

自然言語処理と知識表現

止井 潤一
(京大・工学部)

§1. はじめに

自然言語の解析は、与えられた単語列としての文を受け取り、その文の持つ「構造」を明示的に表現する何らかの表現を作り出す過程である。人工知能における自然言語理解の研究においては、この自然言語の解析において、現実世界に関して人間の持っている様々な知識が関与していることを明らかにしてきた。

しかしながら、自然言語表現そのものについての知識、すなわち、文法的な知識も、当然のことながら、自然言語理解には大きな役割を果たしている。ここでは、文法的知識とは、必ずしも、名詞・動詞・名詞句といった品詞情報にもとづく統語的な知識だけでなく、ある言語表現がどのようなメッセージを聞き手に伝達しようとしているかに関する総ての知識を指すことにする。事実として同じ内容を表現する文であっても、文の焦点・話題等が異なる場合がある。これまでの人工知能からの自然言語理解の研究は、あまりにも「外界に関する知識」に焦点をあて過ぎていたために作成されたシステムは非常に *domain specific* なものになり過ぎていた。

本報告では、「外界に関する知識」は直接対象とはせず、一般の言語現象に即した知識をどのように表現するか、また、言語表現と外界知識がどのように関連を持っているかについて整理し、これを計算機処理するためにはどのような機能が必要かについて考えることにする⁽¹⁾。

§2. 言語現象に関する知識

我々が「文法」という言葉から想起するのは、「名詞」・「動詞」・「名詞句」といった「品詞」レベルでの規則である。ここでは、まず、議論に入る前に、自然言語に関する知識として、このような品詞レベルでの規則性以外にどのようなものがあるかをいくつか考えてみよう。

(a) 形態素的規則性：単語の語形変化は、単語内での現象であり、例えば、言語学における文法理論では、品詞レベルでの規則性を抱える「句構造規則部門」・「変形規則部門」とは別に取扱われる。しかしながら、文の解析し、次元の単語列としての文から、その構造を抽出する解析の過程においては、この形態素的規則性も品詞レベルの規則性と同時に取扱われなければならない。例えば、

(1) the gentlemen in the room $\begin{cases} (1-1) & \text{which} \\ (1-2) & \text{who} \\ (1-3) & \text{that were} \end{cases}$

では、(1-3)の関係代名詞節の先行詞が room でなく the gentlemen であることは、この形態素的規則性から決定できる。

(b) 簡単な意味標識：(1)の例文で、gentlemen が単数の gentleman であっても、(1-1)の関係節の先行詞は、the room、(1-2)の関係節の先行詞は the gentleman と決定することができる。(a)と(b)の規則を同時にとらえようとすると、すく

なくとも、名詞句 *the gentleman* や *the room* には、「数」の情報と ±HUMAN の標識が必要となる。

(c) 統語構造から意味構造への写像：能動と受動、使役等の構文構造からわかるように、表層上の言語表現は、さまざまな変形をうけることになる。言語学では変形部門を設けて、一般的な規則で、この表層表現と深層上の意味を関連づけていた。しかしながら、最近の傾向として、変形規則の取扱う範囲をできるだけ少なくし、意味解釈規則でこれを行なうようになっていく。意味解釈規則は、個々の単語個有の規則であるので、辞書中に、多くの規則性が書かれることになる。このことは、今まで、比較的少数の規則によって一般化できると考えられていたものが、各単語個別の規則として定式化されることとなり、Frame等のA.I.研究の動向と一致している。

'Finally, I assume that it is easier for us to look something up than it is to compute it. It does in fact appear that our lexical capacity -- the long-term capability to remember lexical information -- is very large.'⁽²⁾

日本語についても、Case Linking規則⁽²⁾と呼ばれる単語個有の規則で、これを行なおうとする試みがみられる。

計算機処理の観点からすると、言語処理のための規則の多くの部分をこれまでデータと考えられてきた辞書の記述中に埋め込むことになり、ここでも、人工知能の分野で開発されてきた手法が有効になる。例えば、

(2) 太郎は、花子を美しいと思う。

という文では、「思う」は、「NP が S と 思う」という格パターン とるが、「花子」のように本来は、S 中で格助詞「が」でマークされる要素が、格助詞「を」を伴って、S から外へ出されることがある。格助詞「を」のこのような用法は、一連の動詞に個有のものであり、こういった動詞の辞書中に、[NP が NP を S と 思う] という格パターンとともに、「NP を」が S の「が」格に入るという写像関係を記述しておく必要がある。しかも、この現象は、S 中の述部のアスペクト (d) の項参照) が「状態的」な場合に限られており、このような制御条件も「思う」の辞書記述中に書いておく必要がある。

(d) アスペクト：日本語の述部には、「～ている」「～てある」「～はじめる」「～つつある」といったアスペクト形式素がつく。これらのアスペクト形式素に関してすくなくとも、次のような規則性が記述される必要がある。

① 動詞の格パターンの選択

(3) 利用者がパラメータを指定する

(4) 利用者によってパラメータが指定してある

②主述部のアスペクト素性(瞬間, 継続 等)とアスペクト形式素によって決まる述部アスペクト

(5)彼は死んでいる。(結果状態)

(6)彼は、本を読んでいる。(進行, 経験, 習慣, 結果状態)

①を取扱うためには、アスペクト形式素の辞書記述によって、動詞の辞書記述(これ自身がまた処理手順を記述する規則になっている)の一部を修正したり、とり出したりしなければならない。

②の規則化は、主として、主述部のアスペクト素性と形式素間の規則性として捉えられるが、実際には、(6)の例文のように一意に決定できない場合も多い。しかしながら、

(7)彼は、今本を読んでいる。(進行)

(8)彼は、よく本を読んでいる。(習慣)

(9)彼は、いつも本を読んでいる。(習慣)

といった副詞との共起関係によって、その可能性が減少すること、また、

(10)彼は、いろいろな本を読んでいる。(進行)

(11)彼は、朝日新聞を読んでいる。(習慣)

(12)彼は、ジャン・クリストフを読んでいる。(習慣)

といった名詞の属性によっても、その主な読みが決まる。(10)・(11)・(12)のような名詞句と述部アスペクトの関係は、現実世界の知識との接点として重要であろう。しかしながら、副詞の共起と述部のアスペクトの決定は、一般的な言語知識として処理できる必要がある。

述部のアスペクトは、文の間の関係を決定する際にも参照される。

(13)プログラムを実行しているため、…… (原因)

(14)プログラムを実行するため、…… (目的, 原因)

これらアスペクトに関する現象は、日本語において、アスペクトを決定する場合には、単に局所的な単語のまとまりだけでなく、かなり離れた構成要素間の相互関係をうまく捉える必要があること、また、現実世界の知識をどのように言語処理に反映すべきかの示唆を与えてくれる。

(e)文脈処理と構文：日本語における照応現象については、これまでに言語学の分野で比較的良好に研究されている。一般に、自然言語理解の研究においては、文脈処理は、文の統括的・意味的な処理が完了した後に、文の意味内容を中心に、現実世界の知識を参考にして行なわれることが多かった。多くの文脈処理が、現実世界の知識を必要とすることは確かであるが、一方で、文中での語順や統語構

造も重要な役割を果たしている。例えば、言語学の側から提案されている *proceed* と *command* の原則を考えてみよう⁽⁴⁾。この原則は、代名詞は、「その先行詞よりも先行し、同時に、先行詞を含む構成素を統御する」ことはない、という原則である。

(15) 花子は、彼から太郎がかって住んでいた家を譲り受けた。

(16) 花子は、太郎がかって住んでいた家を彼から譲り受けた。

(17) 花子は、彼がかって住んでいた家を太郎から譲り受けた。

(15)は、図1に示すように、代名詞「彼」が、「太郎」よりも先行し、かつ、これを統御する位置にあるために、「彼」と「太郎」とは同一指示とはなり得ない。これに対して、(16)・(17)は、このいずれかの条件が破られているために、同一指示であり得る。

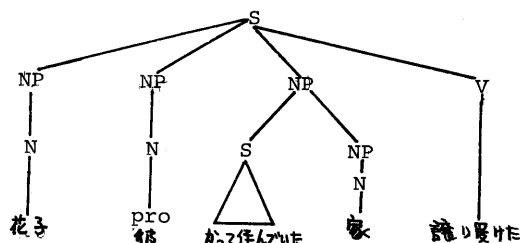


図1. 「花子は、彼から……」の構造

このような条件だけでは、必ずしも、「彼」と「太郎」が同一指示であると断定することはできないが、少なくとも代名詞の指示対象の範囲を限定する役割は果たす。このことは、文脈処理においても、「表層上の語順」(先行の条件)、及び「統語上の構造」(統御の条件)を参照する必要のあること、したがって、文脈処理のように、外界に関する知識に強く依存するような処理においても、言語構造を *cue* として使う必要のあることを示唆している。

(f) 意味処理と統語処理：自然言語理解における意味処理の重要性は、例えば、「の」によって接続される名詞間の意味的関係の決定等を考えれば明らかである。しかしながら、意味処理が重要であるということと、すべての処理が意味によって行なわれるということとは別のことである。重要なことは、統語処理でどの程度のことが行なわれるか、また、どのような統語構造があらわれたときに、どのような意味処理を起動すべきかをきめ細く観察する必要がある。簡単な例を示す。

(18) 日立の本社が東京にある。

(19) 日立が東京に本社がある。

この例は、「象は鼻が長い」と並行的な例であるが、すくなくとも、格助詞「が」でマークされる名詞句が、述部と直接係り受け関係をとらず、文中の名詞句と意味的に関係することを認めなければならない。このような現象は、格助詞「が」あるいは副助詞「は」について生じ、しかも、述部のアスペクトが「状態的」である場合に生じる。また、このように宙にういて「が」名詞句が意味的に関係するのは、述部と「が」格でつながる名詞句である。したがって、このような状

況が文中に生じた場合に、名詞間の意味的関係を調べる適切な処理を起動すればよいことになる。

§3. 計算機処理に必要な機能

§2 で議論したことは、必らずしも日本語や英語の言語現象のすべてを網羅したものでももちろんないし、その定式化にも多くの問題が残されている。実際、これらの現象のための適切な文法モデル(どのような情報を参照すれば良いか etc)を作成することは、それ自体で一つの研究となり得るものであろう。ここで、議論したいのは、このような言語現象を処理するためには、少なくとも §2 で述べたような各種の情報を参照する必要があるということであり、そのために、我々はどのようなソフトウェア体系を用意することができるか、ということである。以下、いくつかの観点から、これを考えてみよう。

[保守の容易性] §2 で述べたのは、言語にみられる規則性の非常に狭い部分を記述したものである。これと同じ程度の複雑な規則性は、いたるところにみられる。これらをすべて把握した後、はじめて自然言語処理が可能になると考えることはできない(このように考えることは、機械処理を全く放棄することにつながる。) 少なくとも、現在抱えられている規則性の範囲内で、計算機処理のためのプログラムを作成する必要がある(このプログラム作成は、作成された文法自体の、気のつかない欠陥を明らかにしてくれるだろう)。

もし、言語のモデルが未完成のまま、計算機処理システムを作成してゆくとしたら、そのシステムは、以後の実験・検討によって頻繁に修正や追加が必要なければならないであろう。Production System にもとづく知識ベースシステムが、evolutional なシステム開発を容易にするためのプログラミング手法であるとしたら、その手法は、自然言語処理プログラムを開発するための手法ともなり得よう。とくに、自然言語の文法記述によく使われる書き換え規則は、Production rule の考え方の基本となったものである。Production System あるいは、書き換え規則の考え方は、個々の知識単位(自然言語処理では文法規則)の独立性を保ち、システムの保守容易性を向上させるという意味において有効であろう。

[記述の汎用性と多重性] これまでの書き換え規則の記述の欠陥は、自然言語の持つ規則性を品詞という単一のレベルで捉えようとしてきたことである。§2 で述べたように、言語の規則性は、品詞レベルだけでなく、

- ① 単・複の区別のような形態素的な情報
- ② ±HUMAN やアスペクトのような意味的な情報
- ③ 特定の格助詞や表層上の単語並びに関する情報
- ④ 統語的な構造・意味的な構造に関する情報

といった、レベルのちがった情報を参照することによって記述される。したがって、文法規則を記述する体系、および処理結果を表現するためのデータ構造は、品詞とそれを基本にした統語構造といった単一の構造ではなく、一つの表現形の中に上記のような各種の情報が有機的に関連づけられて、記述できなければならない

ない。

このような「記述の多重性」の考え方は、人工知能研究の他の分野にもあらわれている。例えば、画像理解における Marr の *Primal Sketch* の考え方も、外部知識という、画像データとは無関係なものに結びつく以前に、画像データがそれ自身で持つ情報をできるだけ落さないように、忠実に記述できる「多重の記述」を持つべきだという主張であろう。例えば、画像データにおいて、Walz の線分に関する規則性を計算機処理に反映しようとするれば、まず、画像データから線分の情報を抽出し、それから、線分に関する規則性を適用することになる。しかしながら、画像の持つ情報は、線分だけでなく、画に関する情報、濃度や濃度勾配に関する情報 etc. があり、しかも、これらが一枚の画面の中に相互に関係し合っ
て混在している。これら別の情報に関する規則性を表現しようとするれば、それらの情報を画面から抽出し、記号化あるいは数量化する必要がある。一枚の画面の持つ情報を、忠実に記号化や数量化しようとする、いくつもの視点から行なわなければならない。また、線分に関する規則の記述が、その線分の周囲をとりま
く面の情報を参照して記述されるとすると、この2つの記号化の結果は、相互に有機的に関係づけられている必要がある。自然言語処理においても同様である。我々は、まず、外部世界の知識という言葉表現とは独立したものと結びつく以前に、言語表現のもつ上記のような様々な情報を有機的に関連づけて表現しなければ
ならない。品詞情報は、あたかも、画像における線分情報のような役割を果たす
ものであり、重要な予掛りとはなるが、それですべての規則性を表現することは
できない。これまでの書き換え規則は、品詞という単一の種類の情報に関する規
則性を記述するものであった。これを拡張する必要がある。

〔プロダクション・システムと言語処理〕 これまで、プロダクション・システムは、知識工学における主要な手法として、各種の応用分野に適用されてきた。とくに、診断システム的な分野への応用が顕著である。何故プロダクション・システムが診断システム的な分野で成功してきたかを考えてみる（あるいは、診断システム以外には適用し難いか、とくに、そのままで言語処理に適用するものが何故困難であるか、を考えてみる）と、プロダクション・システムには、次のような特徴があることがわかる。

診断システムにおける処理は、基本的にもとのデータから判明した情報が *additive* につけ加わる処理であり、つけ加わる順序や、つけ加わった情報間の相互関係は、それほど重要ではない。したがって、データが処理過程全体を通じて、加工されてゆくという通常の計算機処理とは異なる。

診断的システムでは、あるデータ集合から含意されるクラスが決定される。パターン認識の *framework* と同じである。これに対して、航空写真のような画像から、その画像にうつっている対象物についての記述、あるいは、対象物相互間の関係に関する記述を得ようとする、単に情報が *additive* に、既知の情報の集合の中につけ加わってゆくだけでなく、つけ加わる先の「既知の情報集合」自体が構造（画像処理の場合には、空間的位置関係 etc.）を持っているために、この

構造内に適切に新たな情報をつけ加える必要がある。また、規則も、「既知の情報集合」内に、Aという情報が、他の情報とRという関係にあれば、・・・』という形式で定式されることが多い。このような定式化を可能にするためには、プロダクション・システムの規則間のコミュニケーションを支配するWM (Working Memory) にある種の構造を持たせる必要がある。もちろん、述語を使うことによって、通常のWMをそのまま使うこともできようが、この方式には煩雑な操作が必要となろう。このような応用においては、WMが、Blackboardという別紙が与えられる所以である。

言語処理の場合には、この「既知の情報集合」中の要素間の関係が、出発時点では単語の並びであるが、処理が進行するに伴って、本構造となり、さらに複雑になる。とくに、自然言語のもつ構造的なあいまいさのために、「既知の情報集合」の要素間に成立する関係が複数個生じることになり、問題がさらに複雑になる。これまでの書き換え規則によるシステムが使っていた、Recognition Matrixのようなもので、WMに相当する部分を augment する必要がある。

【データとプログラム】 人工知能研究で問題とされた「データとプログラム」の区別が、自然言語処理のプログラムを開発する際、すなわち、自然言語に関する我々の知識を記述する際に、どのような意味をもってしているかについて考えてみよう。「データかプログラムか」の議論を我々なりに整理すると、次のようになる。

これまでデータとみなされてきた知識の多くが、実は、「入力データ」を処理するための手順を規定するという、プログラムとしての性質を持っている。「知識」というデータがあり、それを処理するための基本的な手順がある、と考えるよりも、「知識」そのものがプログラムだと考えられる。ただし、この議論は、どのレベルで「記述」とみるかによって変化する。例えば、論理式集合は、theorem prover によって処理されるデータとみることもできるし、philolog にみられるように、これをプログラムとみることもできる。システムを作成する側からみると、ある記述を行なったときに、その記述形を解釈し処理するための記述(プログラム)を別途自分で用意する必要があるかどうかによって、その記述が「プログラムであるか、データであるか」が決まることになる。

§2 に述べたように、言語表現の規則性の多くは、個々の単語に依存するものが多く、それによって、辞書中に多くの記述を行なう必要がある。システムの evolutionary な改造も、この辞書を通じて行なわれることが多い。もし、辞書の記述を解釈し、処理に反映するためには、プログラムを別途用意しなければならぬとしたら、システムの保守容易性はきわめて悪くなる。すなわち、一般の規則性を記述する文法規則と、個別単語に依存する辞書記述に、大きな区別があって、辞書記述を変更すると、それに伴って、その記述を解釈するための一般の文法規則も修正しなければならぬとしたら、システムの保守容易性はきわめて悪くなる。辞書という、従来はデータとみられていた記述を変更することによって処理手順も柔軟に変更されること、すなわち、辞書記述も一般規則と同じように取扱われることが、システムの保守容易性を向上させるためには必要である。

「辞書の記述を、一般文法規則の記述と同じレベルで扱うようにする」という上記の主張は、「辞書記述の中に、使用者定義のプログラムを埋め込む」という

人工知能研究において従来行われていた主張とは異なる。一般文法規則も辞書記述も同じ枠組で記述でき、かつ、(Prologにおいて論理式をプログラムとみることでできるのと同じ意味で)この記述を解釈実行する機能をシステム側が持つことが必要である、という主張である。

辞書記述や一般規則に、人工知能において従来行われていた「プログラムの埋め込み」を行なうことは、上記の主張とはまた別の意味が必要であろう。すなわち、簡単な意味標識のチェック以上の意味処理や、外部知識の処理を伴う文脈処理を行なうための機能等は、言語表現を処理するための機能とは別の(例えば、推論や解釈の機能)ものがあり、それ専用の記述の体系と処理手順を用意する必要があり。このような機能とリンクは、言語現象を処理するためのソフトウェアシステムからみると、一種の「プロシージャの埋め込み」として実現されることになる。

34. おわりに

現在我々のグループでは、以上のような目的意識から自然言語処理用のソフトウェアシステムを開発している。そのプロトタイプシステムは一応完成し、これによって実験的な文法を記述している。このようなソフトウェアシステムは、実際に文法記述を行ない、その有効性の検証と不備な部分の改良・拡張を不断に行なっていく必要がある。いずれにしても、自然言語処理は、我々が自然言語表現に關して持っている知識を、いかにうまく計算機処理にかかせる形式に記述するか、の研究である。適切な文法規則の作成と同時に、そのような文法を記述でき、処理できるソフトウェアの開発が重要である。

[参考文献]

- 1) 立井・中村: 「自然言語処理のためのソフトウェア構造」, 京都大学数理解析研究所講義録396, 1980
- 2) J. Bresnan: *Realistic Transformational Grammar*, in *Linguistic Theory and Psychological Reality*, MIT Press, 1979
- 3) N. Ostler: *Case Linking; A Theory of Case and Verb Diathesis Applied to Classical Sanskrit*, Ph. D Thesis, MIT, 1979
- 4) 寺津・稲田・山梨: 「日本語における照応現象について(その2)」, 計算機による日本語談話行動の総合モデル化, 昭和54年度研究報告書(代表者: 石綿 敏雄), 1980
- 5) M. Kay: *Functional Grammar*, Xerox, PARC report (1978)
- 6) 中村: 「自然言語解析のための文法記述システム」, 京都大学修士論文, 1981