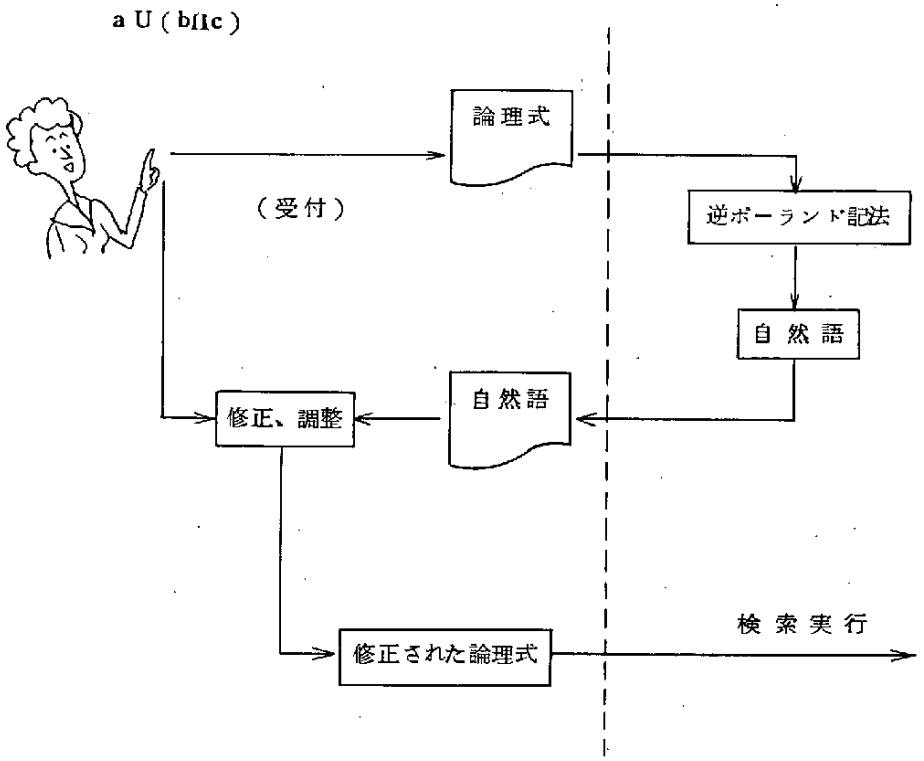


図 Ⅱ－3

を逆ポーランド記法で表記すると、

a b c∩U

と表わすことができる。

カッコが2個以上ある場合でも、上のようにしてすべてのカッコを除去することができ、しかも、カッコの使用で演算順序が示されていた式が、カッコをなくしても、もとの順序通り演算がなされる。式の演算に際しては式の記号を左から右へ一字づつ眺めてゆき、演算記号に出会ったとき、この演算記号がn項演算子であればその前のn項の被演算項についてその演算を行なうことにすればよいわけである。

3．3．4　Pushdown Stack

本節では論理式を逆ポーランド表記されたものへ変換する過程において必要とされるPushdown Stackについて簡単に述べる。この装置の実現は、hardware として行なわれることもあるが普通はプログラムを工夫することによって等価なものを得ている。すなわち、記憶装置のある部分を指定してその場所へつぎつぎと記号を格納してゆくが、それらの記号をとり出すときには最後に入れた記号から順にとり出してこの順を飛びこして先に格納した記号をとり出すことができないよう仕組むことによって実現される。現在使用されている記憶の一部分にこのようなPushdown Stackをつくるには、図Ⅱ－4のように、一次元の配列（たとえばこれをスタックと命名することにすれば）とその配列の成分の位置を数え上げる命名記号(k)を設定すればよい。そしてこのkに最初1を入れた状態でこ

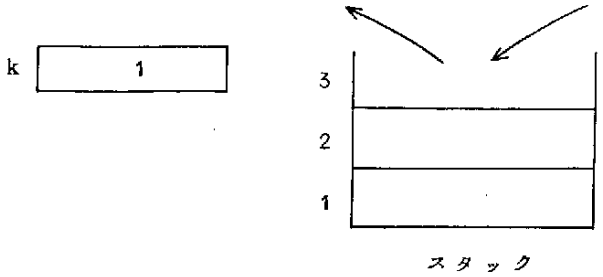


図　Ⅱ－4

れを使用すると、

スタック(k) = v

の形で最初の位置にvが入る。そしてその後、

k = k + 1

とすれば、この場合にはkの値は2となり、つぎのスタック(k) =…では2番目の位置に格納される。

この2つの作用、すなわち、

スタック(k) = v

k = k + 1

をまとめてこの配列に記号を格納するときに使用することにすれば、kの値はつねにそれまでに値が入っている場所のすぐつぎの空いた番地を示すことになる。

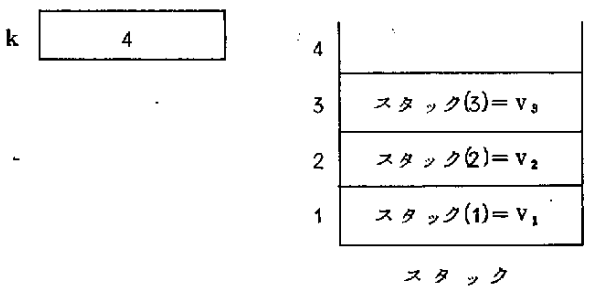


図　Ⅱ－5

例として、図Ⅱ－5はこのようにしてスタックに3記号入った後の状態を示す。この操作をスタックの中に1記号ずつおし入れる(pushdown)という。図Ⅱ－5でこのスタックから最後に入った記号を引き出すには、

k = k − 1

とすればよい。この場合にはk＝3である。

図Ⅱ－6においてつぎにでるのは、スタック(2)の位置にあるv_2であり、今度入れられる位置はスタック(3)であり、現在スタック(3)の位置に何が入っていようともそれは取り出されたのと同じ意味をもつ。したがって、k = k − 1を行なう前に、例えば、

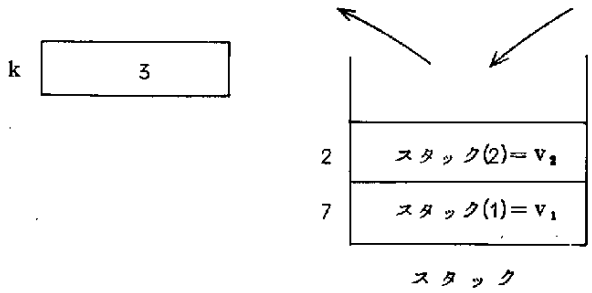


図　Ⅱ－6

u＝スタック(k)

なるスタックの外の記憶場所uにスタック(k)の内容を転移させ、その後にk＝k－1を実行すれば、スタックの中の最近の情報が1つのuの中に入って利用でき、かつスタックの今度利用できる位置が1つ増加したことになる。このような操作をスタックの中から1記号引き出す（pop up）という。

Pushdown transducer を用いる論理式の逆ポーランド記法への変換原理を図Ⅱ－7、そのフローチャートを図Ⅱ－8(a)(b)(c)(d)に示す。

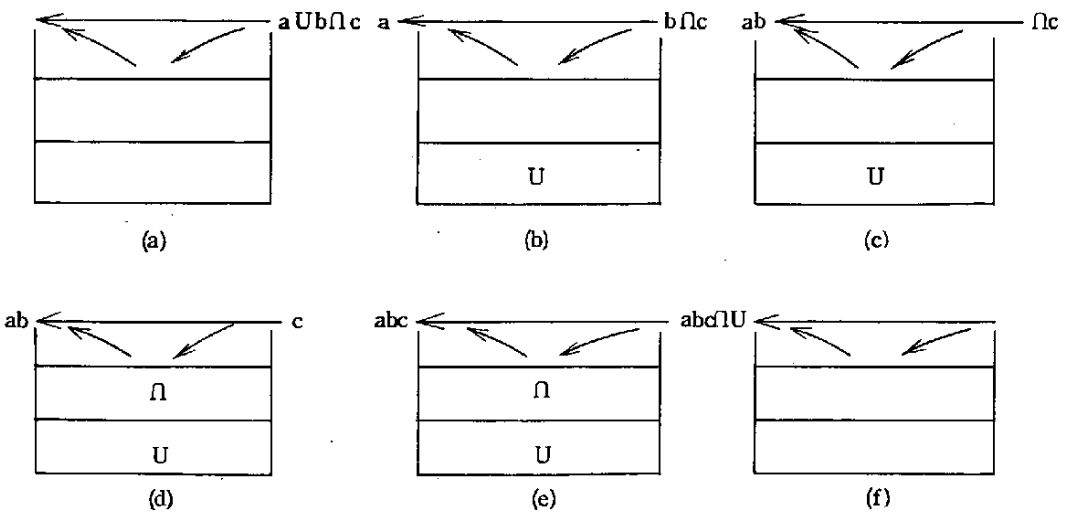


図　Ⅱ－7

3．3．5　流　れ　図　表

論理式をポーランド表記されたものに変換するための流れ図を図Ⅱ－8 (a)(b)(c)(d)に示す。

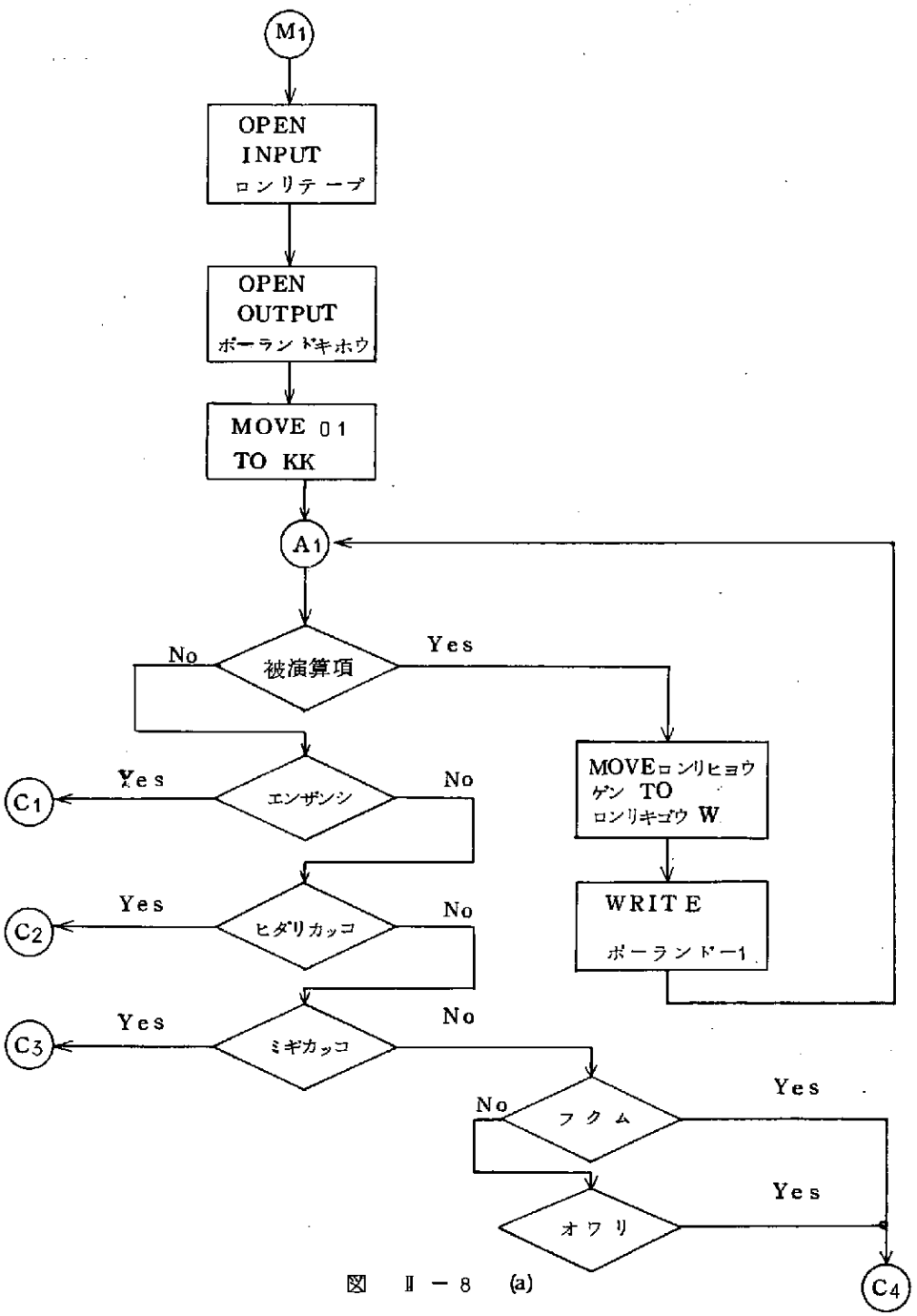


図　Ⅱ－8　(a)

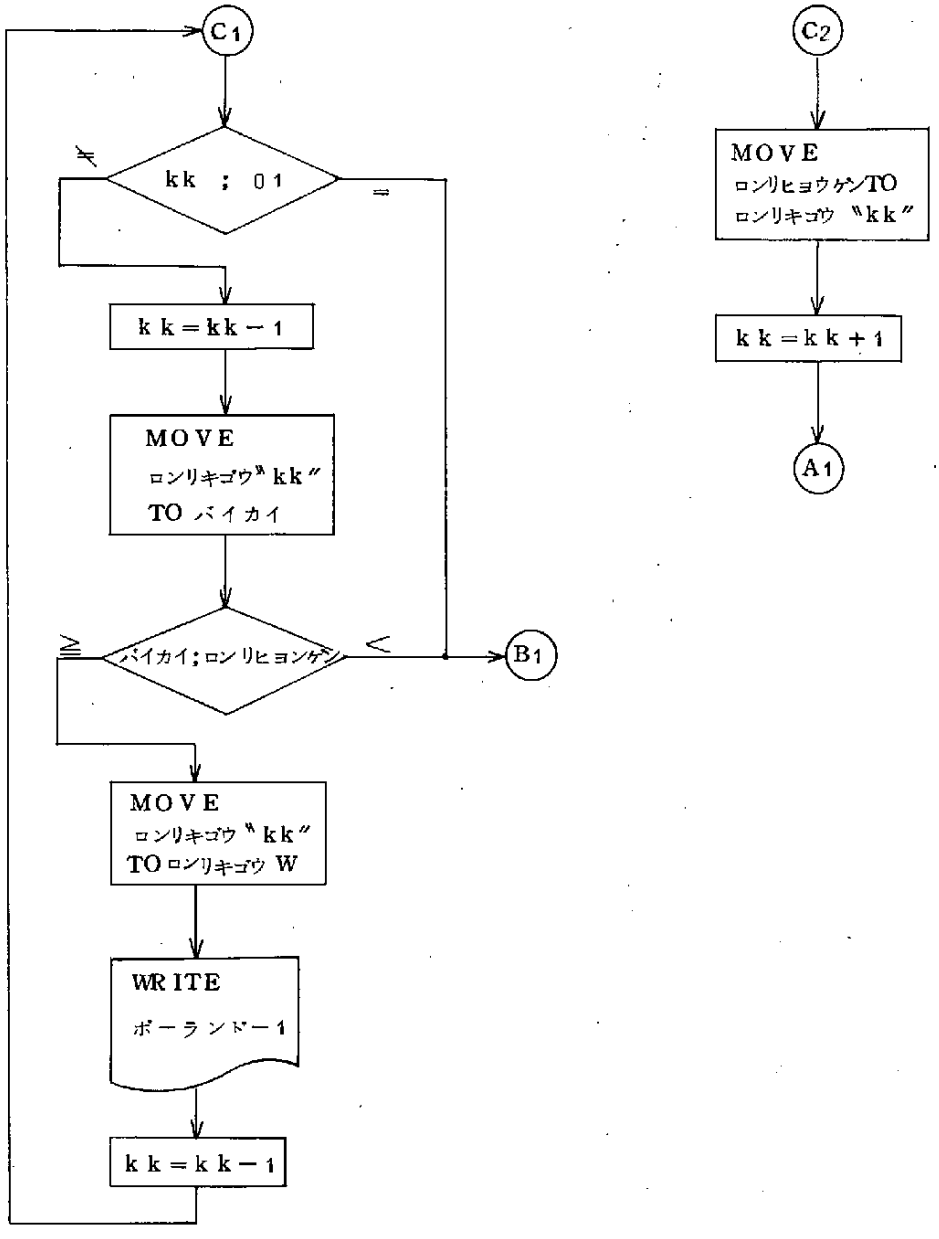

C1
kk ; 01
＜
＝
kk＝kk－1
MOVE
ロンリキゴウ〝kk〟
TO バイカイ
≧
バイカイ；ロンリヒョウゲン
＜
B1
MOVE
ロンリキゴウ〝kk〟
TO ロンリキゴウ W
WRITE
ボーランド－1
kk＝kk－1
C2
MOVE
ロンリヒョウゲンTO
ロンリキゴウ 〝kk〟
kk＝kk＋1
A1

図 Ⅱ－8 (b)

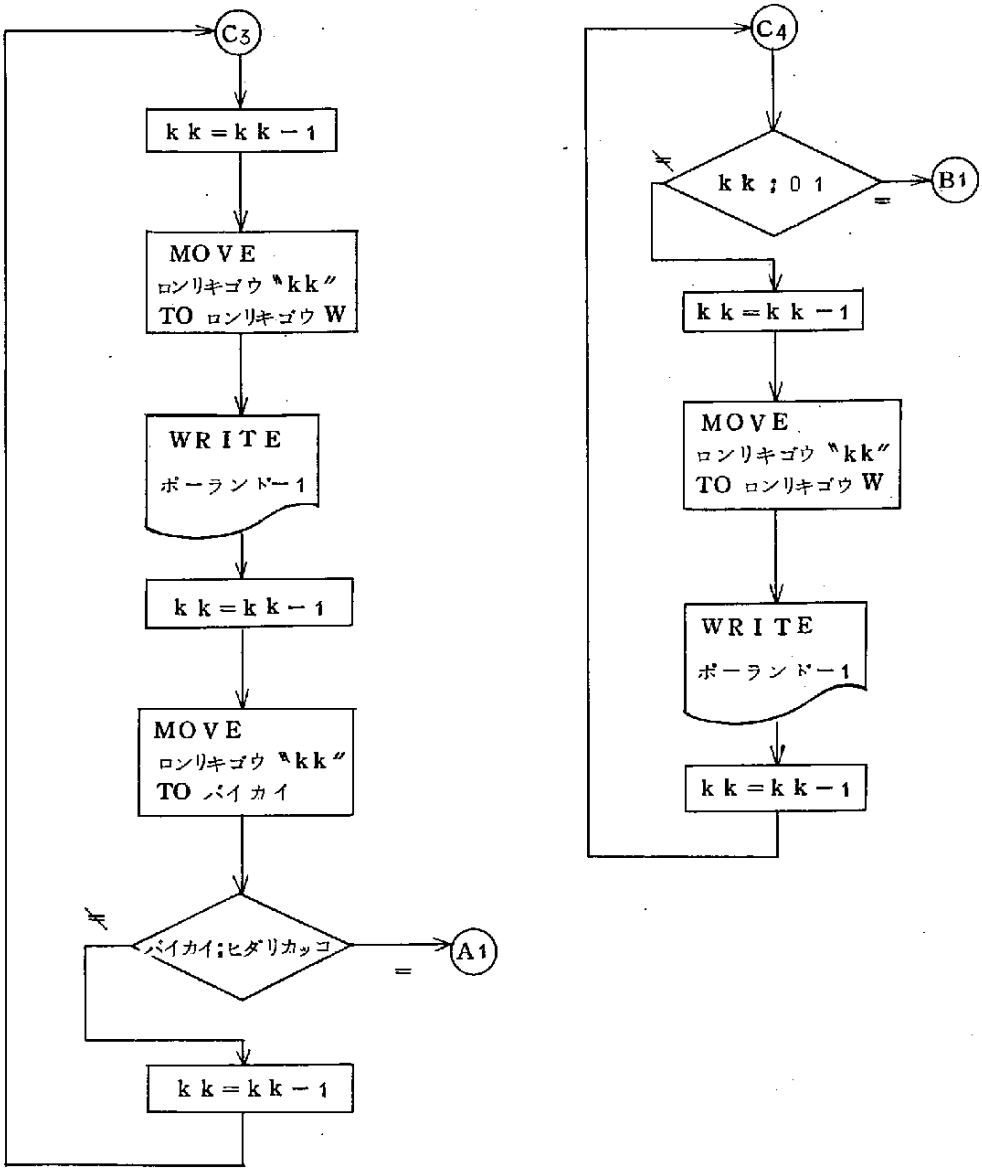

C3
k k＝k k－1
MOVE
ロンリキゴウ〝kk〟
TO ロンリキゴウ W
WRITE
ポーランド－1
k k＝k k－1
MOVE
ロンリキゴウ〝kk〟
TO バイカイ
バイカイ;ヒダリカッコ
A1
k k＝k k－1
C4
k k ; 0 1
B1
k k＝k k－1
MOVE
ロンリキゴウ〝kk〟
TO ロンリキゴウ W
WRITE
ポーランド－1
k k＝k k－1

図　Ⅱ－8　(c)

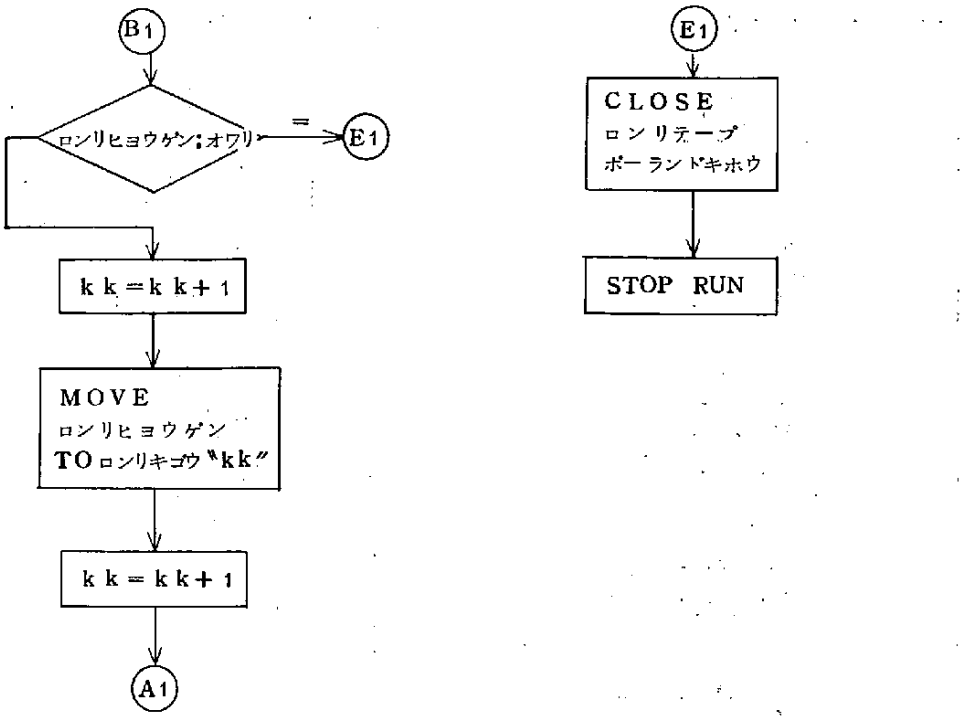


図　Ⅱ－8　(d)

3．4　逆ポーランド記法より自然語への変換

3．4．1　逆ポーランド記法の文法と日本語

逆ポーランド記法では括弧を削っただけ語彙の量は小さくなったが、表現力は通常の論理表記されたものとまったく変るところがなく、演算優先順位を指定する必要はない。したがって逆ポーランド記法では、被演算項、単項演算子(¬)、二項演算子(U,Ω)、関係述語項(⊐)、そして、エンドマーク(．)の5種類の記号が使われるのみである。

逆ポーランド記法で書かれた論理式は、context-free な句構造文法で生成されることができる。この文法をつぎに示す、すなわち、

N::＝ ξ | η |……

K::＝ N | P

P::= K¬|KKU|KK∩

C::= P|KK⊐

S::= C .

S::= (P|KK⊐).

これらの文法より結局生成される、well-formed P あるいはSは、つぎのような形をとる、すなわち、

P::=ξ¬|P¬|ξP¬|ξPU|ξP∩|PP∩|PPU

S::=P.|ξη⊐.|ξP⊐.|Pξ⊐.|PP⊐.

逆ポーランド記法で表記された論理式を日本語に直すためには、演算子にその意味を良く表わす日本語を与えて、論理式を自然語表現すればよいわけである。いま演算子にそれぞれつぎのような日本語を割り当てることにする、すなわち、

¬　　ではない（デワナイ）

∩　　の（）

U　　または（マタワ）

⊐　　含む（フクム）

これらの語に更に複雑な言葉、例えば、デワナイに対して〝コノ単語ヲ含マナイ〟としても、システムとしては何ら変更する必要はないし、適宜選択すれば別に問題はない。

3.4.2 流れ図表

これらの作業を行なう為の流れ図を図Ⅱ−9 (a)(b)に示す。

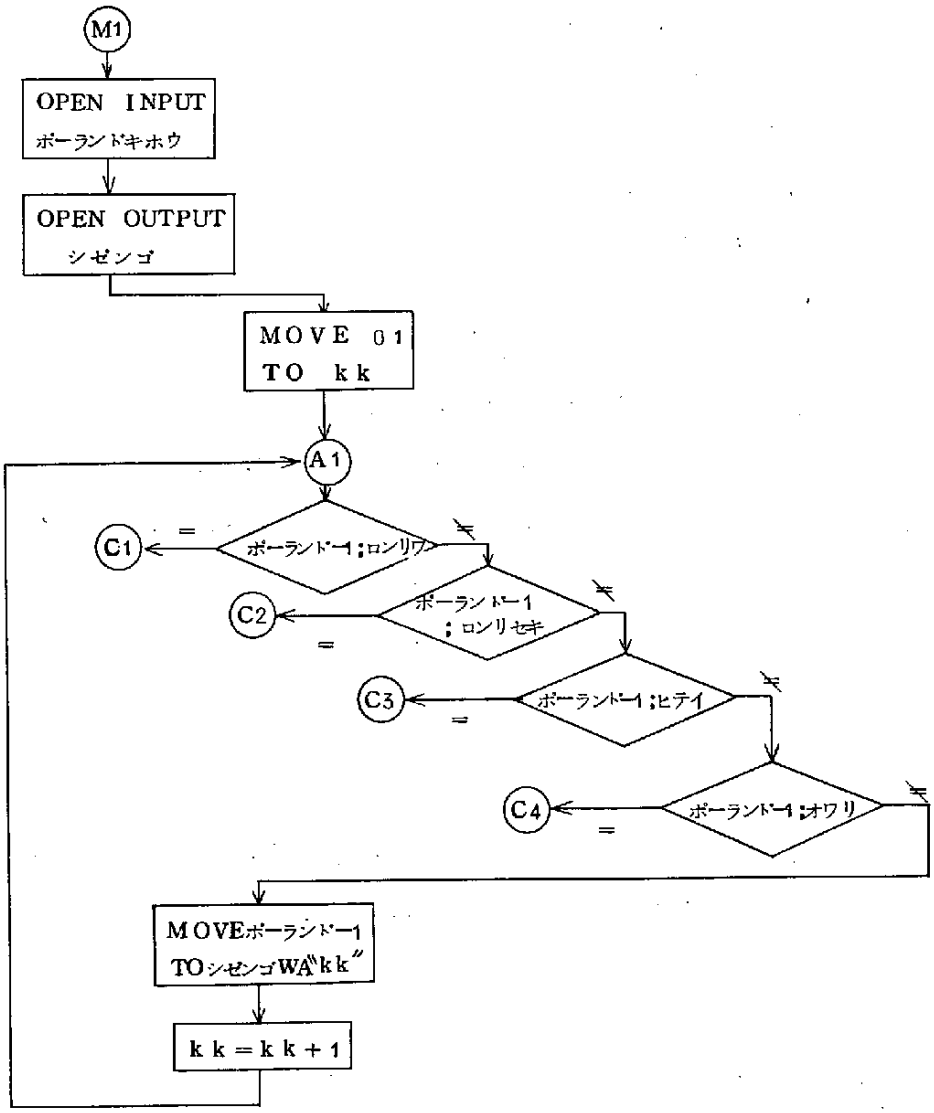

M1
OPEN INPUT
ボーランドキホウ
OPEN OUTPUT
シゼンゴ
MOVE 01
TO kk
A1
C1
=
ボーランド1;ロンリワ
≠
C2
=
ボーランド1
;ロンリセキ
≠
C3
=
ボーランド1;ヒテイ
≠
C4
=
ボーランド1;オワリ
≠
MOVEボーランド1
TOシゼンゴWA"kk"
kk=kk+1

図　Ⅱ－9　(a)

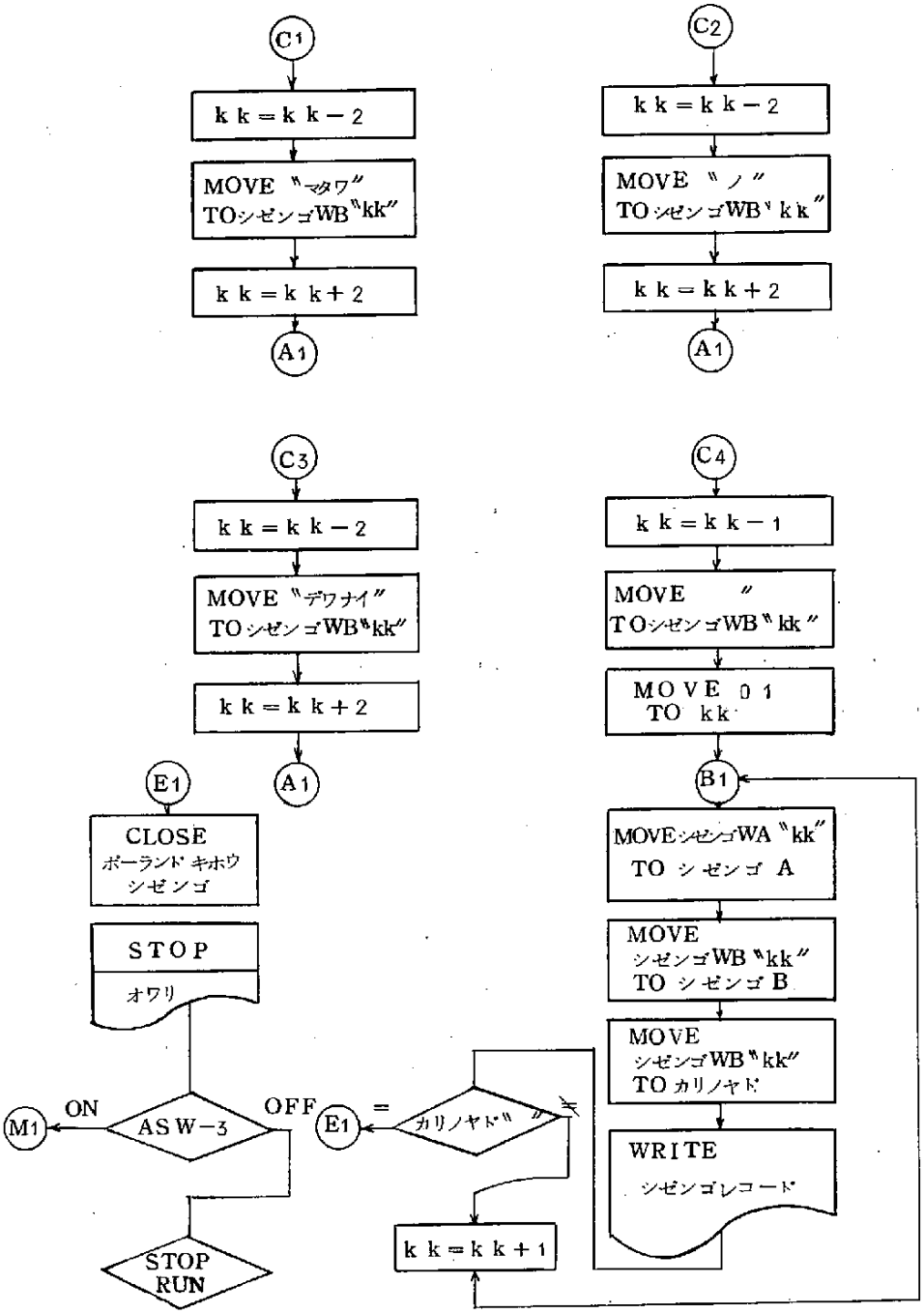

C1
k k = k k − 2
MOVE "マタワ"
TOシゼンゴWB"kk"
k k = k k + 2
A1
C2
k k = k k − 2
MOVE "ノ"
TOシゼンゴWB"kk"
k k = k k + 2
A1
C3
k k = k k − 2
MOVE "デワナイ"
TOシゼンゴWB"kk"
k k = k k + 2
A1
C4
k k = k k − 1
MOVE "
TOシゼンゴWB"kk"
MOVE 0 1
TO kk
B1
MOVEシゼンゴWA"kk"
TO シゼンゴ A
MOVE
シゼンゴWB"kk"
TO シゼンゴ B
MOVE
シゼンゴWB"kk"
TO カリノヤド
WRITE
シゼンゴレコード
E1
CLOSE
ポーランド キホウ
シゼンゴ
STOP
オワリ
M1
ON
ASW−3
OFF
E1
=
カリノヤド"
"
≠
k k = k k + 1
STOP
RUN

図　Ⅱ − 9　(b)

3，5　お　わ　り　に

本節では対話型情報検索システムの一つのタイプについて述べた。

質問照会用の日本語の選択、あるいは、日本語自身の問題、例えば格助詞の問題がより大いに検討を要するが面白い問題だと思う。

4．自然語によるFact Retrieval Systemの構成例－1

4，1　は　じ　め　に

本節では、W. S. Cooperの作った自然語で質問を書き、目的の情報を論理的に演繹するシステムについて述べる。[ ] この論文では最初にFact Retrieval Systemの備えるべき条件が規定され、自然語で質問を許す場合の自然語の処理の困難さ等を論じ、ついで検索システムの大要を決め、形式言語を設定しシステムを組立てている。このシステムは形式的に記述された入力の制御のもとに問題を解いて行くという方向を志向したという点で新しいものといえる。

4，2　Fact Retrieval Systemの備えるべき要素と自然語処理の困難さについて

Fact Retrieval Systemの備えるべき要件とは、(1)特定の目的の為に作られた人工言語によらず、いわゆる自然言語でもって情報を受け入れ、あるいは質問を受け付けねばならない。(2)また、入力文にある内容に対して論理的に演繹が可能でなければならない。つまりシステムへの情報の格納に際しては、自然語で書かれた情報を入力する毎に、システムがこれを適当な形に変換できるようになっている必要があり、また自然語で書かれた文章の中での事項については論理学でいう〝代入の法則〟、〝modus ponen〟などによる演繹が可能でなければならないということである。しかし、自然語の使用を許すとすればつぎのようないくつかの困難さが解決されなければならない。すなわち、

(i)　自然語は不変ではない。したがって自然語のもつ論理的特性、あるいは構文等の正確な記述を行なう事は無意味である。

(ii) 自然語の値域は広すぎる。

(iii) 自然語は、本質的に紛れをもっている。

(iv) 自然語の文章中である与えられた関係が成立するかどうかを決める自己撞着しないような判定基準は存在しない。

(v) また自然語は論理的にも不可解性のものである。

このような困難さは、Fact Retrieval Systemの存在を全く否定するという訳ではない、すなわち、(i)に対しては言語の変化はそんなには早くなく、ゆっくり長年月にわたって変化するものであり、システムの寿命等に比べるとはるかに長いものであり問題はないと考えられる。同じく他の困難についても実用的な観点からの解決を考えることができる。

4．3 Fact Retrieval のアルゴリズム

質問文がシステムに与えられたときには、質問の論理的関係を検出するようなプログラムのもとに、以下のようなアルゴリズムに基づいて答えが出されるものとする。

(1) 格納されている文章の集合を探してみて、それらの文章の論理的帰結が質問文になっているような文章の部分集合を求める。もしそのような文章が見つかれば、〝TRUE〟を印字し、そのような文章が見つからないときは、(2)に進む。

(2) 格納されている文章の部分集合のうち質問文と論理的に矛盾するような部分集合を求める。もしそのような文章が見つかれば、〝FALSE〟を印字し、そのような文章すら見い出せないときには(3)に進む。

(3) 〝UNABLE TO ANSWER〟を印字し、システムは停止する。

1つの例としてシステムの中に〝Birds are vertebrates.〟と〝Birds are not mammals.〟なる文が格納されており、質問文に〝All vertebrates are mammals.〟か否かを問うものがあれば、この文章は、前の二文と矛盾するものであり、〝FALSE〟が印字されるという工合である。

4．4 自然語からの切出し

完全な自然語を使用する事は前述の理由により不可能であるので自然語の部分集合をもって格

納情報、または質問とする。自然語の部分集合の文法を以下に示す。文法中で使用する記号について簡単に説明を加えておく。

； 空白を表わす。

$\cap$ ； 連接を表わす。

・ ； 相連接を表わす。すなわち、

$$X \cdot Y \langle \equiv \rangle \{ x \cap y \mid x \in X,\ y \in Y \}$$

O ； 空系列を表わす。

＋ ； 和集合を作りだすのに用いる。

これらの記号の使用については つぎにその一例を示す。

$$\#A \cap \#MAMMAL = \#A\#MAMMAL$$

$$\#A \cap O = \#A$$

$$\{\#A, \#AN\} = \{\#A, \#AN, O\}$$

$$\{\#A, \#AN\} + \{\#AN, \#SOME\} = \{\#A, \#AN, \#SOME\}$$

これらの記号を用いる情報記述用言語には、

(1) $N = \{\#METAL, \#OXIDE, \#NONMETALS, \cdots\cdots\}$

(2) $A = \{\#SOLID, \#WHITE, \cdots\cdots\}$

(3) $C = \{\#MAGNESIUM, \#SODIUM, \cdots\cdots\}$

(4) $V = \{\#BURNS, \#BURNS\#RAPIDLY, \cdots\cdots\}$

(5) $B = (A^1 \cdot A \cdot \{\#AND\}^1)^1 \cdot A$

(6) $M = \{\#A, \#AN\}^1 \cdot B^1 \cdot C^1 \cdot N$

(7) $R = \{\#WHICH, \#THAT\} \cdot (\{\#IS, \#ARE\} \cdot \{\#NOT\}^1 \cdot (M+B) +$
$(\{\#DOES, \#DO\} \cdot \{\#NOT\}^1 \cdot V)$

(8) $S = M \cdot R^1$

(9) $P = \{\#IS, \#ARE\} \cdot \{\#NOT\}^1 \cdot (S+B) + (\{\#DO, \#DOES\} \cdot \{\#NOT\}^1 \cdot V$

(10) $L = \{\#ALL, \#ANY, \#EACH, \#EVERY, \#NOT\#ALL, \#NOT\#EVERY,$
$\#SOME, \#NO\}^1 \cdot S \cdot P$

このような文法は普通生成文法と呼ばれるものである。最も簡単な例としては、#FUEL

#BURNS 等が考えられ、もう少し複雑な例としては、#ALL#SOLID#WHITE#AND#METALLIC#MAGNESIUM#COMPOUNDS#WHICH#ARE#NOT#WHITE#BRITTLE#AND#COMBUSTIBLE#FERROUS#OXIDE#ARE#NOT#DARK#GRAY#BRITTLE#AND#SOLID#MAGNESIUM#COMPOUNDS#THAT#ARE#NOT#SOLID#WHITE#AND#COMBUSTIBLE#SODIUM#SULFIDES.

等がある。

4.5 論理言語 L^*

論理的演繹を行なう為の論理言語 L^* を設定する。格納情報はもちろん質問文もこの L^* 上で比較され、真偽が問われる。情報はすべてアリストテレスの4つの定言的判断の形に整理されて格納されている。アリストテレスの定言的判断とは、

$\mathbf{A}\,xy$ iff $x \subseteq y$ and $x \neq O$.

$\mathbf{E}\,xy$ iff $x \cap y = O$ and $x \neq O$ and $y \neq O$.

$\mathbf{I}\,xy$ iff $x \cap y \neq O$.

$\mathbf{O}\,xy$ iff $x \not\subseteq y$ and $x \neq O$ and $y \neq O$.

それぞれ全称肯定判断、全称否定判断、特称肯定判断、特称否定判断と呼ぶ。これらを組合わした L^* 上の文の一例として、たとえば、$\mathbf{A}(x \cap y)Z$ などが考えられ、これを〝yであるすべてのxはZである〟という風に読む。つぎに L^* で必要とする普偏集合、およびいくつかの記号について説明を加える。普偏集合としては、#THING を考える、すなわち、

$\#\mathrm{THING} = \sim O$

である。記号については、

$\cap$ ； 集合の論理積

$\sim$ ； 否定

$\wedge$ ； 論理的連接

$\rightarrow$ ； 含意

$|$ ；シェーファストローク

これらを用いて L^* の定義を行なう。

(1) $Variables = N + A \cdot \{'\} + C \cdot \{''\} + V \cdot \{'''\}$

(2) $Constants = \{\#THING\}$

(3)
$$\begin{cases} Terms_0 = Variable + Constants \\ Terms_{i+1} = \{\cap\} \cdot Terms_0 \cdot Terms_0 + \sim\{\sim\} \cdot Terms_i \\ Terms = \bigcup_{i=0}^{\infty} Terms_i \end{cases}$$

(4)
$$\begin{cases} L_0^* = \{A, E, I, O\} \cdot Terms \cdot Terms \\ L_{i+1}^* = \{\wedge\} \cdot L_0^* \cdot L_i^* + \{\rightarrow\} \cdot L_i^* + \{\mid\} \cdot L_i^* \cdot L_i^* \\ L^* = \bigcup_{i=0}^{\infty} L_i^* \end{cases}$$

ここで L^* の文章を上げておく。

(a) A#MAGNESIUM#WHITE'

(b) A#MAGNESIUM#BRITTLE'

(c) A#MAGNESIUM∩#WHITE'#BRITTLE'

(d) →A#MAGNESIUM∩#WHITE'#BRITTLE'A#MAGNESIUM#WHITE'

(e) →∧A#MAGNESIUM#WHITE'A#MAGNESIUM#BRITTLE'

A#MAGNESIUM∩#BRITTLE'#WHITE'

(f) E#MAGNESIUM#BRITTLE'

(g) |A#MAGNESIUM#BRITTLE', E#MAGNESIUM#BRITTLE'

特に、(d)、(e)、(g)は L^* 上の定理の1例である。$\rightarrow \ell_1 \ell_2$ であるとき、文章 ℓ_2 は文章 ℓ_1 の論理的帰結であるといい、また、$|\ell_1 \ell_2$ であるとき文章 ℓ_1 と文章 ℓ_2 は矛盾するという。このような意味において(c)は(a)，(b)の文章の論理的帰結であり、(f)は(a)，(b)というより、(b)を含む任意の文章とは論理的に矛盾するものである。

4．6　L から L^* への写像函数

ここでは、L から L^* への一価写像函数 T_L を構成する。これは情報を格納し、または取り出す時にも必要である。T_L の定義は、

(1) $T_N = \{<\#METALS, \#METAL>, <\#OXIDES, \#OXIDE>, \cdots, <\#SODIUM, \#SODIUM>\}$.

(2) $T_A = \{<\#SOLID, \#SOLID'>, <\#WHITE, \#WHITE'>, \cdots, <\#COMBUSTIBLE, \#COMBUSTIBLE'>\}$.

(3) $T_C = \{<\#MAGNESIUM, \#MAGNESIUM''>, \cdots, <\#SULPHURIC, \#SULFURIC''>\}$.

(4) $T_V = \{<\#BURNS, \#BURNS'''>, \cdots, <\#BURNS\#RAPIDLY, \#BURNS\#RAPIDLY'''>\}$.

(5) $T_B = \{<a, T_A(a)> : a \in A\} + \{<a_1 \frown \alpha \frown a_2, \Pi \frown T_{A(a_1)} \frown T(a_2)> : \alpha \in \{\#AND\}^1\}$
$+ \{<a_1 \frown a_2 \frown \alpha \frown a_3, \Pi \frown \Pi \frown T_{A(a_1)} \frown T_{A(a_2)} \frown T_{A(a_3)} : \alpha \in \{\#AND\}^1\}$

(6) $T_M = \{<\alpha \frown n, T(n)> : \alpha \in \{\#A, \#AN\}^1\} +$
$\{<\alpha \frown b \frown n, \Pi \frown T_{B(b)} \frown T_{N(n)} > : \alpha \in \{\#A, \#AN\}^1\} +$
$\{<\alpha \frown c \frown n, \Pi \frown T_{C(c)} \frown T_{N(n)} > : \alpha \in \{\#A, \#AN\}^1\} +$
$\{<\alpha \frown b \frown c \frown n, \Pi \frown \Pi \frown T_{B(b)} \frown T_{C(c)} \frown T_{N(n)} > : \alpha \in \{\#A, \#AN\}^1\}$

(7) $T_R = \{<\alpha \frown \beta \frown m, T_{M(m)} > : \alpha \in \{\#WHICH, \#THAT\}$ and $\beta \in \{\#IS, \#ARE\}\} +$
$\{<\alpha \frown \beta \frown b, T_{B(b)} > : \alpha \in \{\#WHICH, \#THAT\}$ and $\beta \in \{\#IS, \#ARE\}\} +$
$\{<\alpha \frown v, T_{V(v)} > : \alpha \in \{\#WHICH, \#THAT\}$ and $\beta \in \{\#IS, \#ARE\}\} +$
$\{<\alpha \frown \beta \frown \#NOT \frown m, \sim \frown T_{M(m)} > : \alpha \in \{\#WHICH, \#THAT\}$ and $\beta \in \{\#IS, \#ARE\}\} +$
$\{<\alpha \frown \beta \frown \#NOT \frown b, \sim \frown T_{B(b)} > : \alpha \in \{\#WHICH, \#THAT\}$ and $\beta \in \{\#IS, \#ARE\}\} +$
$\{<\alpha \frown \beta \frown \#NOT \frown v, \sim \frown T_{V(v)} > : \alpha \in \{\#WHICH, \#THAT\}$ and $\beta \in \{\#DOES, \#DO\}\}$

(8) $T_S = \{<m, T_{M(m)}> : \} + \{<m \frown r, \Pi \frown T_{M(m)} \frown T_{R(r)} : \}$

(9) $T_P = \{<\alpha \frown s, T_{S(s)} > : \alpha \in \{\#IS, \#ARE\}\} + \{<\alpha \frown b, T_{B(b)} > : \alpha \in \{\#IS, \#ARE\}\} +$
$\{<v, T_{V(v)}> : \} + \{<\alpha \frown \#NOT \frown S, \sim \frown T_{S(s)} > : \alpha \in \{\#IS, \#ARE\}\} +$
$\{<\alpha \frown \#NOT \frown b, \sim \frown T_{B(b)} > : \alpha \in \{\#IS, \#ARE\} +$
$\{<\alpha \frown \#NOT \frown v, \sim \frown T_{V(v)} : \alpha \in \{\#DOES, \#DO\}\}$

(10) $T_L = \{<\alpha \frown s \frown p, A \frown T_{S(s)} \frown T_{P(p)} > : \alpha \in \{\#ALL, \#EVERY, \#EACH, \#ANY\}^1\} +$

$\{<\#NO\frown S\frown P,\ \mathbf{E}\frown T_{S(s)}\frown T_{P(p)}>:\}+$

$\{<\#SOME\frown S\frown P,\ \mathbf{I}\frown T_{S(s)}\frown T_{P(p)}>:\}+$

$\{<\alpha\frown s\frown p,\ \mathbf{O}\frown T_{S(s)}\frown T_{P(p)}>:\alpha\in\{\#NOT\#ALL,\#NOT\#EVERY\}\}$

T_L を適用した例についていくつかをあげる。

(a) T_L (#FUEL#BURNS)=**A**#FUEL#BURNS'''

(b) T_L (#NOT#EVERY#SULFIDES#ARE#BRITTLE)

=**O**#SULFIDE#BRITTLE'

(c) T_L (#FERROUS#SULFIDE#IS#A#DARK#GRAY#COMPOUND#THAT

#IS#BRITTLE)

=**A**∩ FERROUS'#SULFIDE∩∩#DARK#GRAY' # COMPOUND

#BRITTLE'

つぎにこのような T_L を実行するようなアルゴリズムについて述べる。

(1) 指標 i を 1 に設定。

(2) 入力系列を調べて、L の定義中の i 番目の部分定義によって定義される文法的な類 X_i に属する部分系列 σ を求める。もしそのような系列 σ が見い出せれば、これに X_i という名前を付し、(3)に進む。もし見い出せないときには(4)に進む。

(3) T_L の定義中、i 番目の部分定義によって定義される $T_{Xi}(\sigma)$ を作り、これに対して(2)で行なったと同じように X_i という名前を付す。そして(2)に帰る。

(4) i が L の部分定義の個数より小さいときにはこれを i + 1 にして(2)に帰る。そうでないときには、$T_L(\sigma)$ が完成したのでありこの T_L が適用し尽されたのでこのルーチンを出る。

具体的な例に図 II－10 を示す。

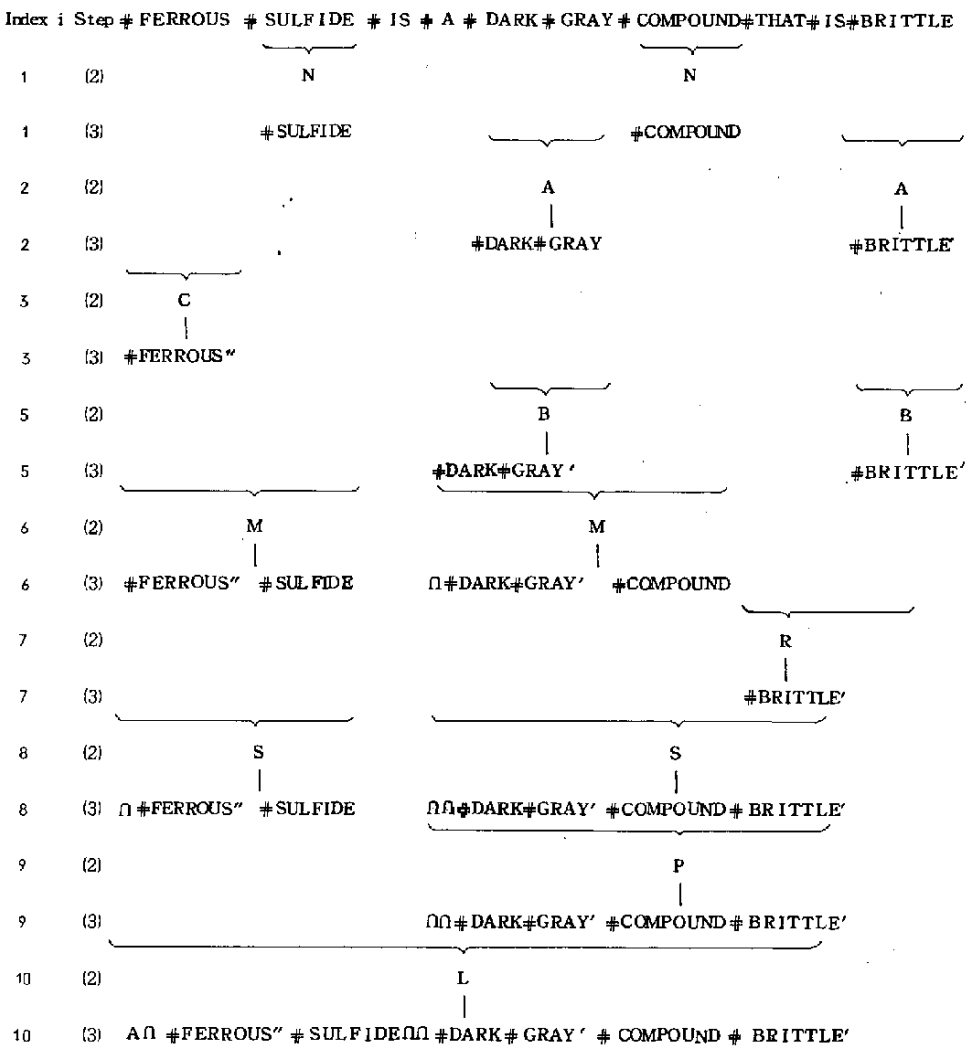

Index i Step # FERROUS # SULFIDE # IS # A # DARK # GRAY # COMPOUND # THAT # IS # BRITTLE
1 (2) N N
1 (3) # SULFIDE # COMPOUND
2 (2) A A
2 (3) # DARK # GRAY # BRITTLE
3 (2) C
3 (3) # FERROUS"
5 (2) B B
5 (3) # DARK # GRAY' # BRITTLE'
6 (2) M M
6 (3) # FERROUS" # SULFIDE ∩ # DARK # GRAY' # COMPOUND
7 (2) R
7 (3) # BRITTLE'
8 (2) S S
8 (3) ∩ # FERROUS" # SULFIDE ∩∩ # DARK # GRAY' # COMPOUND # BRITTLE'
9 (2) P
9 (3) ∩∩ # DARK # GRAY' # COMPOUND # BRITTLE'
10 (2) L
10 (3) A∩ # FERROUS" # SULFIDE∩∩ # DARK # GRAY' # COMPOUND # BRITTLE'

図 Ⅱ－１０

4.7 入力データと検索の例

いわゆるデータ文章としてはつぎのような文章が選ばれた。すなわち

(1) MAGNESIUM IS A METAL

(2) MAGNESIUM BURNS RAPIDLY

(3) MAGNESIUM OXIDE IS A WHITE METALLIC OXIDE

(4) OXYGEN IS A NONMETAL

(5) FERROUS SULFIDE IS A DARK-GRAY COMPOUND THAT IS BRITTLE

(6) IRON IS A METAL

(7) SULFUR IS A NONMETAL

(8) GASOLINE IS A FUEL

(9) GASOLINE IS COMBUSTIBLE

(10) COMBUSTIBLE THINGS BURN

(11) FUELS ARE COMBUSTIBLE

(12) ICE IS SOLID

(13) STEAM IS A CAS

(14) MAGNESIUM IS AN ELEMENT

(15) IRON IS AN ELEMENT

(16) SULFUR IS AN ELEMENT

(17) OXYGEN IS AN ELEMENT

(18) NITROGEN IS AN ELEMENT

(19) HYDROGEN IS AN ELEMENT

(20) CARBON IS AN ELEMENT

(21) COPPER IS AN ELEMENT

(22) SALT IS A COMPOUND

(23) SUGAR IS A COMPOUND

(24) WATER IS A COMPOUND

(25) SULFURIC ACID IS A COMPOUND

(26) ELEMENTS ARE NOT COMPOUNDS

(27) SALT IS SODIUM CHLORIDE

(28) SODIUM CHLORIDE IS SALT

(29) OXIDES ARE COMPOUNDS

この他補助的なデータ文としてつぎのような文章が選ばれた。

(30) METALS ARE METALLIC

(31) NO METAL IS A NONMETAL

(32) DARK-GRAY THINGS ARE NOT WHITE

(33) A SOLID IS NOT A GAS

(34) ANY THING THAT BURNS RAPIDLY BURNS

つぎに検索の例をいくつか示す。

(i) #MAGNESIUM# IS#A#METAL#THAT#BURNS
#RAPIDLY ⊨ TRUE

(ii) #FERROUS#SULFIDE#IS#A#DARK#GRAY#COMPOUND#THAT#IS
#BRITTLE ⊨ FALSE

(iii) # IODINE#IS#A#COMPOUND
⊨THE VOCABULARY OF THIS QUERY-SENTENCE IS DISALLOWED.

(iv) #MAGNESIUM#AND# IRON#ARE#METALS
⊨THE SYNTAX OF THIS QUERY-SENTENCE IS DISALLOWED

(v) #SOME#OXIDES#ARE#NOT#WHITE
⊨UNANSWERABLE ON THE BASIS OF THE STORED INFORMATION.

4.8 むすび

本章では、情報の演繹が可能であり、自然語で情報のやりとりが可能であるようなFact Retrieval Systemの一構成例について述べた。

このようなシステムの構成の仕方は、実用如何は別としても、対象によっては極めて面白いものと思われる。

5　自然語によるFact Retrieval Systemの構成例－2

5．1　は　じ　め　に

4で述べたFact Retrieval Systemでは自然語の使用が許され、情報の演繹が行ない得るように仕組まれていたが、演繹の方向は定まっておらず、いわゆるmaze－runningが行なわれていた。これから述べようとするシステムは道路網等に関する情報検索システムであり、情報の演繹の方向づけが与えられているという点で前者と異なっている。またこのシステムでは入力の質問はＣＦ－文法によって生成されるものであり、道路網等の状態をFuzzy集合の上での値によって記述しているという点が目新しいといえよう。総じて4のシステムが型式論理に主眼が置かれていたのに対してより対象に密着したFact Retrieval Systemといえよう。

5．2　ＰＲシステムのあらまし

ＰＲシステムは、道路網等に関する情報を格納しておき、必要に応じてこれをサービスしようとするシステムである。ＰＲシステムは、図Ⅱ－１１のような構成をとっている。

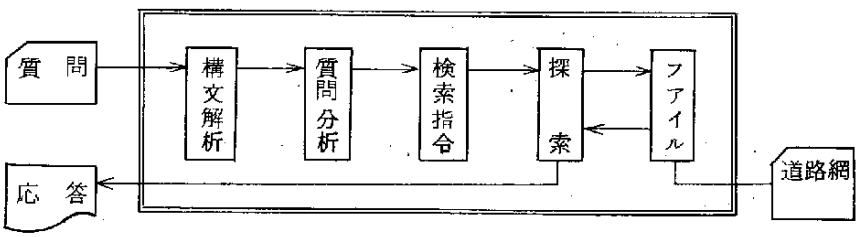


図　Ⅱ－１１

ＰＲシステムでは、通常の情報検索システムと同じように、まず情報要求者からの質問（要求）を、chomsky 流の context－free 文法より生成される擬自然英語の上での検索質問に書き直す。そしてこの検索質問を構文解析(Non－Selective Top－to－Bottom Parsing）することにより探索のアルゴリズムを作成した。また、ＰＲシステムの情報源としては、その接続関係の記述と通行のし易さに限り、通行のし易さは、fuzzy集合の概念によ

ってある分岐点からある分岐点までのそれを与えた。これについては、写像函数を選ぶことによって交通停滞とか混雑を表現できると考えられる。

最適路線探索アルゴリズム構成の前に、このアルゴリズム中で使用する行列の演算を定義し、いくつかの必要な定理を導いている。行列の積を求めるに際しては、演算回数を減らすためWershallのアルゴリズムの一般化を利用している。

システム内からの応答は、プリンター、または、ドラフタックを通じてサービスされる。

以上がＰＲシステムのあらましであり、以後、各項について述べる。

5．3　ＰＲシステムで取り扱う質問

一般に、情報検索システムを構成するに際しては、対象を定めたのちその対象に対してどのような要求が考えられるかを想定し、これに対処できるようにするのが普通である。ＰＲシステムでもこの常道にしたがい、道路網についてどのような質問が考え得るかについて書き出してみることにする。すなわち、

(1)　ある地点からある地点までの道路網如何。

(2)　上の質問において経由点を指定したもの

(3)　同じく最適であるとの附加条件を付けたもの

(4)　そ　の　他

が考えられる。上記の質問を組み合わすことによって更に複雑なものが考えられるが、ＰＲシステムでは、(3)の附加条件を附した(1)、(2)の質問を考えることにした。これらの質問は、つぎのようなchomsky流のcontext free文法によって生成されるものに限ることにした。

S→Show me AB	D→from N
S→Show me AC	E→to N
S→AB	F→via N
S→AC	N→1
A→the best path	N→2
B→CF	.....
C→DE	N→n

この文法より生成される質問の構文解析には、下降型構文解析法、いわゆるNor−Selective Top−to−Bott Parsingを用いて行なう。これらの質問は、オンラインでタイプライターを通じて、または、オフラインでカードを以ってシステムに入れるようになっている。

5．4　PRシステムにおける道路網の状態の記述

前節で述べたように、このような道路網に関する情報検索システムの検索対象としては、単に最適であるといっても、最短路線、最少費用、最短時間等要求に応じていろいろ考えられる。一般に、交通停滞、交通事故、道路の舗装状態等の点から、非常に漠然としてはいるが、ある道路は行き易いとか、非常に行きにくいとか言い表わすことがある。PRシステムでは、この漠然とした道路の状態をfuzzy集合の上での値で表わすことによって種々の要求に答えることにした。この具体的な値としては、基本道路（分岐点を直接に結ぶ道路）が最も行き易い状態にある場合には1を、最も行き難い状態にある場合には0を割り当てることにし、その中間的な状態の場合には0と1との間の実数を割り当てることにする。具体的な一例としては、交通停滞Xmで、その行き易さは $\frac{1}{1}\,T\left(X/100\right)^2$ であるという具合に決めればよいと思う。すべての基本道路の状態は常に行列の上に記述しておき、warshallのアルゴリズムの一般化等を用いて、ある分岐点よりある分岐点まで到達する路線のうち、最も行き易い路線を決定することにする。

5．5　路線の探索のための行列演算

この節では、PRシステムで路線の探索のために使用するアルゴリズムで用いる行列の演算を定義し、必要な定理を導くことにする。

行列A、Bは（n×n）次正方行列であり、A、Bの（i，j）成分 a_{ij}，b_{ij} は、ともに $a_{ij} \geq 0$、$b_{ij} \geq 0$ である。

［定義1］　行列Aと行列Bの和の（i，j）成分は、2つの行列の（i，j）成分のうち大なる方の値とする。$(A \oplus B)_{ij} = \max(a_{ij}, b_{ij})$ と表わす。

［定義2］　行列Aと行列Bの積の（i，j）成分は、(a_{ik}, b_{kj}) K＝1，2，…，n について小なる値のうちで最大のものをとる。

$$(A * B)_{ij} = \max\{\min(a_{i1}, b_{1j}), \min(a_{i2}, b_{2j}), \cdots, \min(a_{in}, b_{nj})\}$$

と表わす。

〔定義3〕　行列のべきは次の様に定義する。

$$A' = A,\ A^2 = A * A,\ \cdots,\ A^K = A^{K-1} * A,\ \cdots$$

$$A^{(1)} = A,\ A^{(2)} = A \oplus A^2,\ \cdots,\ A^K = A \oplus A^2 \oplus \cdots \oplus A^K,\cdots$$

〔定理1〕　行列の和については交換則が成立する。すなわち$A \oplus B = B \oplus A$

〔証　明〕　定義1より明らかである。

〔定理2〕　$(A^K)_{ij} \leq (A^{(K)})_{ij}$　$(R = 1, 2, \cdots, n)$.

〔証　明〕　定義より

$$\begin{aligned}(A^{(K)})_{ij} &= (A \oplus A^2 \oplus \cdots \oplus A^K)_{ij} \\ &= (A^{(K-1)} \oplus A^K)_{ij} \\ &\geq (A^K)_{ij}\end{aligned}$$

〔定理3〕

$$(A^{(K)})_{ij} \leq (A^{(K+1)})_{ij} \quad (K=1, 2, \cdots, n-1)$$

〔証　明〕

1° $K = 1$の場合

$$(A^{(1)})_{ij} = (A)_{ij} = a_{ij}$$

$$\begin{aligned}(A^{(2)})_{ij} &= (A \oplus A^2)_{ij} \\ &= \max\left[a_{ij}, \max\left\{\min(a_{i1}, a_{1j}), \cdots \right.\right. \\ &\qquad \left.\left. \cdots, \min(a_{in}, a_{nj})\right\}\right]\end{aligned}$$

$$\max\left\{\min(a_{i1}, a_{1j}), \cdots, \min(a_{in}, a_{nj})\right\} = a^2_{max}$$

とおくと、

(i)　$a^2_{max} \leq a_{ij}$　ならば、$(A^{(2)})_{ij} = a_{ij}$

(ii)　$a^2_{max} > a_{ij}$　ならば、$(A^{(2)})_{ij} = a^2_{max}$

故に、$(A^{(1)})_{ij} = (A^{(2)})_{ij}$

2° $K = \ell$の場合、但し、ℓは、

$1 \leq \ell \leq n-2$　なる整数

$(A^{(\ell)})_{ij} = (A^{(\ell+1)})_{ij}$ならば、

$(A^{(\ell+1)})_{ij} = a^{\ell+1}_{max}$ とおくと、

$(A^{(\ell+2)})_{ij} = (A^{(\ell+1)} \oplus A^{\ell+2})_{ij}$

$(A^{(\ell+2)})_{ij} = (A^{(\ell+1)} \oplus A^{\ell+2})_{ij}$

$$= \max[a^{\ell+1}_{max}, \max\{\min(a_{i1}, {}^{\ell+2}, a^{\ell+2}_{ij}), \cdots$$

$$\cdots, \min(a^{\ell+2}_{in}, a^{\ell+2}_{ij})\}]$$

同じように

$$\max\{\min(a^{\ell+2}_{i1}, a^{\ell+2}_{ij}), \cdots, \min(a^{\ell+2}_{in}, a^{\ell+2}_{nj}, \} = a^{\ell+2}_{max}$$

とおくと、

(i) $a^{\ell+2}_{max} \leq a^{\ell+1}_{ij} = a^{\ell+1}_{max}$ ならば $(A^{(\ell+2)})_{ij} = a^{\ell+1}_{max}$

(ii) $a^{\ell+2}_{max} > a^{\ell+1}_{ij}$ ならば $(A^{(\ell+2)})_{ij} = a^{\ell+2}_{max}$

故に、$(A^{(\ell+1)})_{ij} \leq (A^{(\ell+2)})_{ij}$

よって、$1 \leq K \leq n-1$ の整数Kに対して、

$$(A^{(K)})_{ij} \leq (A^{(K+1)})_{ij}$$

が成立する。

〔定理４〕 $(A^{(n)})_{ij} = (A^{(n+K)})_{ij}$ $(K = 1, 2, \cdots)$

〔証　明〕 $K = 1$ のとき

$$(A^{(n+1)})_{ij} = \max_{Kn=1}^{n} \max_{Kn-1=1}^{n} \cdots \max_{K_1=1}^{n} \{\min(a_{ik_1}, a_{k_1k_2}, \cdots,$$

$$a_{k_{n-1}k_n}, a_{k_nj})\}$$

$1 \leq S \leq n$ なる任意のSについて、$K_S = j$, $K_S = i$, $K_S = K_r$

$(1 \leq r \leq n)$ の場合について考えてみると、

1° $K_S = j$ のとき

$\min(a_{ik_1}, a_{k_1k_2}, \cdots, a_{k_{n-1}k_n}, a_{k_nj})$ は、つぎのようになる。すなわち、$\min(a_{ik_1}, a_{k_1k_2}, \cdots, a_{s-1\,j}, \cdots, a_{k_{n-1}k_n}, a_{k_nj})$

一般に $\min(a, b) \leq a$ であるから上式は、$\min(a_{ik_1}, a_{k_1k_2}, \cdots, a_{k_{s-1}j})$ に含まれる。

すなわち、A^s の (i, j) の成分である。

2° $K_S = i$, $K_S = k_r$ に対しても同様にして、任意の S'、S'' $(S', S'' \leq n)$

について$A^{s'}$，または$A^{s''}$の（i，j）成分に含まれることを示すことができる。

故に$A^{(n+1)}$の（i，j）成分は、A^{r}（$r \leq n$）に含まれ、$A^{(n+1)}$の（i，j）成分は、$A^{(n)}$の（i，j）成分である。以下これを繰返すことによって証明を行なうことが可能である。

この節では、行列の演算を定義し、探索のために必要ないくつかの定理を導いた。

5．6　道路網と最適路線の定義

ある分岐点より出発して、同一分岐点を2度以上通ることなく、有限個の分岐点を経由して、目的の分岐点まで到達する路線のうち最適なものを決定するために、道路網への拘束とその記述をつぎのように取り決めることにする。

(1) 分岐点iより分岐点j（$i \neq j$）まで、他の分岐点を通過することなく、直接到達可能であるとき、分岐点iと分岐点jは隣接しているといい、隣接する2つの分岐点を結ぶ線P_{ij}を基本道路とよぶ。

(2) 孤立した分岐点は存在しないこと、すなわち、任意の分岐点より有限個の分岐点を通過して他のすべての分岐点に到達可能であること。

(3) 分岐点iより分岐点jまでの基本道路の行き易さの値をw_{ij}で表わす。ただし、$0 \leq w_{ij} \leq 1$，$1 \leq i$，$j \leq n$。

$w_{ij} = 0$（$i \neq j$）；分岐点iと分岐点jの間には基本道路が存在しない。

$w_{ij} = 1$（$i \neq j$）；分岐点iと分岐点jの間の基本道路は、最も行き易い状態にある。

演算の都合上、$w_{ij} = 0$としているが、分岐点iより出発して他の分岐点を通ることなく分岐点iにもどる基本道路が存在するときには、図Ⅱ－12のような仮想的な分岐点を考えればよい。

(4) 分岐点iより分岐点jまで、分岐点$k_1, k_2, \cdots, K_{\ell-1}$を通過して到達可能であるとき、基本道路の系列$P_i k_1$，$Pk_1 k_2, \cdots, Pk_{\ell-1} j$を分岐点iより分岐点jまでの路線とよび、$P(ik_1k_2 \cdots, k_{\ell-1}j)$で表わし、この路線の長さを

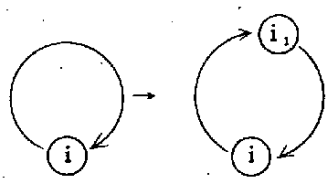


図　Ⅱ－12

ℓとする。

路線P($ik_1k_2\cdots, k_{\ell-1}j$) の行き易さの値w($ik_1k_2\cdots, k_{\ell-1}j$)はその路線に含まれる基本道路の行き易さの値のうち最小のものであるとする。すなわち、

$$w(ik_1\cdots, k_{\ell-1}j) = min(w_ik, wk_1k_2, \cdots, wk_{\ell-1}j)$$

であるとする。

(5) 分岐点iより分岐点jまで相異なる$\ell-1$個の分岐点を通過して到達する路線のうち、最大の行き易さの値を持つ路線を長さℓの最適格線とよぶ。長さℓの最適路線の行き易さの値を$W'(i, j)$で表わすと、

$$W'(i, j) = \max_{k_1, k_2, \cdots, k_{\ell-1}} (W(ik_1k_2\cdots, k_{\ell-1}j))$$

となる。

(6) 分岐点iよりjまで到達するすべての路線のうち最大の行き易さの値を持つものを、分岐点iより分岐点jまでの最適路線とする。これをPost(i, j)で表わし、その行き易さの値をWopt(i, j)で表わす。すなわち

$$Wopt(i, j) = max(W'(i, j), W^2(i, j), \cdots, W^n(i, j))$$

とする。

5．7 最適路線決定のアルゴリズム

道路網の行き易さの値を、各行を出発分岐点、各列を到達分岐点とする行列に書き込み記憶装置に格納することにする。この行列式を$A = [wij]$とする。

今、分岐点iより、分岐点jに到る最適路線Popt(i, j)を決定する。

このために、通過する分岐点の系列を格納するための1次元配列PATH(n) に出発分岐点番号iを入れておく。すなわち、PATH(o)：＝i つぎに最適路線Popt(i, j)の行き易さの値Wopt(i, j)を求めるわけであるが、これは4の定理3および4によって

$$(A)_{ij} \leq (A^{(2)})_{ij} \leq \cdots \leq (A^{(n)})_{ij} \leq (A^{(n+b)})_{ij}$$

であるから

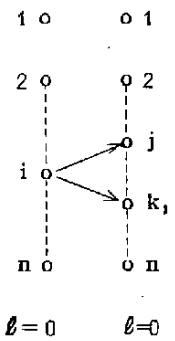
$$Wopt(i, j) = (A^{(n)})_{ij}$$

で与えられる。$(A^{(n)})_{ij}$の値を求めるには、順次、行列の積$A^2, A^3\cdots, A^2$を求めて、その和

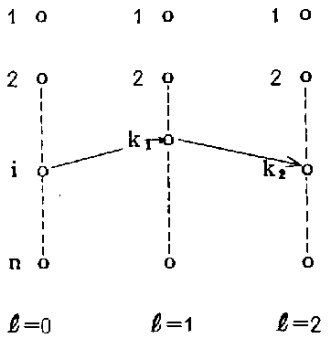
を取れば得られるが、PRシステムでは演算回数を減少させるために、Warshallのアルゴリズムの一般化を利用している。路線の探索は、つぎのようにして行なわれる。

出発分岐点iに隣接し、しかもその行き易さの値が、$(A^{(n)})_{ij}$ よりも大きいかあるいはその値に等しい基本道路 P_{ik} を求める。このためには分岐点番号の小さいものより順にその値 w_{ik_1} と $(A^{(n)})_{ij}$ の比較を行なっていけばよい。$w_{ik_1} \geq (A^{(n)})_{ij}$ なる基本道路 P_{ik_1} が得られれば、PATH(1) $:= k_1$ とし、k_1 と jとの比較を行なう。もし $k_1 = j$ ならば、長さ1の最適路線が求まった。続いて $k_1 < k'_1$ なる基本道路 $P_{ik'_1}$ について $w_{ik'_1}$ を比較する。また、$k_1 \nless j$ ならば、k_1 に隣接し、$(A^{(n)})_{ij} \leq W_{k_1k_2}$ なる基本 Pk_1k_2 を探索する。

以上の様にして分岐点の系列が、



図Ⅱ−13 長さ1の最適路線



図Ⅱ−14 $k_1 \nless j$ の場合

PATH$:=h_m$ $(1 \leq m \leq n-1)$ まで決定したとする。つぎに分岐点Kmに隣接しその基本道路の行き易さの値が $W_{k_mk_{m+1}} \geq (A^{(n)})_{ij}$ となる分岐点 K_{m+1} を探す。K_{m+1} が得られればその K_{m+1} がそれまで通過して来た分岐点 PATH(0), PATH(1), …, PATH(m) に一致しているかどうかを調べる。もし一致していなければ分岐点 K_{m+1} を PATH $(m+1) := K_{m+1}$ として格納する。K_{m+1} がj に等しいかを調べ、もし $K_{m+1} = j$ ならば、長 $m+1$ の最適路線がみつかったことになる。$K_{m+1} \nless j$ ならば、上の操作を繰返す。

また、もし、K_{m+1} がすでに通過した分岐点のいずれかに等しければ、$K_{m+1} \leq K'_{m+1}$ なる基本道路を調べる。$K_{m+1} = n$ となれば、$K_{m+1} < K'_{m+1}$ なる分岐点 K'_{m+1} は存在しないので $(m-1)$ 番目の分岐点までもどり、m番目の分岐点については、$K_m < K'_m$ なる分

岐点K′mより調べればよい。

これらの操作をくりかえして行ない、m≦n，$K_n = n$ となれば（n－2）番目の分岐点までもどり、（n－1）番目に通過した分岐点をK_{m-1} とすると、$K_{n-1} < K'_{n-1}$ なる基本道路Pk_{n-1} K'_{n-1}の選択を行なう。

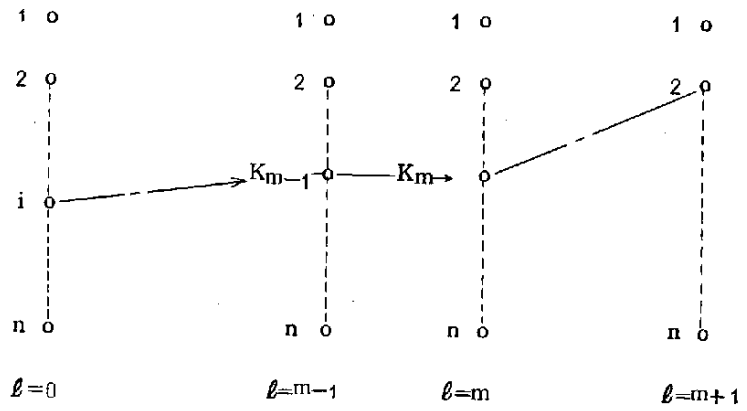


図　Ⅱ－15

以上の操作をくり返して行なってゆくと、最後にk_1の探索にもどる。$k_1 \leq n$ なる分岐点への基本道路$Pi\ k_1$ がすべて調べつくされたとき、この探索アルゴリズムは終了する。

このアルゴリズムに従うと、分岐点iより分岐点jまで到達する最適路線が決定される。

5.8 むすび

PRシステムと呼ぶ道路網等に関する情報検索システムについて、質問、あるいは、最適路線決定のアルゴリズム等について簡単に述べた。

質問の充実化についてもそれ程問題はなく、文中のCF文法の拡張で処理できると考えられるし、それに附随して最短路線、最少費用路線等用のアルゴリズムを格納すればよいと考えられる。

6 結　論

Ⅱでは、質問形式の分類とそのあるべき姿、また質問分析の方法とこれに関連した日本語の

とり扱いの一例について述べた。その他格納情報、質問文の双方に形式言語を設定し情報の演繹を許したシステムについて、ついでより対象に密着した形でのシステムの構成等について述べた。

現在の時点においてFact Retrieval Systemの構成はCase Study 的にならざるを得ないと考えられる。

Ⅲ （意味の位相性を使用した自然語応答システムの解析）

Ⅲ （意味の位相性を使用した自然語応答システムの解析）

1 緒　論

現在計算機で非数値の情報（例えば、概念、意味の遠近、包含関係など）を表現し、処理を行なう場合、分類表、対照表、類語集など、関係のあるものを列挙したリストによっていることが多い。しかしこういう方法は意味的な取扱いにはあまり適しておらず、膨大な記憶と、長い処理時間を要し、性能のよいQ－Aシステムを構成することが困難である。

そこで、意味の位相的性質に着目し、頭の中にあると考えられる意味空間を模型的に作り、それを使用することにより、よりきめ細かい処理の行なえるシステムを考察検討した。

2では、意味の位相性と、その性質を表現する意味空間について、更にその性質を用いての人間－機械系のＭＡＣ使用の為の、会話応答システムについて、その構成と、その長所、短所について論じた。

3においては、日本語の言語学的性質と、機械処理上考慮しなければならない性質の解析を行ない、次に意味の位相的性質を使用したシステムに基ずく、意味理解、解釈の実験を行なった。これは日本語の意味空間の代数的性質をもとに、単語を要素とする意味空間としてのシソーラス、そして単語を組合わせて出来る表題を、意味空間内にその遠近関係を満すように配置することにより論文の意味空間を構成し、論文の概念抽出実験を行なったものでその結果を述べる。

4においては、位相型に表現されたデータをもとに、自然語またはそれに近い応答をなす処理システムの代数的性質、意味の解読過程を明らかにし、次にこのような意味空間を作成する方法について、その最良のものを実験結果から解明した。

また、これらをもとに、人間－機械系の会話応答過程を分析し、命令・質問・応答等のシステムへの入力に対し、内部処理について応答する方法を解明した。

2　位相型自然語応答システムの意義

現在計算機で、数値でない情報・概念、ことばの取扱いは、辞書編集順に単語を並べたり、何等かの分類表に従ってデータを類別し、グループ毎に列挙したファイルを用いて行なっている。このような方法は、その分類体系からみると樹枝状のデータ構造をもつ情報処理システムといえる。しかしこのような構造では、概念の同一のものの選択とか、ある概念と包含関係にある概念の抽出とか、意味的に近い概念とか、ある条件を満たす概念、合成概念などを表現したり，求めたりする、いわゆる高度の情報処理には不十分である。そこでこの欠点を補うシステムとして、位相型のデータ構造をもつ情報処理システムを提案する。なお樹枝状と、位相型の表現法の得失については、Ⅳ．3で述べる。

2．1　位相型データ構造

2．1．1　意味空間

位相型データ構造とは、情報（ことばの表わす概念）の間の意味的な各種の関係（遠近関係、包含関係、概念の合成則など)が満たされていて、その上で種々の意味的な表現や処理を行なえるデータ構造を意図している。

ファイル構成は後に述べるとして、このようなデータ構造は、次のような意味空間の上で論ずればよい。

〔定義２1.1〕　意味空間　$I=(S, U)$

ここで、Ｓは意味空間を構成する集合、Ｕはその上での近傍系である。

〔定義２1.2〕　情報処理　Ⓗ　$I^m \rightarrow I$

ここで、$m=1, 2, \cdots$ 上限は無限迄存在するが普通は、$m=1, 2$ が殆どである。

この意味空間を計算機に取扱わせようとすれば、近傍系だけでは、取扱いにくいので、定義2.1.1に「近傍系は分離公理も満たす」という条件を付加して、距離空間として取扱う。

その場合、Ｓは一般に R^n で表わされる。ここでＲは実数で、ｎを意味空間の次元という。

またこの時情報処理は、Ⓗ；$(2^R)^m \rightarrow 2^{R^n}$ となる。

2．1．2　意味空間構成の為のデータ

このような意味空間構成の為のデータは、次のような方法により得ることができる。

(1) 各言葉（単語、句、節、文、文章など）の間の意味的な遠近、包含関係を色々な判定基準に従い、アンケートなどで、計量心理学、計量社会学的方法によって統計的に求める。また特定の判定基準ではなく、全般的な立場から同様に遠近、包含関係を統計的に求める。

(2) 現用中の各種シソーラス（類語集）、辞書、辞典、各種体系分類表、各種階層分類表などから、意味的に近いもの、正反対のもの、同系統のもの、包含関係、上位、下位関係などを求める。分類表はⅣ3．1参照。

(3) 各種の体系分類表、階層分類表などから、ある判定基準に従った概念（ことば）の配列順を求める。

2．1．3　意味空間の構成

以上のような方法によって得られたデータをもとに、意味空間の構成を次のように行なう。

(1) 遠近関係（類似度）の順序を崩さないで、出来るだけ低い次元の空間内に、ことば（概念）を配置する。

(2) 遠近関係（類似度）の比の値を崩さないで、出来るだけ低い次元の空間内に、ことば（概念）を配置する。

(3) 因子分析を行ない主要因を抽出、各要因を直交または斜交する座標軸としての空間を構成する。

(4) 同一対象物の体系分類が異なった分類基準によるものが幾種類かある場合、それらを各座標軸として、ファセット分類、コロン分類的な空間を構成する。

以上のような諸手法により多次元意味空間を構成する。(4)の場合は、因子分析等により空間を圧縮しないと、各座標軸は意味的に直交しているとは限らない。

2．1．4　意味空間ファイルの構成

統計的手法または、関係のあるものの列挙法、分類法などにより、前節の諸手法を用いて意味空間が構成されると、次はそれを使いやすいように表現するーファイル構成が問題

となる。その構成法には次のようなものがある。

(1) ハード（金物）で、抵抗固体回路を作り、概念の存在すべき各位置にプローブをつけて意味空間を構成する。この場合、距離は抵抗値の大小で表わされる。この方法は三次元の意味空間迄なら実現出来るが、それ以上の次元だと、そのままでは物理的に不可能となる。

固体回路だと、内部からのプローブが出しにくいので、抵抗回路網にする方法もある。このようにすれば、四次元以上の意味空間も抵抗回路網の配線の仕方次第で可能になる。しかし演算速度をあげてゆくと、そのリード線の差が、伝播時間に効いてきて、高速処理はある限界でおさえられる。このような方法で構成したものが、ハードの連想記憶である。

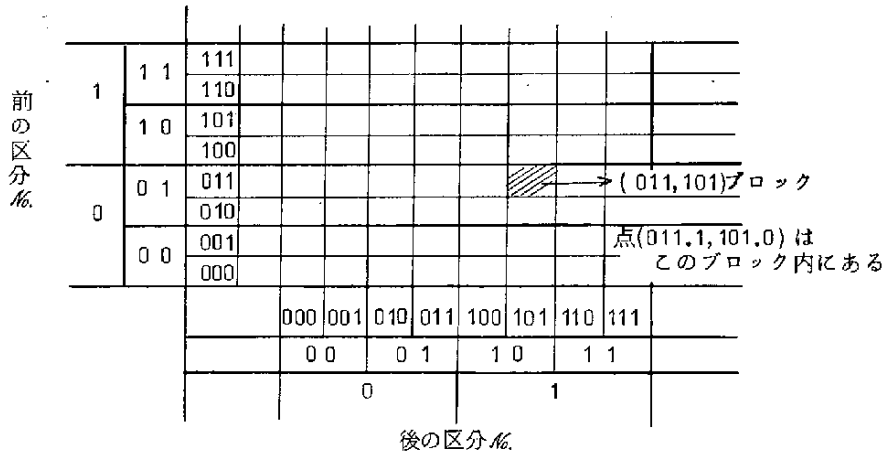
抵抗固体回路がら、抵抗回路網へ表現しなおすと、概念の存在点が不連続になる為に、ファイルの維持を行なうときに、抵抗を入れたり、取ったり手数が複雑になる。

これを防ぐ方法として、碁盤の目状の多次元メッシュ抵抗回路網を作り、該当点にリードを接続するようにすればよい。

(2) 抵抗回路網と、枝に重みのついたグラフ表現するもので、接続行列（IncidenCe Matrix）の形でファイルとして持つ。このような表現は、遠近関係は枝の重みを多段階にすれば断続の二項よりは細かい表現が可能であるが、距離空間の中に概念を配置するのではなく、点を長さの異なる枝で結んだことになり、近傍空間の良さを十分発揮できない。ファイルの維持更新も、あまりやりよいとはいえない。

(3) 空間内に配置した各座標値をそのまま持つ方法。この場合近くにある概念（座標値）ばかり集めてファイルする。この空間を幾つかの部分空間、例えば各象眼毎に分割し。その中にあるものは概念点をまとめてファイルする訳である。

この方法は空間を、多次元体系的ファセット分類し、概念点は各座標値表示をしたことになる。このファイルのアクセスには、質問点の座標が入るファセット（ブロック）をランダムアクセスで選び、その中の点の座標値を求める方法で、このファイル構成を図Ⅲ－1に示す。



例．2次元意味空間（意味空間を2次元3層構造に分割）例えば電子計算機の座標が、（０１１．００１，１０１．１０）であるなら、それに近い概念を求めるには、ブロック（０１１，１０１）の部分意味空間へアクセスし、その部分空間内にあるデータ点との距離を計算する。検索をするのにファイル全部を走査する必要はない。

点Ａ（０１１．１，１０１．０）

点Ａを含むブロック：Ｂ（０１１，１０１）

ブロックＢと距離１のブロック：（０１０，１０１）（０１１，１１０）

（１００，１０１）（０１１，１００）

ブロックＢと距離２のブロック：（１０１，１０１）（０１１，１１０）

（００１，１０１）（０１１，０１１）

（０１０，１１０）（１００，１１０）

（０１０，１００）（１００，１００）

次第に点Ａより離れてゆく。

図Ⅲ－１　意味空間の File 構成

２．１．５　意味空間の使用法

意味空間の使用法としては、次のような2方法がある。

(1) 意味空間をシソーラスとして使う方法

この場合使用時により更に2分される。

(i) データファイル作成時に、意味空間を参照し、意味的な遠近関係をとらえたり、意味空間の構造（算法）をもとに、データファイル（ことばや、論文題目など）を作成する。抄録用に使用するといってよい。

(ii) 使用時（検索時）に意味空間をシソーラスとして使用する。これは質問分析用に使用することである。

また抄録時、検索時とも使う方法もある。

(2) データ、そのものを意味空間に配置してしまう方法。ここでデータとは、論文検索の場合だと、各論文がデータとなる。

なおこのとき、意味空間（ことば）と、データの意味空間を重ねあわして使う方法を、別個にしておいて (1) の方法で意味空間（ことば用＝シソーラス）を使う方法とある。

(1)の使い方をやる場合には、データは必ずしも位相型データ構造をとってなくてもよく、樹枝状データ構造をしたデータに対して、位相型シソーラスを使ってひくこともできる。

2．1．6　自然語応答システムへの利用

自然語を使った応答システムでは、非数値情報＝概念情報の解釈が自由に行なえることが大切である。即ちシステムへの自然語入力の意味解釈，そして必要な意味処理、結果と使用者に出力するときの文構成に必要である。そしてこれらのことを行なうには、データ間の関係を表現している意味空間と、その中での意味処理をとり扱う代数系があればよい。

会話，応答過程の分析については4．4で、樹枝状データ構造と比較した得失については Ⅳ-3 で述べる。

空間内での意味処理については、3.2、4.2でのべる。

意味の包含関係も論じようとするならば、定義2.1.1のSを Fuzzy 集合にとればよい。

2．1．7　位相型自然語応答システムの意義

概念などというものは、完全な体系分類とすることは不可能であり、それを強引に分類すると、同一概念が違う分野の中に表われたりする。また概念同志の関連性にしても、有無の二値では割切れないのが普通である。

その点樹枝状データ構造のファイルで意味処理をすると、さがしている概念が二つに分

かれていて。その一方を落とすとか、関連したものがあるにも拘ず、そこへ辿る枝がない為にたどれないということが起きる。例えば KWIC 表現されたリストから、ある事項の載っている論文をさがそうとするとき、その事項の専門用語（キーワード）を全部いえない人には、シソーラスで専門用語をひいてさがさねばならず、その時一つも専門用語を知らないなら、シソーラスも使えない。またシソーラスを使っても、ひいた語の関連語以外は、さがそうにもさがせない。

位相型データー構造ファイルは、この欠点を補ったところに存在意義があり、意味の近いものはシソーラスをひくことにより求まるので、検索にも、同概念の別表現でも有効に役立つ。

3　論文概念抽出実験

3．1　日本語の言語学的性質の解析

3．1．1　日本語の機械処理上の問題点

高速大容量の電子計算機が発達し、広い意味での情報処理といわれる分野に使用され始めると共に、取扱う対象としての自然語、及びマン・マシン・システムの通信手段としての自然語の持つ重要性が認識されて来た。

それと同時に、今まで気がつかなかった言語処理上の本質的な問題点が明らかになって来た様に思われる。この様な言語処理の一般的な困難さと共に、自然語として日本語を用いるという実用的な意味に於て、英語等とは違った意味での処理上の難点がいくつか考えられる。

(1) 字数の多い事。日本語の処理に於て最も問題となるのは、この点である。日本文に現われる字種を考えてみると、漢字、平かな、片カナ、英文字、欧字（ギリシヤ文字等）数字、記号等 をあげる事ができる。特に漢字はその数が非常に多く（新漢和大字典によると１４２７０字である。）しかも、同音異議語が多い等の表記上の問題点も多い。しかしながら、現在では昭和２１年に告示された当用漢字（１８５０字）の枠内で使用するたてまえになっており、これに若干の使用頻度の高い表外漢字及び一般的な人名、地名等の個有名詞の漢字を考慮に入れても、字数としては３０００字を一応の上限とみてさしつかえないと言われている。この様な字数の多さが問題になるのは、入出力装置の点である。とり扱い得る字数が多く、しかも高速度、高品位の入出力装置の開発が急務である。漢字を取り扱える入力装置として現在使用されているのは、漢字テレタイプである。マン・マシン応答システムで使用する場合は入力データは少量であるから、文をまずカナで入力し、応答モードの中で漢字を決定し入力する方法もある。又漢字をコード化するなど様々な方法が考えられるが、最終的に人間とマッチングのとれた大量の入力が可能となる為には、Ｏ．Ｃ．Ｒ．（光学的文字読取装置）の開発がまたれる所である。しかし、この様な、図形も含めて文字のパターン認識の問題は非常に困難であり早急に解決されるとも思われない。

出力装置について言えば、漢字テレプリンタ、写植機が実用に供されているが、いずれも２～５ CPSと速度が遅い。計算機の出力として十分な速度と、印刷物としての品質の高い電子式の高速プリンタあるいはデイスプレイ装置が現在開発されつつある。

(2) カナ書きに伴なう問題点

現在、計算機の和文入力としては、カナ書き文がほとんどである。又近い将来に於ても、Mon－MaChine 応答システムに於ける人間側からの入力は、カナ書き文に頼らざるを得ないものと思われる。ところが、日本語では字音語（普通、漢字のみで書かれる語彙）の割合が、３０～４０％もあり、これを表音文字のみで綴った場合、明らかに読み取り難くなる。その理由としては、

i） 単語の切れ目が判別し難くなる。

ii） 同音異義語の区別がつかない。

iii） 外来語のカナ表現が一意的でない場合がある。

この３つがあげられる。

(i)について考えれば、我々は普通、漢字かなまじり文に於ては、漢字とかなの字くばりによって視覚的に単語を区別するのは容易であるし、又話し言葉に於ては、適当な区切りとアクセントの規則によって単語の区切りを行なっている事が知られている。解決策は、文の「分かち書き」であるが、問題は、この分かち書きの規則を定める事が困難な点にある。又複合語、外来語を表現する場合に中黒（中点）を入れるが、この入れ方の規則も同時に問題となる。

分かち書きの方針としては、次の３つを採用する事が、最も適当である様に思える。

a） 自立語は分かち書きをしないが、複合語である場合には、中点を打つ事ができる。

b） 付属語はすべて分離して書く。

c） 付属語で分割法のわからないものは、あまり分割しない。

ここで日本語の文法を幾つか比べてみると、単語の区切り方が学者によってずいぶんとまちまちであるのに気づく。例えば、「行カセタカラ」という語句をとると、

松下大三郎博士の文法では、全体で長い一語となり、

山田孝雄博士の文法では、「行カセタ＋カラ」と切り２語とする。

時枝誠記博士の文法では、「行カセ＋タ＋カラ」となる。

中点の例

ドウリヨクロ	ドウリヨク・ロ	動力炉
ヒコウキ	ヒコウ・キ	飛行機
ニホンゴ	ニホン・ゴ	日本語
ジョウホウショリ	ジョウホウ・ショリ	情報処理

カナ書き文による分かち書き

（もとの文）「付属語はすべて分離して書く」

（分かち書きをしないカナ書き文）「フゾクゴハスベテブンリシテカク」

（文節で区切った分かち書き）「フゾクゴハ　スベテ　ブンリシテ　カク」

（付属語を分離した分かち書き）

「フゾクゴ　ハ　スベテ　ブンリ　シ　テ　カク」

問題点（ii）の同音異義語については、話し言葉の場合と同じである。機械に前後の文脈の意味内容を判断させ異義語の中から選択させなければならない。同音異義語の問題は言語に本質的なものであるが、日本語の場合、漢字の音の少ない不合理さが災となって、その音韻表現がひどく不合理なものとなっている。一例を示すと、

「キシヤ　ノ　キシヤ　ガ　キシヤ　デ　キシヤ　シタ。」

（貴社の記者が汽車で帰社した。）

これの対策は、つとめて漢語よりも大和ことばを使う様に心がける事である。

問題点（iii）については、例えば、ヴォルトと　ボルト、といった様なものである。これに対しては、標準語の対比できるシソーラスを作成しておけば良いし、特に技術文献等では、英文まじりカナ文にする事も考えられる。

3．1．2　日本語の文法的特徴

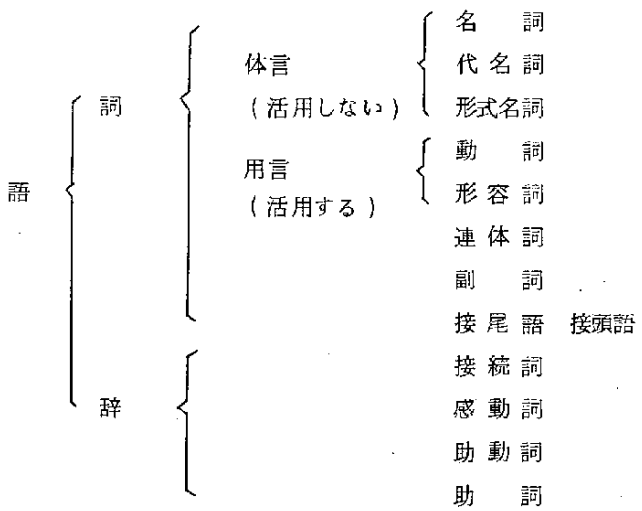
日本語の特徴というものを発音の面、語彙の面等、様々な面からとらえる事ができるが、ここでは文構成の立場から特に英語と違った点について簡単に述べる。

(1) 品詞分類

日本語の文法が、言葉の系統のちがうヨーロッパ語の文法の見方では処理しきれない

事は当然であるが、近代の日本語文法はまず、ヨーロッパ語の文法を下じきとして出発した事は事実の様である。そして、この様な直輸入では間に合わないという事から近年日本語に適した文法論が展開されて来た。

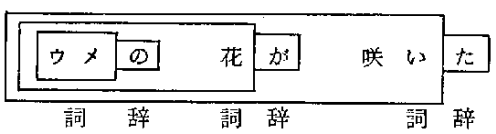
構文分析はまず単語の区切りを決め、品詞決定をした上で行なうものであるから、品詞分類はきわめて重要なものであるといえる。ここでは、時枝文法による品詞分類を示す。



ここで、辞とは概念化の過程を経ない語であり、詞とは概念化の過程を経た語であるといえる。例えば、「花だ」という場合には花という客観化された内容を表わす語（詞）に，話し手の断定の気持を表わす「だ」という語（辞）がついて1つの文ができあがる事になる。

(2) 文の構成

詞と辞の区別といった点から日本語を見る場合これは日本語文法の特徴的な点であるといえる。先の「花だ」という例に示した様に、日本語の文は辞によって詞が包みこまれてゆくところの、いわゆる「入れ子式」の文法構成になっていると言える。例を示すと、



この文構成の形を形式言語でよく使用される形で表示すると次の様になる。

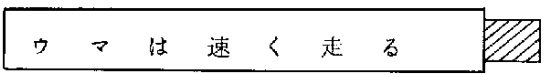
<詞>：＝<体言>｜<用言>｜<連体詞>｜<副詞>｜<接頭語>｜<接尾語>

<辞>：＝<助動詞>｜<助詞>｜<感動詞>｜<接続詞>｜<零記号>

<詞>：＝<詞>＋<詞>｜<詞>＋<辞>

<文>：＝<詞>

ここで、記号（：＝）は、右辺のものを左辺でおきかえる意味であり、右辺の縦棒は「ＯＲ」の意味である。又、零記号の辞というのは陳述を表わすものであり、例を示すと、



詞　　辞

となる。

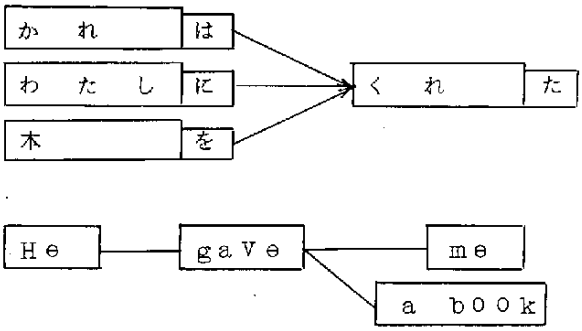
ここに示された様に入れ子式の文法構成を持つということは、明らかに向構造文法としての取り扱いができる事を示している。

この事は又、言いかえれば、計算機言語のＡＬＧＯＬと同じ、Ｃ．Ｆ．Ｌ．としての取り扱いができる事を意味している。この事は、計算機で取り扱う為には、有利な性質である。又、Ｃ．Ｆ．Ｌ。は逐次処理が可能であるから、会話用言語に適した性質であると言える。

ここで、文中にあらわれる詞相互間の関係を考えると、次の様な特徴が存在する。

ａ）文の基本は述語格である。

「だ」、「です」あるいは零記号で示される陳述の助動詞によって統一された詞は述語格と呼ばれるが、日本語に於ては、文の中心は述語格であり一般に主語、目的語、補語と呼ばれる語句は、述語の内容をより詳しく規定する為の一種の修飾語格にすぎない。しかも、この修飾語格相互は全く対等の立場にある。この点から言えば、主語は必要に応じて現われるのであって、主語のない文も、元来あるべきものが省略されたのではない事になる。この点英語とは非常に異なっている。



b）述語格は文の最後に現われる。

C）修飾語は常に被修飾語の前に来る。

すなわち、Aの語句がBの語句に従属しているとすれば、Aは常にBの先に立つ。これは日本語の語句の並べ方における統一した大きな法則である。例えば「白い花」という語句に於て、「白い」は「花」に従属している。又、「咲いて いる」の様な場合にも文法上は、「咲いて」は、「いる」に従属している。なぜなら、「咲いている」という語句が、どの様な言葉に続いていくかは、ひとえに「いる」によって決まるからである。

d）対象語格

英語等に見られぬ便利な表現として、「日本は 山が 多い」 の種があるが、この場合の「は」は主語を表わすのではなく、対象語格とでも言うべき性質のものである。

3．2 文章型シソーラスの構成

3．2．1 意 味 空 間

遠隔制御会話応答システムにおいて、MAC使用を有効に行なう為には、4.2でも述べるように、その使用言語は、出来るだけ自然語に近い方が望ましい。その為には、命令、会話に表われる単語の意味論的な解釈を行なう為のシソーラス、あるいは辞書が必要となる。従来のシソーラスは、離散的な系において意味解釈を行なって来たが、ここでは、もっと能力の高いシソーラスを考え、次のように定義する。すなわち、「シソーラスとは、各言葉の概念の位相関係を表現する意味空間である。」と定義し、意味論的解釈は、会話

文をその意味空間に写像することであると考える。

このシソーラス＝意味空間という考えは、現在の常識とは、かなりかけ離れた考えのようにみえるが、連続系で面的なアプローチにより意味解釈を行なっており、従来のシソーラスをも包含した形で、より能力の高い意味解釈に適している。

3．2．2　意味空間の構成実験

(1) 構成法　4．1 で述べるように、意味空間は$I=\{S\omega\}$なる近傍空間であり、ωは近傍系であるが今の場合これを距離空間と考えてもよい。言葉の間の意味の遠近関係を定める方法には、4．3 で述べるようなものがあるが、ここでは、アンケートにより人間の頭脳から抽出する方法を採用する。意味という極めて主観的なものを客観的かつ定量的に規定する方法には

1) Semantic Differential Method　（S－D法）

2) Non - metric Factor Analysis　（非計量要因解析法）

3) Puoximity Analysis　（類似度解析法）

などがあり、主として行動科学、計量心理学の方面で研究されて来た。これらの方法の相違は得られたデータがどのような種類のものかによって生ずる。ここでは類似度解析法を用いて意味空間の構成を行なう。

類似度解析法には (イ) 1952 年に W．S．Torgerson により提案された方法と (ロ) 1962 年に R．N．Shepard により提案された後に J．B．Kvuskal により改良された方法とがある。(イ)は、空間内の元の間の距離はそれらの間へ類似度に比例するという制限を設けている。(ロ)はその制限をゆるめて、それらの間の単調性を、保持すればよいとしている。これらの方法は、主として音声や色調関係といった物理的な刺戟の間の類似度をもとにして空間を構成するのに用いられていたが、ここでは言葉の間の意味の遠近関係による空間の構成といった新しい分野への応用を試みている。そのために、ここでは (ロ)の方法についてパラメータの、変更や一部の式に修正を加えて、意味空間の構成に適した類似度解析法を用いてみる。

・　類似度解析法

一般に、n個の点をそれらの間の$n(n-1)/2$個の距離関係を満足させながら多次

元空間内に配置するには、（n－1）次元の空間が必要である。しかし情報空間においては、情報間の絶対的な距離よりはむしろ、距離の大小関係の方が問題となる。そこで距離関係を保持すればよいという条件にゆるめ、のちに定義する歪を空間の再現度の尺度と考え、次元数を変えた場合の歪の変化から、最も適当な次元数を定めようというのがこの方法の基本的な考えである。

n個の情報 1,……n と、それらの間の類似度 δ_{ij} が与えられると、歪は次式によって定義される。

$$S = \sqrt{\frac{S^*}{T^*}} = \sqrt{\frac{\Sigma\,(dij - \hat{dij})^2}{\Sigma dij^2}} \qquad (3.2.1)$$

ここで、dij は情報 X_i と X_j の間の距離で、次式によって与えられる。

$$dij = \left[\sum_{l=1}^{t} | Xie - Xje |^r\right]^{\frac{1}{r}} \qquad (3.2.2)$$

又 $\hat{dij}$ の値は、$\hat{dij}$ が δij と同じ順序でなければならないという制限の下でSを最小とするような数である。詳しくいえば、$\delta ij < \delta_{kl}$ なら $\hat{dij} \leq \hat{d_{kl}}$ となるような制限である。

（3.2.1）式から明らかなように、Sは、nt 個の変数の関数として次のように表わされる。

$$S = S\,(X_{11}, X_{12}, \cdots\cdots X_{it} \cdots\cdots X_{n1}, \cdots\cdots X_{nt}) \qquad (3.2.3)$$

ただし、Xij は情報iのj座標

このSを最小にするには、最大傾斜法を用いればよい。すなわち、任意の配置からはじめて、傾斜を計算し、それに沿ってデータを適当な距離だけ動かす。この操作を繰り返して、適当な情報空間を得る。この方法について、もう少し詳しく説明すると

歪は相似変換の下では不変であるから、空間の配置を、その中心が原点にあり半径が1となるような超球面上に分布させておく。

さて、t次元空間において、n個の点X_1，……X_n より構成される空間の配置Xが得られたと仮定しよう。X_i の座標を X_{i1}，…，X_{it} とする。すべての X_{is}（i＝1，……n，S＝1，……t）を配置Xの座標と呼ぶ。Xにおける勾配をgによって与え、その座標を g_{is} とする。Xをステップサイズαに比例する距離だけ、gに沿って動かすことによって新しい配置を得る。この新しい配置X^1 の座標は次式によって与えられる。

$$X^1_{is} = X_{is} + \frac{\alpha}{mag(g)} \cdot g_{is} \qquad (3.2.4)$$

mag(g) は、gの相対的な大きさを表わし、次式で与えられる。

$$mag(g) = \sqrt{\sum_{i,s} g^2_{is}} \Big/ \sqrt{\sum_{i,s} X_{is}{}^2} \qquad (3.2.5)$$

$$g_{kl} = S\sum_{ij} (\delta^{ki} - \delta^{ki}) \left[\frac{\hat{d}ij - \alpha ij}{S^*} - \frac{dij}{T^*} \right]$$

$$\times \frac{|Xie - Xje|^{r-1}}{\alpha ij^{r-1}} \; Signvm(xil - xjl)$$

ステップサイズαは、各繰り返しにおいて変化する。その値は、次のように与えられる。

$$\alpha present = \alpha previous \cdot (angle\ factor) \cdot (relaxation\text{-}factor) \cdot (good\ lock\ factor)$$

ここで、$angle\ factor = 2.0^{(\cos\theta)^{3.0}}$ （3.2.6）

θ ：現在の勾配と前の勾配の間の角度

$$\text{relaxation factor} = \frac{1.3}{1+(5-\text{Stepratio})^{5.0}}$$

$$5-\text{Stepratio} = \min\left[1, \left(\frac{\text{現在の歪}}{\text{5回前の歪}}\right)\right]$$

$$\text{good lock factor} = \min\left[1, \left(\frac{\text{現在の歪}}{\text{前の歪}}\right)\right]$$

gを現在の勾配、g″を前の勾配とすれば、

$$\cos\theta = \frac{\sum_{i.s} g_{is}\, g''_{is}}{\sqrt{\sum_{t.s} g^2_{is}}\ \sqrt{\sum_{t.s} g''_{is}}}$$

このようにして、繰り返し計算を進めて行くと、最小の歪が得られる。

(2) 空間構造解析実験

4.2のモデルに基づき、単語、句レベルの意味空間を構成し、その空間中での処理算法を抽出する実験を行なった。対象としたのは、論文題目によく表われる連語という合成法である。以下その実験アルゴリズム及び結果について述べる。

(i) 実験方法

図(Ⅲ－2)(Ⅲ－3)に実験アルゴリズムを示す。連語、という算法を(3.2.7)のように仮定する。

$$\mathbf{x}_3 = A\,\mathbf{x}_1 + B\,\mathbf{x}_2 + \ell \qquad (3.2.7)$$

$\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$ は単語A、B、及びそれから作られる連語ABの座標を表わすt次元ベクトルである。ここでA、Bはテンソルであるが、次元軸相互間の意味的な直交性が保たれているものと仮定し対角成分以外は0であるとする。したがって(3.2.7)式は次の様になる。

$$\begin{bmatrix} X_{31} \\ X_{32} \\ \vdots \\ X_{3t} \end{bmatrix} = \begin{bmatrix} a_{11} & & 0 \\ & a_{ii} & \\ 0 & & a_{tt} \end{bmatrix} \times \begin{bmatrix} X_{11} \\ X_{12} \\ \vdots \\ X_{1t} \end{bmatrix} + \begin{bmatrix} b_{11} & & 0 \\ & b_{ii} & \\ 0 & & b_{tt} \end{bmatrix} \times \begin{bmatrix} X_{21} \\ X_{22} \\ \vdots \\ X_{2t} \end{bmatrix} \qquad (3.2.7)$$

(3. 2. 3)式は、次の様なt個の独立な回帰模型に分解出来る。

$$\left.\begin{aligned} X_{31} &= a_{11}\ X_{11} + b_{11}\ X_{21} \\ X_{32} &= a_{22}\ X_{12} + b_{22}\ X_{22} \\ &\vdots \\ X_{3t} &= a_{tt}\ X_{1t} + b_{tt}\ X_{1t} \end{aligned}\right\} \qquad (3.2.8)$$

上のような構造模型について母数推定を行なうのであるが、ここでは次の3つの型に分けて母数を回帰する。

(イ) 軸毎に回帰をとったもの。

(ロ) 単一の連語データについて回帰をとったもの。

(ハ) 以上全部の組み合わせについて回帰をとったもの。

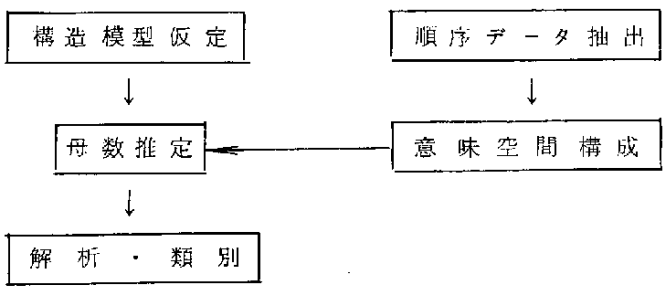


図 Ⅲ－2 文章型シソーラス構成アルゴリズム

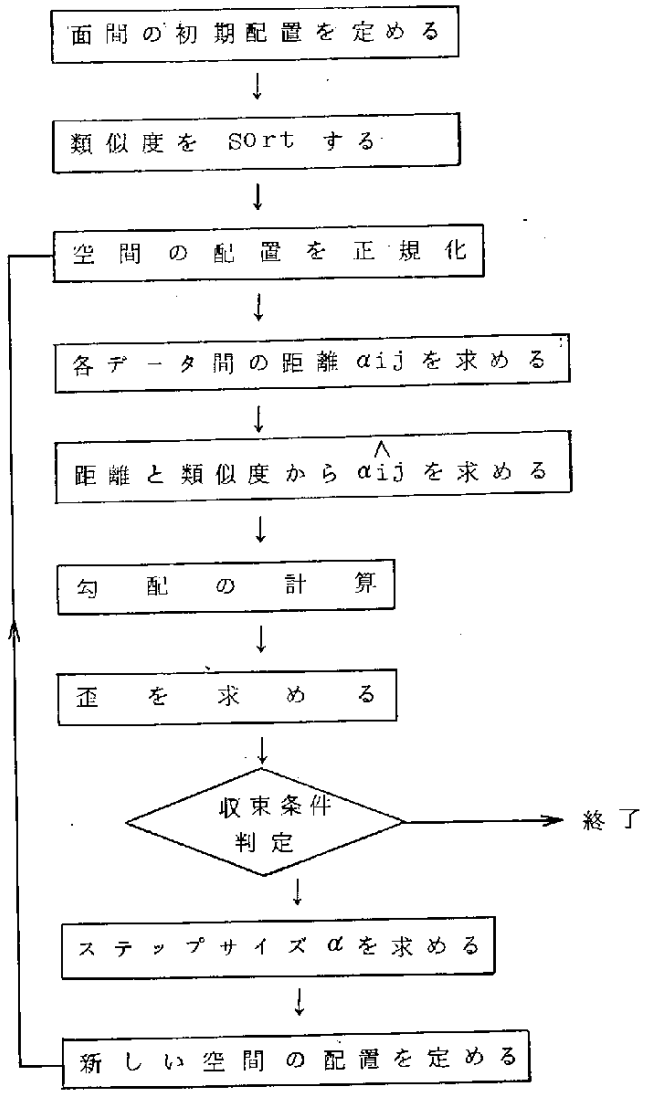


図 Ⅲ－3　意味空間構成アルゴリズム

(ii) 実験結果

電子通信学会誌より単語及び連語を選びそれらの類似度を求めて意味空間を構成した。その際の次元数と歪の関係を図Ⅲ－4に示す。又、母数の回帰の結果を図Ⅲ－5、図Ⅲ－6に示す。

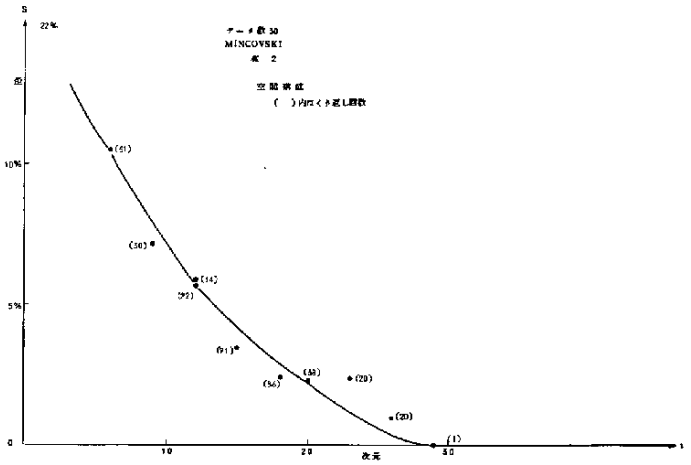
a，b，$^{a}/_{b}$の分布は、次のような性質をもつことが明らかになり、それによって

「連語」という演算は、（3.2.9）のような構造を持つことが明らかになった。

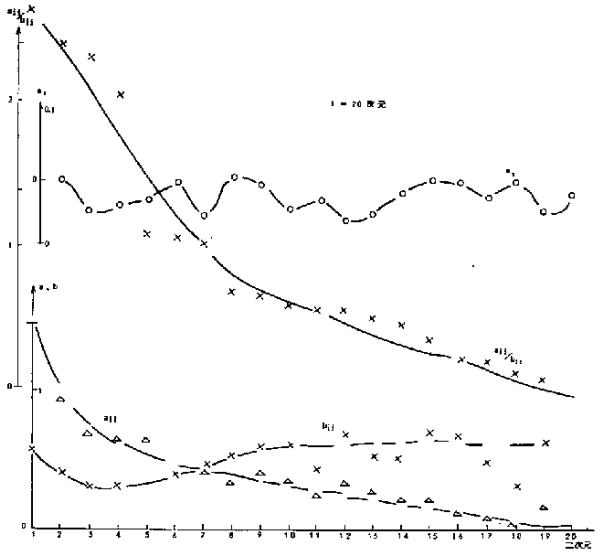
(1) a/bのは大部分が1より小である。このことは、連語においては、一般に後の単語が意味的に重みを持っていることを示している。

(2) a/bの分布は、3つの類に類別出来る。このことは「連語」という算法に3つの種類があることを示している。

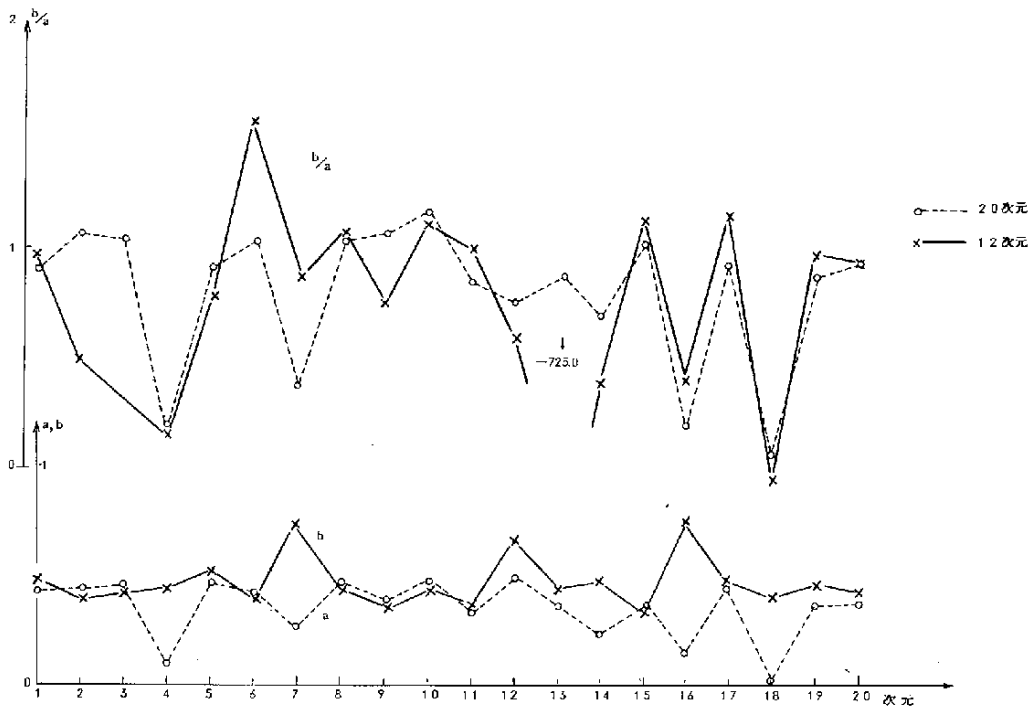
(3) aの分布は、a/bの値に従って1より大なる所から0近くまで単調に減少しており、bの値は次元軸の変化にかかわらず、ほぼ一定の値を持つ。



図Ⅲ－4　意味空間における二次元数と歪の関係



図Ⅲ－5　二次元数毎に母数を回帰した結果



図Ⅲ－6　連語毎に母数を回帰した結果

以上のことから、連語という演算は、次の様な構造を持つと推論出来る。

$$X_{3i} = a_{ii} X_{1i} + b_{ii} X_{2i}$$

$$= b_{ii} (\lambda X_{1i} + X_{2i}) \qquad (3.2.9)$$

ここでは、単語レベルの意味空間における「連語」という算法を導き出す実験を行なった。この方法は、句、節、文のレベルの意味空間における、前置詞や接続詞などによる句、節、文などの結合を与える算法を求める場合にも適用出来る。

参 考 文 献

o R.U.Shepard,"Metric Structure in Ordinal Data"J.of Mathematical Psychology,3,287~315(1966)

o J.B.Kroscal "Moltidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis" Psychometrika - Vol,29.No 1,March,1964

o J.B.Kroskal "Nonmetvic Moltidimensional Scaling, A Nomerical Methad." Psychometrika - Vol,29,No 2, Jone,1964

o R.N.Shepavd,"The analysis of Rvoximitiesi Moltidimensional Scaling with an Unknown Distance Fonction L," Psychometrika - Vol,27,No 2,Jone,1962

3.3 論文概念抽出実験

従来、論文の概念を表わすには、階層分類にしたがい分類コードを与えたり、限定されたKey wordの作る属性空間内に写像したりする方法がとられてきた。ここでは、論文のもつ概念の間の遠近関係に基づき、前節で述べた類似度解析法を用いて概念空間を構成し、その空間内へ論文を写像することにより、論文の概念を抽出する方法について、その実験結果を示す。

3.3.1 論文概念抽出実験方法

概念空間の構成

概念空間は、前節で述べた意味空間において、その元として論文の概念を選んで構成したものに相当する。

類似度解析法を用いて空間を構成する場合、n個の元の位置を決定するためには、n(n-1)／2個の距離関係が必要となる。すなわちnの増加に従い、必要なデータはnの自乗に比例する。又計算時間も、ほぼnの自乗に比例する。したがって、元の数が多

くなった場合意味空間をそのままの形で構成することは、time comsumingな仕事となる。そこで意味空間を部分空間に分割し、類似度解析法により、各部分空間を構成した後、それらの部分空間を合成して、意味空間を構成することが考えられる。その方法としては、次の2つがある。

(i) 樹枝状構成法　言葉のもつ意味の間に、包含関係を規定しその包含関係によって各言葉を樹枝状に配置する。その言葉を含むレベルの異なった言葉が（n－1）個あれば、その言葉をレベルnの言葉と定める。レベルRの言葉をレベル（R－1）の言葉で類別し夫々の類に含まれる言葉によってレベルRの言葉の部分空間を構成する。これらの部分空間を、包含関係によって結び合わせることにより意味空間を構成する。

(ii) 位相型構成法　この方法は(i)の方法のようにレベルの異なった部分空間の間の関係を単に包含関係のみで記述するのではなしに、レベルRの部分空間を、夫々の部分空間の元の間の距離関係を保ちながら多次元空間内に配置し、レベル（R－1）の部分空間を構成する。この手順をRecursiveに行なうことにより、意味空間の構成を行なうものである。

位相型構成法には、次の2つがある。

(イ) 回転を用いて合成を行なう。

標準点を、遠方に固定しておき、それらとの意味的な遠近関係を考慮しながら部分空間を合成する。この場合空間の全体的な圧縮、伸長を行なう必要がある。

(ロ) 回転を用いずに合成を行なう。

各部分空間に共通な標準点を挿入しておきそれらと遠方に固定した標準点との距離を考慮して部分空間を構成し、それらを重ね合わせて合成を行なう。

DOWment RetVieValの段階の概念空間としては (i) の方法で合成したもので十分と思われるが将来 Fact Retrieval を要求された場合には (ii) の方法で構成した空間が必要となる。(ii)の方法については実験を行なっていないが、困難な方法ではないと思われる。又極座標表示にしておけば、回転などは簡単に行なえる。

3．3．2　論文概念抽出実験結果

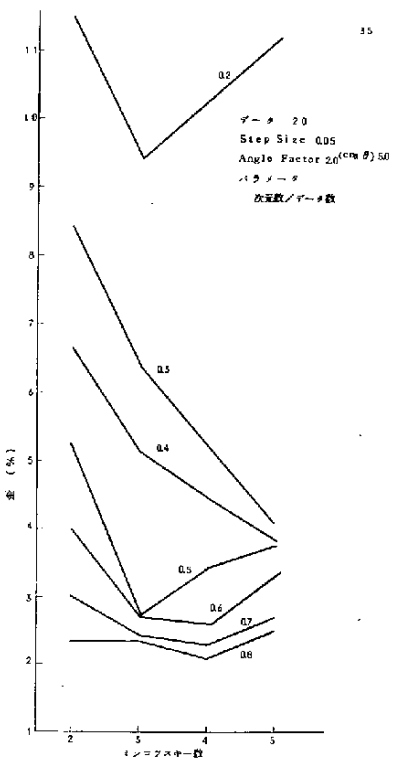
電子通信学会誌及び Communnication of ACMの発表論文について、その概念

を抽出する実験を行なったので、その結果について述べる。

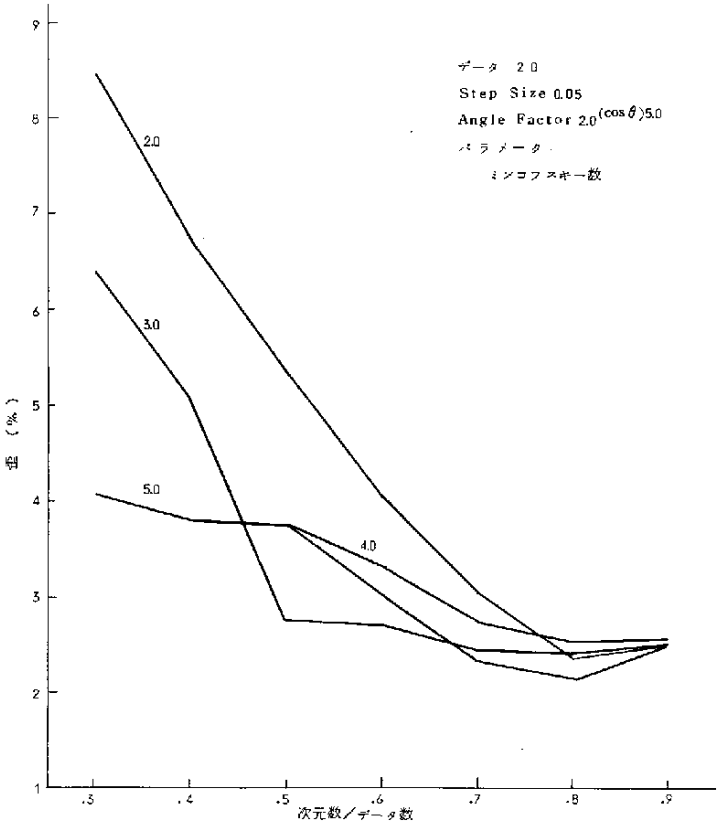
・　類似度解析法におけるパラメータ決定

前節で述べた、類似度解析法のパラメーターを決定するために、C．of ACMの発表論文75件について、アンケートによって類似度を求め概念空間構成実験を行なった。その結果を図 Ⅲ－7 ～図 Ⅲ－15 に示す。計算時間の関係でデータ数の多い場合については、あまり実験を行なえなかったが、それらの結果については以下の結果から十分類推出来ると思われる。なおデータ数が50個の場合の結果を図 Ⅲ－15 に示す。

・　ミンコフスキー数　このパラメーターは、空間を構成する情報の性質によって定められるべきものである。図 Ⅲ－7　Ⅲ－8 より4次元以上の場合は、次元数が増加すると、歪は減少している。しかし3次元では、4次元の場合より歪が少なくなっている。2次元まできげると、歪は極端に悪くなる。この傾向は、R＋0.5（Rは整数）の時に著しい。とくに、ミンコフスキー数が1.5と4.5の場合には、6次元の時の歪と全く等しくなっている。とりわけ、4.5の時には、3次元で歪が1.8％となり収束している。このこと



図Ⅲ－7　ミンコフスキー数の変化



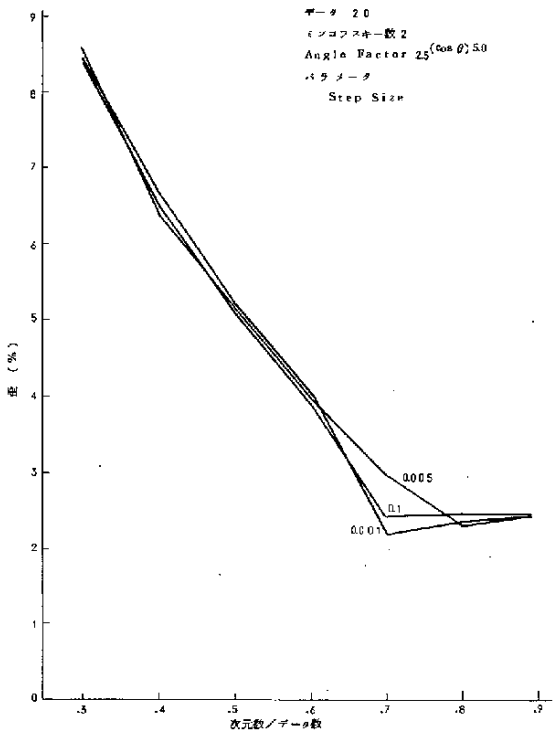
図Ⅲ－8　ミンコフスキー数の変化

は検索を行なうにあたって、検索時間の短縮を記憶容量の節約という点で、明るい見通しを与えている。

図 Ⅲ－7　Ⅲ－8 より、3次元では、ミンコフスキー数を大きくすればそれだけ低い歪が得られている。4.5次元では、ミンコフスキー数が3～4の時に低い歪が得られミンコフスキー数がそれ以上になっても、またそれ以下でも歪が大きくなる。6次元以上になると、ミンコフスキー数が3以下では、収束しない場合があるが、それ以上にすると、収束しミンコフスキー数を変えても歪の値は影響をうけない。

以上のことからCommunication of A.C.M の発表論文によって構成される情報空間における、距離関係を定めるミンコフスキー数は、4前後に選べばよいことが分かる。このことは、最初に述べたことがらが裏づけている。

・ ステップサイズの初期値　このパラメータは、空間の配置を変化させる際に、各デー



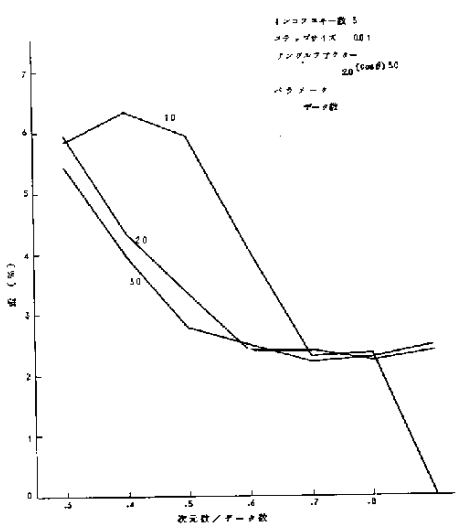
図Ⅲ－9　Step Size の変化

タの移動量を決定するステップサイズに初期値を与えるものである。この値をあまり大きく選ぶと収束点になかなか達せず発散してしまう恐れがある。実際0.2位に選ぶと発散してしまった。しかしあまり小さくしすぎても、収束するまでに余計な繰り返し計算が必要となるため収束時間が遅くなる。そこで、0.1，0.01，0.001，0.05の4つの値を選んで実験を行なった。

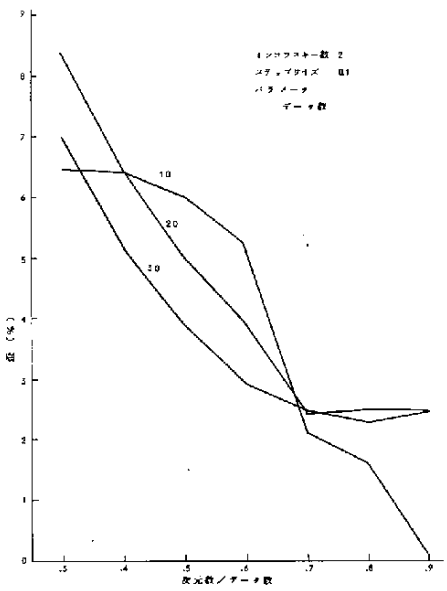
図Ⅲ－9にデータ数が20個の場合を示している。この場合には次元数が多くなれば歪が小さくなるという、傾向が顕著に表われている。また0.05の場合に14次元で収束しなかったことを除けば、0.1，0.01，0.05の3つの場合、その歪の値はほぼ同じ値をとっている。結局、初期値を0.001～0.1の間の値に選んでおくかぎり得られる歪の値は変らない。したがって、Local Minimum に落ち込まないように気をつければ、この値の選び方には、さほど問題はないと思われる。

• データの個数　データの数が増加すると、それらを配置する空間の次元数もそれにつれて当然増加することが考えられる。しかし、実験によると、データ数に比例して次元数が増えるのではなく、ある程度以上にデータ数が増えると次元数が飽和する傾向を示すように思われる。この事実は、Key word などについては、よく知られている。図Ⅲ－10、Ⅲ－11には、横軸に次元数とデータ数との比をとって、データ数の変化による歪の働きをとらえている。ミンコフスキー数が2の場合について表わした図Ⅲ－11の歪はミンコフスキー数が3の場合の図Ⅲ－10の歪より全体に悪くなっている。

データ数が１０個の場合と、２０個又は３０個の場合とでは、低次元における歪の変化に違いが生じている。すなわちデータ数が１０個の場合には、５次元以下で飽和の傾向を示している。一方、２０個又は３０個の場合には、逆に、歪の増加の割合が大きくなっている。これをみるとデータ数が増加すると歪も悪くなるようであり、検索には好ましくないようであるが、実際に検索を行なう際に使われる空間の次元数は、データ数の４～６割の



図Ⅲ－１０　データ数の変化

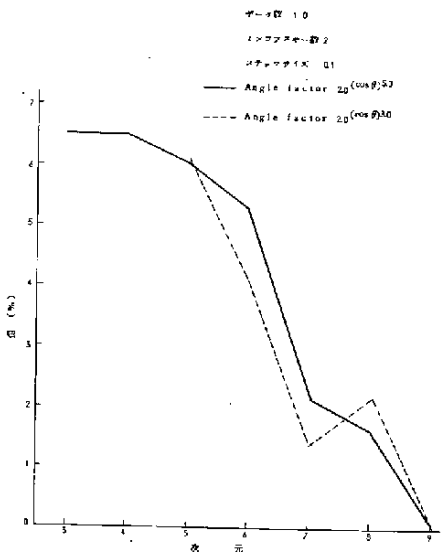


図Ⅲ－１１　データ数の変化

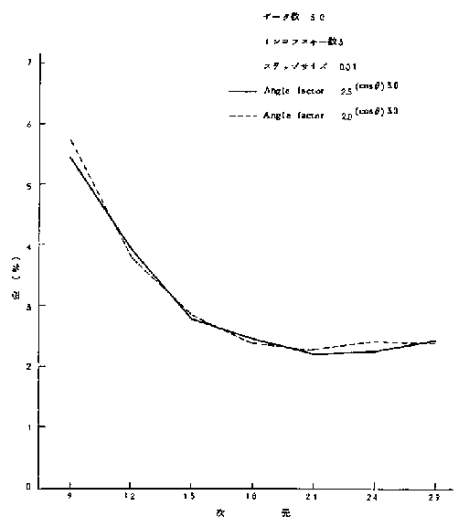
あたりである。このあたりでは、データ数を多くするにつれて、歪は小さくなっており、この傾向をミンコフスキー数を変えることによって強めることも可能である。さらに収束範囲にある次元数のデータ数に対する割合も、データ数が増加するに従い減少している。

これらのことから、データ数が多くなってもそれ程多くの記憶装置の容量を必要とすることはない。

・ Angle Factor　このパラメーターの影響については、図 Ⅲ－12に指数



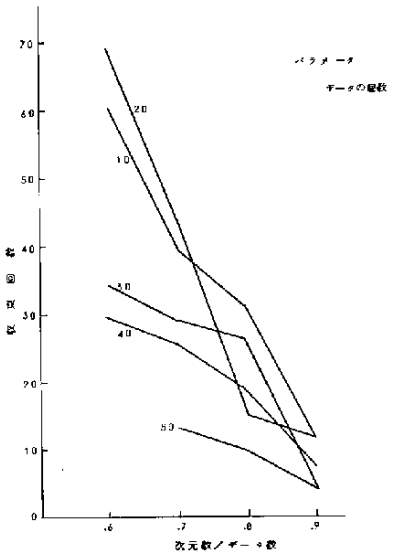
図Ⅱ－12　Angle Factor の効果



図Ⅱ－13　Angle factorの効果

部を変えた場合　図Ⅲ－13には、基数部を変えた場合について示す。指数部を変えた場合には少し影響が現われるが、基数部を変えても歪は、変化しない。あまりデータをとってないので断定は出来ないが基数部に1.5と2.0，3.0、指数部には3.0と5.0を選んで行なったなかでは、Angle Factorを2.0 $\cos\theta^{3.0}$ の形にするのが最も良いと思われる。Kruskalの指示により最初Angle Factorを4.0 $\cos^{3.0}$ としてみたが歪の変化が激しく、うまく収束しなかった。

・　収束時間　いままで低い次元数で歪の少ない情報空間を構成するには、パラメータをどのように定めてやればよいかという点について、考察を進めてきた。ここでは、計算時間という問題点について、コンピュータサイドから検討を加え、この方法を採用する妥当性を確かめる。



図Ⅲ－14　収束回数

データ数を変えた場合に収束するまでに要する計算回数がどのように変化するかを、図Ⅲ－14に示す。これは、パラメータを色々変化させた場合の平均の値を、とったものである。これによると、データ数が20個の場合を除いて、データ数が増加すれば、収束に要する計算回数が少なくなっていることが分かる。

つぎに、計算時間について考察する。

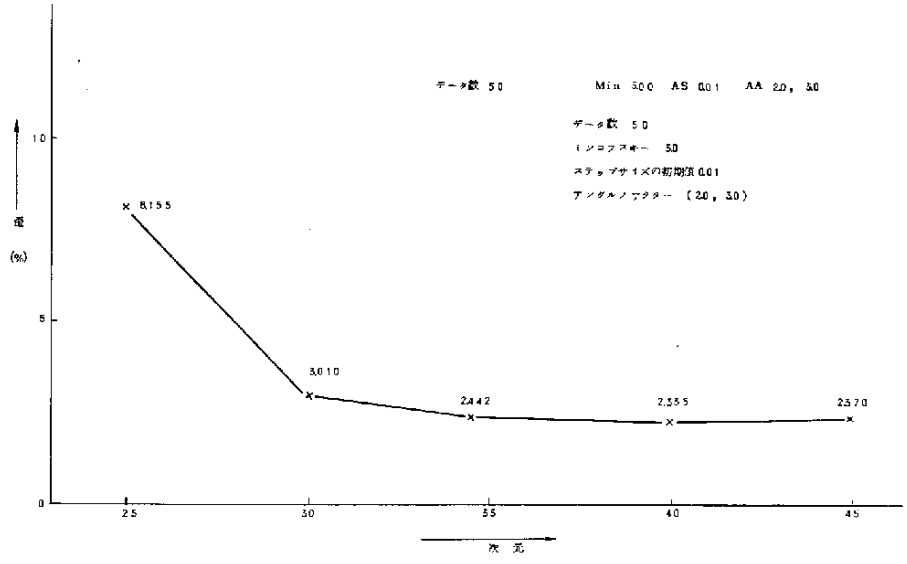
表Ⅲ－1　データ数による計算時間

データ数	10	20	30
計算時間	1.2秒	6.6秒	28.2秒

表Ⅲ－2　データ30個の時の計算時間

次元数	27	24	21	18	15	12	9
計算時間	58	35	31	28	26	20	15

表 Ⅲ － 1 にデータ数による一回の繰り返し計算を行なうのに必要な計算時間、表 Ⅲ － 2 には、データ数が３０個の場合に、次元数によって計算時間が変化する様子を示している。表 Ⅲ － 1 において、データ数が増すと計算時間は、急激に増大している。この増加の割合を、繰り返し回数の減少で相殺することが出来れば理想的であるが、これらの積の値は、データ数の増加とともに増大すると考えられる。この値の変化についてはデータの不足から述べることは出来ないが、次のように予想することは出来る。 図 Ⅲ － 1 5 のデータ数が、５０個の場合を見ると、増加の割合が他と比較して非常に穏やかである。また、収束する次元数のデータ数に対する割合は データ数の増加と共に減少する傾向にあり 表 Ⅲ － 1 より次元数が減れば、計算時間も短かくなっている。したがって、前に述べた積の値の増加は、指数関数的ではなく、むしろ対数関数的であり、データ数がある程度以上になると、増加の割合は非常に穏やかになると思われる。よって、この方法を情報検索のための情報空間の作成に用いることは、妥当であろう。



図Ⅱ－１５ 概念空間構成実験

・論文概念抽出過程の自動化について

類似度の作成過程を自動化する為 (イ)アンケートによる方法と(ロ)論文の標題と内容梗概

から専門語を抽出し、属性空間を作り、そこにおける各論文間の Hamming 距離から求める方法とについて比較検討を行なう。

概念空間の部分空間を求めるために電子通信学会誌 1968 年度の掲載論文（約300件）についてUDC分類により分類をとった。このうち、UDC621・396（無線通信）の分野の論文16件 表 Ⅲ － 3について term-document Incidence Matrix を作った。これの一部を図 Ⅲ － 16 に示す。この Matrix より求めた類似度表及びアンケートにより求めた類似度表に基づいて構成した空間の歪の変化について 図Ⅲ－17 に，又検索能率について 表Ⅲ－4 に示す。その結果得られた空間自体は、どちらも歪の点では、同等で大差はなかった。空間の良さは、検索能率により判断

表 Ⅲ－3 実験用文献

文献番号		著者	論文題名	UDCコード					
67	24	横井 寛 佐藤 敏雄 山田 松一	衛星通信大口径アンテナの利得測定	・B・ 125－132	621	396	4		
226	B－89	宮川 洋 赤間 洋明	Mout of N多周波通信方式についての2.3の考察	61－B,10 475－482	621	396	41		
227	90	更田 博昭 大久保 栄	デイジタル複合変調を用いたマイクロ波PCM制御回線	・ ・ 483－490	621	396	41		
252	101	立川 敬二	マイクロ波PCM方式の回線設計法	・ ・ 537－544	621	396	41		
176	66	岩井 文彦 宮川 達夫 八塚 弘	新しい全固体マイクロ波中継装置の一提案とその実験	・B・ 381－337	621	396	41	029	64
10	2	渡辺 宅治 中村 孝	広帯域トランジスタバリオロッサ	・B・ 9－ 16	621	396	665	52	
13	5	大久保元晶	異方性円柱状プラズマ内の電気双極子からの放射	・ ・ 32－ 37	621	396	674	3	
119	43	松尾 和昭	十字形ダイヤポールを用いた飛しょう体追尾アンテナプレイ	・B・ 247－253	621	396	674	3	

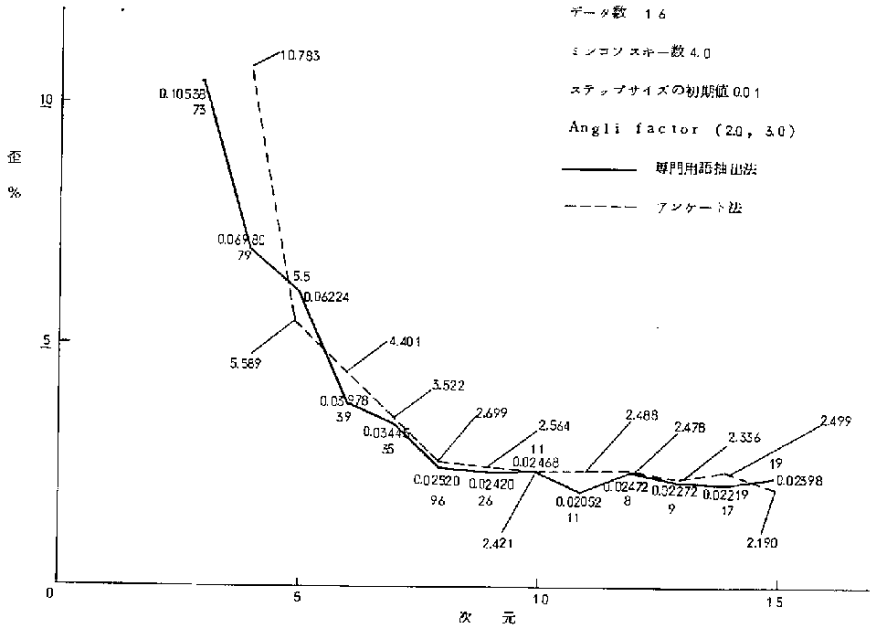
文献番号		著者	論文題名	UDCコード					
34	13	伊藤 洋 虫明 康人	異方性プラズマ中のアンテナインピーダンス	・B・ 69－75	621	396	674	3	011.21
253	102	手代木 扶 永井 淳	励振誤差を含むアンテナアレイの解析	・B・ 545－552	621	396	677	3	
95	35	後藤 尚久 関口 利男	半波長ダイポールアンテナを素子とする不等間隔Broadside	・B・ 201－206	621	396	677	31	
122	46	後藤 尚久	Endtire Arrayの素子とSuper-gain効果について	・B・ 268－272	621	396	677	32	
251	100	後藤 尚久	最大利得をもつEndfire Arrayの素子の更効入力電力	・B・ 531－536	621	396	677	32	
67	24	横井 寛 佐藤 敏雄 山田 松一	衛星通信用大口径アンテナの利得測定	・ ・ 125－132	621	396	677	832	2.091
48	16	清水 康敬 末武 国弘	多段形整合負荷の新設計法	・ ・ 83－94	621	396	69	016	
201	79	稲場 文男 小林 尚郎 市村 勉 森久 光雄 平良 賢剛	Aスコープ方式レーザレーダによる大気伝搬特性の測定とエコー波形の解析	・ ・ 425－431	621	396	962		

専門用語＼文献No.	10	13	34	48	67	95	118	119	122	176	201	226	227
増巾器	1									1			
アンテナ		1	1	1	1	1	1	1					
ダイポール						1		1	1				
PCM												1	1
ビーム							1	1			1		
サイドローブ						1	1						
人工衛星					1	1		1					

図 Ⅲ－16 接 続 行 列

表Ⅲ－4　検　索　能　率

空間の次元数	類似度の作成法	再現率	適合率
10	アンケート	73%	85%
	接続行列	96%	100%
6	アンケート	63%	71.5%
	接続行列	86%	100%
4	アンケート	50%	58%
	接続行列	75%	75%



図Ⅲ－17　概念空間の歪の変化

出来るが、　表 Ⅲ－4　より再現率については差がないが、適合率は（ロ）の方法がよかった。したがって、非専門語の辞書を与えてやれば、論文概念を自動的に抽出することは可能である。

4　位相型言語応答システム

4．1　日本語の意味解読過程の検討

計算機に自然語を扱わす為には、その構造を知る必要がある。自然語に近いレベルに於ける形式言語モデルとしてはChomskyの句構造、変形規則からなるものが代表的である。しかし、これは文の生成過程のみで処理過程は含まれていない。よって意味の理解処理過程を形式的に取り扱えるモデルが望まれる。いわゆるプログラミング言語と呼ばれる機械用の言語は、その付号体系の構造的性質（シンタックス）と，そしてそれが意味するところの機械的、数学的行動（セマンティックス）が、あらかじめ明確に記述されているという点で、我々が日常使いなれた自然語とは異なっている。自然語と機械語間の翻訳や自然語を機械で処理する為には自然語のシンタックス及びセマンティックスを記述する必要がある。しかしながら、これらを形式的に記述することは非常に困難である。我々は意味あるいは概念というものが理解できたという事は、意味を表わす高次元空間（意味空間）内の領域が指定できたという事であると仮定し以下のようにチヨムスキーの文法を参考に、自然語のレベルでの処理過程のモデル化を行なう。このモデルの妥当性は3．3で行なった実験等によって確認された。

4．1．1　チヨムスキーの変換生成文法について

チヨムスキーは1957年に出版した小著“Syntactic Structure”で初めて生成文法という考えを提出した。これは自然言語の持つ文法的な規則性を客観的に形式化して記述するにはどうすべきかという問題に対する一つの具体的な解答であると同時に、人間の言語能力をオートマトン理論と結びつけて説明したところに意義がある。チューリング・マシン、プッシュダウンストア・オートマトン、有限オートマトンが夫々チヨムスキーの0型、2型、3型の文法と等価な事が証明されている。彼の考えでは言語学は、シンタックス、意味論、音韻論よりなる。音韻論は形式言語に直接関係がないので省く。意味論は情報の通信手段として言語を見た場合大切であるが、まだあまり大きな結果は得られていない。シンタックスに関しては変型文法理論で既成の分類的文法を句構造文法として解釈し、それに文と文との関係およびその写像という考えを導入した。その考えでは、シンタックス部分は句構造部分と変形部分から成り立つ。句構造部分に含まれる規則は狭

義の規則でＰＳ規則と呼び、これより有限個のＰマーカーを得る。これに対し変形部分の規則をＴ規則と呼び次の様なものからなる。

1) 順序変換　　A＋B→B＋A

2) 挿　　入　　A＋B→A＋C＋B

3) 削　　除　　A　　→A

4) 取　　換　　A＋B→C＋B

5) 膨　　張　　A＋B→D＋F＋B

6) 置　　換　　A　　→B

7) 併　　合　　二つの単文から重文，複文を生成

句構造部分で得られたＰマーカーは基本Ｐマーカーであり、これらに上述のような変形を施こし派生Ｐマーカーを得る。変形には必須なものと選択可能なものがあり、ＰＳ規則と必須のＴ規則だけを使って出来る記号列を核文と呼び、これに対し他のものを派生文と呼ぶ。核文と派生文の識別をＴ規則適用の目的としており、変形部分は帰納的能力を持つ。以上が当初の文法のあらましであるが、上述の場合，核文と派生文に分ける必然性がうすく区別の方法がむつかしい。これらの点を改善した新文法では、文法の構成要素は前に述べたと同じく３つからなる。

1) 文に骨組を与え、それに意味と音声を与える為の情報を与える構文部分。

2) 与えられた骨組を解釈して意味を持たす意味論部分。

3) 骨組を音声に変え、実際の発話に直す音韻論部分。

構文部分の書換規則（ＰＳ規則）は次の３種からなる。

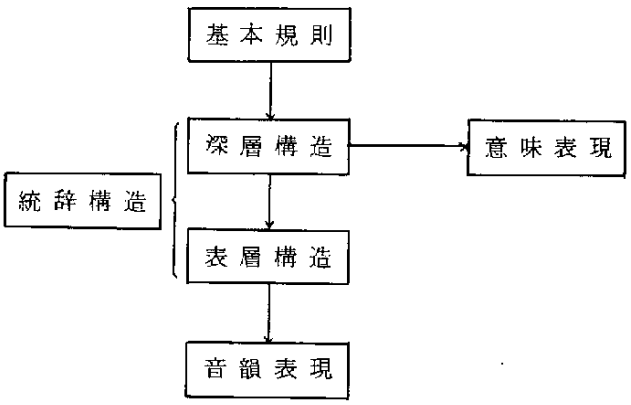
1) 従来の枝わかれ規則

2) 新らしく導入した下位分類規則

3) 辞書

変形部分は変形規則の集合からなり、その働きは取換、削除、付加の３種の組合せであり書換規則で得られた構造（深層構造）へこれを作用させれば、表層構造が得られる。表層構造は普通使用している文の構造であるが、深層構造は生成過程の情報を含ませている為に、△とか＃などの記号を、最終記号列に含む。表層構造は音韻部分により音声にかえら

れ、深層構造は意味部分により意味解釈が施される。この２つの部分に必要なあらゆる情報は既に構文部分によって得られており、音韻部分も意味部分も構文部分で得られたものに何等新しいものを付加しない。又、深層構造から表層構造への写像に関しても意味は全く変化しない。



図Ⅲ－１８　言語学の構造

（チョムスキーの新説による）

４．１．２　言語処理系のモデル化

生成文法は一般に句構造規則と変形規則とからなる。音韻規則は記述言語では直接関係ないので以下言及しない。表現レベルとしては、表層レベル、深層レベル、意味レベルの３つを考える。ここで、表層レベルは実際に使用されている言語のレベルであって、深層レベルは表層表現の生成元または最簡形に相当するものである。意味レベルは意味空間（3.2で説明）に対応する。言語のレベルにおける処理は、表層レベルと深層レベルの間の写像と、深層レベル中での写像である。真の意味での情報処理は、深層レベルと意味レベルの間の写像と、意味レベル中での写像である。これを深層レベルの処理と呼ぶことにする。意味レベルと深層レベルとのつながりは、意味レベルにおける情報空間中で、ある閉領域を占める概念に語を割り当て、文の意味は各語の概念の文法による合成で定まる。

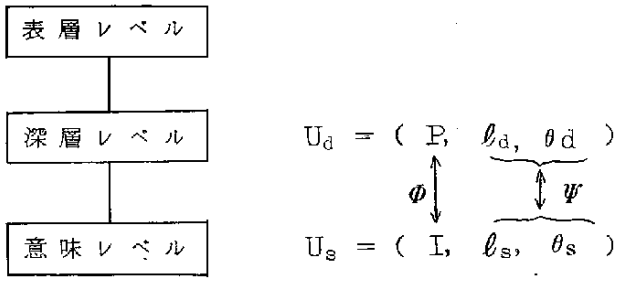


図 Ⅱ－１９　言語処理系のモデル

言語のレベルに於ける処理は、その言語を生成する文法を通して考える。文にその生成過程の情報を含め持たせたものはＰマーカーで表現できる。従って言語レベルでの処理とは、「各文に対応するＰマーカーに種々の演算を施し、求めるＰマーカーを得る事」と考えられる。以下検討してゆく言語系については、ｍｅｔａ　言語と対象言語が別々に存在するとは考えない。文の持つ情報は次の二種のどちらか一方である。

a) 文はそれに対応するＰマーカーで１つの情報を表わす。

b) 文はＰマーカー間の分解合成を行なう写像関数であるか、その様な写像関数あるいは写像関数のパラメータを指定する。

これはＰマーカーによるｍｅｔａ表現であり、疑問文、命令文等文法Ｇにより生成されるＰマーカーのすべての集合をＰとする。

Ｐマーカーの処理としては次の様なものが考えられる。

a) ２つのＰマーカーから分解合成写像によって必要なＰマーカーを得る。

b) 単一のＰマーカーの変形、１つのＰマーカーに作用子を施す事により異なるＰマーカーを得る。

c) ２つのＰマーカーの一致検出。

言語のレベルに於ける処理系はＰを台として、内的算法　$I_d = \{ f, g, h, \cdots \}$ と外的算法　$\theta_d = \{ \omega, \cdots \}$ を持つ代数系 U_y を形づくると言える。

次に深層レベルの処理について説明する。深層レベルは文の生成元にあたるＰマーカーの集合よりなり、意味レベルは多次元位相空間よりなる意味空間である。深層レベルの処理は、

i）深層レベルと意味レベルの間の写像と、

ii）意味レベル中での写像である。

情報空間は、$I=\{S, \omega\}$ なる近傍空間であり、ここで、SはRⁿ（n次元空間）を構成する集合である。なお、この集合としては、

i）従来の集合論の集合

ii）Fuzzy 集合

等のとり方がある。ωは近傍系であるか、もし分離公理も成立する場合には距離とし、空間は距離空間となる。 ii）の集合の場合には、分離公理は閾値をパラメーターとして含む新しい公理を設定しなければならない。

表層レベルGの語彙Vは有意語彙V_Pと文法語彙V_Gとからなる。文法GのV_PとIの閉領域とを結びつける写像Φ；$V_P \rightarrow 2^{R^n}$が各単語に意味を割り当てる写像であり、Ψ；$V_G \rightarrow \Omega U \ell$なる写像は書換規則により$V_P$と$V_G$を結びつけたり、$V_P$と$V_P$を$V_G$で結んだ時に、意味レベルでは$\Phi$（$V_P$）にどの様な写像を施して、出来上った句とか文の概念を求めたら良いかを指定する。Φ、Ψは表層、深層レベルの処理を関連づける基本写像であるが、情報空間同様各人が頭の中に持っていると考えられる。

4．2　意味空間の代表的性質

ここでは前節 4．1 に於いて提案された仮説について、その骨子となるところの深層レベル、意味レベルに於ける内的算法ℓ、外的算法θ及び相互レベル間の要素，算法の写像関数Φ,Ψを抽出する一般的なアルゴリズムについて考察を行なう。まずこの様な算法として具体的にどの様なものが考えられるか、そして抽出される算法の持つべき、或は、持つであろう性質等について考え、実際にこれらを抽出する手順を考える事とする。

4．2．1　演算の性質

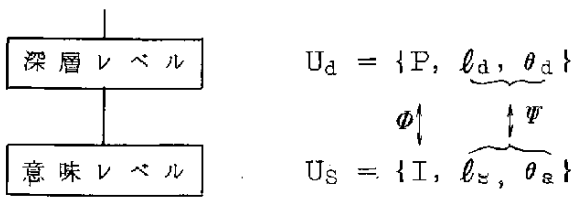
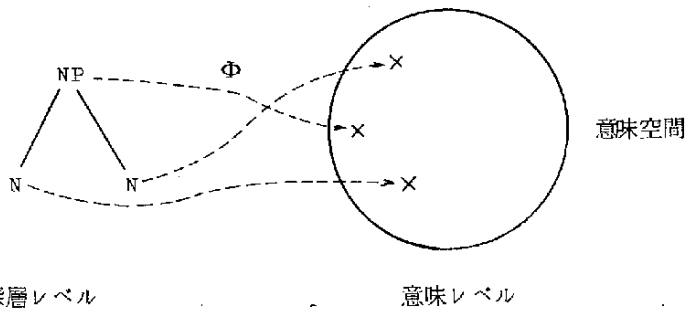


図 Ⅱ－20　　新しい言語処理系

まずΦについて、その意味を考える。深層レベルに於けるその要素Pは P マーカーの最簡形すなわち単語、句、節、文等が、それぞれのレベルで対応したものと考えられる。Iは意味空間{S, R}であり、一般的にはn次元近傍系をなすが 特にここでは実際の取扱い易さの為、n次元距離空間での超球面上であるとする。そうするとΦは単語、句、節、文等を、それぞれのレベルで情報空間上に写像する写像関数であると考えることができる。そして構成された意味空間に於いては、それらの要素間の意味的な関係(例えば遠近関係包含関係等である。)が表現されている必要がある。従って、この様な写像関数Φの1つとしては 類似度解析を用いた多次元配列法による空間の構成が採用できる。すなわち、単語、句、節等の要素に対してそれらの間の意味的な遠近関係を計量心理学的手法を用いて人間の脳より抽出し、その順序関係を保持する様な空間を多次元配列法を用いて構成すれば良い事となる。この場合、Pの要素に対しては意味空間上の座標を指定するt個(tは意味空間の次元数)の数値が対応する。又この対応は一般に多義語などの場合を考えると一意的な対応ではなく、非決定的な対応をなすものである。



図Ⅲ－２１　意味空間

ℓ_d, θ_d について考えると、これらはPに対する二項及び一項演算子である。ℓ_d すなわち内的算法もしくは二項演算子と呼ばれるものは、P×PからPへの写像である。最も簡単な場合としては、例えば「連語」を挙げる事ができる。つまりPの要素である単語と単語を「連語」という演算子で結びつけると、又それはPの要素となっている。θ_d すなわち外的算法もしくは一項演算子と呼ばれるものは、P×ΩからPへの写像である。ここでΩは外部作用団である。最も簡単な例としては、「否定」をあげる事ができる。Ψにつ

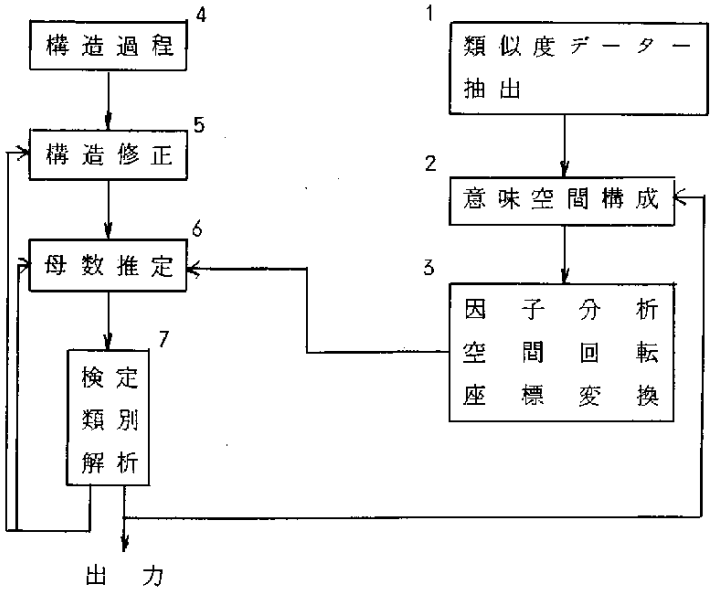
いては、ℓ_s, θ_s と ℓ_d, θ_d との関係を示す対照表を作成する事に帰着される。ℓ_s, θ_s については、ℓ_d, θ_d に対応して情報空間上で行なわれる演算である。これは ℓ_d, θ_d に対応して定式化した形で抽出する必要がある。これを抽出する為にはなんらかの方法で統計的な手法を用いる必要がある。しかも意味空間は一般に多次元であり本質的に扱う変数は多変量となる。従って多変量回帰論の手法を用いる。つまり、データーの中に成立すると思われる演算に対し、何らかの構造を仮定し、その中に含まれる母数を標本より推定するという方法である。抽出される実際の演算の望ましい性質としては次の事が挙げられる。

(1)類似度解析による多次元配列法を用いた意味空間は、概念相互間の意味的な遠近関係の単調性をのみ保持する空間であり、非常に自由度が大きい。従って、空間の初期配置又は収束過程の違いによって全く異なった空間配置となる。抽出される演算はこの違いによって構造、母数の変化しないものが望ましい。

(2)なるべく簡単な形のものが抽出、実際の演算を行なう為の時間が少なくてすむ。特に演算の抽出の段階では構造模型として、正規線型性を仮定する必要がある。(1)とも合わせてこれらの問題に対しては適当な座標変換を用いれば解決できる場合がある。

4．2．2　演算抽出の為の手順

ここでは前節に於いて述べた方針に従って ℓ_s, θ_s を定式化するアルゴリズムについて考察する。方法としては、深層レベルのPマーカーの最簡形及びそれに算法βを施したものをΦにより、I_s 上に写像し、I_s 上の対応点の位置関係から統計的に ℓ_s, θ_s を抽出する。



図Ⅲ－２２　　意味空間に於ける算法抽出の手順

図 Ⅲ－２２　に演算抽出の手順を示す。以下番号をつけた１つ１つのブロックに対して説明を加える。

(1)　類似度データー抽出

３．２節で説明した通り、ｎ個の情報点に於いてそのｎ(ｎ－１)／２個の対に対してその相互間の意味的な近さという尺度に従って、順序づけ、あるいはランクづけを行なうものである。

(2)　意味空間構成

同じく 3. 2 で説明した類似度解析による多次元配列法を用いれば良い。この場合次の２通りの方法がある。

a)　データの持つ順序関係の単調性をのみ保持する様に空間を構成する。

b)　データの持つ順序関係の比率を保持する様に空間を構成する。

a)よりb)の方がより強い束縛条件であるといえる。従ってｂ)による空間の方が構成次元数が高くなると同時に、より多くの情報を含む。しかし、この順序関係の比率が測定の段階で被験者より正確に抽出され得るものかどうかを考える必要がある。a)の方は、この情報を無視したもので、より低い次元数で構成可能となる。両者の得失は抽出された演算の

適合度等を考慮して論じ得るものである。

(3) 因子分析、回転、座標変換

すでに述べた様に抽出される演算は空間の収束形状に関係なく一定である事が望ましい。又演算は意味的に考えて何らかの意味に対応する因子軸に関して簡単な形あるいは一定の形を持っている事が予想される。従って、抽出された演算が簡単な形状になる様にあるいは演算を抽出すべき空間をなんらかの因子について解釈を与えた形に回転、構成しなおすという意味で座標変換を行なう必要がある。因子をぬき出す手法としては、成因分析、因子分析、空間を回転する方法としては、クアーテマックス、バリマックス、オブリマックス等がある。

(4) 構造仮定

これは回帰模型を設定する部分である。電子計算機による反復計算の労さえいとわねば相当複雑な構造式に対し母数を推定する事もできるが、理論的に最も取扱い易いのは変動の正規性、変量間の線型性を仮定した正規線型回帰模型である。この一般型は次式で表現できる。

$$y_\alpha = \beta x_\alpha + \theta_\alpha$$

しかしながら、この様な線型性の仮定は，例えば演算がある軸を中心に回転するといった様な場合、又特に意味空間が球面上に正規化されているという条件も考え合わせる時、不適当なものとなる。これに対する解釈法としては次の様なものが考えられる。

a) 情報空間を球面上に正規化しない。

b) 回帰模型の従属変数の高次の項を新たに従属変数としてつけ加え形式上線型回帰模型として取扱う。

c) 適当な変数変換を行ない、回転に対し線型演算が成立する様にする。

例えば、この場合であれば、極座標変換を施せば線型演算で回転が表現できる。

$$\begin{cases} x_1 = \gamma \cos \theta_1 \\ x_i = \gamma \cos \theta_i \cdot \prod_{j=1}^{i-1} \sin \theta \\ x_n = \gamma \prod_{j=1}^{n-1} \sin \theta \end{cases}$$

a) については、球面上に正規化するという条件は検索に便利な性質であるので、この様

にする事が有利とは言えない。

c）に関しては空間が歪を持つので大きな距離にある点間の距離の単調性が保持されなくなる。しかし、局所的には単調性、比率は保持されている。又、この様な座標系で単調性を保持できる様に最初から空間構成を行なえばよい。b）の方法は変数の数が非常に増加するので取り扱い難くなる。

(5) 構造修正

ここでは(7)に於いて検定、類別、解析を行なった後より適切な回帰模型を設定する。例えば検定を行なった後適合性が悪ければ、より多くの母数を含んだ構造模型を仮定し、又必要のない母数が存在すればそれを切りすてた模型を設定しなおす。

(6) 母数推定

正規線型回帰論の手法を用いて未知母数を推定する。この時、標本は1つの演算に関し施されたものと、施された結果をそれぞれ空間上に写像したものの座標値とする。

(7) 検定、類別、解析

この部分に於いては、得られた演算式を実際に施して、それが深層レベルで演算を施してからΦで意味空間上に写像したものとどれだけ適合するかを検定する。同時に得られた母数を解析して演算自体が類別されるかどうかを検討し、類別されるならその規準によって分けられた各々の類についてそれぞれ独立に構造式を仮定しなおして母数を推定する。

4．3 論文概念抽出法の解析

4．3．1 論文概念抽出方針

自然語のレベルで情報検索を行なう為には、論文の概念を見出し語のレベルで離散的にとらえるのではなく、3．2でも述べた様に文章のレベルで連続的な概念としてとらえる必要がある。情報（論文）をこの様なとらえ方で表現するには、情報のもつ色々な概念（意味）が、空間内の座標軸の組み合わせで表わされかつ情報間の概念的なつながりがそれらの座標軸に何らかの演算（回転や変換）を施すことにより表わされる様な空間を構成しなくてはならない。この様な情報（概念）空間を構成しておけば、従来の様に見出し語とそれらの間の理論結合で情報を表現する属性空間に比べ、より密度の濃い検索を行なうことができる。ここでは、人間の情報処理過程を分析することにより、概念空間の構成方法

の解析を行なう。

人間の情報処理過程は、非常に複雑で明確に記述することは、困難であるが、次の３つのパターンの組み合わせを考えることができる。

(1) 属性とその値を与えて属性空間内の一点としてとらえる。

$$I = (P_1, P_2, \cdots\cdots, P_i, \cdots\cdots P_u)$$

ここでP_iは属性P_iの値

(2) 他の情報との順序情報がある属性について与えられており、それにより情報の位置をとらえること。

$$I = \{ I \mid P_i > P_i^A ; P_i, P_i^A \in P_i \}$$

ここでP_i^AはAについての属性P_iの値

(3) 他の情報との順序情報が概念的に与えられており、それにより情報の位置を定める。

$$I = \{ I \mid I_A - I \mid > \mid I_B - I \mid \}$$

ここで$\mid \quad \mid$は情報間の類似度を表わす。

(1) (2) (3) のパターンに、共通な性質として、順序情報を基にして情報をとらえていることがわかる。又これらの情報を記憶する場合には既に貯えられている情報のうちのいくつかと新しい情報との意味の遠近関係を調べることにより格納場所を定め、呼び出す場合には、概念を表わすベクトルを概念空間内に想定し、そのベクトルに各情報を写像することにより順次をとり出していると思われる。

この様に順序関係を基にして空間を構成する方法には、計量心理学の分野で種々の方法が研究されているが（３．２．１参照）、ここで与えられる順序情報の性質からKruskal氏の類似度解析法を少し修正した方がよいと思われる。又その正当性は、３．３で述べた実験結果より明らかである。

４．３．２　論文概念抽出方法

概念空間を構成する際の問題点としては、

(1) 部分空間への分割及びそれらの合成法

(2) 論文概念の間の類似度の作成法

の２つがある。(1)、(2)の問題を解決するためには、まず論文よりディスクリプターを抽出

する必要がある。ディスクリプターの抽出法としては

(i) ディスクリプターとして使用する言語(概念)を限定する。

(ii) ディスクリプターを限定しない。

方法とがある。(i)の方法では、論文の概念を表現するものがディスクリプターとして選ばれておればよいが、そうでなければ論文の概念を抽出することはできない。又使用言語が限定されているからＭＡＣで使用する際には不便である。又変更修正が行ない難い。したがって、頻繁にアクセスされ、又変更が要求されるようなファイル、すなわち部分空間構成のためのディスクリプターとして不適当である。そこで(i)の方法を、部分空間への分割する際に用い、(2)の問題の解決の為に (ii) の方法を用いる。部分空間を合成する際の方法として　3．3　で2つの方法を述べたが、樹枝状合成法を用いる場合が(i)、位相型合成法を用いる場合が (ii) に相当する。

・(i)の方法の１つであるＵＤＣ方類を用いる部分空間に分割する方法を示す。まず２つの論文概念の間の包含関係を次のように定める。すなわち、C_A が C_B を含有するならば、C_B のＵＤＣ指標を C_A も必ず 持っていると定める。この包含関係から次のような Ｍａｔｒｉｘ が得られる。このＭａｔｒｉｘ を用いて論文概念の位相関係を定めることにより部分空間への分割が行なわれる。

・論文概念の間の類似度を作成する方法としては、

(イ)論文よりディスクリプターを抽出し、それらの作る属性空間内へ論文を写像し、各論文間の距離によって類似度を作成する。

(ロ)論文より抽出したディスクリプターに、それが表われる場所、例えば、タイトルであるとか、結論の部分であるとかにしたがって重みをつけその重みの和の値によって類似度を定める方法などがある。

(ロ)の方法を用いる場合には、数種類の辞書が必要であり、重みの値を定める方法も実験による以外には、適当な方法がない。よって直ちにそのメリットを云々することはできないが、上に述べた問題点が解決されれば検討する価値のある方法と思われる。そこで、ここでは(イ)の方法について検討してみる。ディスクリプターを抽出する方法としては、

(a)論文全文を対象にして、専門語の出現頻度の統計をとり、ある閾値をこえたものをディスクリプターとして選ぶ。

(b)論文の題目及び要約を対象として専門語の出現頻度の統計をとる。

(c)論文の題目中の専門語を選ぶ。

などが考えられるが、いずれの方法を採用しても専門語とは何かという問題が生じる。その選択法としては、

①非専門用語集（論文の概念を表現しない用語を集めたもの。英語では、A, The, On, Of・・・・・　日本語では、結論、概略などの一般用語が相当する）を作り、論文中に表われる自立語とくに名詞句などのうち非専門用語集にないものを専門語として抽出する。

②その分野の論文中に表われる語の統計をとり、その統計を参考にして、あらかじめ専門用語集を作っておき、その中に表われる語及びそれから作られる連語、名詞句などを専門語とする。

③人間が行なう場合は、人間がその論文の概要を表わすのにふさわしいと思う語を、専門語とする。

完全に自動化を行なう場合には、専門用語数より非専門用語数の方がはるかに少ないと思われるので、①の方が望ましいと考えられる。しかし現在論文全体を読み込ませ得るようなＯＣＲ装置等がないので③の方法をとらざるを得ない。次に論文のどの部分からディスクリプターを抽出するかという問題であるが、(b)、(c)の方法について実際に行なった結果、(c)の方法では、２つ以上の論文に共通に出現する専門用語は極めて少なかった。したがって(a)の方法によりディスクリプターを抽出してもそれから論文間の類似度を求めることは，不可能である。(b)の方法を用いれば、同一の専門語が共通出現する確率は高くなるが、類似度を求めるためには、連語を分解したり、近い概念を表わすと思われる専門用語を同一視したりする操作が必要な場合もあった。(a)の方法を用いるのは、かなり冗長があると思われるので(b)の要約を論文全体から重要と思われる文を抜き出してそれを列挙して作りそれを対象として専門語の出現頻度の統計をとり、ディスクリプターを選べばよいと思われる。次に、このようにして抽出したディスクリプターから論文概念の遠近関係を求める方法を考える。まず論文Ｗを次のようなｎ－ｔｕｐｌｅ で表わす。

$$W = (t_1, t_2, \cdots\cdots t_i, \cdots t_u)$$

ここで t_i の値の定め方により、Wは次の3つの方法で表わされる。

ⓐ $t_i = \{0, 1\}$ Wがディスクリプター T_i を含めば $t_i = 1$

その他は0，

ⓑ $t_i = [0, 1]$ t_i は T_i がWに表われる頻度確率を表わす。

$$t_i = N(T_i \mid W) \Big/ \sum_{j=1}^{n} N(T_j \mid W)$$

$N(T_i \mid W)$ はディスクリプター T_i のWにおける出現頻度を表わす。

ⓒ $t_i = [0, 1]$ t_i は上式のNに N' を代入した式で与えられる。N' は出現頻度の統計をとる際に、T_i が現われる場所、例えばタイトルに現われるとか、結論に現われるとかによって重みをつけて加え合わすことを意味する。

上のようにして定めたn-tupleについて、これをn次元空間中のベクトルと考え、内積、距離、角度などを求めることにより類似度を定める。いずれの値を選んでも順序関係は変らないので計算に便利なものを選べばよい。Wの形としては、ⓒが最もよく概念を表わしていると思われるが重みのつけ方が難しい。いづれの表わし方をするにしても、Wの長さが1になるよう正規化しておけば類似度を求める計算が楽である。さらに T_i の間の相関関係を規定しておけば、Wの概念をよりよく表わすことができる。

3．3 において述べたようにアンケートによって類似度を求めた場合と、上記の③(ii) ⓐの方法により自動的に類似度を求めた場合の比較を行なったが、その結果として、自動的に行なった方がよりよい論文概念の抽出を行なえることが分った。③(ii) ⓐの方法は、自動化の最善の方法ではないので、OCR装置の開発等、いくつかの問題点が解決できれば、論文概念抽出の自動化は十分可能である。

4．4 会話、応答過程の解析

計算機のオンラインMAC処理における、会話、応答過程は、計算機サイドで考えると(1)質問、命令、応答の分析 (2)処理 (3)応答の3段階に分れる。以下その各過程について更に詳しく論ずるが、そのプロセスのフローチャートを、図Ⅲ－23に示す。

4．4．1 質問、命令、応答の分析

これは更に、(1)入力の構文分析 (2)入力の内容解釈の2段階に分かれる。

(1)入力の構文分析　これは統辞（Syntax）処理であるが、入力としては、文（文字の連鎖の系列）、図形、音声、接触などがある。しかし計算機の内部では、一つの言語体系があり、全部の入力はその言語に翻訳される。この翻訳が、構文分析となる。次に各種入力について、更に詳しく見てみよう。

(i)文入力の場合、カード、テープ、文字読取器用特殊文字、磁気インクで書かれた文字、ブラウン管からライトペンなどによる文字入力、などを入力媒体とし、文字綴りで入力する場合である。なお簡単なシステムでは、文字といっても数字しか取扱わないものも多い。ライトペンで文字入力というのは、まだみられないが、将来はぜひ必要となろう。また音声は、ソノタイプなどで文字綴りに変換すると、以下は文入力として同様に論ぜられる。構文分析の過程は次のようになる。

①入力文をまず文字毎に、システムで取扱かえる（許される）文字かどうかをチエックする。ここではねられたからといってすぐ誤りとなる訳ではないが（例えば「艸という字はない」という文は正しい）、その場合は、対象言語としてではなく、メタ表現したもので、以下の取扱いが異なってくる。

②次に辞書（これはSyntax　辞書で、Semantic　はまだ問題にしない。）をひき、文中に表われる単語が、今問題としているシステムの言語体系の中に存在するかどうか、をチエックする。

③最後に、システムの言語の文法をもとに、構文分析を行なう。会話用言語としては、FORTRAN、COBOLなどより自由度の高い言語を使いたいので、構文分析を行なった結果得られるPマーカ（Derivation Tree）が唯一つだけ存在すればよいが、二つ以上存在する（あいまいさがある）ときには、複雑になる。前後の文脈により、正しいPマーカを推定する能力をソフトに与えておくと、それがやりやすくなる。

使用言語は、ある制限をつけたCSLか、CFLのレベルであるから、もし計算機のハードが、それと同等のもの（LBAか、PDSメモリをもつ機械）ならば、構文分析した結果は，直ちに実行アルゴリズムがわかることになる。例えば、バロースのB5000シリーズでは、構文分析した結果を、ポーランド記法に従って配列しただけで、もうプログラムとなる。ところが機械がより下のレベルのものだと、その機械がわかる言葉、アッセ

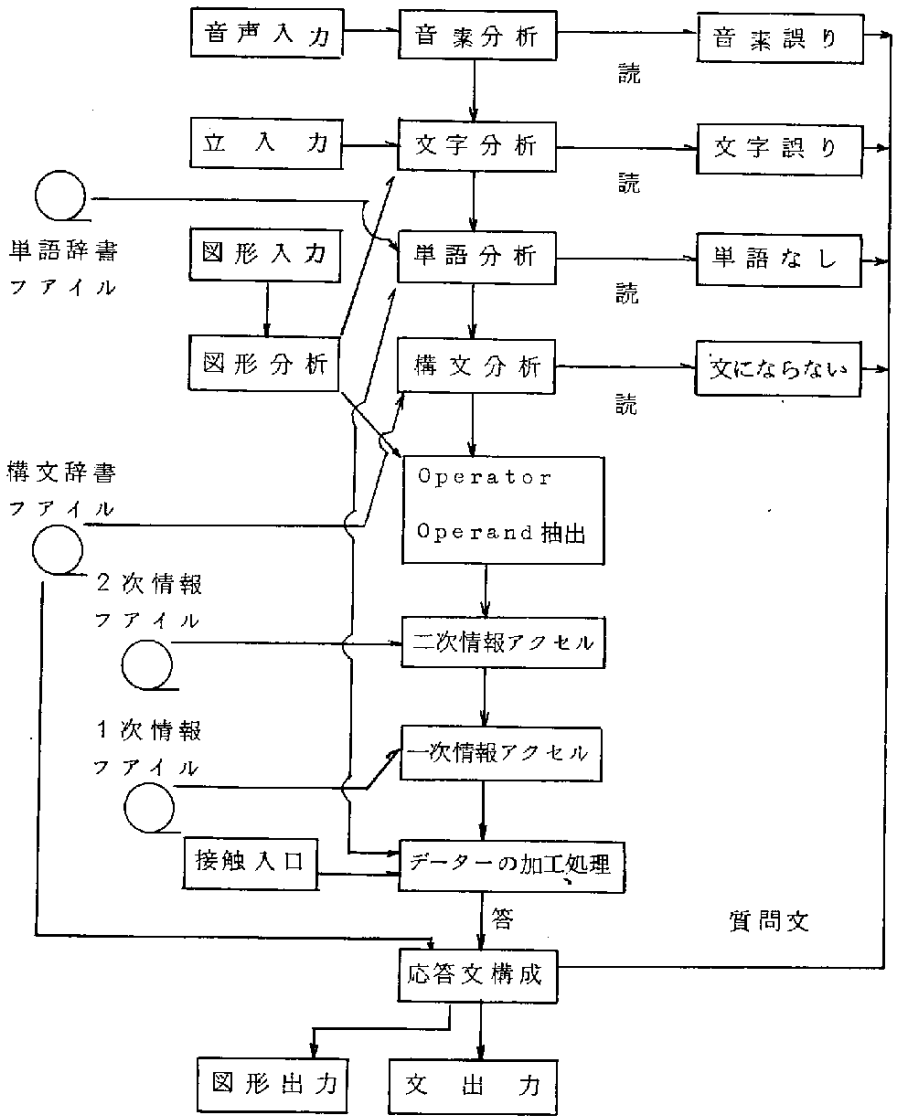

音声入力
音素分析
音素誤り
読
立入力
文字分析
文字誤り
読
単語辞書
ファイル
図形入力
単語分析
単語なし
読
図形分析
構文分析
文にならない
読
構文辞書
ファイル
Operator
Operand抽出
2次情報
ファイル
二次情報アクセル
1次情報
ファイル
一次情報アクセル
接触入口
データーの加工処理
答
質問文
応答文構成
図形出力
文出力

図 Ⅲ－２３　　会話、応答過程

ンブラか機械語にまでおとしてやらなければならない。これについては⑵で論じる。

構文分析をやっているうちに、あいまいさ（Ambiguity）や不明な点があれば、計算機から使用者へ質問を出して、正しいＰマーカを選択する為の情報を得る。（４．４．２参照）

また①②の過程で認められない（存在しない）文字、単語、文を読みこんだ時も計算機は質問を出す。この場合には、４．４．２の処理を行なわず、直ちに「，「×　×」なる字、語は認められないので、綴り間違いではないか？「×　×」なる語句をもう一度正しく入力せよ。」というような応答を出す。エラーメッセージを出して止まるようでは、対話型ＭＡＣ使用としてはあまり役に立たない。

入力文のタイプは、何時もよく使っているルーチン・ワークとか、計算機の熟練者ならその入力は命令型ばかりである。――― 冗長な記述がなく、必要最小限の完全なアルゴリズムからなり、もし金物がＰＤＳメモリ付きのものなら、そのままかけることができるような記述となっている。――― が、思考過程，定理証明のシュミレートなど、使いなれない使い方をする時には、冗長入力、質問型の入力が増加する。これは発見的処理を行なおうとすればさけられない点である。

またこのような会話を行なっていると、計算機から、あいまいさや不明確な点に関する質問が出され、それに対して、使用者からの応答入力が存在する。

結局、計算機への入力としては、命令、質問、応答の３種のタイプが考えられる。

現在のコンパイラの文型だと、データ記述は平叙文であるが、処理要求はすべて命令文でしか書けない。ＴＳＳでＭＡＣ使用を行なう為には、一般の人にも使い易くするように、質問文、応答文も使えるソフトを開発しなくてはならない。なおここで意図しているのはその外見的なSyntaxが疑問、応答型だというのではなく、その処理プロセスが最初から明らかにはなっておらず、試行錯誤を繰り返しながら、それを求めていくことであり、処理プロセスは、Deterministicではなく、Non-Deterministicである。

対話型では特に、不自由さを我慢して人間が、不具の機械にあわせるのではなく、機械が人間にあわせるべきである。

(ii) 図形入力の場合　ブラウン管にライトペン入力（Inquiry Display）というのが現在の入力媒体であるが、他にグラフ入力として、カーブ・リーダもある。図面管理としては、ホログラムも取扱えるようにするのが今後の課題であろう。これをオンラインで使用できると、非常に色々のことが可能となる。また現在のところオフラインであるが、マイクロフイルム、マイクロフイッシ等も、そのままオンライン入力ができるようにすべきである。

図形入力のレベルとしては、

①ある処理のマクロ命令のトリガとしての入力。これは手動ではなく光で動くスイッチとして機能する。最も低いレベル入力。

②幾つかの要素の中から、どれかを選択する入力。これは分岐点に於ける行先選択を行なうものである。①②は(iv)の型の入力の変型とみられる。

③グラフなどで数値、関数形を読み取る入力。グラフの縮尺、目盛から、入力数値を読みとったり、グラフ中の連続図形から、入力数値系列をＡＤ変換で読みとり、(2)の過程または、４．４．２の過程で、その関数の形（表現）を定めたりする。関数形の場合は、ＡＤ変換して、デジタルで比較するより、各種の関数形発生ライブラリをハードで持っておき、それらをディスプレイ装置上に再現させ、回転、移動、縮少拡大をアナログ的に自動的に行なわせ一致をとらし、関数形の形、パラメーターの数値を求める方がよい。精度をあげる時には、チエビシエフ近似等デジタルを使わねばならないが、図は図でアナログ比較するのがよいと思われる。

④ディスプレイ装置へライトペンの文字入力。　命令．データの入力をライトペンでブラウン管から行なおうとするもの。文字わくに入れ特徴抽出をすれば、文字に変換可能。

⑤図形でのものの入力。　図形から何らかの情報を抽出したり、図形を認識（分類）したり，図形の抽象化など、パターン認識の為の視覚入力であって、図形処理に於いて最も取扱いにくいものである。

以上①～⑤の入力により、①②は命令文に、③⑤はデータに、④は文字綴り（命令、データ）に変換され、以下は(1)と同様に論ぜられる。

(iii)音声入力　音声入力のレベルとしては、

①何種類かの処理のマクロ命令トリガの入力。

②会話としての文入力　がある。

これらの場合入力の最小単位が音素である点が(1)と異なり、音素の認識を行なえば、後はそれを組合わせた文字、単語、文と同様な話ができる。音素を認識し、それを組合わせた文字を打てるソノタイプの開発が必要。

(iv)接触入力　これはオルターネイトスイッチなど、人間が外から計算機のハードの接続を変えるもので、開始、中止、終了や分岐選択命令などの機械語に翻訳されると、以下は(1)と同様に論ぜられる。

(2)　入力文の内容解釈　質問分析といわれているもので、入力から制御命令とデータを分離抽出する過程である。換言すると、入力文を、命令コードとデータからなる命令語の系列に翻訳する。その型として次のものがある。

(I)数値計算の場合は、命令コードは、＋－×÷↑などの代数算法、データは数値となり、数式が命令語となる。

(II)属性値ファイル（例、給与計算の為の社員の基本給、歩合給、勤務時間数など）の属性値をデータとする時は、命令コードは、特定属性値の抽出とか、書換えとかになる。

(III)ことば（概念）をデータとする時には、それを分解翻訳して、意味空間を媒介に、(I)(II)の表現になおすのが命令コードとなる。この過程を真の情報処理といってもよく、その処理は一番難しい。

(IV)ことば（概念）をデータとするが、命令コードとして、意味の近いもの、包含関係にあるものの抽出、意味同等のものの列挙などがある。

4．4．2　処理

質問、命令、応答の分析を行なうと、処理アルゴリズムとデータが抽出されるので、アルゴリズムに従って処理を行なう過程である。処理は(1)二次情報のアクセス　(2)一次情報のアクセス　(3)データの加工の3種がある。

(1)二次情報のアクセス　計算機記憶部のファイルは大容量となるので、ファイル記録アクセスの為の索引となる二次情報が必要。また記憶容量の関係で、データそのものは入れてなくても、他所にある記録のリストを持つのが普通である。これらの二次情報リストから

必要な一次情報アクセスのてがかりを検索する過程で、この結果が直ちに答となる場合文献検索、研究者検索などがある。

この処理は、一致検出、大小比較、ランダムアクセス、論理和・積・不定等で行なう。

⑵一次情報のアクセス　(1)で検索した結果をもとに、データそのものを検索する過程である。これは百科辞典をひくのに相当し、いわゆるデータ検索である。この処理は(1)と類似の演算やマスクによる特定部分抽出で行なう。

⑶データの処理　(2)で必要データを抽出したが、そのデータを、処理アルゴリズムにより加工、処理する過程であるが、科学技術計算の場合などでは、サブルーチン、ライブラリを使用するなら(1)(2)の処理を通るが、そうでないものは、入力に必要全データが含まれており、最初から(3)の処理を行なうこの過程が計算機処理といわれているものの狭義のもので、(2)で求めたデータファイルから属性値最大のものの抽出、指定属性値をもつ個体の抽出などファイル加工と数値計算の代入式の繰返しとからなる。これは事項検索といわれているものに相当する。

また発見的（henristic）な処理を行なうのもこの過程である。これは先に目的を計算機に与えておき、それを解くのに必要なものを、会話を通して求めて行くものである。

4．4．3　応答（編集と応答）

前述の処理を行なうと、求める結果が得られた訳であるが、そのままでは人に受け入れられる表現形式をとっていないので、会話型の出力に編集しなくてはならない。出力の型としては(1)会話型文体　(2)図形表示　(3)表、リスト表示　(4)音声その他がある。

(1)会話型文体出力　入力のSyntaxの誤り（ミススペル・文法ミス）、入力文にあいまいさがある場合、命令、データに欠けた部分がある場合（発見的思考）、そのわからない所を知る為の質問を作成し出力する。また処理結果が非数値の文字表現された記録とか、数値解で一つ一つに注釈が必要なものとかの場合処理結果の骨組に、文法にあう肉をつけて、人が読みやすい文に作成する。

⑵図形表示出力　処理結果が数値解でグラフ表示できるものや、結果が図形の場合、また会話文をブラウン管で表示する場合、ブラウン管またはXYプロッタで出力するものなど出力にあうよう作図編集する。

(3)表、リスト表示　処理結果が数値解の時変数をパラメータとして表、リスト表示する。

(4)結果を表現した会話文を、音素の合成により音声出力として出す。またエラー出力のベルなどもこれに属する。

5 結　　論

2でのべたように、意味の位相性を考慮し、ことばの位相的なデータ構造を表現出来るような情報処理システムを考え、自然語に近い言葉による意味解釈を行なわすシステムを設計した。その設計に基ずき、単語の意味空間、論文概念空間の構成実験を行なったが、良好な動作を示した。

またそれをもとに、自然語での会話応答システムの処理過程を論じた。

意味の位相性とは、非数値の言葉で表現する概念の間の遠近関係、包含関係などが、すべての概念間で定義され、意味的に近いものはその近傍系として求まることをいい、体系分類や階層分類やシソーラスのように、その意味的な関連の強さを有無の binary表現する表現法に対立するものである。

3．1では、日本語の言語学的性質を論じたが、日本語は英語など印欧語に比べ字数が多く、特に漢字が多く表現が複雑なため、それらを直接入力しようとすると、入出力機器が非常に複雑になる。入力はカナで、出力は、デイスプレイ装置を用いて漢字で出すのが適している。次に入力にカナを使うと、単語の切れ目が判別しにくくなり、同音異義語の区別が不可能となり、外来語のカナ表現が一意でないものが多く、非常に読みにくくなる。その為にカナ綴りの標準文法を分かち書きのルールなど、早急に定めなければならない。

日本語の文法は、詞と辞で対になり、それの入れ子式構造で文を生成する。

3．2では「シーソラスとは、各言葉の表す概念の位相関係を表現する意味空間である」と定義し、意味空間を構成するプログラムを作成し、シソーラスの構成実験を行なった。

シソーラス＝意味空間という考え方は、聞きなれない人には奇異に感じるかもしれないが、従来のシソーラスが線的なものだったのに対し、これは面的にとらえており、従来のシソーラ

スを包含している。

意味空間は類似度解析法という。計量心理学的な手法を「ことば」という新しい分野へ適用し、意味の遠近関係の順序関係を変えないで、許容歪範囲内で、出来るだけ低い次元に圧縮された意味空間を構成した。

3．3では、3.2で単語で行なった意味空間の構成を拡張したものとして、論文の意味で同様な意味空間を構成する実験を行なった論文数が増加すると、論文の間の関係を調べる手数は、2乗に比例して増加するので、部分空間を構成し、次にその部分空間を要素として一段上の部分空間を構成する…… というように Recursive に空間を構成することにした。空間構成には、種々のパラメータを変えて収束の早いものを求めたが，距離空間のノルムとしては、4乗をとればよいことが判明した。

また論文概念の抽出を、人間がやるのではなく、自動化する方法について実験を行なった。これは、論文の見出しと内容梗概から、専門語を抽出し、属性空間を作り、そこに於ける各論文間の Hamming 距離を類似度として空間を構成した。人が行なったものと、自動化したものを使い検索をやったところ、歪は同じで、再現率は同じ位であったが、適合率は自動化した方が良かったので、自動化することが可能であることがわかった。

4．1においては、日本語の意味解読過程を検討し、その代数系モデルをのべた。Syntax（統辞論）と Semantics（意味論）は、表層－深層が 文法で定まる言語系に支配される統辞論の成立つ部分であって、いわゆる言語的性質というのは、このレベルでの話である。これに対し、深層－意味とよばれる2レベルが、本稿で問題にしている意味論の成り立つレベルであって、深層レベルというのは、文を生成したその生成過程情報を総合表現したPマーカを要素とする一つの言語レベルであって、意味レベルというのは、意味空間である。意味処理というのは、深層のPマーカと意味レベルの情報空間とを結ぶ写像であり、また意味空間中での写像である。

単語は意味空間中の点または領域を指定し生成規則は、意味空間中での領域の合成法を指定する。

4．2では意味空間の代数的性質について論じた。これは、書換規則により、単語から句、節、文を作っていくのを、意味空間上でみると、単語の概念を表す領域から合成概念領域を求

めることに相当しているから、意味空間内での領域の合成を行なう算法を抽出し、その構造をとらえようとしたものである。

書換規則として、連語生成則を選び、単語およびそれから出来る連語のなす意味空間を構成した。次に連語生成則に対応する意味空間上の算法を、合成概念点は、単語概念点の線型結合で表現出来ると仮定し、空間の実際の座標をもとに回帰分析を行った。その結果連語の場合はこの仮定が成立つことがわかった。

4．3においては、3．3の所で行なった実験結果を参考に論文概念抽出法の解析を行ない、最も良好な抽出法を求めた。

意味空間を部分空間に分割するに際しては、デイスクリプタとして使える言語に制限をつけ論文概念の間の類似度をとるのには、ディスクリプタには使用に制限をつけない方法がよい。また空間の歪、呼出率、適合率から判断して、概念抽出は自動化しうることも判明した。

4．4においては、以上の意味解釈実験をもとに、会話、応答過程の解析を行なった。この過程は (1)質問、命令、応答の分析と (2)処理 (3)応答の3段階に分かれ、一番問題となるのは質問の分析であって、この場合はその処理アルゴリズムが明確にされていないので、計算機にある程度の発見的な処理を行なわせるような能力をもたせなくてはならない。

Ⅳ （システム構成）

1. 緒　　論

Ⅱ、Ⅲで、樹枝状表現のデータ構造をもつ意味解読システムとそれを用いた、情報検索システム、および位相型表現のデータ構造をもつ意味解読システムとそれを用いた文献抄録システムを論じたが、それをもとに、出来るだけ自然語に近い言葉を使つた検索システム、そして、情報検索など非数値情報の処理も行なえるオンラインＴＳＳ自然語会話システムの構成法とその例について述べる。

2においては、オンラインＴＳＳシステムの現状を概観し、その構成上の問題点をあげた。

Ｑ—Ａシステムで問題となるのは、アルゴリズムのしつかりした命令形としての質問入力ではなく、普通の形の質問文が入つてきたときに、計算機がその処理アルゴリズムを自身で求めなければならないところにある。

現在動いているシステムの主なものについて、概観したが、意味論的な処理の行なわれているものは、見当らないようである。

またＴＳＳ会話応答システム構成上の問題点：端末機器、スケジューリング、ファイル構成、会話用言語についても検討した。

3においては、樹枝状データ構造をもつ自然語応答システムと、位相型データ構造をもつ自然語応答システムについて、そのデータ表現法を主に論じた。

樹枝状データ構造とは、体系的分類とか、階層性分類のように、概念の間の関係を樹枝状にとらえるものであり、シソーラスなども関係があるかないかをｂｉｎａｒｙでとらえるから樹枝状構造をなす。

これに対して、意味の遠近関係、包含関係などを近傍系として捉え、その関係を近傍空間または、距離空間の上に意味空間として表現しようとするものである。

4においては、3の意味取扱いシステムをもとに　オンラインＴＳＳ会話型処理の出来るシステムの、これからのあるべきシステム構成について検討した。

今後のＴＳＳ会話用応答システムは、情報処理センター網に加入し、リンクして動作することになると考えられるので、他センターのデータバンクの利用法が大きな問題となる。

データバンクの持つべき情報と、他センターの共通公開情報のオンライン利用法について論じた。またシステム構成上の問題点についても、その解決策を論じた。

5においては、Ⅱで論じた意味解読法と、4で論じたシステム構成から、文献情報検索システムを実際に構成し、その動作を調査し、検索システムの評価を行なった。

これは文献情報の情報空間（意味空間）を米国計算機学会の学会誌論文をもとに構成した。そして検索質問は、標準点とよばれる点との関連度で、情報空間内に写像し、その点に近い所にある文献を検索した。

色々とパラメータを変えながら、最も良いシステム動作を行なうように改良していった。

2. 遠隔制御会話応答システムの現状と構成上の問題点

2．1 質問応答システムの現状

question-answering system (Q-A system)の現状と問題点について述べる。Q-Aシステムを設計する際には、情報の蓄積又は、そのような蓄積を行なう方法そして質問が行なわれた時にその蓄積から適当な情報を抽出する方法を与えておかねばならない。その蓄積は簡単な叙述文の形で表わされた情報に基づいて作り上げられる場合もあるし、前以つて用意されたデータ構造である場合もある。どちらの場合においても人々が通常互いに情報伝達を行なつている言葉で表わされた情報を含んでいる。Q-A systemの基礎である情報の蓄積は、同じ情報を意味する任意の文章のモデルと考えられる。もちろん、ここで〝同じ情報を意味する〟とは情報量が同じということではなしに、意味的な内容が同じであることを意味する。現在自然語の意味的な解析は、まだ暗中模索の状態であるので，このQ-A systemにおける蓄積情報の内容を最も効果的に調べる方法は、このシステムにいくつかの質問を行ない、システムの意味解釈機能から間接的に知ることである。システムの蓄積情報は、叙述文の集合に対するモデルである。というのは、その蓄積情報の内から抽出される情報はその文章において許される情報（実際には同一でなくてはならないが）少なくともある部分に定められた方法で対応づけられている。このようなモデルの最も有利である点は、そのモデルから望む情報を同一視したり、抽出したりすることが、全文に対してそのような、操作を行なうよりずつと簡単だという点である。Q-Aシステムは、種々のこのようなモデルを用い、様々な段階の成功を修めながら開発されてきた。

以下にＱ－Ａシステムの実施例について述べる。

２．１．１　Ｑ－Ａシステムの実施例

いくつかの計算機プログラムが、意味論的な検索システムの目的や結果と関係して書かれてきた。これらのＱ－Ａシステムは、任意の意味情報のモデルを使用していないし、意味論的な検索を行なえる程一般的な主題を取り扱つていない。しかし、それぞれのシステムは興味ある特徴を持つているので、遠隔制御会話応答システムの設計に参考となるものと思う。

(1)"Baseball"　ＩＰＬ－Ｖで書かれたこのプログラムは、野球に関する質問に答えることが出来る。例えば

```
input  :  "How many teams played in 8 places in July?"
output :  MONTH=JULY
          PLACE number of=8
          TEAM  number of=3 : YANKEES,
                              TIGERS, REDSOX
```

蓄積情報は、前以つて選ばれた階層構造に従つて並べられた関連のある野球の結果を総て含んだリスト構造から成つている。このモデルを自動的に修正する機能は、もつていない。各質問は、望む情報のリストとそのモデルとを比べ合わせ空白の部分をうめる。そして、完全な最終リストをプリントアウトする。

このプログラムの大部分は、質問文を質問リストに翻訳する仕事にさかれている。この仕事には、辞書を引いて見出し語を探す、慣用語を同一視する、文法的解析を行なう、冗長を除去する、などがある。辞書は、その言葉の品詞、慣用語、そして意味とから構成されている。ある特定の言葉に対してのみ現われる意味は、プログラムの構文中で、標準語に変換される。このようにして、主題の特殊な性質を簡単にすることが出来、他の場合なら非常に困難であると思われる問題も簡単に解決出来る。情報の構造が固定しているこのモデルは、質問リストの空白を埋める操作が容易に出来るように配列されている。

"Base ball"システムは、種々の形の質問に答えることが出来るので、知的な行動を行ない得るかのような錯覚を与える。しかしながら、特殊な主題についての限られた情報が固定したデータ構造の中に、前以つて並べられている必要があるし、データは階層的な順序

関係に適応するものでなくてはならない。このような体系を真の知的なシステムに必要な広範囲な情報を取り扱うのに便利なように一般化することは不可能である。

(2) 'Question-Answering Routine' LISP で書かれたこのプログラムは、英語の単文の集合に基づいて、ある種の簡単な質問に正確に答えることが出来る。例えば、

```
input : ((AT SCHOOL JOHNNY MEETS THE TEACHER) (THE
          TEACHER READS BOOKS IN THE CLASSROOM))
        (WHERE DOES THE TEACHER READS BOOKS)
output: (IN THE CLASSROOM)
```

文のモデルは、主語、動詞、目的語、場所、時の5つの要素から成るリストである。このモデルは、corpus 中の文と質問文に対して構成される。質問リストが各文章リストと比較され、もし適当な一致した文章が見つかったならば、正しい解答が元の corpus から対応する文章が抽出される。このシステムには、次のような欠点を持つている。5つの基本的な要素以外の文章に含まれている情報と解析出来ない文章は無視されている。質問文中の言葉は corpus 文中のものと同じでなくてはならない。全corpus に対するモデルが各質問に答えるために線形に探索されなくてはならない。しかしながら、新しい文章が附加出来るように創造され拡張されたモデルそして、質問に答えるのを助ける中間形を許すモデルの考え方は、知的な人間的なシステムの基本的な特徴

(3) 'SYHTHEX' JOVIAL で書かれたこのプログラムは、Golden Book Encyclopedia のようなものに含まれた情報の質問に答えることが出来る。例えば

```
input : 'What do birds eat?'
   (somewhere in the encyclopedia)  :'Worms are eaten
     by birds'
output: Birds eat worms'
```

このプログラムは、全ての言葉を構造的(構文的)重要性をもつた function word (例えば、the, is, do, what) と意味論的重要性をもつた content word (実際には function word でないものを content word とする) とに類別する。先ず corpus (encyclopedia) に全ての content word の出現頻度に基づいて索

引付けがなされる。この索引はcorpusそれ自身と殆んど同じ広さの領域を占める。質問がなされた時、質問と共通のcontent wordの出現頻度が最も大きい文章がcorpusから選ばれる。この時点で選ばれた文章のうちどれが質問に対する解答を与えるのに適しているかを決定するため、面倒な文法的解析が行なわれる。このシステムでは、モデルを全く使つていない。完全なcorpusが元の形のままで保持されており必要に応じてその索引が用いられる。情報は使用しやすい形に前以つて処理されていないから、質問に答える際に必要となる文法上の解析は大変複雑なものとなる。Kleinは、文法の規則のいくつかは、corpusから自動的に作成出来、いくつかの文章からの情報は、構文的な方法を用いることにより、結合出来ることを示している。

これらの'dependency grammar'によつて作られた言語の関係は、意味論的解析によりもつと簡単に見つけられるだろう。このような意味的関係に基づいたモデルは、質問応答過程を単純化出来るだろう。

(4)' SAD-SAM: Sentence Appraiser and Diagrammer, and Semantic Analysing Machine' IPL-Vで書かれたこのプログラムは、基本英語で書かれた任意の文章をメカとして受け入れ、それから血縁関係に関する任意の情報を抽出する。そして'family tree'にこの情報を付加する。

例えば、input : John, Mary's brother, went home'

effect: John と Mary は両親の共通集合を割り合てられる。すなわち、それらはfamily tree の共通の頂点のdescendantとして表わされる。その文法は、血縁関係を認識する際に、自然語の考慮すべき部分をとり扱えば十分である。著書は、質問応答について詳しく考えてはいないが、血族関係モデルは、樹枝状モデルに直ちに適応出来る。又、特殊な質問に対しては、殆んど普通に答えることが出来る。

このシステムは、非常に特殊な仕事のために設計されたモデルの効果を例示している。Lindsay は血族関係のみが興味あるものと決め血族関係の'natural'モデルが存在することを見抜いていた。残念ながら、異なつた種類の情報に対しては、異なつた形の'natural'モデルが必要である。もつと一般的なシステムにおいては、各々の主題分野を表わすのに最も適切なモデルを用いることは可能である。例えば血族関係に対しては、樹

枝状構造、空間の関係に対しては、デカルト座標、などのように。

しかしこのようなシステムは巨大な組織問題をかかえた複雑なものとなるだろう。ＳＩＲシステムは、種々の特殊モデルの利点のいくつかを備えた１つのモデルを基礎としている。そして蓄積の場合と同様に一定不変の処理過程を許す。かつ、人間の会話において表われる任意の事実を検索出来ねばならない。

いくつかのＱ－Ａシステムが特殊な問題を解くためとか、特別な機能を例示するために開発されてきたが、計算機の知的な理解動作を与えるための直接的な接近法を構成しているものはない。又、種々のモデルが用いられているが、意味論的な関係を与えているものはない。

２．２　遠隔制御会話応答システムの問題点

会話応答システムの問題点については、前節で述べたので、ここでは主として遠隔制御の問題についてふれる。

２．２．１　端末機器の問題

遠隔制御会話応答システムは、オンラインリアルタイムで、人間と計算機とが会話形式で情報の交換を行ないながら処理を行なつていくので、人間と機械のインターフエイスとなる端末機器が問題となる。オンラインシステムの端末機器は、その使用目的から(i)特殊目的用（特定の目的、例えば、列車の座席やホテルなどの予約装置、工場などにおけるデータの収集装置などに使用される。）(ii)一般会話用（利用者と計算機とが会話的に情報のやりとりが出来る。）(iii)リモートバッチ用（遠隔地のコンソールからカードスタックをオンラインで計算センターに送り、その処理は通常のバッチ処理で行なわれる。）(iv)グループ通信用（軍のシステムや証券取引所の株価表示、競馬場の売上げ状況表示などに利用）に分類出来るが、ここでは、(ii)の一般会話用について考える。現在よく用いられているものには、タイプライタ式のものとキーボードとＣＲＴデイスプレイを組み合わせたものがある。これらでは、使用言語や入出力情報の形式は、特定のものに限定されず、open-ended なサービスが行なわれる。しかしこれらの形式では、人間と計算機との情報のやりとりを十分に行なうことが出来ないので、グラフイックデイスプレイコンソールが開発されているが、コストの問題が残されている。今後の開発が期待されているものに音声応答装置がある。これが完成すれ

ば、人間と計算機の会話は、非常に楽なものとなる。

2.2.2 スケジューリングと時間割り当ての問題

オンラインリアルタイムシステムでは、利用者とターンアラウンドタイムを同調させる必要がある。ターンアラウンドタイムがあまり速すぎると、利用者のレスポンスが遅れ、計算時間の無駄が生じる。又遅すぎると利用者が計算機を専有している感を抱かせない。これを解決するためにＴＳＳ等の技術が研究されている。

2.2.3 ファイルの問題

多数の利用者がプログラムやデータを収めるために、大容量で応答速度の速い、ランダムアクセスのファイルが必要である。又、そのファイル・メインテナンスの問題としては、私的なファイルの機密保持（パス・ワードの問題）○共同利用ファイルの保持　○ファイル相互の干渉の防止　○各端末からのファイル変更がシステムのファイル全体に影響を及ぼさないことなどがある。

2.2.4 会話用言語の問題

理解しやすい言語である必要があるので、なるべく自然語に近いものが望ましい。又、オンラインでは、速いコンパイリングが主要な条件となる。このために１パスにしたり、ロードの時にリンクするとか、インクレメンタルなコンパイラにするなどの方法が答えられている。

2.2.5 その他の問題点

スワップ用の記憶装置の開発、通信系における符号形式などの標準化、異常事態に対する処置の問題などがある。

参考文献

○ M·Minsky "Sewantic Infovwortion Puocessing"

○ 大野　豊 "オンラインシステムの概説" 情報処理学会　vol·8, No 6, １９６７

3. 樹枝状型自然語応答システムと位相型自然語応答システム

樹枝状型自然語応答システムとは、そのシステムでの意味のとらえ方を、ことば（ことばの表す概念）を意味的な包含関係や、意味の近いもののグループ化などにより、意味の関係を樹枝状に配置したもので、その樹枝状図は、体系分類や、辞書編集順の列挙であることが多い。

このような意味のとらえ方のもとに、データのファイル化、会話応答時の意味解釈、検索時の連想などを行なうシステムである。

これに対し、位相型自然語応答システムとは、そのシステムでの意味のとらえ方を、ことばの表わす概念の意味的な遠近関係を、もう少し連続的な位相的構造としてとらえ、その関係を意味空間に表現する。この空間を何等かの形で処理しやすいファイル（ハードの連想記憶、ソフトによるランダムアクセス的なファイル）に構成し、それをもとに、会話応答時の意味解釈、連想などを行うシステムである。

3．1 樹枝状自然語応答システム

樹枝状自然語応答システムにおいては、意味のとらえ方、表現方法を、ことばの表す概念の意味的な包含関係や、意味の近いものを繰返しグループ化して、まとめるやり方で、意味の関係を樹枝上に配置したものであり、意味的な包含関係、上位下位概念は、体系的分類の階層性およびシソーラス（類語集）のＢＴ（広義語）、ＮＴ（狭義語）によつて表現する。その表現例を図Ⅳ－１に示す。

意味的な遠近関係は、分類表によつては、あまり正しく表現されないが、分類表でどの程度離れているかで、ある程度推察する。シソーラスでは、ＲＴ（関連語）で表現する。しかしこの方法では、関連があるかないかの二値関係しか表現出来ず、より連続的な遠近関係を表現することが不可能である。この点が樹枝状自然語応答システムの大きな欠点である。もう少し遠い意味の遠近関係を知ろうとすれば、シソーラスのＲＴを繰返しひいてゆくことにより行なえるが、シソーラスをひく操作はかなり手数がかかりよい方法とはいえない。

デンシケイサンキ

UF　コンピュータ

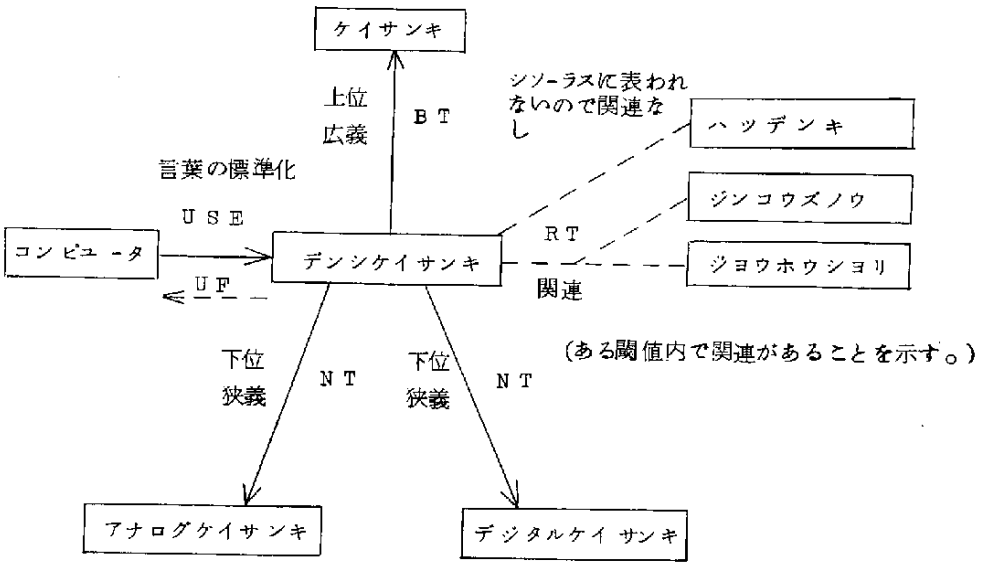
NT　アナログケイサンキ

　　デジタルケイサンキ

BT　ケイサンキ

RT　ジョウホウシヨリ

　　ジンコウズノウ



図Ⅳ－1　シソーラスの表現とその樹枝状構造

意味の包含、上位下位概念を表わす体系的分類による樹枝状表現としては、次のようなものがあげられる。

(1) 日本十進分類（ＮＤＣ）

(2) 国際十進分類（ＵＤＣ）

(3) デューイ十進分類（ＤＤＣ）

(4) コロン分類（ナンガナターンによる）（ＣＣ）

(5) 科学技術文献速報、分類表．索引項目表（日本科学技術情報センター）

(6) 帝国図書館図書分類表

(7) 東大中央図書館図書分類表

(8) アメリカ議会図書館分類表（ＬＣ）

(9) 展開分類（ＥＣ）カッター

(10) 国際分類表ライダー

(11) 件名分類ブラウン

(12) 書誌分類ブリス

(13) 中国科学院図書館図書分類

(14) 中国人民大学図書館図書分類

(15) ソ連邦図書館書誌分類表

(16) 産業分類　日本標準産業分類

(17) 産業分類　国際標準産業分類

(18) 日本標準職業分類

(19) 日本標準商品分類

(20) 国際特許分類（ＩＰＣ）

(21) 日本特許分類、実用新案分類

(22) 米国特許分類

(23) 西独特許分類

(24) 英国特許分類

(25) 機械工業海外情報分類

(26) ＡＳＭ－ＳＬＡ分類（米・冶金学分類）

(27) 仏統計経済ドキュメンテーション十進分類

これらの分類からいえることは，

(1)完全な一次元体系分類を行なうことは不可能である。階層性は殆ど全部のファイルが持つているが、体系分類ではない。

(2)一次元階層では体系分類が不十分なので、その対策として多次元体系分類（完全な体系ではなく、多次元階層分類というべきもの）を行なったものがある。これはまたカテゴリによる分類表ともいえる。分類の (4) (25) (26)などは、そのような分類とみてよい。

(3)一次元または多次元分類だけは、複雑な内容の表記には十分でないので、それらを補助的な結合子などで結合し、もう少し詳しい表現も出来るようにしてある。この結合子としては、WRUのロール・インデイケータ、ＡＩＣＨＥのリンクとロールなどがあげられる。また結合子ではないが限定子的な働きをするものとしては、ＵＤＣを始めとするこの系統の分類に大体ついている。

(4)以上のような分類の組合わせをやつてみても、各人により、同内容を違う言葉で表現することがある。特に表現内容をコートではなく、より自然語に近い言葉で行なおうとすると、用語の統一が問題になる。その結果シソーラス（類語集）を持つ。シソーラスとしては、次のようなものがある。

ＥＪＣ　（米国技術者連合会議）

ＮＡＳＡ　（米国宇宙航空局）

ＡＩＣｈＥ（米国化学工学会）

ＡＳＭ　（米国金属学会）

ユートラム、ヨーロッパ原子力共同体

ＡＳＴＩＡ（米国防省ドキュメントセンター）

Ｒｏｇｅｔシソーラス（英語一般）

結局(1)(2)においては、データの入れものの指定、(3)(4)においては、個々のデータの簡潔な名札づけ的な要素が強い。

このような表現方法から考えると、意味的に近いものなどを調べる為には、列挙式または、階

層式分類で、求めるものが入つていそうな区分を調べ、後はそこに入つている内容を(3)(4)などの関連の有無、有るとするとどんな関係かを一つ一つ照合しながら求めるものをさがすことになり、あまり能率はよくない。

その理由は、自分の知りたいものは、分類表でいえばどの分野に入るかを知らなくてはならないが、なれてなかつたらそれを調べるのも面倒で、時にはシソーラスの働きを借りなければならず、分野がわかつても、そこに収納されたものの中で自分の必要なものをとり出す為には、その分野の全データを調べねばならない。また分類が完全な体系分類でないので、一つ自分の知りたい分野がわかつたからといつて、他には自分の知りたい分野がもう無いということを保証されず、もれがありうる。

応答会話システムはリアルタイム（オンライン）で動作するのが普通なので、処理時間は、人間の思考速度にあつたものでなくてはならず、長く待たせるようであつてはならない。そして、意味的処理は、検索が総てではなく他の処理も色々と行なう（Ⅲ・4・4参照）ので、検索にはより少い時間しかさきえず、全列挙式のリストを計算機にひかせたり、人間が外部リストをひいてコードに変換するのは時間的に無理があり、後者の場合はコード変換を行なうことにより前述のような情報損失を起すおそれがあるので、出来るならさけたい。

3．2　位相型自然語応答システム

位相型自然語応答システムとは、Ⅱでものべたが、意味のとらえ方を、ことばの表す概念の遠近関係、包含関係、上位、下位概念や概念の同一なものの判定ができることとしてとらえ、これらの関係を満すような意味空間を考え、その空間の上で、同一概念の選択、ある与えられた概念と包含関係にある概念の抽出とか、意味的に近い概念、与えられた条件を満す概念、合成概念など、概念の加工、処理のできるそういう空間を構成することにより、樹枝状データ構造では出来なかつた概念の加工、処理が行なえるようにしたものである。

このような条件を満す空間（意味空間）は、近傍系であるが、近傍系では、計算がやりにくいので、分離公理も入れて距離空間として考える。

このような考え方で言葉の表す意味の間の関係をとらえようとしているものとしては、次のようなものがある。

(1)自動抄録法（P・H・Luhn の方法）の修正

これは対象が、沢山の文章からなる論文よりのものでないと行なえないが、抄録したい文の集合を計算機に読ませ、文中の各専門用語の出現頻度を数える。専門用語の選択は、非専門用語リストがあり、それにないものを専門語として取扱う。次に専門語を含む文章をとり出し、専門語の含まれる語数がある閾値に上の文をとり出して並べたものを抄録とする。以上がLuhnの方法であるが、これをもとに論文間の意味的な関連度を調べるには、専門用語の出現頻度の相関係数を計算し、それでもつて関連度とする方法が考えられる。また論文間の包含関係は、表われる専門用語のシソーラスによる上下比較により統計的に上下関係を判定する。

(2)連想係数決定法（H・E・Stiles）の応用

これは単語間の相関度（連想係数）を、それらの単語を使う文献数の相関度で計算しようとするものである。

自動的に統計抽出する方法はまだ他にもあるが省略する。

(3)多数の人にことば（概念）の間の遠近関係、包含関係のアンケートをやつてもらい、それから統計的に関連度を抽出する。

(4)多数の辞書、シソーラスの意味の遠近、包含関係から、単語間の遠近関係を求める。

以上のような方法で、意味の位相的性質に関する情報がぬき出されたら、それをもとに因子分析を行なって、因子軸を座標軸としてとり空間を作つたり、意味の遠近関係の順序関係または，遠近比の比の値を保持するような空間を作る。（Ⅲ・2・1・3参照）

なおこれの簡略化した形として、グラフ表示または、接続行列表示したものがあるが、その場合は、概念の数をnとすると、n×nの行列が必要で、検索するにも、一応全概念との関連度をとつて調べなければならない。

これに対して、空間圧縮した意味空間は、検索による誤差が気にならない程度の歪を許すなら概念の数の2／3～1／2の大きさの次元に圧縮出来、記憶容量が少くてすむ。また、Ⅲ・2.1.4でのべたように、空間をファセット分類しておくと、検索時における照合も、全データとする必要がなく、意味的に近いブロックから調べてゆけ、その意味的な連想による他概念を求める過程は、好きな所で打切れる。その為冗長な照合が不要となる。

3．3　樹枝状型データ構造をもつ自然語応答システムと、位相型データ構造をもつ自然語応答システムの比較

3．1および3．2で、二つのデータ構造をもつ自然語取扱いシステムについて論じてきたが、両者の特徴を比較して、表Ⅳ－1に示す。

表Ⅳ－1　樹枝状自然語応答システムと位相型自然語応答システムの比較

	樹　枝　状	位　相　型
意味的な遠近関係、包含関係表示	体系分類 辞　書 一覧表リスト シソーラス	意味空間 座標 距離 空間の分割 部分空間の体系分類（Facet）
連　想	シソーラスのRT NT．BTでたどる。	大まかには、Facetでたどり詳しくは、問題点（質問点）と各データ（概念）との距離を求め近いものをさがす。
連想の度合いの調節	RT・NT・BT　でたどれるのみ、かなり段階的で、きめ細かな調節は不可能	連続的に可能
問題点（質問点）の表現	シソーラスに表われる用語でないと、後、意味的に近いものをさがすのはむつかしい。体系分類からひけるのみ。	問題点を空間内に写像しなくてはならない。その度の言葉－空間対照表及び言葉の組合せ方と意味空間中での意味合成やり方の対照表が必要、しかし、それがあると自動的に写像可能
ある概念へのアクセス、その手数	・シソーラスをひき、体系分類のどこにあるかを知る。 シソーラスがランダム・アクセスならはやいが、シーケンシャルファイルな	ことばと意味空間の中の座標とを対照表にしたリストをひき、部分意味空間へアクセス。 KWICを直接空間に構成すれば、空間

	樹　　枝　　状	位　　相　　型
	ら非常に長い手数がかかる。 ・直接体系分類をひくなら全ファイルのルック・アップが必要。 ・KWICなどをひくときには、さがしたい論文を表わすような専門用語を思いうかべなくてはならず、シソーラスをひいても、もれがないとは保証されず、ひき方に欠点が残る。	へアクセスするだけで、後は、部分空間中で距離を計算することにより、全部をルックアップすることなく、関係の近いものから求めてゆくことができる。 無駄なルックアップをしなくてすむ。
維時，更新 変　更	体系分類は追加は可能、変更は不可能（変更するとファイルも殆ど全部変更しなくてはならない。）	少しなら意味空間へ追加可能、沢山になると再構成を要す。

4. システム構成

前節の意味取扱いシステムをもとに、オンライン会話型処理の出来るシステムの、これからのあるべき、システム構成について検討する。

4．1 対象業務

遠隔制御会話応答システムの行なうべきサービスとしては、次のようなものがあげられる。

(1)計算機と対話しながら、発見的な問題を解いてゆく。科学計算、数学証明、各種設計等。

(2)これからの遠隔制御会話応答システムは、情報処理センター網に当然加入するから、

中央にある大きな、メインデータバンクの情報（公開共通情報）を利用する。

必要事項の検索、各種予測、計画評価。

(3)各地区にあるサテライトデータバンクの公開共通情報を（他地区のも）利用する。

必要事項の検索、各種予測、計画評価。

(4)各端末局の有する私有公開情報の利用（有料のこともある）データバンクにおくほどの共通性

のない公開情報および自局で管理するほうがよい公開情報、ライブラリはＰｒｉｖａｔｅ Ｆｉｌｅとなる。必要事項の検索等。

(5)メインデータバンクまたはサテライトデータバンクに格納してある自分の私有非公開情報の利用。自局の記憶容量が小さいとき、中央のデータ・ロッカーを利用する。

(6)特定の仕事を行なうオンラインリアルタイムシステム。列車、航空機の座席予約、普通預金業務、生産管理、在庫管理など。

(7)データの交換。　電信電話交換

為替管理、ＳＡＧＥシステム

(8)データの収集と分配。交通管制、航空管制

(9)リモート・バッチ処理。端末局から、中央ＣＰＵを使い、バッチ処理を行なう。

以上のうち(2)～(5)は情報処理網に関連したもので、(6)～(8)は、会話型ほどの自由さを必要とせず、単能の機械である。

この大規模システムは、当然時分割使用システム（ＴＳＳ）となるが、その場合、上記のＦｏｒｅｇｒｏｕｎｄ Ｊｏｂに対し((9)など Ｂａｃｋｇｒｏｕｎｄ Ｊｏｂに近いものもあるが)、Ｂａｃｋｇｒｏｕｎｄ Ｊｏｂとして、処理周期の長い大きなルーチンワークなどのバッチ処理を行なう。以上のシステムの持つべき、公開共通情報の種別、用途、処理などの一覧表を表Ⅳ－２に示す。また非公開私有情報について表Ⅳ－３に示す。なお(9)については、4.3.2.（１０）、4.3.3. 参照。

４．２　システム構成

対象としている計算センターは、各地区の情報処理センターと、中央情報処理センターからなる情報処理センター網に加入する。

各地区情報処理センターは、その地区で通用するローカルなデータを、付属のサテライト・データバンクに持つ。また全体的なデータは、中央情報処理センターのメインデータ・バンクにファイルされる。範囲を図にとるなら、ＮＩＳＴなどはこのようになる。情報処理センター網を図Ⅳ－２に示す。

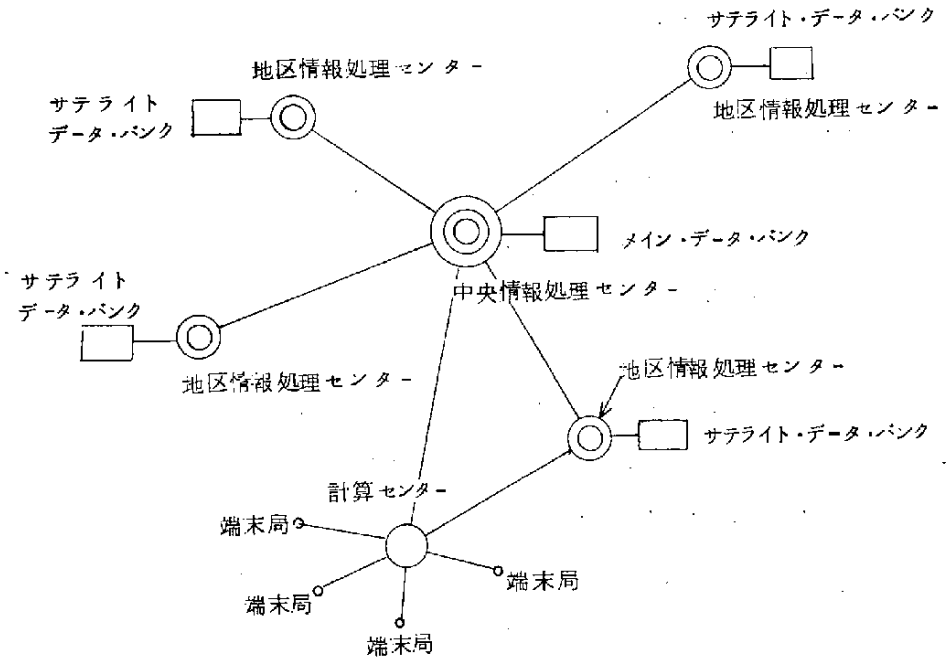
計算センターは、入出力制御のサテライト・コンピュータを通じ、通信回線制御装置を経て

表 Ⅳ－2　公開共通情報一覧

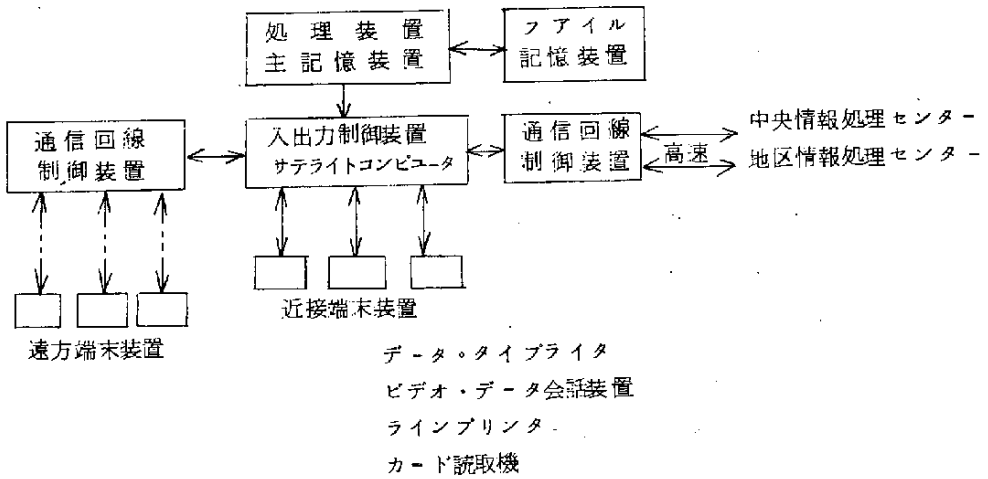
	情報種別	情報用途概略	情報の形	処理内容	サービスの型
A	経済動向 国民生活 財務統計	需要予測、設備計画 商品計画 マーケティング	統計 調査報告	情報検索 統計計算	2
B	労働動向	雇用計画	統計 ニュース	情報検索 統計計算	3 (2)
C	特殊技能者動向	雇用計画 委託研究、委託業務	統計 一覧表（技能／所在／勤務先） ニュース	情報検索	2 3 4
D	商品現場、物価 荷動き、株価	購売 営業	ニュース	情報検索 統計計算	2 3
E	製品、部品、材料	購売、営業、設計 マーケティング	カタログ資料 仕様書	情報検索 統計計算	2 3 4
F	生産技術	設計、製造	方法、データ	情報検索 計算	
G	管理システム	経営管理システムの設計	方法、データ	情報検索	
H	文献、論文 研究報告、特許	研究開発 設計	二次情報リスト 本文	情報検索	2 3 4
I	百貨辞典 辞書	知識源 発見的思考	辞典、辞書 類語集	情報検索 意味処理	1

表Ⅳ－3　非公開私有情報一覧

情報種別	情報用途	共に使う公開情報	情報の形	処理内容	サービスの型
生産量、決算 販売記録 受注資料	生産計画 販売計画	A、D E	データ、統計	ファイリング 計算	5、6 (2) (3) (4)
人事、要員資料 給与資料	人事管理 労務管理 給与計算	B、C	データ、統計	ファイリング 計算	5(2)(3) 9 (4)
在庫情報 (資材、部品、製品)	在庫管理 購買計画	D、E	データ	ファイリング 計算	5、6 (2) (3) (4)
技術資料 (設計、構成)	生産技術管理	F H I	方法 データ	情報検索 ファイリング 計算	1 (2) (3) (4)
財務諸表 各種計画	経営管理 意志決定	G ABD	データ 統計	計算 heuristic な決定	1 (2) (3) (4) 5



図Ⅳ－2　情報処理センター網



図Ⅳ－3　計算センターの構成

高速伝送回線で地区、中央の情報処理センターへ接続され、必要に応じ各地のデータバンクおよびライブラリを使用出来る。

なお交換機能とか、データバンクの蓄積に関しては、中央一ケ所に集中してしまうのは、中央の障害で総てが止まるので、危険であり、信頼性の点からいつて、網目状にし、機能も中央集権にせず、分散させなくてはならない。計算センターの構成を図Ⅳ－3に示す。

使用者は、入出力制御装置を通じ、データタイプライタ、ブラウン管データ会話装置、ラインプリンタ、カード読取、さん孔機から入出力できる。

また計算センターを通じ、中央、各地のデータバンクおよびライブラリをオンラインで使用できる。この場合、必要全データを計算センター迄送つてもらいそこで処理する方法と、データバンクをもつ情報処理センターの計算機をリモート・ジョブで使用する方法とがある。対象データが大容量の場合は、自分の計算センターで、プログラム作成、デバッグを行ない、それらを目的地の情報処理センターへ送り、そこで4.1の(1)、(9)の処理を行なう。

4．3　オペレーテイングシステムの構成

オペレーテイングシステムは　(1)制御プログラム　(2)処理プログラム　(3)支援プログラムから構成される。

制御プログラムは　(i)データ管理　(ii)ジョブ管理　(iii)タスク管理からなり、処理プログラムは、(i)言語処理プログラム　(ii)サービスプログラム　(iii)アプリケーションプログラムからなり、支援プログラムは　(i)システム開発用プログラム　(ii)システム保守用プログラムからなる。

オンライン・リアルタイム、遠隔制御会話システムにおいては、システム資材の有効な利用を行なう為に、ジョブ管理（スケジューラ）、タスク管理（スーパーバイザ）に関連した入出力管理プログラム（ＩＯＣＳ）が大きなウエイトを占めてくる。

また制御命令としての、コマンド、支援プログラム（エラー検出など）の能力も備えた、言語翻訳プログラム（プロセッサ）をもつ会話型言語も大きなウエイトを占める。また多数の利用者のプログラム、データを保存しなくてはならない為、ファイル構成も大きな問題点となる。また主記憶装置の容量はそう大きくないので、多人数が同時に共用する為にその時分割、容量分割の使用法（スケジューリング）も大切である。

以下各部について更に詳しく検討する。

4．3．1　コマンド

コマンドの機能としては、(1)登録、終処理　(2)翻訳　(3)実行　(4)使用法指導　(5)情報処理センター網とのInterruptの命令　(6)ファイル、ライブラリの保存、引用、加工等がある。

各機能について概説すると

(1)登録、各処理は、I／O、使用ファイルの指定や、料金勘定などを行なう。

(2)翻訳　ALGOL、FORTRANのコンパイルを行なわす。

(3)実行　コンパイルされたオブジエクト、プログラムのスタート、ストップなどを行なう。

(4)使用法指導　コマンドの使用法がわからないとき、また誤つて使つたとき、正しい使い方を指導する。

(5)情報処理センター網とのInterruptの命令。

必要なデータが自分の計算センターになく、他のデータバンクにある場合、それをオンラインで使う為の命令。使い方には、そのデータを処理する部分プログラムだけ、データバンクをもつ情報処理センターへ送り、そこで実行させ、結果を返送してもらい自分のファイルに格納し、計算を続ける方法と、データバンクのデータを自分のファイルに転送してもらい、自分の所で実行するやり方がある。このコマンドは、現在のTSSシステムのコマンドには無いので作らなくてはならない。

しかしもう一歩進歩したシステムでは　ファイルがどこにあろうとも、プログラムのステップ数と、データ量からどちらで実行したらよいかを判断し、使用者は実行命令を出すだけで、計算機が適当な場所を選んで実行するようになる。

(6)ファイル・ライブラリの保存、引用、加工。

他人のファイルのコピー、計算結果の保存、その他各種加工。

現在のTSS用コマンドは、非常に簡略化、記号化されたものが出廻つているが、これはあまり使いやすくない。その理由は、コマンド命令の後に、沢山の変数やパラメータをつけなくてはならず、そのつけ方が単に列挙するような方法で、簡単にお覚えられるようなSyntax構造をしていないからである。

コマンドは、もつと自然語に近いSyntax構造をもち、省略による暗黙の標準型指定も可能にし、計算機からはその使用法を詳しくコーチしながら実行し→(4)それに対する人間の入力は出来るだけ簡単に、かつ自然語のSyntaxにあう型で入力することもできるようにすべきである。そうすれば未熟使用者は、コマンドを対話形式で使え、熟練者は簡潔にも、複雑にも使えるシステムが構成される。

また現在のコマンドは、固定形であるが、自己拡張形コマンド――使用者が次々とコマンドを定義し、それをコマンドとして使用出来るシステム、その為には、最初必要最小限の自己拡張形システム記述言語をもてば、後は各人に適したコマンドが作成出来る。―― にするとずつと使いやすくなる。

また(3)、(5)に両方とも関係するが、ForegroundとBackground との相互移行命令をもつこと。Foregroundは、アルゴリズム発見用、プログラム作成用、デバッグ用、中途検索などに使用し、出来た大きなプログラムとかファイルの処理はBackgroundにまわり、性能のよいコンパイラと大容量メモリを使い処理を行ない、結果をその処理センターでLPから打出して表を作成したり、結果をForegroundに持ち帰つて対話処理を続行したりする。プログラム作成とか、発見的思考は計算センターで行ない、そのプログラムの実行に中央情報センターのデータ・バンクの大データが必要な時には、オンラインで中央情報処理センターのForegroundへプログラムを移し、次に、ForeからBackgroundへ移行し、プライオリテイの高いバッチ処理をしてもらい、結果をもつて、Foregroundに戻り、オンラインで自分の所の計算センターのForegroundへ移り処理を続行するという使い方もある。

4．3．2 会話用言語（プロセッサ）

会話用言語は、自己拡張形システム記述言語から出発して出来たコマンドであれば非常に使いやすいが、ここでは一般に会話用言語の持つべき性質について、列挙する。

(1) 人間がプログラムを組まなくても、機械と対話をやつていくうちに、発見的に（heuristic）そのプログラム（アルゴリズム）を構成することができること。

(2) その為に、コンパイル、実行の切換えが何時でもでき、任意の命令や 部分プログラムの実行が出来ること。

(3) その為に、コンパイラは、インタプリテイブな動作も行なえること。一般にインタプリテイブなコンパイラで翻訳すると、プログラムは最簡化されない（冗長のある性能の悪いプログラムとなる）ことが多いので、別に最簡化できるコンパイラもあること。インタプリテイブな動作が出来るという点では，現在ＴＳＳで最もよく使われているＦＯＲＴＲＡＮよりはＡＬＧＯＬ、ＰＬ／１の方が良い。

(4) ソースプログラムの一部の修正、変更がオブジエクトプログラムの局所的な変更ですむこと、その為にも (3)が必要

(5) 文法ミスに対しては、常時デバッグを行ない、エラーメッセージではなく、訂正の仕方または、幾つかのうちのどれかを選択するような形、またはエラー語を問返すような動作をし、ミスの度にマニュアルをみなくてもオンライン、リアルタイム使用ができること。

(6) アルゴリズムの発見を行なう為に、それ迄の会話内容の蓄積と、それらの色々な組合せを自動的に行ない、その組合わせで、何かを計算するアルゴリズムが出来たら、そのアルゴリズムを出力するような使い方が出来る事。

(7) 中途半端な言語ではなく自然語に近いこと。また省略による標準型選択ができること。

(8) 会話はForeground Job で行うので、コンパイル、インタプリトの処理時間が、大体人間の思考速度とあつていること。あまり遅いと使いにくい。

(9) (1)および(6)の操作を可能にする為に、この会話用言語で、リアルタイムの情報検索が大きな待時間なしに行なえ、その後編集も自由に行なえること。検索範囲の拡大、縮少、変更等も容易に行なえること。

4・3・3 主記憶のスケジューリング

情報処理センター網に加入すると、自分のセンターの主記憶を、他の計算センターの端末使用者が使うこともある。（自センターのデータバンクの大容量ファイルを他センターの端末使用者が使う時に、大容量ファイルを全部伝送するより、プログラムを伝送し、結果をもらつた方が安上りになるとき使われる）

そこで次のようなスケジューリングが必要。

(1) Foreground Job の占有領域の動的再割当

上記のような場合、プログラムは、計算センター Foreground ⟶ 他情処センターの

Foregroundからそこのbackgroundと移つてゆく。ForeからBackgroundへ移る理由は、ForeよりもBackのコンパイラの方が能率が良く、Foreの衛生コンピュータより、Backの主コンピュータの方が速度が速く、また大容量の記憶とファイルを使うのは、Backgroundの方がやりやすいからである。このような操作をする為には、主記憶のForeground領域とBackground領域を動的再割当により交換できれば、転送する必要もなくうまくゆく。またこれは常識であるが、リモート局からの大容量記憶を必要とするプログラムが入つたり、大容量記憶を要するBackgroundのバッチ処理が入つてきたりすれば、ForeとBackgroundの境界は動き、動的再割当が行なわれ一時的に割当比が変る。

(2) Foreground Job用領域内で、各端末局の主記憶のとりあいによる動的再割当。

大容量記憶を要する端末からのプログラムが入ると、その優先度が高い時は、並行して実行している他のForeground Jobを補助メモリへ出して大領域を再割当して実行さす。

4.3.4 ファイル構成

システムとして持つべきデータファイルは、表Ⅳ-2、表Ⅳ-3に列挙した。

各使用者毎に、私有ファイルが必要、このうち非公開情報（データ、ライブラリ）には、パスワードなどの機密保持が必要。

自己拡張型のコマンド、会話用言語プロセッサの共通部と私有部の分割管理、実行時には両者併せて使えなければならない。

オンラインリアルタイム使用する為に、ファイルアクセスは速くできないとならない。特に検索など、大容量ファイルから必要データを迅速にとり出せなくてはならぬ。ファイル構成は、Ⅱ、Ⅳの2参照。

5. 文献情報検索システムへの応用例

5.1 文献情報検索システムの概要

ここでは、文献情報を対象とした検索システムの構成を行なう。このシステムで取り扱う情報

は、文献情報のうちでも学会誌等論文形式のものとする。質問言語は、自然語まで許す。このシステムでもつファイルには、次のようなものがある。

(1) 一次情報ファイル

オリジナル情報を蓄えたもの。磁気テープに収める。

(2) 二次情報ファイル

一次情報ファイルから、題名、著者名、引用文献名などの属性を抜き出してファイルしたもの。

(3) 索引ファイル

論文の題名、あらすじ、などから専門用語を抽出し、その出現頻度を数えたもの。

(4) 概念空間

索引ファイルから、論文間の類似度を求め、類似度解析法を用いて構成した空間。

質問分析は、質問文を概念空間内へ写像することにより行なわれるが、その過程は、人間と計算機とが会話を行ないながらなされる。この方法には、次のものが考えられる。

(1) 樹枝状解析法

利用者の質問概念を明確にするため、計算機のもつ概念のカテゴリーを利用者に示し利用者の持つ概念を一致するものを選ばせる。この操作を概念のカテゴリーを次第に細分しながら、recorsiveに行なうことにより写像を行なう方法である。つまりトリー構造のファイルのある頂点から始め、そのdecendantをPathを選択しながら求めていく方法である。利用者の持つ概念が、計算機の持つ概念と一致している場合は、問題ないが、その他の場合には、連想などの機能が必要となる。

(2) 位相型解析法

(i) 文章型シソーラスを用いる方法

質問文を単語レベルの意味空間から始めて次第にレベルの高い意味空間に写像して行き質問文析を行なう。この際、夫々のレベルでの意味空間における元と算法の指定を利用者と会話を行ないながら行なう。この場合、意味空間の内容を利用者が理解しておく必要がある。

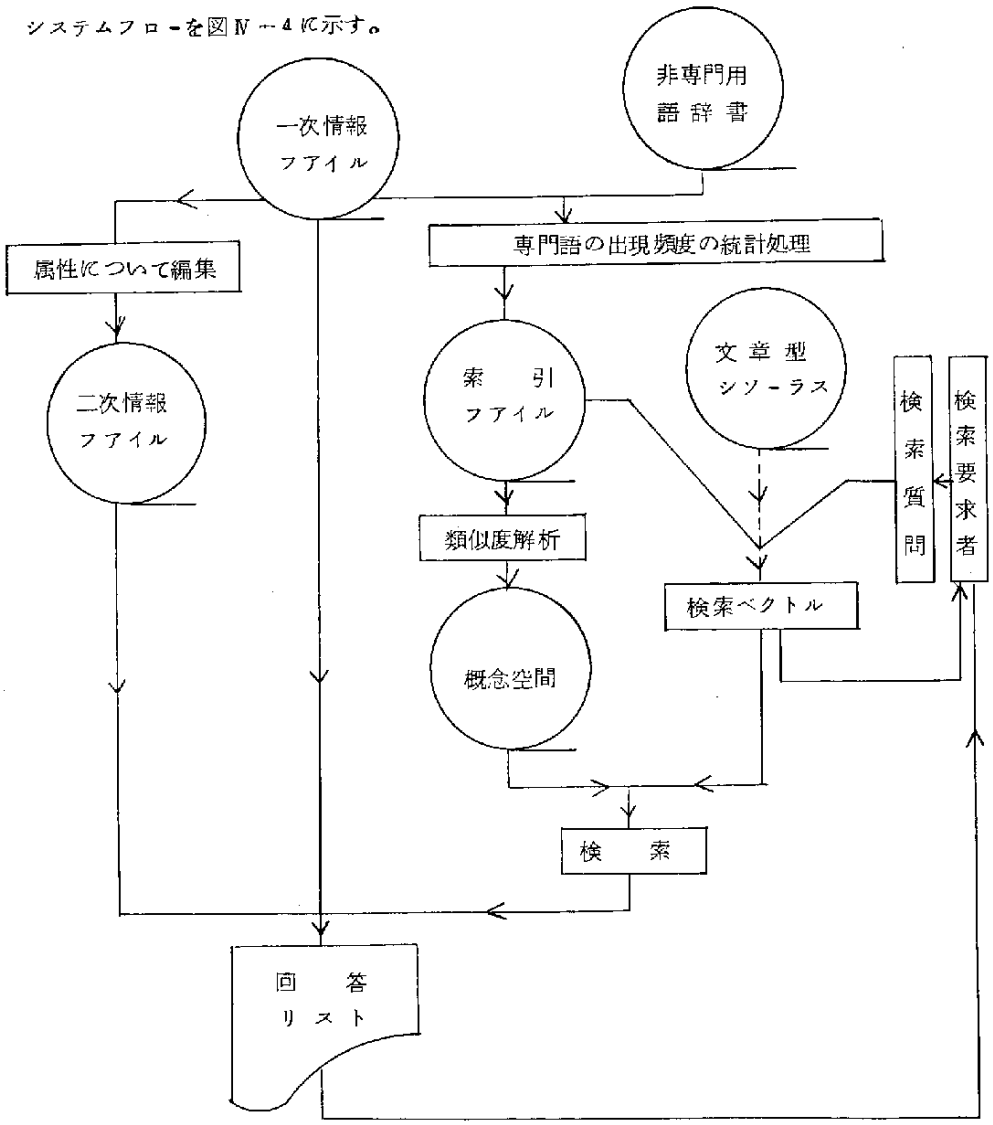
(ii) 標準点を用いる方法

質問文を属性空間に写像し、属性空間に、あらかじめ定めておいた標準点との類似度を求

める。この標準点のリストに、質問文との類似度を添えて利用者に示し、会話を進めながらこの類似度の値を質問概念に一致するように訂正していく。このようにして得られた類似度の集合を質問として概念空間内に写像する。この場合には、利用者が概念空間の内容を理解している必要はない。

探索は、概念空間内に写像された質問点との距離の近い論文を抽出することにより行なわれる。実際には、概念空間を質問ベクメルと垂直な超平面で分割することにより行なつている。

システムフローを図Ⅳ－4に示す。



図Ⅳ－4　検索システム・フロー

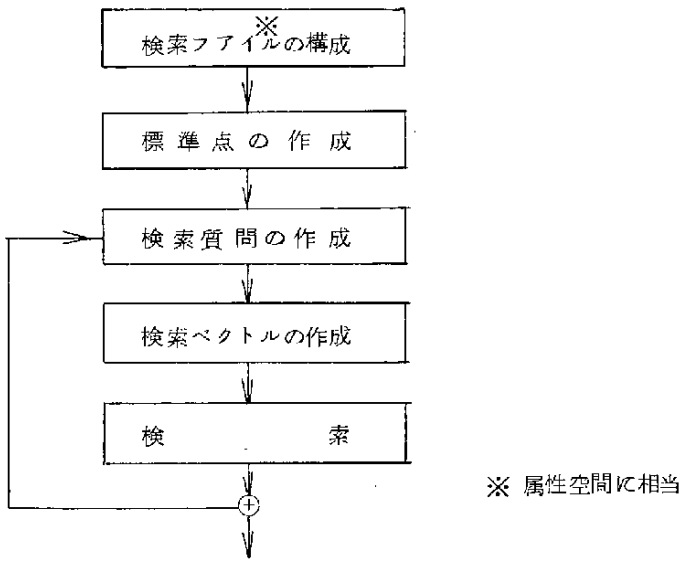
5・2 検索アルゴリズム

図 Ⅳ－5 に検索手順を示す。まず類似度解析法を用い検索ファイルの構成を行なう。つぎに、検索ファイルの代表的な情報を選び、それらを標準点とする。標準点との類似度から検索質問を作成し、それと標準点とから質問ベクトルを作成する。最後に情報空間を、質問ベクトルに垂直な超平面で分離し、超球面上の情報を抽出して検索を終わる。

標準点の作成方法としては次の3つが考えられる。

(1) 与えられた情報を人間が調べ代表元となりそうなものを選ぶ。

(2) 情報空間において、隣り合う情報の距離が等しくなるように分布している情報を選ぶ。すなわち、まず原点を中心とした半径1の超球面の表面を、等面積のいくつかの領域（標準点の個数に等しい数）に分ける。その領域の内に、丁度1つずつ点が入るように一様分布点を発生させる。この超球面を情報空間と重ね合わせ一様分布点とデータ点を定める。もしこの内積の



図Ⅳ－5 検索アルゴリズム

値が適当なデータ点がない場合には、もう一度一様分布点発生させ、同様の手順を行なう。もつと厳密に行なうには、先程の内積の値が1となるようなデータ点を探す。もしなければ空間を回転させ1となるようなデータ点を探す。次に、他の一様分布点に移り、先程の一様分布点とデータ点を一致させながら、空間を回転させ一致するデータ点を探す。この手順を繰り返せば、求めるデータ点が得られる。しかし、次元数が多くなれば、回転の軸が非常に多くなり、

事実上この方法は不可能である。前半に述べた方法がより実際的であるが、質のよい一様分布点を発生させる方法は、現在のところ、見当たらない。

(3) 類似度より求める。すなわち、情報の類似度の分布を調べる。そして、分散を求め分散の小さなものを標準点に選ぶ。

3つの方法のうち (1)の方法で標準点を選ぶのが最も理想的であるが、情報が多くなつた場合無理が生じるし、情報の範囲が広くなると、かなりの数のエキスパートを集めなければ不可能である。この実験では、計算時間などの条件から (3)の方法で標準点を選んでいる。標準点の選び方により、検索質問が、空間内に写像される位置が異なつてくるのでこの選び方は、検索能率などに大いに影響を与える。

検索質問は、自然語で与えられ、それを情報空間内へ写像し検索ベクトルを作成する。写像関数を、きれいな形で与えたかつたが、そこまで研究が進まなかつたので、検索質問は、次のように与えられるものとする。標準点と検索質問との類似度を調べて次のような検索質問ベクトル$\mathbf{R}$を作る。

$$\mathbf{R}=(\delta_1, \delta_2 \cdots, \delta i, \cdots \delta m) \quad (4.\ 2.\ 1)$$

ここに Siは検索質問と標準点 iの 類似度。

質問ベクトル$\mathbf{Q}$は、検索質問ベクトル$\mathbf{R}$に線形演算を施して、次のようにして作られる。標準点の座標マトリックス$\mathbf{S}$とすると、まず$\mathbf{R}\times\mathbf{S}$によつて$Q'$を求める。

$$Q' = (q'_1, q'_2, \cdots, q'_i, \cdots, q'_m)$$

$$= (\delta_1, \delta_2, \cdots, \delta i, \cdots, \delta m)$$

$$\times \begin{pmatrix} x_{11} & \cdots\cdots & x_1 i & \cdots\cdots & x_1 t \\ \vdots & & \vdots & & \vdots \\ x j_1 & \cdots\cdots & x j i & \cdots\cdots & x j t \\ \vdots & & \vdots & & \vdots \\ x m_1 & \cdots\cdots & x m i & \cdots\cdots & x m t \end{pmatrix}$$

$$= \mathbf{R}\times\mathbf{S} \qquad (4.\ 2.\ 2)$$

ここで、xjt は、標準点 jの i座標。

また、 $q'_i = \sum_{j=1}^{m} \delta i x j i$

さらにQ'を正規化してQを求める。

$$Q = (q_1, q_2, \cdots\cdots, qi, \cdots\cdots, qn)$$

$$= N(Q')$$

$$= N((q'_1, q'_2 \cdots\cdots, q^i, \cdots\cdots q'n))$$

ここで

Nは正規化関数、すなわち

$$qi = \frac{q'i}{\sqrt[r]{\sum_{1}^{t} q'^{r}_{i}}} \qquad (4.2.3)$$

最後に、検索ベクトルQと各情報との内積を求め、その値が閾値Pₒより大きいデータを抽出する。すなわち、情報空間マトリックスを、𝕀とすると，𝕀×Qから、n-tuple Pを求める。

$$P = (p_1, \cdots\cdots, pi, \cdots\cdots pn)$$

$$= \mathbb{I} \times Q^T$$

$$= \begin{pmatrix} x_{11}\, x_{12} \cdots\cdots x_{1T} \\ x_{21} \cdots\cdots\cdots\cdots \\ \vdots \quad \cdots\cdots\cdots\cdots \\ xn_1 \cdots\cdots\cdots xn^T \end{pmatrix} \begin{pmatrix} q_1 \\ \vdots \\ q_t \end{pmatrix} \qquad (4.2.4)$$

このPにおいて、その要素を大きいもの順に並べたP'を作り、その要素がpより大きいものの添字を順にとり出す。それが抽出すべきデータ番号である。

以上の手順によつて検索が終了する。

5．3 検索実験結果

前節で述べたアルゴリズムにしたがつて、プログラムを行ない 検索実験を行なつたので、その結果について考察する。

検索システムに要求される役割の一つに、情報と情報要求との照合を行なうことが挙げられる。しかるに、質問と合目的文献とを比較すると、不必要な文献が検索されていたり、必要な文献が検索されていなかつたりする誤差が生ずる。これらの関係を表わす尺度として、次の係数が一般に採用しているので、ここにも用いることにする。

$\alpha = \frac{R}{C}$　　再現率（Recall Factor）

$\beta = \frac{R}{L}$　　適合率（Relevance Factor）

ここでL：検索文献集合の文献数

C：関連文献集合の文献数

R：検索された関連文献数

データ数、標準点の数、歪などによる再現率、適合率の変化の様子を表Ⅳ－4に示す。ここでその値は2回の検索の平均値を与えている。

表Ⅳ－4　検索効率

データ数	次元数	標準点の数	歪	再現率 α	適合率 β
20	8	8	6.7%	100%	90.9%
20	6	8	19.1	0	0
20	6	15	19.1	100	80
30	18	20	2.5	100	100
50	35	15	3.0	78.9	93.7

再現率については、どのようなシステムにおいても、蓄積している情報を全部とり出すことにより、これを100%にすることが可能である。しかし、適合率は、システムにより、そのとり得る最大の値が定まつており、これを100%にすることは、困難である。しかし、このシステムにおいては、歪が2.5%以下の空間を用い、標準点を適当な数だけ選んでやれば、適合率を100%とすることは、不可能ではない。このことを表Ⅳ－4のデータから確かめてみよう。

このシステムの検索効率を左右する要因として、情報空間の歪、標準点の個数とその選び方を挙げることが出来る。　表Ⅳ－4によると、歪が3％程度の質のよい空間を用いても、標準点の数を空間の次元数より少なく選ぶと、適合率が94％、再現率が79％となり、検索効率は良くなく、逆に歪が20％の質の悪い空間を使つても、標準点の数を次元数よりずつと多く選んでやれば、適合率80％、再現率100％と、検索効率を改善することが可能である。しかし、この場合でも、標準点の数を次元数とほぼ同じくらいに選ぶと、適合率、再現率ともに0％となり、検索を行なうことが不可能となる。したがつて、歪の大きい空間を使うと、確かに検索能率は悪くなるが、それよりも標準点の選び方によつて、より大きく左右されることがわかる。すなわちこのシステムにおいては、検索質問をいかにして、うまく情報空間内に写像するかが最も重要なポイントである。

つぎに、データの付加、削除など、情報空間のメインテナンスの問題について考えてみよう。データの付加は、検索質問を情報空間内へ写像した時の操作と同じように、次式により与えられる。

$$X'_{iot} = N\left(\sum_{j=1}^{n} \delta_{ioj} \cdot x_{jt}\right)$$

ここで

X'_{iot} は　付加する情報 io の t 座標値

x_{jt} は　情報 j の t 座標値

δ_{ioj} は　情報 io と情報 j の関連度

N は　正規化関数

このようなデータの付加、あるいは削除により、空間の歪は当然変化するものと思われる。しかも、歪は、悪化する場合が多いだろう。よつて検索効率も悪くなるだろうが、標準点の選び方により、十分改善出来ると思う。したがつて、情報空間の再構成は、半年か1年に1度程度行なえば十分である。

情報質問は、（4．2．2）、（4．2．3）式により、情報空間内へ写像されるのであるが、もつといろいろな操作、例えば、回転とか座標変換とかを許した方が、能率のよい検索を行なう事が出来るかも知れない。しかし、この操作は、非常に複雑で、処理時間も多くかかり、不可能

である。そこで、このシステムでは、線形演算だけを選んだのであるが、この方法でも 表 Ⅳ－4 から分かるように、能率のよい検索を行なうことは可能である。

データ数が、２０、３０、５０個の場合について計算時間を 表 Ⅳ －５ に示す。

表Ⅳ-5 検索時間

データ数	２０	３０	５０
検索時間	1.2秒	4.6	8

この検索時間の中には、background job としてバッチ処理される操作の為の時間も含まれているので、realtimeの処理に要する時間は、上の値の５０～６０％になるものと思う。さらに、適当なスクリーニングを施しておけば、これを短縮することも可能であるが、このままでも、オンラインで、リアルタイムに検索結果を得るには、十分に短い時間である。

結局、表 Ⅳ －４ の実験結果から結論されることは、歪が７％程度の、出来るだけ次元数の低い空間を選び、標準点の数を次元数より十分多くとつてやれば、検索効率、検索時間、空間構成時間、あるいは、記憶容量などの点からいつて、有効な検索が行えるだろう、ということである。

6. 結　　論

Ⅲでのべた意味解釈機構をもとに、オンライン会話型システムの構成法について論じた。特に位相型データ構造をもつシステムについて詳しく論じた。このシステムは、今迄普通に行なわれている樹枝状データ構造のファイルをもつシステムに比べ、意味的な取扱いが豊富に出来る点で秀れている。

またそれを基に、位相型データ構造のファイルを持つ文献情報検索システムを実際に構成し、検索実験を行なったところ、良好な動作を示した。以下更に詳しく細部の結論をのべる。

２においては、現在動作しているオンラインＴＳＳシステムの現状を概観した。

また会話応答が自然語で出来ること、という要望に従い、そのようなシステムを構成する為に、今迄実験されたＱ－Ａシステムについてその動作特性を調査した。

これらのQ－Aシステムは殆どリスト処理言語で書かれており、Stringの取扱いは、やはり、リスト処理言語でやるべきことを示唆しているようである。

これらのQ－Aシステムにおける質問文の分析は、固定された質問リストのわく組があり、質問文の中の必要な単語を該当するわくの中に埋めていつて質問分析を行なうもの、依存関係による分析（Dependency 分析）を行なうものなどがある。しかしこれらのQ－Aシステムは、特殊な問題を解く為とか、特別な機能を例示する為に開発されたもので、計算機の知的な理解動作を与える為の直接的な接近法を構成しているものはない。

又、種々のモデルがあがつているが、意味論的な取扱いを行なっているものはない。

また遠隔制御会話応答システムの構成法を検討した。構成上問題となる点は、端末機器、スケジューリングと時間割当、ファイルの構成と維持、会話用言語などである。

3においては、樹枝状データ構造ファイルをもつ自然語応答システムと、位相型データ構造ファイルをもつ自然語応答システムの比較を行なった。

位相型データ構造をもつ自然語応答システムは、樹枝状データ構造をもつ自然語応答システムに比べ、連想などのように、非数値の情報の記述とか処理に適していることが示された。

樹枝状データ構造とは、体系的分類、階層的分類のように、ことばの表す概念（情報）の間の意味的な関係を樹枝状にとらえるものであり、シソーラスなども関係が有るか、無いかを二値でとらえ、関係のあるものを上位（BT）、下位（NT）、同等（USE、UF）、関連（RT）などで表現する為やはり樹枝状表現といえる。

これら樹枝状の分類を現在使われているものを相当広い範囲にわたつて調べてみたが、完全な一次元体系分類をなしているものは存在しないことがわかつた。そして一次元階層分類を行なっているものは、かなり沢山存在した。体系的分類は、どの要素もその樹枝の中のどこか一ケ所かつ唯一ケ所のみに分類されるもので、階層分類は樹枝状に分類されるが、同一概念が樹枝の違う所に二カ所以上表われてもよいものである。一次元分類では不十分なので、多次元分類、また分類しただけでは詳細を表わしきれないので、シソーラスと、ロール。リンクなどの関係子を添えて表現しているが十分ではない。

位相型の場合は、近傍空間、距離空間の上に概念（情報）を配置するが、このようにすると、連想など意味的な処理が非常にやりやすくなる。またその連想もその連想範囲を自由に変えられ

る。このような意味空間のファイル構成は、意味空間をファセット分類的な部分意味空間に分割し、その各部分空間に属する点は列挙する方法をとれば、連想のような意味的に近いもの、包含関係にあるものを求めるときなど、質問点の入る部分空間を求め、その部分空間内の点とで近いものを求め、もう少し離れたところのも求めたいときは、隣接する部分空間内の点を調べればよく、非常に有効に近い概念を求めることができる。

4のシステム構成においては、前節の意味取扱い法を参考に、オンラインＴＳＳで、会話型処理の出来るシステムの、将来あるべきシステム構成について検討した。

その詳細は次の通り。

今後のオンラインＱ－Ａ応答システムは、Ｑ—Ａを行なう為に大容量記憶を必要とするので、必然的に、情報処理センター網にリンクして使うことになるので、他センターのデータ・バンクをオンライン使用するという、複雑な事態が生じてくる。この場合もし使用するデータが大容量なら、データを自分の端局の属する計算センターにまで、データ伝送回線を通じて呼び寄せコピーして使うよりは、プログラムを、データのある処理センターへ伝送して、そちらで処理をした方が安上りになる。結果は伝送回線で自局の計算センターへ戻してもらい、処理を実行する。

このような処理方法を考えると、情報処理センター網を、時分割、空間分割で使用することになり、その場合のプログラムの実行場所の選択を自動的に行なわす方法の確立が望まれる。

またデータバンクに多種の情報を格納することにより、各種予測、計画などが行なえるようになる。

しかし最も重要なことは、計算機と対話しながら、アルゴリズム未知の問題を発見的（heuristic）に解くような機能を持たすことであり、また自己拡張の出来るシステム記述言語を開発することである。

5においては、位相型データ構造をもつ文献情報検索システムを設計し、米国計算機学会の学会誌の論文を対象に、意味空間を構成し、それをもとに情報検索実験を行なった。

この情報検索システムは良好に働き、その良さが確認された。

論文は多次元超球面上に意味の遠近関係を満足するように配置した。そして歪がある一定限度を越えない範囲で空間の次元を縮少していつた。空間は大体６～７割程度に縮少できる。（歪を2.5％におさえたとき）

検索に際しては、質問分析を行なわないとならないが、それは空間内に標準点を配置しておき、その標準点と質問点との関連度をもとに情報空間の中へ、質問点を写像した。

検索実験結果によると、歪が7％（これは空間を4割程度にまで圧縮した状態）程度の出来るだけ低い次元の空間を選び、標準点の数を次元数より十分大きくとつてやれば、検索効率、検索時間、空間構成時間および記憶容量の点からいつて、有効な検索が行なわれることがわかつた。

この方式で問題となるのは、論文間の関連をとるのに手数を要することと、情報空間構成に大容量の計算機を必要とすることである。

Ⅴ　（結　　論）

数値計算などバッチ処理技術は、ほぼ確立されたといつてよいが、ＭＡＣ使用による非数値の情報の処理（連想思考、情報検索、機械翻訳、発見的・創造的思考）部門についてはまだ不十分である。われわれは、このような処理の行える日本語による応答系システムを開発する為に、二つのデータ構造のとらえ方に基き、樹枝状データ構造自然語応答システムと位相型データ構造自然語応答システムを実験的に構成し、かなり好成果を得た。

以下に、遠隔制御処理システム構成上の問題点とその対策、二種のデータ構造自然語応答システムの構成法などに関する結論をのべ今後の研究の資料としたい。

1. システム構成

各レベルの情報処理センターを伝送回線で接続し、情報処理網を構成する。各レベルのセンターには、その地区で必要な情報を蓄積したデータバンクを備える。

オペレーティング・システムは、バッチ処理のみのＯＳに比べ、(1)情報ネットとのリンクによる、システム資材の有効な管理を行う為の入出力管理プログラム、どのデータ、プログラムをどこへ伝送し、どのセンターのＣＰＵを用いて処理するのが有効かを判断したり、ジョブ、タスクのスケジュール等を含む。(2)Machine Aided Cognition を可能にする為のコマンド、会話用言語、(3)共通公開情報、私有情報のファイリングと管理（私有情報の機密管理等も含む）、(4)大容量データのファイリングとアクセスなどの諸機能を持たせなければならない。

コマンドの機能としては次のものが必要である。(1)登録、経処理　(2)翻訳　(3)実行　(4)使用法指導　(5)情報網へのＩｎｔｅｒｒｕｐｔ命令　(6)ファイル、ライブラリの保存、引用、加工、命令へ

(1) 会話用言語

発見的、思考錯誤的な処理が行えることが必要である。その為にコンパイラはインタプリテイブな翻訳、任意の命令、部分プログラムの実行が出来、プログラムの修正が局所的修正で可能であり、それらの処理が人間の思考速度とあつていなくてはならず、オンラインリアルタイムのＩＲが出来ることが必要で、それらの条件を満す言語を構成しなければならない。

ファイル管理については、各データ構造により異るので、各システムでのべる。

現在動作しているオンラインＴＳＳシステム、また実験された各種Ｑ－Ａシステムの構成と動

作特性を概観すると、次の事項が結論される。

自然語を取り扱う為には、リスト処理の可能な言語でなくてはならない。質問分析は、固定質問枠組の中に該当するものを選び出して入れてゆくもの、依存関係による分析を行うものが殆どで、何れも特殊な問題を解く為とか、特別な機能を行なわすものであつて、計算機に知的な理解動作、意味的な取り扱いを行なわすように、直接的な接近をしているものはない。

遠隔制御会話応答システム構成上の問題点は、端末入出力機器（漢字、仮名、図形の入出力と、計算機および人間の処理速度に整合のとれた伝送回線）、タスクなどのスケジューリングと、時間割当、ファイルの構成と維持法、会話用言語などの構成法などがあげられる。

今後のオンラインＱ－Ａ応答システムは、実時間Ｑ－Ａを行う度に、即時アクセスの出来る大容量記憶を必要とする。その度に、情報処理センター網にリンクし、各センターの持つデータバンクを、またより高度のものでは、他センターのＣＰＵを使用しなければならない。どこのＣＰＵを使うかということは、伝達すべきデータ、プログラム量と伝達費用、各ＣＰＵの使用料により決定する。

この場合、情報処理センター網を、時分割、空間分割で使用することになり、その場合のシステムの制御を、独立式（各センターが独立して制御）、統一式（処注網の各センターの動作を、一つの専用システムで統一的に制御）に行う方法があるが、何れにしろ、プログラム（タスク）の実行場所の選択を自動的に行なわす方法を確立しなければならない。

また、中央のデータバンクに多種の公開情報を格納することにより、各種予測、計画などを、安上りに、短いターンアラウンド・タイム（ほぼリアルタイム）で行えるようにすべきである。その場合、公開情報はリードオンリメモリに格納し、使用者によりその内容を荒されないようにしなければならない。

しかし最も重要なことは、計算機と対話しながら、計算機の処理速度、人間の創造性とその長所を生かし、アルゴリズム未知の問題を発見的、試行錯誤的に解ける機能を持たすことであり、自己拡張の出来るシステム記述言語（会話用言語として）を開発しなくてはならない。

2. 位相型データ構造と、樹枝状データ構造

樹枝状データ構造とは、体系分類、階層分類のように、ことばの表す概念（情報）の間の意味的な関係を樹枝状にとらえるものであり、シソーラスなども、関係の有無の二値でとらえ、関係のあるものは、上位、下位、同等、関連などの関係子で表現するからやはり樹枝状表現といえる。

位相型データ構造とは、意味の遠近関係、包含関係など位相的性質を表現するのに、近傍空間、距離空間の上に、概念（情報）を配置するもので、位相的性質を満すように配置すると、連想などの意味的な処理が非常にやりやすくなり、その連想範囲も自由に変えられる。位相型の場合、その情報空間のファイル表現が問題となるが、情報空間をファセット分類的な部分空間に分割し、その各部分空間に属する点は列挙する方法をとれば、質問点に対応する情報空間中の点または領域を求め、それの入る部分空間を求め、意味の近いものを求め、もう少し離れたものを求めるなら、距離的に近い部分空間をたどつて求めればよく、最初に無駄な走査なしで、近い概念を求めることができる。連想の度合いも連続的に可能である。

樹枝状の場合、完全な体系分類は行えず、シソーラス、ロール、リンクなどの関係子を添えても、連想は二値でしか行えず、繰返し行つたとしても段階的で、きめ細かな調節は不可能である。

質問点（問題点）へのアクセス方法は、樹枝状では、体系分類または、シソーラスに表われる用語で行うが、必要なものすべてを列挙出来る保証はない。位相型では、言葉一空間領域の対照表と、言葉の組合せ方とそれに対応する意味空間中での意味合成の行い方の対照表が必要であるが、それがあると自動的な意味解釈が可能である。

表現レベルは、(1)自然語またはそれに近い形式言語を用いる表層レベル、(2)検索、処理可能ならしめる為の表現言語、例えば逆ポーランド記法とか、文生成過程迄表現したＰマーカなどで表現される深層レベル、(3)意味の位相性を表現する意味レベル、例えば近傍空間、距離空間とし各領域にことばの表わす概念を割り当てることにより情報空間として取り扱うの三レベルが考えられる。(1)(2)は統辞論で話ができるが、(3)は意味論を対象とする。

その両者はそれぞれ、表層言語代数系、深層言語代数系として数学的に記述できる。

3　樹枝状データ構造応答システム

データは体型分類または階層分類をとつているとし、そのような分類に従いファイルされる。これへのアクセスとしては、事項検索に応用するとして、自然語に近いContext Free の形式言語を考え、この形式言語で書かれたプログラムを用いて、データへアクセスしたり、加工、処理を行おうとするものである。樹枝状データ構造を最も簡単に利用する方法としては、各階層の分岐枝をすべて列挙して、質問者に必要な枝（分類項目）を選択させ、次々と下層へ下りてゆき、最後に求めるデータを得る方法がある。

もう一つの方法は、質問を形式言語で記述し、検索系の言語（逆ポーランド記法等）に翻訳し処理を行い、その質問とか結果を再び逆に変換し、自然語に近い形式言語になおし、質問者にフィードバックし、質問意図と合つているか、答が目的に叶つているかを調べてもらい、その結果を再び入力するという処理法で、次第に依頼質問そのものへ約束させ、答を得ようとするものである。

各階層で該当範疇を選択しながら情報ファイルを分割してゆくやり方は、情報ファイルそのものの知識さえあれば、形式言語により情報ファイルの切り出し方を記述する方法を知らなくてもすむ。しかしシステムの占有時間が指数関数的に増大し、また境界の明らかでないものは、分類、アクセスが不可能である。

これに対し、形式言語で情報ファイルの一点または部分集合を記述する方法は、その生成文法をよく知つていなくては、使いこなせず、素人むきとはいえない。

結局質問の表現法としては、次のようなものが望ましい。

(i) 形式言語の能力を極限に高めた形としての自然語の使用を許す。

(ii) 自然語に近い形式言語をとり、その言語で許される適当な文を選び、文中の空白の所に必要な語句を記入する、いわゆるL－言語の形式をとる。

(iii) 対話形式の積極的利用を行う。

以上のような観点からすると、樹枝状データ構造を用いて検索などの処理を行なう為には、情報ファイルの点、部分集合の正確な記述が最も重要なので、対話の一形式として、形式言語で書かれた質問文を、Spatax-Oriented Transducer で、検索系の論理表現に翻訳し、再びそれを自然語に翻訳しなおし、質問者に照会し、質問の修正、検索語の調整を行つた後に検

索に移るというオープン情報検索システムを考え、その翻訳プログラムを作成、検討した。

翻訳プログラムは、論理式を逆ポーランド表記に、また逆ポーランド表記された論理式を日本語に翻訳する手順について検討した。

日本語の語順は逆ポーランド記法とよく一致している。

ファイルの部分集合の合成指定法としては、否定、論理和、論理積、包含関係を用いる。

以上は、質問の分析法であつたが、実際の検索システムについて、情報の演繹を行える二方式についての構成例を示した。

一つは、W・S・Cooper の型の事項検索システムで、この方式は演繹の方向が定まつておらず、すべての組合わせをチエックする方式である。情報はアリストテレスの四つの定言的判断の形で表現され、形式論理処理を行う。

もう一つは、演繹の方向づけが与えられるもので、道路網等の状態をFuzzy集合の上での値で記述し、最適経過を求めるものである。

現在においては、事項検索システムは、対象を限定しないと、取扱えないと考えられる。

4. 位相型データ構造応答システム

意味の位相性を考慮し、ことばの位相的なデータ構造を表現出来る情報処理システムを考え、自然語に近い言葉による意味解釈を行うシステムの構成法を検討した。その構成法に基ずき、単語、意味空間、論文概念空間の構成実験を行つたが、良好な動作を示した。

またそれをもとに、自然語での会話応答システムの処理過程を論じた。

意味の位相性とは、「ことば」で表わされる概念の間の遠近関係、包含関係をいい、これらの間の関係が表現されたものを、位相型データ構造をもつデータという。

自然語応答システムを構成するには日本語の特性が明らかにされないとならないが、日本語は印欧語に比べ字数が多く、特に漢字が多く、それらを直接入力しようとすると入出力機器が複雑になるので、入力はカナで、出力は、デイスプレイ装置を用いて漢字で出すのが適している。印刷も出来れば漢字が取扱えれば良いが、それが出来るまでは、カナ綴りを使わざるを得ない。その場合のカナ綴りの標準文法（切れ目の入れ方）を早急に定めなくてはならない。

位相型データ構造をもつ意味空間は、樹枝状データ構造のシソーラスに対応する。シソーラスは線的なものであるのに対し、意味空間は面的あるいは、空間的なものである。

類似度解析法という計量心理学的な手法を「ことば」という新しい分野へ適用し、意味の遠近関係の順序関係を変えないで、許容歪範囲内（2.5％以下）で出来るだけ低い次元に圧縮された意味空間を構成した。データ数が50個程度の場合、その実現次元数は、データ数の4～6割程度で構成出来た。その際、距離のとり方としては、ミンコフスキー数4程度が良い（2点間の距離を差の4乗の和の4乗根で定める）ことが判明した。

この意味空間は、単語のレベル、論文のレベルなど数種のレベルで構成実験を行つた。

論文概念の抽出にあたつては、人がその関連度を判定する代りに、機械に行なわせる試みとして，論文のあらましに表われる語のうち、非専門用語だけ捨てて、専門用語の生起統計をとり、それから関連度を求めるやり方で行い空間を構成する実験を行つた。人が行つたものと、自動化したものとでは空間の歪はほぼ同等で、検索に使用した時も孫色なしに使えたので、OCRなどで読み込ますなら自動化も可能である事がわかつた。

論文数が増加すると、論文間の関係を調べる手数は2乗に比例して増加するから、部分空間の再帰的なつみあげで構成してゆかねばならない。

以上のような位相型データ構造をもつ文献情報検索システムを設計し、電子通信学会誌や米国計算機学会誌の論文を対象に、論文概念空間を構成し、それをもとに情報検索実験を行つた。この情報検索システムはパラメータにより呼出数が加減出来、良好な呼出率、適合率を示した。

検索に際しては、質問分析を行なわないとならないが、その方法には二通りあり、一つは入力文を構文分析し、意味空間の構造を参考に、文の該当領域を求めるもので、他方は空間内に標準点を配置しておき、その標準点と質問点との関連度をもとに情報空間の中へ質問点を写像するものである。実験は後者により行つた。

検索実験結果によると、歪が9％（圧縮率4～6割）程度の出来るだけ圧縮された低次元の情報空間を選び、標準点の数を次元数より十分大きくとれば、検索効率、検索時間、空間構成時間および記憶容量の点からいつて有効な検索が行なわれることがわかつた。

この方式で問題となるのは、論文間の関連をとるのに手数を要することと、情報空間構成に大容量の計算機を必要とすることである。

昭和45年3月　発行

発行所　財団法人 日本情報処理開発センター
東京都港区芝公園21号地1番5
機械振興会館内
TEL 03(434)8211(代表)

印刷所　株式会社 関西桜井広済堂
大阪府豊中市麻田88番地の2
TEL 068(53)2991(代表)