

構文的予測の分析とその構文解析への応用

Analysis of Syntactic Predictions and Its Application to Parsing

武舎 広幸 東京工業大学理学部情報科学科*
Hiroyuki MUSHIA

*Dept. of Information Science, Faculty of Science, Tokyo Institute of Technology, Tokyo 152, Japan.

1986年5月14日 受理

Keywords : predictive parser, deterministic parser.

Summary

We focus our attention on syntax of natural languages, particularly English, and predictions or expectations that can be made solely with syntactic information during the sentence recognition process are analyzed in detail. We also present machine executable mechanisms that enable proper handling of analyzed aspects and a description method of the mechanisms as grammar rules.

We reinforce Kuno's predictive analyzer by introducing five types of predictions. For each type of prediction, we discuss and present its necessity, its description method, and recognition mechanism. We make use of three kinds of stacks whose behavior is specified by grammar rules in an extended version of Greibach normal form. The extended mechanism provides us with a simple yet still powerful means for recognition of sentence structures. The recognition method can be seen as a deterministic one if we permit looking some words ahead.

We also investigate other factors that affect the predictive recognition process, i. e., preferences among syntactic ambiguities and necessary amount of look ahead. These factors as well as the proposed handling mechanisms of predictions are tested by analyzing four kinds of articles. In our experiment, about seventy percent of sentences are recognized and looking two words ahead seems to be the critical length for the predictive recognition.

1. ま え が き

われわれが普通の文章を読む際、途中で想定しなかった要素に出会い前に戻って読み直すことはまれである。これに対し、現在の計算機による構文解析では、多くの場合後戻りを繰り返しながら処理が行われる。人間が行っている処理を簡単に計算機上を実現できれば、より自然で効率的な自然言語処理システムの構築が可能になるであろう。

本論文では、このような立場から、人間が文章を読む際に行っている構文的予測に注目しその分析を行

い、これを後戻りをしない直線的な構文の解析に役立てる方法を提案する。また、その方法に基づきパーサを作成し実験を行ったので、その結果についても報告する。四つの記事を対象にした実験では、70%程度の構文をとらえることに成功している。

予測に注目した構文解析法として、久野は、『予測分析法』^[1]を提案した。しかし、この方法をそのまま用いて構文解析をすると、頻繁に後戻りを行ってしまう。この原因は、久野の提案した方法(および規則)が人間の行う予測を自然に表現したものになっていないことにある。本論文では、人間の文の理解の過程の分析を基に、具体的に久野の方法と人間の処理との相違を指摘し、また、必要な拡張を提案する。

本論文では、言語の(特に英語の)構文的側面についてのみ考察し、その範囲内での構文解析の方法を提案する。ここでは、前置詞句や形容詞節の修飾先など、意味的な解析が必要なものは、その一つの可能性を提示するにとどめる。これについては、第3節で具体例をあげて詳しく説明する。

以下の議論は、筆者の人間の英語の文章理解の過程に対する(主観的)観察に基づいている。筆者の母国語は日本語であり、英語を外国語として学んできた。筆者の考えでは、われわれがどう英語の文章を理解しているかは、われわれがどう日本語の文章を理解しているかよりも容易に観察ができる。なぜならば、われわれは、英語を意識して学んできており、その認識の過程をより客観的に観察できるからである。

本論文の以下の構成は次のとおりである。次の節では、予測のさまざまな側面を、それを計算機上でうまく処理する方法に力点をおいて分析する。それにつづいて、第3節では、作成したパーサによる実験の結果を見る。最後の節で結論を述べる。

2. 予測の種類

われわれが英語の文章を読んだり聞いたりする際、常に次に何がくるかを予測しながら進んでいく。しかし、この予測にはさまざまなタイプがある。この節では、この予測のタイプについて考察する。以下の各小節で、次の三つの観点からそれぞれのタイプについて論ずる。

- (1) そのタイプの予測の説明とそのタイプの予測の存在を裏付ける例。
- (2) そのタイプの予測を表現する規則の記述法。
- (3) そのタイプの予測を使って文の構造を認識する手順(計算機上での実現)。

本論文で提案する構文の認識の方法(手続き)は、文法規則によって完全に記述される。言い換えると、文法規則と認識のメカニズムが一体となって初めて有効となる。文法規則は動作メカニズムを知ったうえで書かれなければならない。

2・1 必須予測

本小節では、最も単純であり、以下の議論の基となる『必須予測』について論じる。このタイプの予測は、久野の予測分析法で用いられた予測と同じタイプのものと考えることができ、久野^[1]はこのタイプの予測だけでほとんどすべての文法規則を記述した。

普通の英文では、前置詞の後にはほとんどまがいがなく名詞句が、冠詞の後には主名詞(head noun)を含む名詞句の構成要素が続いて現れ、人間もこれらの予測の助けをかりて構文を認識している。このような予測は、通常の句構造文法規則、特に Greibach 標準形の規則によって自然に表現される。たとえば、以下の規則は、(他)動詞句がどのように認識されるかを表現するものと見ることができる。ここで、NP-ART は冠詞に続く名詞句の構成要素を表す。

- (1) VP → vt NP
- (2) NP → article NP-ART
- (3) NP-ART → noun

これらの規則は、右辺第1項の終端記号(単語あるいは品詞)に出会うと、第2項以降の要素を予測すると読むことができる。右辺第2項以降の要素(NP, NP-ART)は、動詞句や名詞句を形成する上で必要な要素であり、以下に続く語によって、その予測が満たされなければならない。

Greibach 標準形の規則による文法の記述は、予測された要素を積んでおくスタックを使うことによって簡単に文の構造の認識に用いることができる^[1]。以下では、この Greibach 標準形による文法規則の記述と、スタックを用いた構文の認識方法を基にし、これを以下の各小節で導入された新しいタイプの予測を扱えるように徐々に拡張していく。ここで述べる枠組は、単純でありながら効果的な解析方法を提供する。

2・2 任意予測

本小節では、『任意予測』を導入する。この種の予測は、名詞句や動詞句などの後置修飾語句の把握に用いられる。

前小節で、Greibach 標準形による規則の記述が、予測的な認識のプロセスを自然に表すことを見た。しかし、規則の記述内容がわれわれの予測を反映したものになっていないと、これに依存した解析は頻繁に後戻りを行ってしまう。特に予測をしすぎないよう規則を記述しなければならない。これを次の例文と規則を用いて説明する。なおここで、PP_CL と ADJ_CL はそれぞれ分詞節および形容詞節を表す。

- (1) There lived a girl.
 - (2) There lived a girl named Kaguya-hime.
 - (3) There lived a girl who loved her parents very much.
- (A-1) NP → article NP-ART
 - (B-1) NP → article NP-ART PP_CL
 - (B-2) NP → article NP-ART ADJ_CL

上の三つの例文は、それぞれ規則 (A-1)、(B-1)、(B-2) を使ってその構造を認識できる。しかし、人間の認識の過程との比較という観点から見ると、(A-1) と (B-1, 2) の間には大きな違いがあることがわかる。A-1 は、前小節で見たように、われわれの予測を自然に表現しているものと見ることができる。しかし、B-1 や B-2 の規則は、名詞句の先頭でその名詞句が分詞や形容詞節によって修飾されることを予測していることに相当し、われわれが行う予測をうまく表現していない。われわれが名詞句の後ろに続く分詞や形容詞節の存在を認識するのは、名詞に必須な主名詞が出現したあとで、それらの修飾節(句)の始まりの要素に気がついてからである。必須な主名詞が現れた時点で、われわれは、そこで名詞句が終わってしまってもよく、また、続いて後置修飾節(句)などが現れてもよい状態で進んでいくと考えられる。

このような状態を表現するために『任意予測』を導入する。たとえば、上の三つの例文は、次のような規則を使って解析する。なおここで、文法規則と認識のメカニズムとの対応を保つため、規則の記述を拡張し『シフトフラッグ』を導入する。

品詞	旧予測	フラッグ	新予測
(C-1) article	NP	→ t	NP-ART
(C-2) noun	NP	→ t	*NP-N
(C-3) rel_pro	NP-N	→ nil	ADJ_CL
(C-4) (vt ptp)	NP-N	→ nil	PP_CL

ここで、rel_pro、(vt ptp) は、それぞれ、関係代名詞、他動詞の過去分詞形を表す。また、*がついた予測 NP-N は任意予測であり、名詞句に必須な主名詞となりうる要素はすでに現れており名詞句がそこで終わってもよい状態になっていることを示す。たとえば、(C-1) の規則は、次のように解釈する。

IF NP を予測しているとき(スタックの最上位の要素が NP のとき)に冠詞に出会った、
THEN 新たに NP-ART を予測し、(シフトフラッグが 't' なので) 次の語の処理に移る。

任意予測は、(C-3) や (C-4) のように必要に応じて後置修飾する節などの予測と入れ替わる。例文(1)のように名詞句がそこで終わってしまう場合は、スタックからただ取り出されるだけで次に進む。言い換えると次の規則が適用されることになる。

NP-N → ε

シフトフラッグが有効となるのは (C-3)、(C-4) の場合である。これによって、次の語の処理に移ることなしに予測を交換することができる。これにより、人間が名詞句が終われる状態で過去分詞や関係代名詞

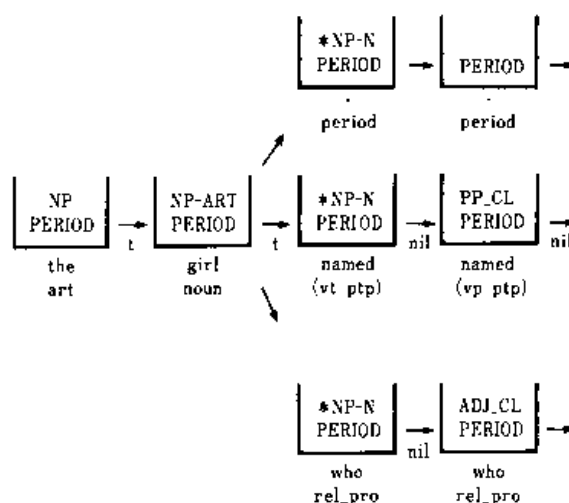


Fig. 1 Handling of the optional prediction.

に出会って初めて分詞 (PP_CL) や形容詞節 (ADJ_CL) を予測する様子を自然に表現することができる。Fig. 1 に、これらの規則を使った上の三つの例文の解析の過程(スタックの状態変化)を示す。この Fig. 1 において、スタックの下にその時点で参照している語およびその構文的分類(品詞)を示し、矢印の下は、シフトフラッグを示す。

2-3 束 予 測

本小節では、構文要素のいくつかを一度に予測することを許す『束予測』を導入する。ここでは、be 動詞の後に続くうる構文要素を例に、このような種類の予測が必要であることを示す。次の小節では、このような予測が等位接続詞に続く要素の解析に有効であることを示す。

動詞句の先頭の be 動詞の後には、名詞句・形容詞・前置詞句・過去分詞句などさまざまな構文要素が現れる可能性がある。われわれの文の認識の過程で、“am”, “are”, “was” といった単語に出会ったとき、これらいくつかの可能性の中から、その特定の場面に現れる要素を前もって予測はできない。一つの可能性としては、次に続く単語を見て、初め何らかの要素を予測するということが考えられる。しかし、次の例のように、be 動詞の後に、理論上無制限の長さをもつような挿入が現れうることから、この段階で何らかの予測をして次に続くうるものの可能性を絞っていることがわかる。

(1) He was, as we had expected, in the park.

(2) He was, as we had expected, running in the park.

このようにいくつかの可能性をもったゆるやかな予測を表すため『束予測』を導入する。束予測では、可能性のある構文要素をまとめて扱い、あとでこの中から適当なものを選択する。可能性のある予測の集合を新たに一つの予測として独立に扱うこともできるが、次小節で見る等位接続詞に続く要素の表現や、動詞の後に続く要素の多様性を考えると、新しい予測の導入は不適当と考えられる。

束 (bunch) 予測は、次のように記述する。なお、ここで (VP fnt) および (VP ing) はそれぞれ定形の動詞および ing 形の動詞で始まる動詞句を示す。

(be fnt) (VP fnt) → t
[bunch (NP) (ADJ)(PP)((VP ing))]

*VP MOD

束予測は、全体が一つの予測として扱われ、他の予測と同じようにスタックに積まれる。それがスタックから取り出されると、現在参照している語などを参考にして、そのうちから一つが選ばれ、あたかもその選ばれた予測がスタックに積んであったかのごとく扱われて処理が行われる。

2・4 and スタック

本小節では、等位接続される要素を処理するために、新たなスタック『and スタック』を導入する。ここで述べる方法は文献 [2] や [3] で述べられているものと類似しているが、and スタックと前小節で導入した束予測によって単純に処理が行える。and スタックは予測を積んでおくスタック (以後『予測スタック』と呼ぶ) とまったく同じ深さを持ち、現在満たそうとしている予測のリストを記憶している。等位接続詞に出会うと、このリストの要素を束にして予測することによって、後続の構文をとらえることができる。

等位接続詞の出現は、普通予測することができず、その出現が特別の操作を引き起こすきっかけとなる。これを次の文を例にして見る。

- (1) Mary had a little lamb and a kitten.
- (2) Mary had a little lamb and washed him everyday.
- (3) Mary had a little lamb and she was always with him.

この例から明らかなように、たとえば、節の先頭でその後に等位接続されたもう一つの節が現れるといったことを予測することはできない。したがって、次のような規則を使っては、予測的な処理は実現することができない。

(A-1) NP → NP and NP

(A 2) VP → VP and VP

(A-3) S → S and S

また、他の多くの単語ではその単語を見ただけで次に続く要素がかなり限定できるのに対して、and を見ただけでは、ほとんど何も予測ができない点で、著しく異なっている。and に続く要素の予測に影響を与えるのは and 自身ではなく、それ以前に構築されているものである。

上の例文では、and の前で、名詞句 (NP)、動詞句 (VP) および節 (S) が作られており、それらが再び現れている。われわれは、文の認識の過程で現在どんな構文単位の要素を構築しているかを記憶していると考えられる (Fig. 2 参照)。

and スタックはこの記憶の役目をするものであり、これにより上のような文の構造を、簡単に把握することができる。Fig. 3 に、予測スタックと and スタックの状態の変化を示す (なお、この Fig. 3 では、不必要な詳細は省略してある)。状態 (ii) において、最初の予測 S は、先頭の語 "Mary" の参照によって (参照している語を示すポインタを進めずに) NP と (VP fnt) に取って代わられる。このとき、and スタックの下段には、予測スタックの下段 (VP fnt) に対応して ((VP fnt) S) という予測のリストが置かれる。これは、予測スタックの (VP fnt) という予測を満たそうとしている間、(VP fnt) と S という二つの予測を満た

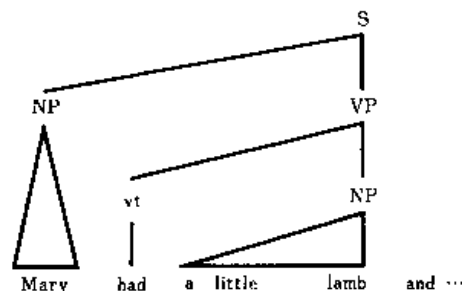
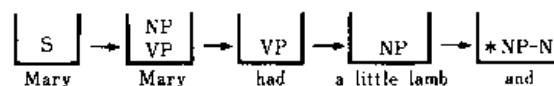


Fig. 2 Structures constructed before the conjunction.

prediction stack



and stack

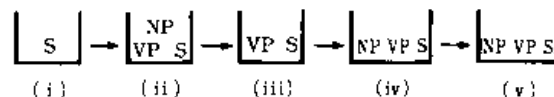


Fig. 3 Relation between the and stack and the prediction stack.

そうとしている(その二つの構文単位を完成しようとしている)ことを示している。

同様に処理が進み, and の直前では, (NP (VP fnt) S) というリストが and スタックに積まれていることになる。and に続く構文をとらえるためにここで必要なのは, NP-N に代わってこのリストを, 次のような束予測としてスタックに置くだけでよいことになる。

[bunch (NP) ((VP fnt)) (S)]

and に続く語によってこれらの中から適当なものを選ぶことによって, その構文構造の認識を行うことができる。

なお, この方法によって次のような文も認識することができる。

Mary looked for and found the unicorn.

この文の and の前では (NP PP (VP fnt) S) というリストが and スタックの最上位要素として作られる。このうちの (VP fnt) が使われて and に続く要素の構造をとらえることができる。

上記の操作を行うときに適用される規則は, 下のよう記述する。

and ?P → t (special and_stack)

ここで, ?P は, この規則の適用が, 現在の予測(予測スタックの最上位要素)に依存しないことを示す。

2・5 挿入構造

ある種の語はわれわれに挿入構造の始まりを示し, すでに出された予測を満たすことを一時的に中断させる。ある種の副詞・前置詞・副詞節などがこれらの挿入構造として現れる。また, コンマはこれらの前後に現れることが多い。次の例で引用符で囲った部分が挿入構造である。

- (1) There are economic risks and "generally" a lack of available date.
- (2) He adapted "for the use of linguists" the existing system of formalization.
- (3) He was, "as we had expected," running in the park.

これらの構造を規則で表現するために, 下の表記法

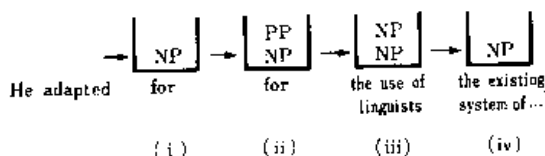


Fig. 4 Handling of insertive structures.

を用いる。

(A-1) adverb ?P → t

(A-2) preposition ?P → nil PP

(A-3) comma ?P → t ADV_CL

これらの規則は, 等位接続詞を扱う規則と同様に, 現在参照している語が品詞部の条件さえ満たせば, NP-ART などを除くほとんどすべての予測に対して適用可能となる(冠詞の後に前置詞句などが挿入されることはない)。ただし, 等位接続詞の場合と異なり, 予測スタックの最上位要素は取り出されず, その上に新たな予測が(もしあれば)積み上げられる。

たとえば, (2)の文章は, Fig. 4のような経過をたどって認識される。状態(i)では, 動詞“adapted”の目的語である名詞句(NP)が予測されている。(i)と(ii)の間で, (A-2)の規則が適用され, 新しい予測PPがスタックに積まれ, 予測NP(およびそれを含む動詞句(VP))を満たそうとすることは, いったん中断される。新しい予測PPが満たされると, (iv)の状態になり, 動詞“adapted”の目的語のNPの処理が行われる。(3)の文を処理する際には, “,”が(A-3)の規則を起動し, “as we had expected”が副詞句としてとらえられる。この後に続く“running in the park”の部分は, 2・3節で見たように, 束予測の中から現在分詞で始まるものが選ばれて処理される。

2・6 NP スタック

本小節では, さらにもう一つスタックを導入する。これを『NPスタック』と呼び, 形容詞節などその一部に名詞句の欠けた要素を持つ構造の把握に用いる。この考え方は, 広く用いられている(文献[4]など)が, 人間の行っている処理との対応を考えた場合も妥当な方法と考えられる。われわれは, たとえば, 完全な節の予測と名詞句の欠けた節といった予測を無関係に扱っているわけではない。むしろ, 関係代名詞などに会ったときに, 続いて名詞句の欠けたものが現れるということ記憶し(言い換えると, 名詞句をスタックに積んで), それを後で必要になったときに取り出すといった操作を行っていると考えられる。このとき使われるスタックを『NPスタック』と呼ぶことにする。

次のように規則を記述して, 上記の操作を行う。

rel_pro ADJ_CL → nil +S

ここで, Sの前に付いている‘+’によって, 名詞句が一つNPスタックに積まれ, 後で必要になったときにそこから取り出される。

3. 実 験

前節で述べた予測についての考察を基にパーサを作成し、いくつかの文章について実験を行った。実際のパーサでは、このほかに、単語単位の先読みと、構文的に曖昧な解釈の間の優先度 (preference) を使っている。本節では、まず、次の小節でこの二つの要因について触れ、続いてパーサの出力の表現方法を解析の手続きとの関係から示し、最後に実験結果を示す。

3・1 先読みと優先度

人間が内的に行っている局所的な決定の遅延を反映させるために、先読みを導入する必要がある。ここでは、Marcus^[5]の決定性パーサと異なり、単語をその単位として用いている。

われわれが文を理解する際、その途中でそれまで構成したすべての構文構造の子細をはっきりと記憶しておらず、構成しようとしている要素のほうを強く意識してと思われる (これは、文献 [6] にも述べられている)。いったん、次にくるであろう構文要素を予測してしまうと、われわれの関心はその新しい予測を満たすことに移行してしまう。現在構築中の構文要素を完成するために、振り返って前に作り上げた要素を意識することは少ない。構文要素単位の先読みを許すことは、この振り返って前の要素を見ることを許すことに対応する。この枠組においてはたとえば次のような規則の記述が許されてしまう。

IF 次の要素が名詞句で、その次の要素が動詞句

THEN その名詞句と動詞句を文Sの子とせよ。

この規則では何語か前に作られた名詞句が再び参照されている。このような枠組は、人間の内的モデルとしては強力過ぎるように思われる。ここでは、Marcusのパーサで構文単位の先読みが用いているものの役目を、適切で一般的な予測と、限られた長さの語単位の先読みによって代用する。

作成したパーサでは、この先読みの長さを変えて、必要な長さを検討した。なお、パーサの実際の動作は、処理の単純さと効率を考慮して、この長さの範囲で後戻りを許す形で実現されている。つまり、文の i 番目の単語の処理が始まると、 $i-k$ 番目の語の処理で決定した構造は決して取り消されない (ここで、 k は先読みの長さ)。処理がここへ戻った場合は、解析に失敗したものとみなしている。

構文的な曖昧さの間の優先度は、以下のように扱っている。多品詞語の品詞は、順序づけられている。文法規則は、大きく A・B 二つのグループに分けている。挿入構造をとらえる規則や特殊な規則が B グループには入っている。方策はまだ固定していないが、いずれにしろ、各単語の品詞の一つ一つが順番に従って試みられ、また、A グループの規則が B グループの規則の前に試みられる。

3・2 構文木の構造

このパーサの出力は、Fig. 5 のような木構造である。なお、出力は、解析木の高さが、いたずらに大きくならないように、たとえば、NP-ART といった中間的構文単位は省略して表示する。先に述べたように、前置詞句の修飾先など、意味的な解析が必要なものは、その一つの可能性を提示するにとどめる。これらの構造は、前節で示したもののうちの、任意予測によってとらえるもの、and スタックによってとらえるもの、および挿入構造としてとらえるもの、のいずれかとして処理される。

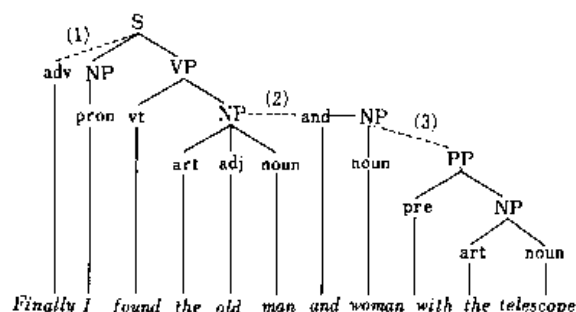


Fig. 5 Tree structure constructed by the parser.

Fig. 5 において、(1) は挿入構造として、(2) は and スタックを使って、(3) の前置詞句は任意予測によってそれぞれとらえられる。これらの構造を『ペンディング構造』という。ペンディング構造のうち、並列構造の場合は、それに先行する適当な予測に並列な要素としていったん扱われ、挿入あるいは、任意予測によってとらえられる後置修飾節・句などは、先行する予測を修飾するものとして、把握される。しかし、この例で、(3) の前置詞句が上のレベルの動詞句を修飾するという解釈も、(3) のリンクを上に移動するだけで得ることができる。実験では、必要ならば、このようなペンディング構造の移動を行って正しい構造が得られる場合は、解析に成功したとする。

3・3 実験結果

前節で述べた予測のさまざまなタイプに対応した処

Table 1 Result of an experiment.

	total	success	ratio
article 1	85 sentences	65 sentences	77 %
2	85	70	82
3	54	42	78
4	84	54	64

理に、3・1節で述べた先読みと優先度を加えた処理をするパーサを作成し、いくつかの文章を入力して実験を行った。Table 1にその結果を示す。先読みの長さは2語である。入力した文は、それぞれの記事の先頭から、タイトルなど文になっていないものを除いたものである。記事1はソフトウェアのマニュアル、記事2は経済関係の抄録記事、記事3、4は異なる日本人の書いた計算機関係の論文^{[7],[8]}である。実験の方法は、まず記事1、2について基本的な規則を作り、それをもとに文をパーサに入力し、その結果を検討しルールを修正することを何回か繰り返した。記事3、4については規則をわずかに加えただけで優先度などの変更は行っていない。

失敗の原因は、主に、(1)文法的なチェック(数の一致など)の不足によるもの、(2)品詞の優先度と規則の優先度の関係によるもの、(3)予測のメカニズムの改善を必要とするものがある。この中で、特に次の例のように(1)と(2)が関連しているものが多い。

(A-1) Actors model conceptual or physical entities which...

(A-2) ... a simple message queue sorted in the arrival order is ...

(A-1)では、“Actors”が後続する名詞“model”を修飾するものとして、また(A-2)では“queue”を他動詞としてとらえてしまった(“sorted”は目的語となる名詞を修飾すると思えるため、先読みでも排除できない)。これらは、“model”では名詞が動詞より、また、“queue”では動詞が名詞より優先度が高くなっていたことと、数の一致のチェックを行っていないことが原因となっている。改善の方向としては、文法の細かいチェックを行うことと優先度を見直すことが考えられる。後者については、特に使われ方の差が大きいもの(例としては、“can”や“will”の助動詞と名詞など)には大きな効果があるが、対象文の分野や個人的な差も考慮する必要がある。このため、まず、文法的なチェックをできるだけ細かくするほうが効果的と思われる。

現在のメカニズムでは、次のような例文をいつも正

しく解析することができない。

(B-1) Rules in Greibach normal form are suitable for expressing our predictive recognition process and these rules are ...

(B-2) Laying stress upon their proper handling by computers, we discuss ...

(B-1)では、“and”の直前で節(S)も名詞句(NP)も作られており(andスタックの最上位要素にあるリストの中にSもNPもはいつている)、次の“these rules”が節の主語の名詞句になるのか、“process”を主名詞とする名詞句と並列になる名詞句なのか先読みの範囲ではうまく判定できない。また、(B-2)のような文の先頭では、ingで始まる動詞句が主節の主語になる可能性を否定できない。これは、to不定詞で始まる文についても同様である。実験した文章では、すべてが分詞構文やto不定詞の副詞的用法であり、これらを優先しておくことで多くの場合好ましい結果が得られると考えられるが、根本的な解決とはならない。これらをいつも正しく解析するためには、新たなメカニズムを導入する必要がある。拡張の方向としては、andスタックによって記憶されている、構築されたばかりの要素のリストを使って新たな処理を行うことが考えられる。

次のような文に現れるコンマや括弧の処理は、後にさまざまな要素が現れむずかしい。これらについてはより詳しい分析が必要である。

(C-1) An actor is always in one of two states, active or inactive and ...

(C-2) We focus our attention on syntax of natural language, particularly english, and ...

(C-3) Behaviors of computer systems, ...are result of highly concurrent (independent) activities.

(C-4) Each actor has its own clock (frame of time reference) which is ...

(C-5) ... where X is the specified memory area (or a variable).

先読みの長さや解析の成功・失敗との関係は、記事1、2について、先読みの長さを3語にしたときと2語のときと結果の比較を行った。長さを3語にしたことによって失敗が成功に変わったものは、両方の文章のうちともにわずかに1文ずつであった。いったん誤った方向にいつてしまうと、わずかな後戻りだけでは修正がきかないようである。

行った。その結果、7割程度の文を期待どおりに把握することに成功した。

4. む す び

英文を理解する際に、出される構文的予測を後戻りをしない直線的な解析に役立てるという観点から分析し、六つの異なったタイプの予測があることを指摘した。また、それぞれのタイプの予測を表現する文法規則の記述法、およびそれを構文の認識の手続きの記述として解釈する方法を示した。その枠組は、非常に簡潔でありながら効果的な解析方法を可能にする。

この方法に基づき、実際にパーサを作成して実験を行った。この実験では、予測的な処理に関連して単語単位の先読みと、曖昧な構造の間に優先度を導入して

謝 辞

指導教官である米澤明憲助教授には、この原稿に関して数多くの有益なコメントをいただきました。ここに深く感謝いたします。

日ごろディスカッションをしていただく米澤研究室および木村研究室の皆様にもさまざまなコメントをいただきました。ここに感謝いたします。最後に、本研究のために貴重な辞書の使用をお許しいただいた、リソースシェアリング(株)にも感謝いたします。

◇ 参 考 文 献 ◇

- [1] Kuno, S.: The Predictive Analyzer and a Path Elimination Technique, Commun. ACM, Vol. 8, No. 7, pp. 453-462 (1965).
- [2] Winograd, T.: Understanding Natural Language, Academic Press, New York (1972).
- [3] Boguraev, B. K.: Recognising Conjunctions within the ATN Framework, pp. 39-45, in Automatic Natural Language Parsing (K. S. Jones and Y. Wilks, eds.), Ellis Horwood Limited (1983).
- [4] 今野総, 田中徳積: 左外置を考慮したボトムアップ構文解析システム, コンピュータソフトウェア, Vol. 3, No. 2, pp. 19-29 (1986).
- [5] Marcus, M. P.: A Theory of Syntactic Recognition for Natural Language, MIT Press, Cambridge (1980).
- [6] 橋田浩一, 山田尚勇: 自然言語の認識過程の心理学的なモデル, 自然言語処理技術シンポジウム論文集, 情報処理学会 (1984).
- [7] Musha, H.: A New Predictive Analyzer of English, Proc. of COLING '86, pp. 470-472 (1986).
- [8] Yonezawa, Y. and Matsuda H.: Towards Object Oriented Concurrent Programming, 第6回ソフトウェア科学・工学における数理的方法研究集会資料 (1984).

[担当編集委員: 石崎 俊]

著 者 紹 介



武舎 広幸

1980年国際基督教大学教養学部理学科卒業、83年山梨大学工学部計算機科学科修士課程修了。リソースシェアリング入社。85年東京工業大学理学部情報科学科博士課程入学在学中。現在米同オハイオ州立大学留学中。情報処理学会、ソフトウェア科学会、ACM、ACL会員。