自然言語処理 35－2
（1983. 1. 25）

自然言語理解の枠組について

白井 英俊
（東京大学 工学部）

1. はじめに

自然言語理解への取組み方は、大きく次の二種に分類できると考えられる。

A. 自然言語の言語学的側面に注目した研究。
B. 言語学的知識よりも人間のもつ常識、推論を重視し、それらの説明のために導入された知識の枠組（図式）に基づいた研究。

A.では特に、アナフォラ解析、限量詞のスコープの問題、自然言語の統語的・意味論的曖昧さ等がその研究の焦点となる。その観点から、意味をどの様に表現すべきか、推論はどの様な役割を果たしているかが研究されている。例えば、モンテギュ文法（以下、MGと呼ぶ）の枠組では、構文解析木と意味表現との間に直接的な写像関係を仮定し、それによって得られた意味表現（論理式）の違いによって限量詞のスコープの問題等による曖昧さを説明している*。また、白井[2]においては、推論は「現在の文と先行文脈との間の意味的な関係を求める」ためのものと考えられている。

一方、Bでは、意味表現とは図式の記述に他ならず、理解とは「新しい情報に対してどの様な図式を思い起こし、それとの照合を行ない（slot-filling）、その情報を図式と整合させ、図式から得てれる『期待』を使える事」を意味している。[3] このアプローチでは、推論は図式を提供し、新情報と図式との照合や期待の生成に用いられる。

A, B共にそれぞれ長所短所はあるが、言語理解の枠組としては、これらを融合したものが適切であると考えられる。[4] 全く何もない所からは理解の生じる余地はない。そういう意味で、何らかの「前理解」（以後モデルと呼ぶ）は必要であり、理解プロセスは、モデルの逐次的な修正・更新のプロセスであると考えるのが妥当であろう。

本論文では、この様な観点に立って、著者の考えている言語理解システム（以後単にシステムと呼ぶ）の枠組について述べる。基本的にはMGの様に、構文解析木から意味表現を写像規則によって生成する。但し、日本語の特性[5]から、直接MG流の論理式を生成するのではなく、中間表現を用意する。この論理式の解釈はモデルに依存して行うが、ここでのモデルは、文によって生成された対象物（モデルのobject）をどんどんモデルに取込むという意味で動的である。ここで強調しておきたいのは、モデルの更新・修正はad hocに行なわれるのではなく、絶えずモデルが無矛盾（consistent）となる様に行なわれるべきであるという事である。これを保証するメカニズムはTMS[6]と呼ばれている。これにより、得られた結論に対する説明の生成やプログラムのデバッグが容易になる事が知られている。[7] これを組み込んだ知識表現言語（のひな型）を作成したので§3で説明する。[12]

2. 理解システムの枠組

2.1 構成

ここでは、システムは話し手が与え

*勿論、それればかりではなく、例えばアナフォラ解析に対する優れた研究も行なわれている。[1]

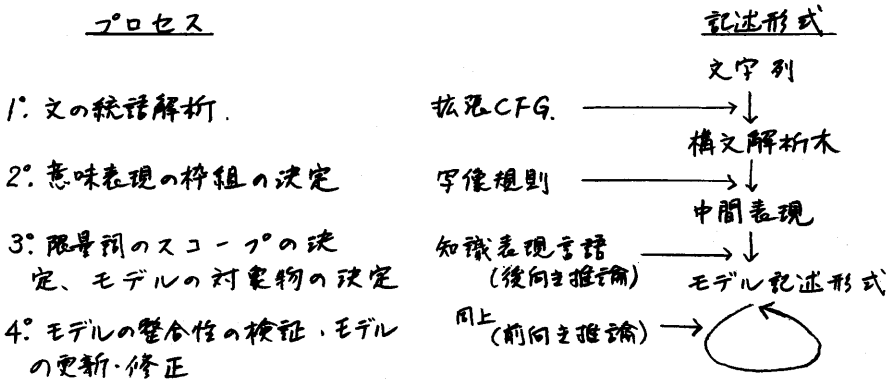

図1. 言語理解におけるプロセスと記述形式

る文を有意味な文と仮定し、その連続体であるテキストから話し手が伝えようとする情報を抽出する事を目的とするものとする。さらに、システムは話し手が誠意を持っている事を仮定し(Griceの会話の公準)、文の順序や語彙の選択に理解を容易にするための手がかりが十分に与えられているものと仮定する。*

この様な仮定に立つテキスト理解システムの構成は図1の様なものと考える。実際には、1°~4°のプロセスは相互に関連し、一つのプロセスが完了した後に次のプロセスが動作するというものではない。著者ら[7]は以前に、統語処理だけでは対処不可能な曖昧さが生じた時に意味処理を行なって効率を高める様なシステムを提案した。図1の構成においてもこの考えを拡張したものが採用できると考えている。

2.2. 中間表現の必要性

図1の枠組で、構文解析木から直接モデル記述形式に変換せず、中間表現を設定したのは次の観察に基づいている。

(1) 桜は育てやすい。
(2) 桜が咲いている。
(3) 桜が欲しい。
(4) 桜は日本中にある。

(1)~(4)の「桜」は、いずれも同じ言語表現で表わされているが、意味的には異なっている。英語では冠詞の区別により総称か否か、specificか否か、等の情報がある程度得られるが、日本語では助詞の違い、述語の性質や時制、アスペクト等の関係から決まるものと考えられる。

Carlson[8]の様に、述語の意味記述にその様な情報を記述しておく方法もある。しかしその様な知識は言語学的知識というより常識の範疇に入るのではないだろうか。次の例を考えられたい。

(5) (犬が家の中に入って来た。) 犬は大きかった。
(6) (次に恐竜の時代が来た。) 恐竜は大きかった。

(5)、(6)の下線部の指示物は、括弧の中の文があれば、それぞれ

犬 = 前文で生成された「犬」(個体)
恐竜 = 「恐竜」の典型例(prototype)

---

* この手がかりがいかなるもので、それにより実際にどういう影響があるのかという研究は今後の課題である。

である事は明らかだが、無い場合でも違いが感じられる。これは「犬」と「恐竜」の知識によるものである。この様に、Carlson流に写像規則（単語の意味記述）によって処理するよりも、知識として処理した方がよいと思われる。

実際には、統語処理の段階でも、常識に頼らざるを得ない部分がある。良く知られた例では、「の」で括られた名詞句同士の係り受けがある。

(7) 門の大きな家，大きな門の家
(8) 角の大きな家，大きな角の家

上例の「門」と「家」、「角」と「家」の関係がそれである。この様に、言語学的知識も含め、知識は扱い易い記述形式で表現しておき、それを扱うプロセスを共同作業させる、という枠組が望ましいと考えている。

2.3. 中間表現の概略

構文解析から写像規則により得られる中間表現は、文の(格)構造と、各々の名詞句等の表層での順番（スコープ）を表現するものである。その統語論は図2.の様になっている。

例えば文(1)に対する中間表現は、下図の様になる。ここで、「る」、「容易」、「育てる」等は、知識表現のことばである。図3は

(る (容易 *b (育てる *b (は (λ (*f) (桜 *f)]

と等価な構造を表わしている。中間表現でこの様な表記法をとらなかったのは、語順の情報を保持したいためで、この語順情報は後に名詞句が限量詞付きであると分かった時点で、そのスコープを定めるのに用いられる。即ち、図3の様な記述形式は、括弧の外から内へ向かって、名詞句がどんな限量詞が付きうるのか、個体を指すのか、等という事を決定していくのに役立ち、内から外へ束縛リストの並びの順に限量詞のスコープが定められる。

紙幅の関係上、詳細は省略する。

2.4. モデル記述形式とそれへの変換

モデルに登録される対象物は、文中で生成された対象（MGでいう所のタイプ1のobject）と、文が主張している事実並びにそれが前提／含蓄している事実（タイプ0のobject）である。TMSが無矛盾に保とうとするのは「事実」間の関係であり、「対象」間の関係（アナフォラ！）を扱うのが文脈推論[7]である。文脈推論自身は知識として知識表現言語のプログラムとして表わされる。

```
(λ nil (る *a)
   (*a (λ (*b) (容易 *b *c)
          (*c (λ nil (育てる *b *d)
                 (*d (λ nil (は *e)
                        (*e (λ (*f) (桜 *f))]
```

図3. 文(1)の中間表現（但し"]"は括弧省略）

```
<記述形式> ::= (λ <変数リスト> <述語リスト> . <束縛リスト>)
<変数リスト> ::= () | (<変数> . <変数リスト>)      ;<変数>は先頭に*のついた英数字列
<述語リスト> ::= () | <述語文> | (and <述語文1> ... <述語文n>)
<述語文> ::= (<述語名> . <変数リスト>)
<束縛リスト> ::= () | (<変数束縛> . <束縛リスト>)
<変数束縛> ::= (<変数> . <記述形式>) | ((<変数> . <定数>) | ...
                                        (nil . <記述形式>)
```

図2. 中間表現の統語論

ここで注意しなければならないのは、アナフォラ解析は一旦モデル記述形式が生成された後に行なわれるという事である。(但し、文内で定められる様な場合を除く。)何故なら、アナフォラ解析の結果(例えば文(5)の括弧内の文の「犬」と次の文の「犬」がcoreferentであるという情報)をもTMSで扱う対象とした方が良いかもしれないからである。

述語の性質、時制、助詞等によって名詞句の素性がどう定められていくかは興味深く、今後研究の焦点の一つとなろうが、ここでは省略する。一つの例としては文献[9]がある。

紙幅の関係で、例をあげてこの節を終える。図3の中間表現からは次の様に二通りのモデル記述形式が得られる。

1°. (タイプ0) S1:
(→ (man x)
(→ (桜 y) (育てる x y)))

2°. (タイプ0) S2: (→ (man x) (育てる x 桜p))
(タイプ1) 桜p: associated_with S2

(注: (man x)が導入されたのは、「育てる」の制約による。また「桜p」は「桜」の典型例.

図4. モデル記述形式の例

3. 知識表現言語

我々が知識表現(プログラミング)言語に対して要請するのは、主として、次の3点である。

1°. 人間の持っている知識を効率的に表現でき、それを基に推論を行える事。
2°. 文から生成された中間表現をモデル記述形式に変換するという様な手続きも記述できる事。
3°. モデルを無矛盾に保つためのメカニズム(TMS)のインタフェースとしても働く事。

1°, 2°だけであればPrologで十分である。全部をPrologで書けばよいのかも知れないが、モデルを無矛盾に保つための制御機構としては、chronological backtrackingよりもdependency directed backtrackingの方が適している。ここではインタフェースのデータ(ルール)をTMSと共有するというシステムを考える。

3.1. TMS

まず、TMSの概略を述べる。TMSは一つの主張に対し真偽値の代わりにinまたはoutという状態を与える。inはその主張を信じている事に対応し、outは信じていない事に対応する。状態を変化させるには「証明」を付けなければならない。証明とは、主張間に、ある依存関係を定める事に対応している。証明は通常、(SL: Supporting List)

(SL (n1, …, ni) (m1, …, mj))

という形をしており($n_1, \cdots, n_i, m_1, \cdots, m_j$はすべて主張)、$n_1 \cdots n_i$がすべてinで、$m_1 \cdots m_j$がすべてoutの時に有効となる。

ある主張を前提(絶対にin)とする事もでき、また、$n_1, \cdots, n_i$がすべてinとなった時は矛盾である、という事も指定できる。

矛盾が起こった時には、TMSはその矛盾を解消する様に動く。具体的には、その矛盾を支えている主張の依存関係を調べ、inとされていたものをoutにする事により矛盾が生じない様にするのである。矛盾が解消できない時には、その旨を使用者に知らせる。

我々のTMSは文献[6]に基づいて作成したが、基本的にはDoyleのシステム[5]と同じである。但し、次の点等で幾分異なる。

1°. 全体を複雑にしているCP型証明を認めていない。

2°. 従って、NGというノードを作らずに矛盾の解消を行う。*

3°. パタンマッチング（unification）によって新しくできたパタンをもTMSの中に取込み、その依存関係をも考えた状態割付けを行う。

3.2. インタフェース

前節の説明で明らかな様に、TMS自身には推論能力はない。従って推論を実際に行うインタフェースが必要である。DoyleのシステムにはAMORD[11]という推論システムがあり、そのルールは一種のLISP関数で、

(Rule (<fact> <pattern>) . <body>)

という形をしている。これは、

1°. 利用者は<body>でパタン（TMSのデータである「主張」）を生成し、これに対し正しくSL型証明を付けなければならない。

2°. ルール自体の真偽はTMSでは扱わない。

というものであった。それに対し、我々のシステムではルール（Prologの「主張」に相当）は変項を含まない原子論理式と前向き推論型のものはTMSの対象となるので、非常に柔軟で強力である。

以下、我々のTMS用インタフェースの概略を説明する。

このシステムは基本的には演繹推論システム（deductive retriever）であり、ルールが一つのプログラムの役割を果たす。ルールをデータベースに格納するには、ADD（ルールを証明なしで導入）、PREMISE（前提として導入）、ASSUME（ルールに相当するパタンをin、(NOT ルール)というパタンを生成してoutとし、互いに依存関係をつける）という述語が用意されている。その他に、ルールは次の3種に分けられる。

A. 一般のルール
(述語 引数1 … 引数n)
例 (IS FIDO DOG)
(LOVE *x *y) ; *のついているのは変項.
(LISP lisp式)
(AND ルール1 … ルールn)

B. 前向き推論型ルール (A, Bをルールとする)
B-1. (→ A B)
B-2. (→ A (ASSUME B))
B-3. (→ (OUT A) B)
B-4. (→ (OUT A) (ASSUME B))
等.
B-2, B-4 は Reiter[37]の表記法によれば $\frac{A:MB}{B}$, $\frac{\neg A:MB}{B}$ にそれぞれ対応する。

C. 後向き推論型ルール
(← A B)

(但し、B型とC型のルールを互いに埋め込む事は許されていない。)

この他、ユーティリティとして、パタンの理由を返すWHY等が用意されている。

以上で明らかな様に、言語としては、Prologを包含しているが、ユーティリティ等が十分ではない。更に、知識表現言語として見た場合、フレームの様にある事物が関与している関係を一まとめにして記述する能力に欠ける。この問題を含めて、モデル中の対象物の扱いは次で説明する。

3.3. モデルの扱い

アナフォラ解析の要点は、先行詞となるべきものが、モデルに登録されて

*この為、記憶領域は小さくてすむが、時間がかかる。これについては文献[10]を見よ。将来、NGをつける事を考えている。（NG: No Good）

いると考えられる事にある。MGでは、文の意味記述である論理式を評価した時には既に、モデル中に対象物が存在しているという仮定をおいている。しかし、実際の談話解析ではむしろ、文の処理が進むにつれて「引起こされた」対象物をモデルに登録し、後で参照された時に検索できる様管理する必要がある。しかも、そういう対象物が<u>ある</u>というだけではだめで、それが<u>どういう性質を持っているか</u>、という事まで検索可能でなければならない。

現在の所は、タイプ0のパタンをルールとして登録し、タイプ1のobjectは別にまとめ(リストで表現)、ある文から生成されたobjectや、あるobjectが関係するパタン(ルール)を検索できる様にしている。

フレーム様の知識構造については、(デフォルトを含めて)ルールとして容易に書き下せるので、マクロを作って扱う事にした。

4. 結び

ここで論じた枠組は、人間の理解過程の一つのモデルである。「完全な」モデルを目指すには、学習の研究が必要なのは言うまでもない。

このモデルでの最大の問題点は、モデルの整合性を保つための計算量と、モデルの拡大に従うメモリの増加である。前者については、停止性は保証されているものの[6]、実際に応用を考えるには後者の問題と合わせて深刻である。組織的な「忘却」メカニズムの導入(パタンに対するGBC)等が必要となるであろう。但し、法令文の無矛盾性の検査等に応用範囲は広いと思われる。

謝辞 日頃御指導頂く伊理正夫教授、文脈推論の道を開いてくれた横尾英俊氏、TMSのインプリメントに協力してくれた鈴木浩之氏に感謝する。

[1] Bach, E. and Partee, B. H., 'Anaphora and Semantic Structure', Proc. of the Chicago Linguistic Society, Vol. 16 (1980)

[2] 白井英俊、「文章理解と意味結合関係」、計量国語学、Vol. 12, No. 7 (1980) pp. 308-320.

[3] Schank R. C., 'Language and Memory', Cognitive Science, Vol. 4, No. 3 (1980) pp.243-284.

[4] Webber B. L., A Formal Approach to Discourse Anphora, Ph. D. thesis, Harvard Univ. (1978)

[5] Doyle, J., Truth Maintenance Systems for Problem Solving, MIT AI Lab, TR-419 (1978)

[6] Charniak, E. et al., Artificial Intelligence Programming, Lawrence Erlbaum Assoc. (1980)

[7] 白井英俊、横尾英俊、「統語処理と意味処理を同時に行うシステムについて」、情報処理学会計算言語学研究会資料 21-2 (1980)

[8] Carlson, G. N., Reference to kinds in English, Ph. D. Diss., Univ. of Massachusetts, (1977)

[9] 横尾英俊、文脈推論を含む日本語理解システム、東京大学工学系修士論文 (1980)

[10] Steele Jr., G. L., The Definition and Implementation of a Computer Programming Language Based on Constraints, MIT AI Lab, TR-595 (1980)

[11] de Kleer, J. et al., AMORD: A Deductive Procedure System, MIT AI Lab, Memo-435 (1978)

[12] 鈴木浩之、論理の一貫性を重視した文脈処理、東京大学工学系修士論文 (1982)

[13] Reiter, R., 'A Logic for Default Reasoning', Artificial Intelligence, Vol. 13, No. 1-2 (1980) pp.81-132.