

機械翻訳*

和田 弘**

まえがき

外国語の読み書きに苦勞すると、翻訳が自動的にできないものかと思う。戦争で暗号を解説するために知識が築かれ、その後電子計算機が出現した。英人 A. D. Booth (1946) はこの機械を使えば翻訳ができそうだと示唆し、米人 W. Weaver が同調した。そして言葉の意義、外国文の分析、外国文と自国文との共通点などがまず紙上で研究され、小規模ながら計算機で単語の翻訳が実験され始めた。

1949年には Booth が Birbeck College でフランス語を英訳する実験に成功した。

アメリカでは 1952 年に MIT で機械翻訳の国際会議が開かれた。1954 年から雑誌 Mechanical Translation が刊行されている。1954 年には Georgetown 大学で、ソ連語 250 語を入れた単語辞典と、構文辞典とを使って、ソ連文が翻訳された。

ソ連は 1954 年に研究を始めた。1956 年の国際会議で発表した内容は、言語学的解析にまで及んでいたのに、西欧側は驚かされたようだ。

わが国では電気試験所が 1958 年の始めから研究にかり、専用機“やまと”を製作し、これを使って 1959 年 2 月に英文和訳の実験^{1,2)}に成功した。

1. 概 説

1.1 現 状

その後各国で研究が益々活潑に行なわれている。アメリカの National Science Foundation は定期的にこの状況を調べている³⁾が、それによると主なものは次のとおりである。

(a) アメリカ

大学では、さきに記した Georgetown のほか、Harvard, M.I.T., California, Washington, Wayne State などは、いずれもソ連語の英訳に従事している。Georgetown, Calif., Washington は支那語も対象としており、最近では、Harvard⁴⁾, Washington State

で日本語も対象にしている。

National Bureau of Standards でもソ連語について長期計画をもって進んでいて、その成果は高く評価されている。

民間機関では IBM を筆頭に、Ramo-Wooldridge, Rand. などが研究を行なっている。

いずれのところでも使っている機械は主に 704, 709, 7090 などである。各所はそれぞれ専門分野についての単語を豊富に揃えることで協力体系ができています。また計算機なしで研究をしているところも少なくない。

(b) ソ 連 圏

モスクーでは、言語学、計算機、数学の 3 研究所の他、Moscow Inst. でも行なわれている。Leningrad では、28 カ国語を対象として一旦共通な言葉に改めて、それをソ連、独、インドネシア語に翻訳する研究が行なわれている。研究者数では世界最大と信ぜられている。

この他、Georgia, Lithuania でソ連語を自国語に翻訳する試みのほか、Rumania では英語を翻訳することを企てている。

(c) 欧 州

英国では前記 Birbeck College のほか、Cambridge にある研究所が著名である。最近 National Physical Lab. を始め、二三の大学でも研究を始めたが、英語自身の研究が多い。

フランス、イタリーは比較的最近になって始め、ともに二三カ所で活動している。

その他ベルギーの Brussel, ドイツの Aachen でも研究されている。

(d) そ の 他

メキシコでは、英語、ソ連語をスペイン語に訳すことを、アメリカの大学と協同して行なっている。

中共では、北京にある言語研究所が研究に着手している。

1.2 問題の所在

翻訳したい論文は多数の文章 (sentence) からなり立っている。そこで各文章を独立した単位と見做してそれを翻訳した上で、次の文章に移るのが普通である。

* Machine translation, by Hiroshi Wada (Electro-technical Laboratory, Tokyo)

** 電気試験所

一つの文章は多くの単語からなり立っている。ことに欧米語はソ連語を含めて同じ系統から発しているので、文章の中で単語の相互の位置には大きな違いがない。たとえば

I love you. → Ich liebe dich.

そこで原語を逐次機械に入れて、その訳語を打ち出させれば、逐語訳ができるから、機械に便利な単語辞典を作ればよい。なお単語の意味は一つに限られないことが少なくない。“spring”を名詞に限っても、源泉、泉、バネ、弾性のほかに季節の春もある。それらの中から適当な訳語を選ぶことは多義語として大きな宿題だ。これを意味論 (semantics) という。訳語を羅列して糊塗している。

さて英語を日本語に訳そうとする場合は、

I give you some money.

ワタシアゲル フナタ イクラカノ カネ

としたのでは、訳せたとはいえない。彼に与えるのか、彼がくれるのか、大切な点が不明確である。テニヲハを入れて動詞を文章の末尾に廻すくらいことは望まれる。つまり構文の研究が必要になる。これを構文論 (syntactics) といっている。

この立場では単語はその訳語よりも構文上で果している機能の方が注目されなければならない。

機械翻訳においては両者は総合的に研究されねばなるまい。こう考えると研究は

1. 外国語を整理する。
2. 整理された内容を機械に教える。
3. 訳文の合成をする。

の3段階に分けられる。各国の研究状況を見ると、逐語訳から一歩進んで、構文の分析に関心が持たれる時代になっている。自国語を合成することにはまだあまり力が注がれていないようだ。

本文は読者が外国語(ことに英語)と日本語とは承知のものとして、英語の知識を整理して機械にかける要領を中心として述べてみる。

2. 言葉を計算機へ

計算機自体は情報を高速で処理する能力を持っただけであるから、これに翻訳させるには、語学についての知識を機械に適する形で予め教え込み、併せてその使い方を計算機に教えねばならない。後者をプログラムを作るという。

2.1 文字の表示法

計算の際は数字をコード(符号)化して使うが、翻

訳には Alphabet のほかに ? ; などの記号を始め、訳語にいる仮名までもコード化する必要がある。例えば a 010001, b 010010, c 010011 とする。

一旦コードを決めると、a 010001 は 2 進法で読めば 17 となる。同じように b は 18 になる。与えられた文字を調べるには、たとえば a と引き算をして 0 になれば a であることが判るし、引き算の結果の符号が正か負かを知れば、次には a b c 順でどちらの方向の文字と較べればよいかわかる。

2.2 単語辞典

文字をコード化すれば、単語の表示法は自から決まる。単語を辞典で探すことは、一方に与えられた単語と、他方に辞典の中から順次取り出した単語とが、全く一致(引き算をして、全部 0 になる)することで知れる。

計算機の記憶装置に単語の辞典を作る時、まず問題になるのは単語の数であろう。形容詞に —ly をつけて副詞、—ty をつけて名詞になるのは普通の辞典でも別語として取扱われるが、複数を示した —s、過去を示す —ed などは辞典にはない。

これらを全部別個の単語とすると、機械がいかに速くても探すのに時間がかかるし、経済的にも大変だから語数を減らすことが課題になる。単語をただ a b c 順に並べると、一語探すのに平均して全体の半分を調べねばならないし、また追加したい時にも困る。使用頻度の大きい単語から並べるのも一案であろう。

また辞典が用意されたとしても、その中から要領よく探すことは、計算機の普通の使用法と違った技術になるので、研究の対象になる。IBM の G.W. King はこの問題に対して意見を発表^{5),6)}している。

次に単語の語尾を処理したくもなる。こういう要求を記述したプログラムはしばしば使われるものなので、サブルーチン (subroutine) といわれる (2.4 参照)。

次に単語を見出した場合に、辞典の中から取り出される内容は普通の辞典と大差ない筈であるが、機械に便利な姿で記述する工夫がある。整然と並んだ様から、表 (table) とも云われる。

2.3 文型の決定法

文章にはいろいろあるけれども、それらは単文 (simple sentence) と、そうでないものとに分けられる。後者は単文が処理できるようになってからの研究であろう。

単文は主語、述語を含んでいて、その述語の性質で目的語あるいは補語を必要とする。このことは四則演

算が $a \times b$ のように必ず2個の数値を操作するのと違う点であろう。文章では述語の性質によって2語以上の単語の相互関係が示されている。その文法を数式的に表示すると、次のようになる。

- $f_1(s, v_1)$
- $f_2(s, v_2, c)$
- $f_3(s, v_3, o)$
- $f_4(s, v_4, o, c)$
- $f_5(s, v_5, o_1, o_2)$

ここに s は主語、 v は動詞であって、 v_1, v_2 は自動詞、 v_3, v_4, v_5 は他動詞である。自動詞でも補語 c を要する不完全自動詞のあることは、他動詞の場合にも同様で、 v_5 のように直接目的と間接目的語とを必要とするものもある。

機械にこれを確認させるには、この表示を機械的に行なったもの (Table といわれる) と与えられた文章との一致を求めさせればよい。

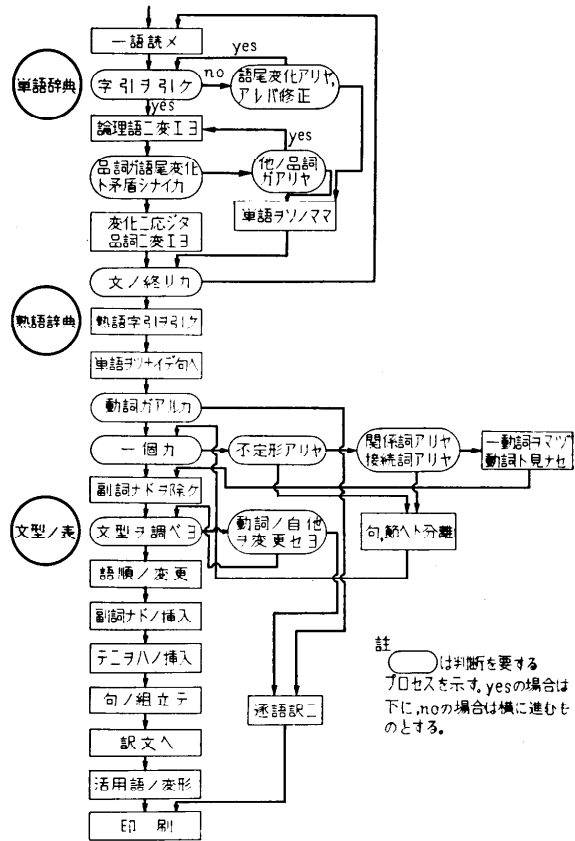
文章には普通は修飾のための語や句が加えられていて、いずれの文型に属するか見出すことは必ずしも容易でない。また単語には品詞 (part of speech) の唯一つでないものがあるから、文型の確認に苦しむこともある。しかし文型を決定することは翻訳にはもちろん、文章を正確に理解するための必要条件である。翻訳というとすぐに訳語が話題になるが、単語の品詞を決めることはそれ以前の作業であろう。

もし何らかの方法で単語の品詞を確定して、前記5文型のいずれかに一致することを確かめれば、それ以降に訳文を合成する方針を樹てるこ

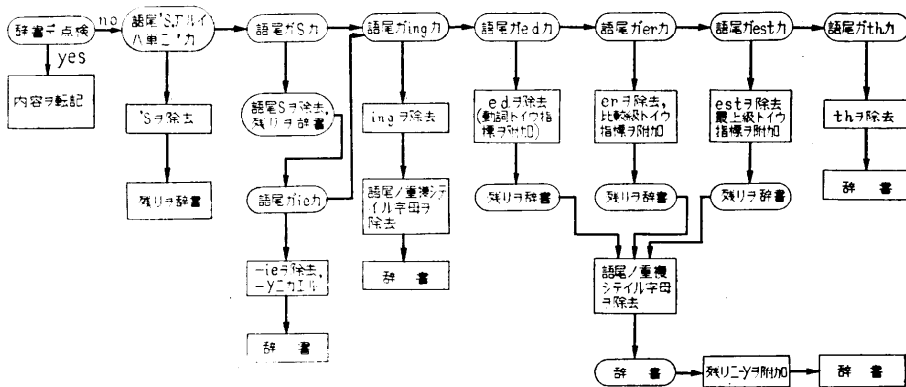
とはそれほど難かしくない。

なお文章には主語、述語、補語など相互の間に一貫した規則がある。たとえば、

This is a book.



第 1 図



第 2 図

では各語がいずれも単数であることを含んでいるし、

Is this a book ?

では、さらに疑問文であることが、Is this と？との2カ所に含まれている。このように一種の redundancy を持つ規則を複雑な文章の構造を決める手掛りとするのを shunting という。

2・4 プログラム

計算機に作業をさせるにはプログラムが必要である。これにはまず意図する内容を分析して、処理法を具体的に表示する必要がある。表示した図を流れ図 (flow chart) という。第1図は1958年に電気試験所での翻訳方針を示した流れ図⁹⁾である。

次に図中の各ブロックの内容を細かく示さねばならない。一例として第1図の右上にある語尾変化を操作するブロックの内容を示すと第2図⁹⁾のようになる。これも流れ図であり、subroutine でもある。要するに、語尾を調べて、該当するものがあつたら削って、字引で見つかれば、その旨を論理語 (3・2参照) に追記する、ということである。

最後に、このような詳細図を機械の持つ“命令”の系列を使って表示しなければならない。命令を機械語 (machine language) ともいう。

われわれが翻訳に当って使いたい命令形の単語は機械に備わっている命令には見当たらないし、これらでは表現できないように感じられる。これが機械翻訳の一つの隘路になっているようだ。しかし、これを機械でやらせようとするからには、命令で表わせるような工夫をしなければならない。

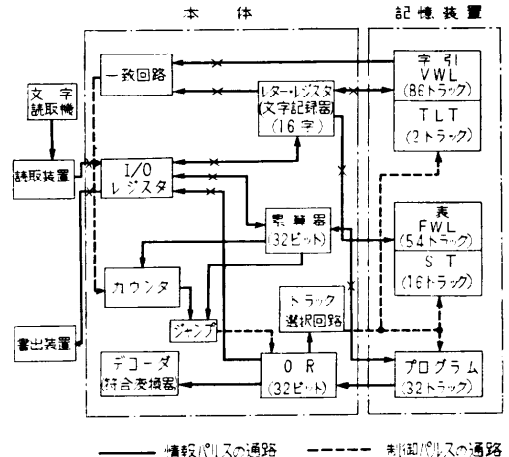
その要領を述べよう。“a×b”なる演算は

$$\begin{array}{r} 23 \\ \times 14 \\ \hline 92 \\ 23 \\ \hline 322 \end{array}$$

が示すとおり、九九、左への桁送り、加算、という操作を積み重ねた結果である。このように単純な操作の集合として記述することを算法 (algorithm) という。従って命令“乗算”がなくても、上記 algorithm ができれば、機械に乗算を行なわせられる。しかし乗算の都度このような記述 (プログラム) をすることは面倒なので、普通は“乗算”と書けば、操作する電気回路が機内に組込まれている。

このことを乗算が hardware でできているという。ない場合は software で行なわせるという。

例えば単語を“字引を引け”ということとは table look-up というので、解法の一例を示すと第3図のようになる。いま調べたい単語が1語入っている文字記録器の内容と、辞典の中の単語を順次とり出したものが一致するまでとり出した単語の見出し数をカウンタで数える。一致した時にカウンタが示す数値に相当するところから、その単語の“内容”を取り出す。最後まで探しても一致しない時は、判断回路が動作して、たとえば語尾変化の操作に移らせる。



第3図

このような操作はプログラム software でも実行できるが、“やまと”では hardware で組込んである。

翻訳でプログラムを作りたい時に、欲しい機械語がなくても前述の“字引を引け”のような擬似命令 (pseudo code) を提供することは、プログラミングにとって大切な課題である。

3. 現実の文章

平常接するところの論文は前述のような簡単な規則だけでは解決されない。構文の分析とか品詞を決めるのに苦しんでいる。若干の例をあげてみよう。

3・1 品詞の決定

一例として M. Corbe¹⁰⁾ の調べた英論文は、文章が48、単語1,014から成り、その中に341種の単語が含まれていた。これらを辞典で調べると、

12	(16)	辞典になし
179	(493)	品詞が1個
150	(505)	品詞が2以上のもの(コンマを含む)
341	(1,014)	

第1表

	形容詞	副詞	定冠詞	不定冠詞	接続詞	感嘆詞	前置詞	代名詞	名詞	動詞	計
a/ad	3	4									7
a/pron	5						16				21
a/n	20							5			25
a/v	12								5		17
ad/定冠			80								80
ad/conj		1									1
ad/prep		5					23				28
ad/n		1									1
不定冠/n				13					4		17
conj/prep							4				4
conj/pron					18						18
conj/n					1						1
pron/n								3			3
n/v									58	44	102
a/ad/conj					1						1
a/ad/prep							13				13
a/ad/pron	1							1			2
a/ad/n	2								1		3
a/ad/v		8									8
a/n/v	8								13	2	23
ad/conj/prep					1						1
a/ad/prep/n							5				5
a/ad/conj/pron					15			2			17
a/ad/int/n	1										1
a/ad/prep/v							14				14
a/ad/n/v	6									1	7
ad/conj/int/pron		3									3
ad/conj/prep/pron		9									9
ad/conj/int/n		3									3
a/ad/conj/n/v	1									1	2
a/ad/int/pron/n							2				2
a/ad/conj/prep/n		2			2						4
a/ad/int/prep/n/v		1			4						5
comma											57

505

であった。この品詞の定められない全体の約半数である150(505)語について、人間が適当な品詞を選定した結果を第1表に示す。

この中で comma の57, the の80を除くと、圧倒的に多いのは名詞か動詞か(n/v)の区別を含むもので、次いで形容詞か名詞か(a/pron., a/n)を区別するものであった。

とにかくこれらの品詞を決めるには、沢山の可能性がある場合を調べなければならない。

元来辞典による品詞わけは本質的なものではなく、用法によって決まる場合も少なくない。たとえば、lamp には名詞しかないが、

He lamps the room.

とすれば、その意味は通ずる。

This man likes ships.

には品詞の数が15あって、ことにn/vが3語も含まれている。

つまり、品詞を決めることは大変困難である。文章の分析が品詞を決めることと分離できない所以でもある。

3.2 単語辞典の見出し語

求める単語が辞典にないと困るから、どんな語でも求められる字引を用意することは大きな問題である。この課題に対しては二つの方向がある。Washington大学のReifer¹¹⁾は単語を網羅してtable look-upで処理させるのが論理的処理を必要としないでよいと主張し、170,000語の辞典を持っている。これに対して、California大学のLambは、単語を分解してlexeme

(語幹, 接頭辞と接尾辞, 語尾変化など)を用意し, これの組み合わせで求められる語を作る方が, 続々現われる新語にも応じられてよいと主唱している。

同大学の RUSDIC¹²⁾ は, 15,000 の lexemes を整備してある。普通単語はこれらの組合せで作られて約 1,141 万語までできる筈だし, 同大学が化学を分担している関係から, 化学用語は, 2,400 万語まで見出せる筈だという。

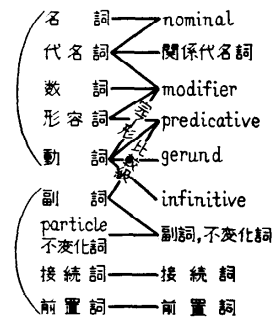
これらの激論にかかわらず, 多くの研究所は中庸を歩んでいる。Harvard 大学の自動辞典¹³⁾では, 辞典にない text 中の単語に人間が処理要領を与えて, 辞典を増強して来ている。現在では 10 万語に及ぶ text を読ませると, 5,000 語 (388 種) が二つ以上の見出しに refer されて (同形異義語 homograph という) 苦しむ他は, 僅かな誤りがある程度で, その信頼性を誇っている。

要するに, 単語 (space と space との間) には, 語尾変化 (inflection), 語幹 (stem) に接頭辞, 接尾辞 (affix), 派生語 (derivative), 複合詞 (compound) などが含まれるので, 経済や時間と関連して, 見出し語 (entry) がどうであったらよいかは, 暫く議論が続くだろう。このように, 単語だけでも多くの問題があるから, それらを括めて形態論 (morphology) という, 辞典の作り方とは別の立場がある。これを拡張して解釈して, 句を含ませる人もある。

字引で見つかった場合 text の単語に代って, そこに記入されている内容は以降機械の中で使われることになるが, 要すれば全部印字させることもあるので, その内容と形式とはかなりの差違がある。各所では, 必要と思う事項を記入した上, 便宜上一定の長さになっている。それぞれの呼称に従うと, Harvard word type では 9×12 字 (1 字は 6 ビット), R-W の computer word は 19×6 字, Wayne 大学の Coding of word は 3×(36 ビットの機械語), RAND Co. の Form は 50 字, 電気試験所の論理語¹⁴⁾は 7×8 ビットである。品詞, 訳語の他に, 形態の分析, 構文の分析で得られた情報を追記することは共通であるが, 原語を入れたり, 原文における位置を記入したり, あるいは修正を書込む余地を含めるものなどがある。これらなるべく短かい字数で表示する工夫がいる。

なお P.L. Garvin は従来の品詞の分類法は機械翻訳には必ずしも適切ではないとして, 第 2 表のような分類法¹⁴⁾を提案した。Part of speech と區別して word class と名付けているが, この説を採用して辞

第 2 表



典を作っている研究者は少なくない。

3・3 文章の分析

単文では文中に動詞が 1 個なのでまだ取り扱い易いが, 現実の文章では 2 個以上含んでいることが多い。こういう文章の持つ思想はいくつかの簡単な文章を組み合わせたものと見なすことができる。この内容を損ねることなく簡単な文章へと分解させる規則を創ることが文法の使命であろう。

普通は, 句 (phrase), 節 (clause), という概念で処理している。

句には, 形式上 on the desk のような動詞を含まないもののほかに, to go のような不定詞, playing ball のような動名詞, および現在分詞, 分詞構文などがある。それらは一面, 文中での働きに応じて, 名詞句, 形容詞句などと分類される。句は単語が連結させられたもので, このことは句が一つの結合語ともいえる因子 (string, package, fused unit などと呼ばれる) となって, 他の単語と同格な立場で単文を構成していると思わせる。

節も同じで, coordinate clause は接続詞でつながれているから, 重文と同じであり, 単文を f と表わせば, $f_1 + f_2$ の形になる。

これに対して subordinate clause は日本人を苦しめやすいが, その clause は形式的には 1) 単文の形を備えており, 2) 句読点, 接続詞, 関係詞がその始まる位置を示している。そこでこの節を見出せれば, 節にも働きによって名詞節, 形容詞節, 副詞節などになるから, この新因子は他の単語とともに, 全体としての単文の規則に従うことになる。

いずれにしても句, 節を見出すことは, 長い文章の中に, 適当な切れ目を入れて, group を作り, 全体として単文の一文型と見なせるようにすることである。

これが構文の分析である。

そこで、もし句節が求められれば、その中に含まれている品詞の曖昧な単語 (v/非 v, n/a など) も解決の手掛りが得られる筈だから、構文の分析と品詞の決定とは相互依存の関係にある。

次にいくらかの例を示そう。

(a) Georgetown 大学

アメリカで実験学派と目され、現に最も実用に近いと思われる Georgetown 大学^{15,32)}では、第4図に示すような方針に則っている。

図で、1, 2 については説明を要しまい。a 熟語を求めることは、1, 2 が一個の単語を調べるのと違うので

-
- ```

graph TD
 1[1. 字引を引く] --> 2[2. 形態の分析]
 2 --> a[a. 熟語の調査]
 2 --> b[b. Syntagmatic 分析]
 2 --> c[c. 挿入句の除外]
 2 --> d[d. 分離]
 a --> 3[3. 構文分析]
 b --> 3
 c --> 3
 d --> 3

```
- 第4図

別掲される。b も同様で、一単語が前後の単語と関連して確認する情報である。shunting として述べたように、主語たる名詞とその述語との間には、数、性などの一致 (agreement); 動詞の性質によって目的語、補語の存在 (government); 副詞が他の語と同格 (apposition) と見なせること; を調査する。c は句が前後と関係なく挿入されている場合には、そのつながりを邪魔するから、見つけて除外するためで、d は強い切れ目で文章を2個以上に分断させるのである。

3. このようにして、文中にある多数の単語を適当に結合して、それら因子の機能と境界とを求めて、総合して行き、最後にはいくつかの因子の中から、主語に相当する因子と述部に相当する因子とに集約する。

#### (b) ソ 連

I.K. Belskaya<sup>16)</sup>によると、ソ連では翻訳を単語の分析、構文の分析、訳文の合成の順に行ない、そのうち構文の分析は次のような順序で行なっている。

1. 動詞の分析
2. 句読点の分析
3. 構文の分析
4. 名詞の分析
5. 形容詞の分析
6. 語順の変更

つまり、まず動詞について、語尾変化とか前の語の品詞などを調べて、細かい分類をする。

構文の分析は3段階に別ける。

第1段では句などを見出すことを目的として、挿入的なもの、形容詞的なグループの前後に印を入れ、その印には等級がある。形容詞的なグループは普通その直前の語と直後の語を調べると、何らかの特長を見出すことができる。

一般に句は前端より後端を確める方が大変である。それでも順次後続する語を調べて行くと、主語である名詞、述語である動詞、等位でない接続詞などが出てくるのでわかる。

第2段では節を固めることを目的として、対等でない接続詞、接続的な名詞なり形容詞、最初の単語などを探す。予め見当のつけてある節の中に2個の述語があるとか、2個の名詞が続いて出る場合などを検査する。

第3段では、以上の調査で得た資料を基として節の前端と後端とを定める。主部と述部とが形容詞的なもの、従属節などで分離されているものを分析する。

以上の操作を全部決まるまで繰返した上、その中にある名詞や形容詞の分析に移る。

#### (c) Indiana 大学

Indiana 大学の F.W. Householder, Jr.<sup>17)</sup> は多品詞語の品詞を決定することに重点をおいている。すなわち、動詞に関係して品詞の確定できないものを次の6種に絞って、その subroutine を作る。

- |                 |                   |
|-----------------|-------------------|
| 1. n/v (現在)     | point, face など    |
| 2. a/v (現在)     | clean, complete   |
| 3. n/v (過去分詞)   | cut, thought      |
| 4. a/v (過去分詞)   | given, interested |
| 5. a/n/v (現在分詞) | meaning, running  |
| 6. 特長のあるもの      | mine, like, well  |

節の分析には、動詞が1個に限ることに注目するほか、文章を末端から逆方向に調べることを提案している。つまり、動詞より後方が複雑であることと、前述の節の開始位置は比較的容易に求まるためである。そして切れ目が起る条件と、すでに動詞が1個あった時に分離できる条件とを詳細に compile しようと試みている。

#### (d) Pennsylvania 大学

文章に含まれている無数の表現を整理して、それらの翻訳形式を用意すれば、辞典と同じようにそれを look up させることで翻訳の一解法になる。このよう

な計画はいくつかあるが、ペンシルバニア大学の Z.S. Harris<sup>18)</sup> の一派は、これに機能的な定義を与えて、英語を整理している。文法的な形式は違っても意味が同じな一対の型は、相互に変換 (transformation) ができるとする。それによると、

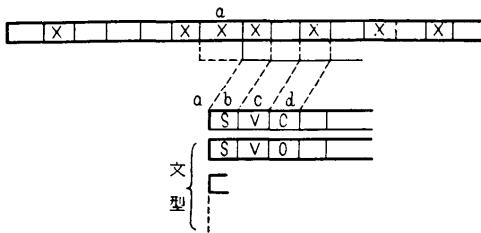
Binary, Unary, および Sentence+Predicate の 3 変換法がある。基本文型に相当するものを kernel と呼んでいるが、Binary は subordinator, coordinator などを含む文で 2 kernels を 1 sentence にする。

Unary は passive, middle などに着目して、1 kernel を 1 kernel+component の形に直す。

また、(1 kernel+component) の形の文を container verb などに着目して複文に直す。

(e) 和田の試案

“やまと”での従来の単文に対する方法<sup>(1) (1.2) (1.3)</sup>は持続する。現在までの要領でまず各単語の品詞を調べる。



第 5 図

Corbe の示すように約半数の語は、品詞が決定する筈である。第 5 図に品詞の決まった語を x で示す。

文章を構成する element は単文であると見做す。そしてこの単文の前後端を求めることに努める。このために連節語を求める。連節語とは、構文上、節と節とをつなぐ単語で、品詞のいかんを問わない。and は名詞と名詞をつなぐ場合は連節語でないが、節と節をつなぐ場合は連節語である。

関係詞, if, then, as, because……などがこれに含まれる。comma の如きもこれに含める。

また単語によっては、品詞が決まるのと同時に、それが主格になるか、所有格であるか決まるものがある。このような要素を、さきの図中に追加する。

次に修飾句などを棚上げすること、単文の場合と同じ。このようにして簡素化された文に対して、別に用意してある基本文形を、

1. 文中に連節語を見出して、これを文形の a とを合せるか。

2. 文中に主格を見出して、これを文形の b とを合せるか。

3. 文中に動詞を見出して、これと文形 c とを合せるか、などで照合させて見て、文形の残余の部分と文章のこれに対応するところが、文形の全領域で合致するかどうか調べる。この調査には品詞の決まっていない語が文形の要求する品詞を含むかどうかが含まれるであろう。

以上の操作を繰返すことによって、文章全体の構文を決定できれば、品詞も同時に決まることになる。

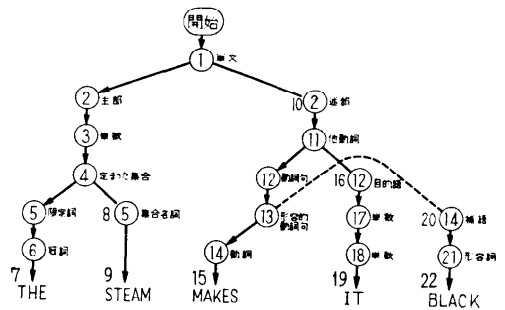
このためには文章が文型の表か、いずれかの address が slide できるように、変更すべきだろう。

なお句については、動詞を含む句の基本形を用意する必要がある。

(f) 文章の創作

M.I.T. の Yngve<sup>19)</sup> は文章構造を別の角度から眺めて、機械に文章を創作させている。

子供の本に十分な文法と単語とを与えておいて、それらを組み合わせて文章を打出させるのである。機械には選択の余地のあるプロセスとないプロセスとがある。



第 6 図

第 6 図で二股に岐れているプロセスは構文要素である。われわれが単語の一つ使う時、後方に何らかの制限が起る場合とそうでない場合とがあるが、前者ではそれを憶えておかなければならない。たとえば主語から始めれば、述部をも記述すべきことを憶えておかねばならない。clearly projected picture というような場合、clearly は名詞を前提としての語であるから、記憶の深さが 3 あると解釈する。

図で右下に走る線に対して左下に走る線の数が、こ

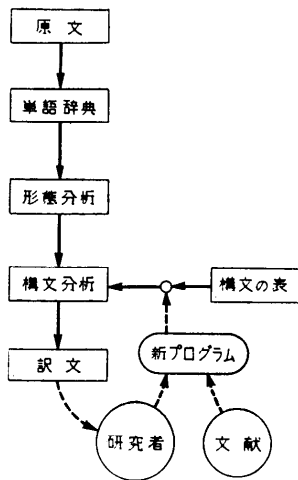


の記憶の深さに対応する。彼はこれを逆行的構造と名づけて、右下へ走る前進的構造と区別している。この考えで現実の文章の深さを調べると、統計的に7までだそうで、この制限に関連して品詞の価値が見直される。

### 3・4 前進させる方策

以上いくつかの実例を述べて来たが、もっと多くの方法が提案され、実験もされている。しかし文章は複雑だから、どの方法でも満足な翻訳は早急には望めそうもない。

ある方法が正しく、他の方法が誤っているという数学の解のようには行かない。紙上で決定版が出るとは思えない。実験をしながら漸進すべきだろう。それに



第7図

は第7図に示すように、訳文を見て難点があれば、プログラムを追加修正するところの循環式の方法が適当であろう。

### 3・5 日本文について

機械翻訳は“現実の外国語を現実の日本語”に直す作業を人間に代って機械にさせることを目標としている。原文に忠実であろうとすると、原文に含まれている単語を全部訳文に含ませ、構文についても忠実であるべきだろう。

定冠詞、不定冠詞、単数、複数の概念は和文英訳の場合に苦しみばよいが、関係詞のように英語に相当する日本語がないものは困る。

当所では単文を単位にして実際の文章に当らうとし

ているので、節は( )をつけて一単語のような表示法をさせている。たとえば

This is the book which is mine.

The computer does not forget whatever he learned.

は

コレガ ホン (ソレガ ワレノモノダ) ダ。

computer ガ (カレガ マナビ タモノハナンデモ) ヲワスレナイ。

と打ち出している。

過日、テレビ“私の秘密”で「私は敵艦見ゆとの警報を発した信濃丸の乗組員でした。」という問題があった。これでは警報を発した当人と誤解も生ずる。( )で信濃丸の説明が入ればよいのではなかろうか。英文を訳す際には特にこういう訳文が作られやすい。日本語に記号類を豊富に採用して、表現を明確にさせる努力は、適切と思っている。文章をなるべく数式的な表現に近づけたいとの理想にも通じているし、和文を英訳させる際にも必要となるのではあるまいか。

## 4. 関連ある研究

### 4・1 中間言語

機械翻訳は当初英語から日本語へというように、特定の国語の間で進められてきた。しかし、どんな国語からでも翻訳できるように拡張したくなる日はあろう。その場合、共通点は多いのだから、それらをまとめて記述できれば労が少なくすむ筈である。あるいは各国が自国語を分析した資料を作って相手国と交換すれば直ちにMTができることは望ましい。

Calif. 大学<sup>20)</sup>では露→英をやっているが、このことを念頭に入れた上でプログラムを作っている。翻訳に必要な言語学上の知識を具体的に記述したものと、それらを頻繁に参照するプログラムとに別ける。

このように分離した方式を採用すると、情報についての知識が増えた時に、プログラムを作り直さなくても追加できるし、またプログラム一つでどんな国語にも応待できる素地が生ずる。

MTはさきに記述したとおり、辞典の問題、形態の問題、および構文の解析によって基本文型(pivot languageと呼ぶ人もある)まですれば、あとは自国語の合成である。これらの問題をどんな形で記述するかに当って、参照すべき一つの基準があって、これがalgorithmに便利な姿であると便利である。また

研究機関ごとに関発したものを対比させようとする場合にも同じ要求が起る。

Esperanto があるように、このような基準となる言語を、中間言語<sup>21)</sup> intermediate language と呼ぶ。ソ連ではことに関心が払われている。

これをどんなものにすればよいかはまだ多くの研究に俟たねばなるまいが、次に述べる述語論理がその方向を示唆してくれれば有難い。

#### 4.2 述語論理

論理学が進んで形式(記号)論理になった。最近すぐれた解説書<sup>22)</sup>が出たから、詳細はそれらを参照していただきたい。

命題論理は、文と文との結合方式を明らかにしてくれたが、述語論理は一つの文の構造を研究して、量化した概念を取扱えるようになって来た。また文が主語と述語だけの他に、いくつかの目的語を含んだ単文についてそれらの言葉の間の関係を追求してくれそうな傾向にある。

まだ未完成ではあるが、文章が数式化できる暁にはその基準となる素質を備えていると信ぜられる。

#### 4.3 Basic English

I.A. Richards<sup>23)</sup> は英語の持つ欠点を改良して世界語としての資格を備えさせるべく、1943年に Basic English を提案した。

それによると、まず単語の数を制限して850におさえ、内750語が名詞と修飾語であって、その他の100語を operator とする。動詞は18語 (come, get, give, go, keep, let, make, put, seem と, take, be, do, have, および say, see, send; may, will) に限定し、その他の動詞は, enter → come in, abandon → give up, help → make help などとする。名詞か動詞かの判断に苦しむ必要がなくなるわけだ。

また about~with などの20語を directive とし、て位置と方向とを表わせることにする。

その他の as, for, all, and, if, so など62語になるべく簡潔な用法を決めてもらえれば好都合である。連節語を含めて。

この動きは言語学自身のことであり、同じような動きは ruly English としてアメリカにもある。

この18の述語ですませられれば、これらを記号で表わすと、述語論理の手法を使えば、文章は名詞、修飾語と記号とで表示することもできるのではあるまいか。

#### 4.4 自動プログラミング

電子計算機の科学用計算、事務計算に必要な program は、初期には machine language を使って刻銘に作られていたが、記号 program を経て、いまや Algol<sup>24)</sup>, Cobol<sup>25)</sup> などによる自動 programming の時代に入ろうとしている。各メーカはこのための compiler を作る競争をしているといってもよからう。

このことは、制限があるとはいえ、自然の言葉を用いてプログラムを書けば、compiler がその内容を直接機械語に翻訳するか、または一度 pseudo code に翻訳したものを interpreter がそれを machine language に直して、計算が遂行されるのである。したがって機械翻訳の研究と compiler の研究とは近い。

このようにプログラムを楽に書く思想が普及して来ると、機械翻訳の分野でもこれに似たものが開発されるようになることは当然であろう。RAND の MIMI C<sup>26)</sup>, Georgetown 大学の SLC<sup>27)</sup> (Simulated Linguistic Computer の略) は、いずれも自己の研究の便宜上開発されたものである。これに対して MIT の COMIT<sup>28)</sup> は単に翻訳だけでなく、数字でない文字をいろいろの目的で計算機で取扱う際に便利のように完成されたもので、要求があれば配布できるようになっている。

#### 4.5 その他

文書はますます氾濫する傾向をたどっているから、MT と同じように実用面から要求されている事項には、情報の保存法と取り出し法 (retrieval), 文書を分析して組織的に配列しておいて、必要の時に容易に探し出せる方法なり組織の研究がある。これに関連して文章の要約 abstract がある。また文書を直接機械に読ませるために文字読取装置の研究などがある。要すれば文献〔7〕を参照せられたい。

#### あとがき

わが国でも MT が話題になるようになって、早くも5年になる。この方面に関心を持つ人が次第に増えて来ていることは喜ばしい。これらの人々はいずれも MT に寄与する専門知識を持っている。言語学者、論理学者、技術者にしても。

しかし同じ分野の人々だけで固まったのではグループとはなっても MT を推進させる力にはなりにくい。

MT をなるべく速く実用に近づけようとするには、これらの専門家の間の研究が協力の方に向く必要がある。そして事実その意志もあるように見受けられて喜

ばしい。

そこで具体的な方策となるが、それには専門を異にする人々が自己の立場で話をするところから一步前進して、共通の場としてプログラムを中心とした話へと移ることがよいと思われる。もちろん言語学、機械、理論などにいく分偏していることは差支えないが、

### 参 考 文 献

- 1) 蓼沼良一: 計算機による英文和訳 I. 電試研報, 624号, 1961/12.
- 1・2) 蓼沼. 五十嵐: 計算機による英文和訳 II. 近刊,
- 1・3) 五十嵐実子: " III. 近刊.
- 2) 電試ニュース: 110, 1959/3.
- 3) N.S.F.: Current R & D in Scientific Documentation # 9. 1961/11
- 4) S.Kuno: Approach to Japanese-English translation, Report NSF-6, Computation Lab., Harvard 1961/6
- 5) G.W.King: Table look-up procedure. IBM J. 5 2, p. 86 1941/4
- 6) J.L. Craft: A Table look-up machine. IBM J. 5 3, p. 192 1961/7
- 7) 和田弘: 計算をしない計算機, 情報処理, 1, 1, p. 19, 1960/7
- 8) S. Takahashi et al.: English-Japanese MT. [29] p. 194
- 9) P.Y. Panov: 自動翻訳 (みすず書房) 1956.
- 10) M. Corbe: Introduction a une syntaxe automatique de l'Anglais. [31]
- 11) E. Reifler: Solution of MT linguistic problem through lexicography. [30] p. 312
- 12) Univ. of Calif: vocabulary coverage of RUS-DIC. 1962/2
- 13) V.E. Giuliano: Research on MT at Harvard [29] p. 163  
M.E. Sherry: Automatic affix interpretation and reliability of the Harvard dictionary, [30] p. 317
- 14) P.L. Garvin: Syntactic Retrieval [30] p. 286
- 15) M. Zarechnak: Current research at Georgetown Univ. [30] p. 63
- 16) I.K. Belskaya: MT method of an Anglo-Russia scheme [29] p. 199
- 17) F.W. Householder Jr.: 7th report on automatic language analysis. Indiana U. 1962/3
- 18) Z.S. Harris: Transformations and Discourse analysis project. No. 1~42 U. of Pa. 1962
- 19) V.H. Yngve: Computer programs for translation. Sci. Amer. p, 68. 1962/6
- 20) S.M. Lamb: Advance in MT. New Scientist 9 p. 260, 1960.
- 21) I. Sakai: Syntax in universal translation. [31] 1962
- 22) 沢田允茂: 現代論理学入門 (岩波新書) 1962.
- 23) I.A. Richards: Basic English. Kegan Paul, Trench, Trubner & Co. 1943.
- 24) Algol-60 Comm. ACM 3 p. 299, 1960.
- 25) Cobol-61, Dept. of Defence, U.S. 1961
- 26) H. Kelly: MIMIC: A translator for English coding. [30] p. 451
- 27) A.F.R. Brown: Flexibility versus Speed. [30] p. 444
- 28) MIT: An introduction to COMIT programming 1961/11; COMIT programmers manual 1961/11; V.H. Yngve: Comm. ACM. 5 1 p. 19, 1962/1  
次の3冊は MT に関連した会議の議事録 (単行本) である。  
[29] Information Processing. UNESCO. 1959/6 Paris での会議  
[30] H.P. Edmundson: Proc. of Natl. Symposium on MT. Prentice-Hall. 1960/2 Los Angeles での国内会議。  
[31] Proc. 1st International Conf. on MT. HMSO 1961/9 英国 Teddington, N.P.L での会議。
- 32) Research on MT. Report of U.S. House of Representatives. 85 th Congress, 2 d session. # 2021. 1960/6

(昭和37年7月4日受付)