自然言語処理 71－4
（1989. 3. 18）

日本語語彙の代数構造と機械翻訳への応用

北村 博
日本ＩＢＭ 東京基礎研究所

日本語の語彙を作用する働き(operator)と、作用を受ける働き(operand)の２つの性質を同時に併せ持つ(operandor)ととらえると、語の意味演算の代数的な特徴が明確になる。まず結合律を満たさない。ここから数学の主要な結果は全部無力になる。語の作用の仕方をＬＦＧ，ＧＰＳＧのような形式で書く必然性がわかる。operandorと考えると構文木のambiguityは当然のことで、同時に存在して意味を作ることがわかる。日英機械翻訳システムＪＥＴＳにこの語の作用素モデルが取り入れられている。その方法も発表する。

Algebraic Structure of Japanese Words and its Application to Machine Translation

Hiroshi Kitamura
Tokyo Research Laboratory, IBM Research
IBM Japan, Ltd. No. 36 Kowa building
5-19, Sanban-cho, Chiyoda-ku
Tokyo 102, JAPAN

Cayley introduced the concept of operandor, which works as operator and at the same time as operand. A Japanese word is taken to be an operandor on meaning operation. Underthis view, an algebraic structure of the Japanese word can be defined. Non-associativeness is the most important characteristic. This means that many of the main theorems of mathematics are inapplicable to language theory. Applying operandor concept, ammbiguity of structure can be accepted, because continuous application of operandor have various structures. This idea is combined with Relational Grammar in the JETS transfer conponent.

はじめ

自然言語文が意味のコミュニケイション手段として同一言語圏内の人間間の非常に優れた方法であり常用されていることは言うまでもない。コンピュータで自然言語文を扱おうとする時、とりわけ機械翻訳のように語の意味、文の意味を扱うことが避けられない時、機械処理のプログラムで扱う"意味"とは何か、その性質は何かを明確化する必要に迫られる。コンピュータで処理できるとは、プログラムで処理ロジックが記述可能であり、それに用いるデータが作成可能である、ということと同値である。処理ロジックがプログラム言語で記述可能であるとは、そのロジックを数学的な記号体系で抽象的に書けるということであり、そうやって整理されたものに対し何らかの数学分野があてはまるとするなら、きっとそれは、微積分とか位相数学ではなく、代数であろうと思われる。逆に自然言語文の意味処理に関し（数学的に）明確に記述された範囲のものはコンピュータで実現可能である。

トリー構造と非結合性

文が言語の基本的な単位である語から出来ており、語にはカテゴリーとしての品詞があり、品詞間には規則としての文法があることは広く承認されている。また文の意味関係を表す構造としてトリーが一般的に用いられている。

他方数学でtreeに関する論文をみると良く引用されているのが、最初の頃1857年に有名な大科学者Cayleyが書いた論文がある。("On the theory of the Analitical Forms called Trees" Phil. Mag. 8 1857 pp 172-176)

これは自然言語の意味モデルに対し示唆に富んでいると思われる。少し紹介すると、Cayleyはoperator(作用素)とoperand(被作用素)の両方の性質を同時に持つ"operandor"を考えた。簡単化した例で書けば、x, y 2変数の多項式から多項式への変換を考える。

A，B，C，D，E，Fをそれぞれ2変数の多項式とする。

$$P=A\frac{\partial}{\partial x}+B\frac{\partial}{\partial y} \qquad Q=C\frac{\partial}{\partial x}+D\frac{\partial}{\partial y} \qquad R=E\frac{\partial}{\partial x}+F\frac{\partial}{\partial y}$$

x,yの多項式uに対し

$$P*u=A\frac{\partial u}{\partial x}+B\frac{\partial u}{\partial y}$$

$$Q*(P*u)=C\frac{\partial}{\partial x}\left(A\frac{\partial u}{\partial x}+B\frac{\partial u}{\partial y}\right)+D\frac{\partial}{\partial y}\left(A\frac{\partial u}{\partial x}+B\frac{\partial u}{\partial y}\right)=$$
$$C\frac{\partial A}{\partial x}\frac{\partial u}{\partial x}+CA\frac{\partial^2 u}{\partial x^2}+C\frac{\partial B}{\partial x}\frac{\partial u}{\partial y}+CB\frac{\partial^2 u}{\partial x\partial y}+$$
$$D\frac{\partial A}{\partial y}\frac{\partial u}{\partial x}+DA\frac{\partial^2 u}{\partial x\partial y}+D\frac{\partial B}{\partial x}\frac{\partial u}{\partial y}+DB\frac{\partial^2 u}{\partial y^2}$$

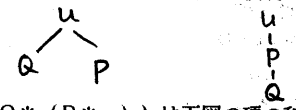
ここでQxP＝ $\left(C\frac{\partial}{\partial x}+D\frac{\partial}{\partial y}\right)\times\left(A\frac{\partial}{\partial x}+B\frac{\partial}{\partial y}\right)$

$$=CA\frac{\partial^2}{\partial x^2}+(CB+DA)\frac{\partial^2}{\partial x\partial y}+DB\frac{\partial^2}{\partial y^2}$$

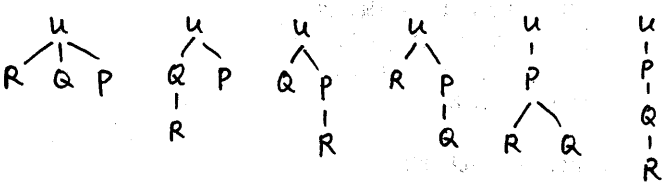
と定義し

$$Q*P=\left(C\frac{\partial}{\partial x}+D\frac{\partial}{\partial y}\right)*\left(A\frac{\partial}{\partial x}+B\frac{\partial}{\partial y}\right)=C\frac{\partial A}{\partial x}\frac{\partial}{\partial x}+D\frac{\partial A}{\partial y}\frac{\partial}{\partial x}+$$
$$D\frac{\partial A}{\partial y}\frac{\partial}{\partial x}+D\frac{\partial B}{\partial y}\frac{\partial}{\partial y}$$

と定義すると、ＱｘＰ，Ｑ＊ＰはoperandorであってQ＊（Ｐ＊ｕ）＝（ＱｘＰ）＊ｕ＋（Ｑ＊Ｐ）＊ｕ
この各項を下のように図で書くと便利なことがある。



Ｒ＊（Ｑ＊（Ｐ＊ｕ））は下図の項の和になる。



ここでこの演算は結合律を満たさない。つまり
Ｐ＊（Ｑ＊Ｒ）≠（Ｐ＊Ｑ）＊Ｒ
これは非常にまずい性質であり、このため数学の主な結果はほとんど使えないことになる。(半群にすらならない、群、環、体にもちろんならないため。)言語も非結合であり例えば
（ＡとＢ）かＣ　（Ａ and Ｂ）or Ｃ
Ａと（ＢかＣ）　Ａ and（Ｂ or Ｃ）
この上下の意味はそれぞれ異なる。つまり言語は非結合性を持ち、ここから言語理論に直接、数学の主要結果が応用できるとは期待できないことがわかる。これが『変圧器の設計』[1]と異なる点だ。非結合数に関しては若干の結果があり、それとＣＦＧの関連については［2］を参照。

人間が文を読むとき、あるいは耳で聞くとき、人間の言語処理系は先頭から順々に語の連鎖を入力として受けていることになる。『雨と風が強まった』は
start－雨－と－風－が－強ま－っ－た－end
人間の脳神経系が『雨』を処理して、その状態に対し『と』が作用する(operator)。もう『雨とが』という文型は現れない。つまり『と』は明らかに状態を変更している。この状態に対し、『風』が作用する(operator)。ここまででも、雨、風の意味内容は組合わさって残っている。つまりこれらは、operandでもあると考えられる。すべての語は基本的にoperandorであろう。そこでCayleyの表現が参考になる。しかし語に対して、もっと具体的な対象空間、作用域、作用の仕方を規定しないかぎり、単にoperandorというだけでは、何もでてこないのは明らかである。そこで日本語の文法をとりいれたモデルを考える。
例えば『子供がパンを食べる』は、operandとしての語(品詞、意味属性)、operatorとしての語(文法状態の変換、構造の変換、意味操作)を意識して書けば、次のようになろう。
『食べる』＊『を』＊『パン』＊『が』＊『子供』＊initial-state

語　属性(品詞 意味)　作用
子供　名詞 人 動物 生まれた者　nil　-----> 名詞句(+意味属性)
小さい 学習する,,,　の＊名詞句A--> 名詞句(+意味属性 +親=A)
A(+子=topic-of-文)
用言句　---->名詞句(+意味属性 用言の空格があれば、或ﾙｰﾙで埋める/なければ『について』関係)
名詞句 -----> 名詞句(+意味属性 複合名詞)

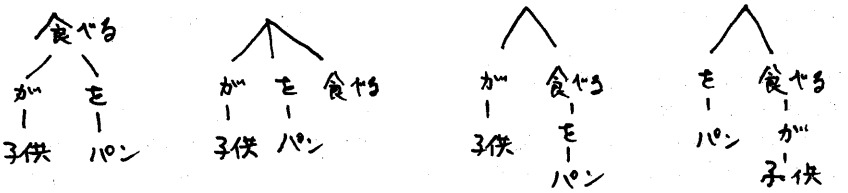
が　助詞 格助詞 接続助詞　nil　------>nil＊文開始(+逆接)
強調 動作主 逆接　名詞句 ----->ga格要素
用言句連体形->nil＊複合文(+逆接)

パン 名詞 物 生産物 食品　　nil　-----> 名詞句(+意味属性)
　　小麦粉をｲｰｽﾄ菌で　　格要素　----> nil
　　発酵させて作る、、　　用言句　---->名詞句(+意味属性
　　用言の空格があれば、或ﾙｰﾙで埋める/なければ『について』関係)
　　名詞句 -----> 名詞句(+意味属性 複合名詞)
　　但し以下の物はパンの細分類
　　黒パン　白パン　ブドウパン
　　フランスパン　ヤマザキパン、、
　　フライパンは別

を　格助詞　目的格　　名詞句 ----->wo格要素
　　その他 ----->nil

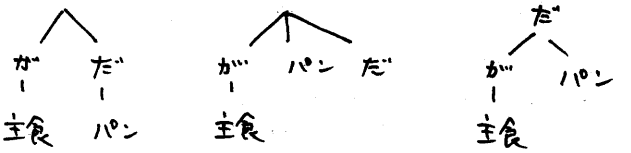
食べる　動詞　動作性　他動詞　　すべて　-----> 用言句(+意味)
　　ga,wo格をとる　　以下の変換も行う
　　人が食品を食べる　　ga格要素---->動作主of食べる
　　生物がXを食べる　　wo格要素---->対象of食べる
　　Xは死ぬ／なくなる　その他 ----->修飾成分of食べる

ここでCayleyの図を上下逆様にして作用されるものを下に書くと、この文の構造は次のようなものが、考えられよう。operandorという見方に立つと、これらの構造は、互いに排他的でなく、この文に対し、同時並列的に存在して良いことになる。

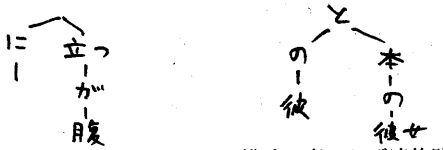


『食べる』自身が意味変化する　　組み合わさって意味ができる　　『パンを食べる』が熟語化した　　『子供が食べる』が熟語化した

パン：格要素 ---> nil がきいて、構造が大幅に減っているが、これにより『彼がパン』(彼のパン)が合文として受入れる余地を捨てている。『主食がパンだ』は

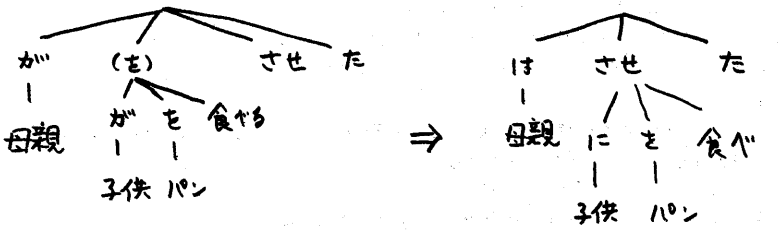


になろう。『腹が立つ』のような熟語は下の成分が強まっていると考えられよう。



同様に『彼のと彼女の本』は下のような構造を考えれば連体助詞『の』が『本』と同格になり形式名詞化するのが良く分る。

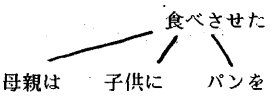
『させる』(使役)のような語の機能を書くには、表現と事象という観点を持ち込む必要がある。例えば
『母親は子供にパンを食べさせた』という文を考えてみる。書き手＝話し手はこの文の作成に当り、
『母親が　『子供がパンを食べる』ことを　させた』を無意識のうちに思い浮かべ、次いで例文を作ると仮定する(いずれにしても検証は不可能)。話し手の中での変化を推定すると



ここから『させる』は属性として、補助動詞 使役 、その作用は次のように書ける。
『させる』の対象の用言---------->用言連用形
『させる』の対象の用言の動作主------->ni格要素
(但し、『させる』は仮定の意識内構造に対し働く)
この逆作用素をinv(させる)と書くことにする。我々の日英機械翻訳システムでは、拡張付属語にたいし、その逆作用素を定義、定式化することでtransfer処理を簡略化している。それに入る前に、語の働きが代数的に明瞭に書けた例を紹介する。

線形代数を用いた文節内分ち書きの文法モデル

表面的な文字の意義通りでのつながりと構造を考えると、発声時のまとまりからみて、『母親は子供にパンを食べさせた』に対し次の構造が妥当であろう。



各文節は　母親／は　子供／に　パン／を　食べ／させ／た　のような小構造を持っている。これを取り出すのに、良く知られているように、形態素解析の文法（分ち書きの文法）が使われる。分ち書きでは、A／（B／C）と（A／B）／Cは同値である。つまり語を作用素とみると、このモデルでは結合則が成立する。事実、ここでは語は、接続文法状態を示す空間（ベクトル空間）上の線形作用素＝行列で表現できる。逆に分ち書きの文法では、語の本質的な性質であるoperandorは取り入れる、あるいは表現することはできないため、意味処理には実質無力であることがわかる。
例を『仕事していたのだ』にとって、語がいかにして行列と対応ずけられるか、またそのことが、形態素解析の手順といかに関係するかを考える。
接続文法状態を最初に定義する必要がある。文節先頭、文節終端候補、カ行5段動詞語幹直後、、、、サ行5段動詞連用形直後、、、連用形イ音便形直後、連用形促音便形直後、、、助動詞『た』終止連体形直後、、、、、等である。語に対し、どういう接続文法状態にたいしてこの語が働きうるか、を定義できる。またその結果接続文法状態がどう変化するかも規定できる。
例えば『た』にたいし
た-a　行った　　助動詞『た』終止連体形
た-b　行きたい　補助形容詞『たい』語幹
た-c　立たない　タ行5段動詞活用語尾
が考えられる。た-a はサ行5段動詞連用形直後が1、または、連用形イ音便形直後が1、または、、連用形促音便形直後が1、である接続文法状態に対し有効で、結果は、文節終端候補、助動詞『た』終止連体形直後のみ1、他は0のベクトルになる。それ以外のベクトルに対しては、0 ベクトルに写像する。この事情を次のようにまとめられる。
(検証は線形代数の知識があれば容易、ここでは省略)

前接続条件ベクトル＝ｂ＝(サ行5段動詞連用形直後、連用形イ音便形直後、連用形促音便形直後のみ1、他は0)
後接続条件ベクトル＝ａ＝(文節終端候補、助動詞『た』終止連体形直後のみ1)
ｔをtransposeとして 行列 Ｍ(た-a)＝$a^{t}b$
た-a ：接続文法状態ベクトル ｖ ---->Ｍ(た-a)ｖに写像する
但し行列演算で＋，ｘ は論理演算 or andを示すものとする。
同様にして、Ｍ(た-b)、Ｍ(た-c)を定義できる。表面的な語『た』に対しては
接続文法状態ベクトル ｖ ---->(Ｍ(た-a)＋Ｍ(た-b)＋Ｍ(た-c))ｖ
でその作用が記述できる。
このようにして、語を接続文法状態ベクトル空間上での線形作用素＝行列に対応ずけることができる。ここで形態素解析は行列演算として書けるようになる。
『仕事／し／て／い／た／の／だ』は次式と同値
Ｍ(だ)Ｍ(の)Ｍ(た)Ｍ(い)Ｍ(て)Ｍ(し)Ｍ(仕事)(文節先頭のみ1、他は0)
＝ (文節終端候補が１、他は自由) .....(1式)

仕事には、名詞、サ変動詞語幹、の２種がある。
ａ11=(名詞直後のみ1、他は0)
ａ12=(サ変動詞語幹直後のみ1、他は0)
ｂ0=ｂ11=ｂ12=(文節先頭のみ1、他は0)
Ｍ(仕事)=ａ11^{t}ｂ11+ａ12^{t}ｂ12

『し』には3種、『て』には4種、『い』には9種、『た』には3種、『の』には3種、『だ』には3種、ある。(我々の基礎になっている[3]において)
Ｍ(し)=ａ21^{t}ｂ21+ａ22^{t}ｂ22+ａ23^{t}ｂ23
Ｍ(て)=ａ31^{t}ｂ31+ａ32^{t}ｂ32+ａ33^{t}ｂ33+ａ34^{t}ｂ34
以下省略。
(1式)に代入して

$$\sum_{\sigma_7\sigma_6\cdots\sigma_1}(a_{7\sigma_7}{}^{t}b_{7\sigma_7})(a_{6\sigma_6}{}^{t}b_{6\sigma_6})\cdots(a_{1\sigma_1}{}^{t}b_{1\sigma_1})\,b_0=e$$

行列演算は結合則を満たすから () を付け替えて

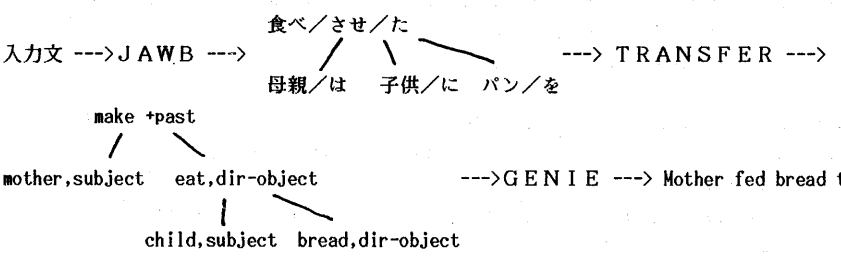
$$\sum_{\sigma_7\cdots\sigma_1} a_{7\sigma_7}\cdot({}^{t}b_{7\sigma_7}a_{6\sigma_6})\cdots({}^{t}b_{2\sigma_2}a_{1\sigma_1})({}^{t}b_{1\sigma_1}b_0)=e$$

ここで () 内は１ｘｍ，ｍｘ１行列の積より結果はスカラー、この場合０か１である。つまり接続可否の論理値になっている。
ｂ0=ｂ11=ｂ12 ｂ0ｂ0=1よりσ_1=(1 2)に対し
^{t}ｂ2σ_2ａ1σ_1= 1なるσ_2のsetが求まる。このσ_2のsetに対し
^{t}ｂ3σ_3ａ2σ_2= 1なるσ_3のsetが求まる。このσ_3のsetに対し
.....
^{t}ｂ7σ_7ａ6σ_6= 1なるσ_7のsetが求まる。
以上を区切りphaseと呼ぶことができる。語選択のcontrolを行えば、形態素解析の語の区切りに対応するからである。
次に(1式)の右辺に注目して
ａ7σ_7は文節終端候補が1でなければならない。これで上で求めたσ_7のsetが縮小できる。このσ_7のsetに対し^{t}ｂ7σ_7ａ6σ_6= 1を満たす条件で上で求めたσ_6のsetが縮小できる。
.....
このσ_2のsetに対し^{t}ｂ2σ_2ａ1σ_1= 1を満たす条件でσ_1を決定できる。
以上を決定phaseと呼ぶことができる。(注)このＯ(Ｎ)の解析のideaは大河内 正明氏による。
以上のprocessで
仕事 .... サ変動詞語幹　　し .... サ変動詞連用形語尾　て .... 助動詞『た』連用形
い .... 『居る』由来の補助動詞　た .... 助動詞『た』終止連体形　の .... 形式名詞
だ .... 助動詞『だ』終止連体形
が得られる。

以上のように分ち書きの文法は完全に線形代数で記述できる。これは語を結合則を満たす作用素として近似したため可能となり、同時に限界をも示している。私見では、日本語ワープロの成功は日本語形態素解析の成功に負っており、その成功は、語の接続を品詞間の接続ととらえたこと、それが広大な言語現象に比較すれば、少量の規則で有効に働いたこと、がある。その原因は、結果的にその体系が結合則を満たしたため、規則が個別的というより一般的に働きえたためだろう。数学は汎用的な論理を対象としているが、その基礎に演算の結合性を前提に組立てている。きっと無関係ではないだろう。
しかし自然言語処理が一歩、意味処理に踏み込めば事情は一変する。そこでの主役は個別的な語の意味とその働き(作用)であり、operandorの世界である。

翻訳処理での応用

我々の翻訳システム[4]は日本語形態素解析、構文解析をする『ＪＡＷＢ』[5]、Transferをする部分、英文生成をする『ＧＥＮＩＥ』[6]、各部分の辞書を管理する部分[7]、lisp上の木構造操作utility『Tree-Tranceducer』[7]等から構成されている。ここでは『母親は子供にパンを食べさせた』でみてみると、



ここでは語の作用モデルをtransferで取り入れた部分につき述べる。語、組み合わされた部分的な表現、を事象に関するものと、話し手(書き手)の評価、態度の表明をしているものとに分ける。『食べ』は『ＡがＢを食べる』という事象の表現であり、基本的に辞書情報が訳語、構造変形を指示すべきであると考えられる。『母親』、『子供』、『パン』もそうである。『に』、『を』も事象間の関係(relation)を示す語である。『は』『させ』『た』は評価を含む語と考えられる。modal,voice,aspect,topicに関する語、表現はこれ等の代表的なものである。『食事／すら』の『すら』のような語、表現も評価を示す。『そして』『しかし』『ながら』等もそうである。

これら評価を示す語、表現は日英両言語間で表層を見ている限り、機械的な対応に困難を感じるものばかりである。

一方Relational Grammar理論([6]のreference参照)では、日英両言語を含む多言語間でcanonical structureは同じであるとしている。そこでこのcanonical structureを前出の"話し手が、文生成の前に思い浮かべる脳内の構造"と考え、日本語解析結果から、日本語のcanonical structreへの変換を行ない、そして日英辞書による事象に関する語、表現の英語への変換をするというアイデアが生まれる。これで英語canonical structureが得られたので、同じくRelational Grammarに基づいて作られている英文生成が処理することができる。しかし、英語側でのRelational Grammarの研究は充実しているが、不明なる筆者の知る限りでは日本語側での研究がない。そこで我々としては日本語canonical structureを英語に合わせて取ることにし、そのように開発している。

評価に関する語、表現につき文生成時の作用を考え、その逆作用を記述したtable,programを作る。現在約500の語、表現につき作った。又、その作用の仕方を分類して12種類にまとめ、処理プログラムを簡単化させた。今の例では

(『た』※『させ』※『食べ』) ※ ((『は』※『母親』) (『に』※『子供』) (『を』※『パン』))

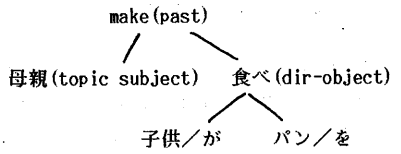
に対し、inv(た) inv(させ) inv(は)を順々にその出現位置で適用する。

inv(た) --->+past

inv(させ)-->new-node("make", vt),用言句(dir-object),ga要素(subject-of-new-node)
ni要素[意味区分:人 組織](ga要素)

inv(は) --->+topic

結果は



参考までに他の例も上げると

inv([動詞連用形]落とす)-->new-node("fail",cvform=inf),動詞句(subject-of-new-node)

fail to (A say B)...>A fail to say B

inv([動詞た形]後で)--->+phrase-relation="after",vform=prog

after going to B

inv([用言た形]ばかりに)--->+phrase-relation="just because"

inv([用言終止連体形]とすると)--->+phrase-relation="if"

inv([用言終止連体形]途端に)--->+phrase-relation="no sooner",親(+phrase-relation="than")

no sooner does he go to bed than he falls asleep

inv([名詞]ごとに)-->new-child-node("every",adj)

every Sunday

inv([名詞[意味区分:数],数詞]までに)--->prep="by"

inv([名詞]に違いない)--->new-node("no doubt",np,there-ins),名詞句(prep="of")

there is no doubt of XXX

現在この枠組で新聞社説に〝挑戦〟しているところであり、まだこの枠組についての結論を出す時点でないことを付言したい。

まとめ

意味処理について、何らかの画期的な理論ができれば、問題が一度に解決するというようなことは、有りえない。単純化された微分方程式がある種の自然現象をうまく近似する、というのと同種のことは絶対に起こらない。自然言語処理は、非常に常識的なことだが、辞書に知識を蓄積し、個々の語の使われ方と、それが翻訳されたときの訳詞分けのｷｰに何がなったかの分析、そして基礎に健全な言語理論、各種softwareの整備、これらによってだけ進歩することができる。現在の機械翻訳について、色々の批判、愚痴、等があるが、我々はもっと胸をはれると思う。こういう世界なのだから、良くやっていると。

参考文献

［1］天野 真家，〝構文解析における種々の問題〟1989-1-13 文法的知識と意味的知識の蓄積管理ｼﾝﾎﾟｼﾞｳﾑ論文集,電子情報通信学会

［2］北村 博,〝CFGの数論への変換〟1985 信学技報vol85No22,〝CFGの生成力評価〟1986 情報処理32全国大会4c-3

［3］大河内 正明,〝仮名漢字変換のための形態素接続規則〟1981 日本IBM TSCﾚﾎﾟｰﾄ

［4］丸山 宏,〝日英機械翻訳ｼｽﾃﾑJETSにおける知識の蓄積管理〟1989-1-13 [1]ｼﾝﾎﾟｼﾞｳﾑ論文集

Maruyama,N. et. al.,"A Japanese Sentence Analyzer",1988 IBM Journal of Research and Development, 32-2

［5］渡辺 日出男、丸山 宏,〝制約依存文法に基づいた対話的な日本語解析支援ｼｽﾃﾑ〟1988情報処理学会自然言語処理研究会69-6

［6］Johonson,D.E.,"The Design of Post-Analysis in the JETS Japanese/English Machine Translation Systems" 1988 Proceedings of the International Conference on Fifth Generation Computer Systems. "The Role of Relational Grammar in the JETS System" 1988情報処理学会自然言語処理研究会68-1 "GENIE:A Transportable English Generator for Machine Translation" 1988 IBM TRL Research Report TR87-1023

［7］野美山 浩 〝自然言語処理のための機械辞書作成支援環境〟1988 情報処理36全国大会3u-6

〝機械翻訳のための木構造変換言語〟1987 情報処理34全国大会7w-1