

| | |
|--------------|---|
| Title | 日本語文の構文 ・ 意味解析システムに関する研究 |
| Author(s) | 池田, 尚志 |
| Citation | 大阪大学, 1987, 博士論文 |
| Version Type | VoR |
| URL | https://hdl.handle.net/11094/1155 |
| rights | |
| Note | |

Osaka University Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

Osaka University

日本語文の構文・意味解析システム
に関する研究

昭和61年9月

池田尚志

内容梗概

本論文は、筆者が電子技術総合研究所において行った「日本語文の構文・意味解析システムに関する研究」についてまとめたものである。

自然言語の解析手法としては、従来、句構造モデルの枠組みのもとで、文脈自由文法を基礎としたさまざまな手法が研究されてきた。句構造文法では、文の部分構造を統語カテゴリとして階層的に分類し、どのように統語カテゴリが組み合わさっていれば正常な文として認定できるかということ、統語カテゴリの生成規則として表現する。構文解析の結果は、ノードに統語カテゴリを、リーフ(末端のノード)に単語を配置した木構造として表現される。このモデルの基調にあるのは、どのような単語の並びが文をなすのかという見方である。従って、英語のように語順が統語関係に重要な役割を演ずる言語を分析する基本的な枠組みとしては自然なものであるといえよう。

ところが日本語は膠着語であって、語順が構文において果す役割は英語とはかなり異なっている。日本語には文節という明確な構文的単位が存在し、文節内の語の並びには強い制約がある。一方、文節相互間の並びには、係りの文節が受けの文節に先行すること、および言語一般の制約としての非交叉条件のほかには、文体／文脈／語用論上の要因を除けば構文上の制約はない。構文関係を表現するには、文節の並び順ではなく、文節中の機能語(格助詞)が重要な役割を果たす。しかし、格助詞もなく、直接に意味上の関係から構文関係が決まることも多い。また英語では、たとえば主語を省略するなどのことは不可能であるが、日本語では、意味的に省略可能な文節は構文上も省略可能である。このように日本語は意味主導型の構造をしていると言える。

本研究は、このような日本語文の構造を素直に反映した解析手法を提案したものである。ここでは、句構造モデルではなく依存構造モデルを用いた。日本語文における依存構造とは、伝統的な国文法でいう係り受けの構造にほかならない。このモデルでは構文解析の結果は、親ノードに受けの語を、子ノードに係りの語を配置した木構造である。これは、句構造がもつばら統語関係を表現しているのと比べると、述語論理様の意味的な関係の表現となっており、意味主導型である日本語文を解析する枠組みとして適している。またここでは、構文解析と意味解析を統合した手法を提案した。一般に、構文解析の後に意味解析を行って、可能な構文図式の中から意味的に妥当なものを選び出すというのが広く行われている手法である。しかし意味上の制約が強く働く日本語文の解析では、構文関係と意味関係を同時に扱っていくほうが有利である。また一般に心理モデルとしても、構文と意味を同時に処理するほうが自然である。

従来の日本語文の解析システムでは、意味解析の段階で動詞の格構造を用いることが広く行われてきた。格構造による解析は、名詞とそれを受ける動詞との間の係り受け関係を解析しているものとみなすことができる。しかし、動詞の格構造による係り受け関係以外の係り受け関係の扱いや、格構造の変形(表層格シフト)、連体埋め込み文の扱い、格構造の意味記述と完全には整合しない

文の扱いなどを含んだ、統一的な係り受け解析としての枠組みを示したものはない。

本研究は、このような観点から、一連の係り受け規則によって構文解析と意味解析を一体化して行う、統一的な係り受け解析方式を提案したものである。

また、句構造モデルに基づく方式では書き換え規則のような規則表示を用いるが、言語現象には個別的な事項が多く、規則表示の形では例外処理などが複雑化する問題があった。本研究では、規則表示の形ではなく、すべての文法規則を内容語、機能語および文節カテゴリの属性として辞書上に記述する方式を提案した。これによって、個別的な言語事実も素直に表現することができるとともに、解析システム本体を簡潔に構成することができた。

第1章は序論で、本研究の背景、動機、目的等について述べた。

第2章では、本論文で提案する構文・意味解析システムの全体構成、基本的な方式について述べた。本システムは、文節構造を解析する部門と、文節間の係り受け関係を解析する部門からなる。係り受け解析は、shift-reduce-parsing を基本的な構図とする方式である。

第3章～第5章で解析システムの各部分について述べた。

第3章では、文節構造の解析手法について述べた。自律語+付属語の図式(表層の文節構造)から内容語+機能語の図式(深層の文節構造)へ移行するための整形規則を導入した。これは、一貫した係り受け方式による解析システムを構築するのに貢献している。

第4章では、格構造規則の構成および格構造変形処理の手法について述べた。格構造を用言だけでなく任意の内容語が持ち得る属性として一般化した。これによって、格構造を内容語の個別的な用法を記述する手段として位置づけることができ、体言や連体埋め込み文などの係り受け関係も統一的に処理することができる。用言については文型と対応づけて格役割を設定した。使役や受身等による格構造の変形は、文型の変換ととらえることができる。

第5章では、係り受け解析について述べた。ある語句へ係るための、またはある語句を受けるための構文・意味上の制約条件を記述する一連の係り受け規則を設定した。格構造も係り受け規則の一つとして位置づけ、統一的な枠組みを得た。文節カテゴリ、機能語、内容語の係り受け規則の記述例を示し、連体埋め込み構造や数量表現、種々の呼応関係などの解析も、この枠組みの中で統一的に処理できることを示した。

第6章では、前章までに述べた方式による解析の結果を、機械翻訳に応用する手法について述べ、解析・翻訳の実験結果を示した。これによって、本方式による構文・意味解析の有効性を確認した。翻訳手法としては、解析された係り受け関係から直接翻訳規則を導く手法を提案した。

第7章では、このシステムの実験環境について述べた。キー順ファイルに直接アクセスできるようにリスプの機能を拡張し、キー順ファイル上の辞書に対するエディタを作成した。プログラムや辞書データ、入力文の管理などを、すべてキー順ファイルをベースとして行うことにより、データやプログラムの修正や追加にも容易に対応できる、簡便な実験環境を整えることができた。

第8章では、本研究の結果をまとめ、文脈処理など今後の課題について述べた。

目 次

| | |
|--------------------------------|----|
| 第1章 序 論 | 1 |
| 1.1 本研究の背景 | 1 |
| 1.2 本研究の目的, 動機 | 5 |
| 1.3 本研究の範囲と論文の構成 | 8 |
| 第2章 構文・意味解析システムの構成 | 10 |
| 2.1 はじめに | 10 |
| 2.2 解析システムの全体構成と解析の基本的手法 | 11 |
| 2.3 解釈の多義性の扱い | 13 |
| 2.4 まとめ | 17 |
| 第3章 文節構造の解析 | 19 |
| 3.1 はじめに | 19 |
| 3.2 前処理 | 20 |
| 3.3 文節構造 | 21 |
| 3.4 文節構造の抽出 | 22 |
| 3.4.1 文節構造の抽出手順 | 22 |
| 3.4.2 接続処理 | 23 |
| 3.4.3 未登録語の処理 | 27 |
| 3.5 文節構造の整形 | 28 |
| 3.6 まとめ | 30 |
| 第4章 格構造の処理 | 31 |
| 4.1 はじめに | 31 |
| 4.2 格構造処理の手順 | 31 |
| 4.3 格構造規則 | 32 |
| 4.3.1 格構造 | 32 |
| 4.3.2 文の種類と格構造 | 36 |
| 4.3.2.1 用言の格構造 | 36 |
| 4.3.2.2 体言の格構造 | 42 |

| | | |
|-------|----------------|----|
| 4.3.3 | 意味特徴 | 43 |
| 4.4 | 格構造の変形 | 46 |
| 4.5 | まとめ | 48 |
| 第5章 | 係り受け解析 | 50 |
| 5.1 | はじめに | 50 |
| 5.2 | 係り受け解析の手順 | 50 |
| 5.2.1 | 係り受け規則の種類と適用手順 | 50 |
| 5.2.2 | 連体埋め込み構造の解析 | 53 |
| 5.2.3 | 後処理 | 56 |
| 5.3 | 係り受け規則のプログラミング | 58 |
| 5.3.1 | プログラミングの環境 | 58 |
| 5.3.2 | 係り受け関係の種類 | 60 |
| 5.4 | 係り受け規則の記述例 | 61 |
| 5.4.1 | 文節カテゴリの係り受け規則 | 61 |
| 5.4.2 | 機能語の係り受け規則 | 65 |
| 5.4.3 | 内容語の係り受け規則 | 67 |
| 5.5 | まとめ | 70 |
| 第6章 | 機械翻訳への応用 | 71 |
| 6.1 | はじめに | 71 |
| 6.2 | 翻訳システム | 71 |
| 6.2.1 | 翻訳システムの構成 | 71 |
| 6.2.2 | 翻訳処理の例 | 74 |
| 6.3 | 解析・翻訳実験 | 76 |
| 6.3.1 | 翻訳規則の記述例 | 76 |
| 6.3.2 | 解析 翻訳実験例 | 81 |
| 6.4 | まとめ | 87 |
| 第7章 | 辞書データベースと実験環境 | 89 |
| 7.1 | はじめに | 89 |
| 7.2 | キー順ファイルへのアクセス | 90 |
| 7.3 | 辞書データベースの構造 | 91 |
| 7.3.1 | キー順ファイル上の構造 | 91 |

| | |
|-----------------------|-----|
| 7.3.2 内部記憶上の構造 | 92 |
| 7.4 辞書エディタと実験環境 | 93 |
| 7.5 まとめ | 94 |
| 第8章 結 論 | 96 |
| 8.1 研究結果の概要 | 96 |
| 8.2 今後の課題 | 97 |
| 謝 辞 | 99 |
| 参考文献 | 100 |

日本語文の構文・意味解析システム に関する研究

池田尚志

第1章 序 論

1.1 本研究の背景

自然言語の研究は古くて新しい研究テーマである。古くは紀元前のパーニニによるサンスクリット語の文法の研究や、アリストテレスによる論理的立場からの研究をはじめ、今日まで巨大な研究の歴史がある^{31),33)}。しかし、電子計算機出現以来の、機械で言語を処理するという手法を伴った研究は、自然言語研究の歴史に新しい時代を画するものである。電子計算機の出現以来、言語を単に論ずるだけでなく、工学上の対象として分析し、加工し、あるいは蓄積・検索するなどしてさまざまな場面で用立てる道が開けてきた。既にワードプロセッサは実用化して広範に普及している。機械翻訳でも部分的実用化をめざしたシステムが商品化されており、さらに進んだシステムの研究が行われている。また、大量の言語データの中から意味的に照合するものを検索するシステム(内容検索)や、入力したテキストの要約を作り出すシステム(自動抄録)、あるいは自然言語で計算機と対話するシステム(質問応答システム/自然言語インターフェースシステム)など、さまざまな応用をめざした自然言語システムの研究が行われている。さらに、音声入力・音声出力による自然言語システムを構築するための、音声認識・合成の研究開発も行われている。しかし、人間の言語処理能力は、人間における情報処理機構の全体と関わっており、その全体的なモデル化という点からは未だ程遠いのが現状である。

(1) 自然言語処理研究の流れ

自然言語処理の研究は、電子計算機が出現した1940年代に、機械翻訳への期待として始まった¹⁵⁾。人工衛星スプートニクによって宇宙開発でソ連に先行されたアメリカは、1950年代の後半から1960年代にかけて、露一英を中心に、自動翻訳の研究に期待し援助した。このことが世界的にも機械翻訳の研究を活発に促すこととなった。しかし60年代の半ば(1965年)、ALPAC(Auto-

matic Language Processing Advisory Committee) が政府に提出した報告書²⁾ (ALPAC レポートとして知られている) では、

- ① 機械翻訳は質、速さ、コストいずれの点でも人手による翻訳に劣る。
- ② 近い将来それらが十分に改善される見込みは無い。
- ③ アメリカでは現在、機械にたよって翻訳をしなければならないようなニーズは無い。

という判断が示される。これに基づいて、アメリカ政府は機械翻訳システムへの過大な期待を見直し、研究開発のための援助を大巾に後退させることとなった。これによって機械翻訳の研究は、アメリカだけでなく世界的にも下火となっていった^{3),11),49)}。

この時期の言語処理は統語論が中心であって¹⁴⁾、言葉の意味に関わる問題に関してはほとんど手をつけるに至っていない。これは当時の計算機の容量、速度、ソフトウェア環境等の制約によるところもあり、また言語理論において、N. Chomsky⁷⁾の生成文法を中心に統語論上の整備が進んだ時期であったことも関連している。

その後 1970 年代にはいって、自然言語処理においては意味理解という本質的な問題に取り組まなければならないという自覚が強まり、意味理解の問題を前面におしだした自然言語理解の研究というよびかたで、人工知能の中心的な研究課題としてとりくまれるようになった^{20),21),26)}。1971 年に T. Winograd が発表した SHRDLU というシステムは²⁷⁾、自然言語理解の研究の出発点となった古典的なもので、積木をいじる仮想的なロボットと英語で対話するシステムである。SHRDLU は、入力された文を理解して、質問文なら自然言語で応答し、命令文なら指示に従ってディスプレイ上で積木を動かす。こうして、言語を理解するとはどういうことであるかということが全体的に追及され、積木の世界の中での非常に限定された領域ではあるが、自然言語によって動作する具体的なシステムとして示された。その後さまざまな対象領域で、多くの質問応答システムが作成され、データベースへの問い合わせや、エキスパートシステムのルールの入力部などへの応用が試みられた^{25),84)}。これらの研究は、言語理解とは何かという探求と同時に、自然言語で計算機に問い合わせる、あるいは自然言語で計算機に指令するというように、人間がより容易に計算機を活用できる環境の構築をめざした応用を指向している。

1980 年代にはいと、特に日本で、機械翻訳の研究が再び盛んに行われるようになってきた⁹⁷⁾。これは、

- ① 日本語と欧米語、特に英語との間の翻訳のニーズはもともと非常に高い。
- ② かつて機械翻訳が盛んに研究された時と比べると、計算機の容量、速度、ソフトウェア環境などが著しく進展している。
- ③ また、この間の自然言語処理技術の進展も著しい。

などの点を背景として、機械翻訳への期待が再び高まってきたものである。民間企業においても研究開発は盛んであり、すでに製品化されているものもある。

以上のように自然言語処理の研究は、機械翻訳システムや計算機との対話システムという夢の実

現を大きな目標として進んできた。そして計算機技術の著しい進歩を支えとして発展し、夢は一部現実のものとなりつつある。しかし現状は、自然言語の仕組みが解明されたという状態からは程遠く、機械翻訳システムにしても自然言語インターフェイスシステムにしても、実現しているシステムは使用条件が強く制限されており、未だ試用段階という状況である。まだまだ多くの試みがなされなければならない段階である。

(2) 自然言語解析技術の現状

自然言語処理の中核技術は、入力文を解析する技術である。機械翻訳システムでは、解析結果を目標言語の表現に変換し、対話システムでは、解析結果を内部の知識ベースの表現に変換して必要な問題解決を行い応答文を生成する(図 1.1)。従って、自然言語システムでは、入力文の解析技術がまず第 1 に重要である。自然言語の解析は、形態素解析、構文解析、意味解析、文脈解析の 4 つのフェーズを経て行われるのが普通である⁹³⁾。

① 形態素解析

形態素解析では、入力文から単語の列を切り出す。(英語の場合は、単語とそれに結合している数や時制などの情報を抽出する。)

英語のように分ち書きがなされる言語では形態素解析には困難な問題はほとんどないが、日本語では分ち書きがなされないので問題が生じる。特に、仮名またはローマ字でのべた書き入力の場合には、多くの曖昧さをともなうのが通例である。

かれがくるまではしらなかった

→ 彼が来るまで走らなかった／彼が車で走らなかった／彼が来るまでは知らなかった／...

これは、構文や意味あるいは文脈まで考慮しないと、完全には解決できない問題である。漢字かな混じりの通常の表記の場合には、このような曖昧さはほとんどなく、実用上問題にはならないが、漢字で書くか、仮名で書くかといった正書法が確立しているわけではないので、問題は残る。

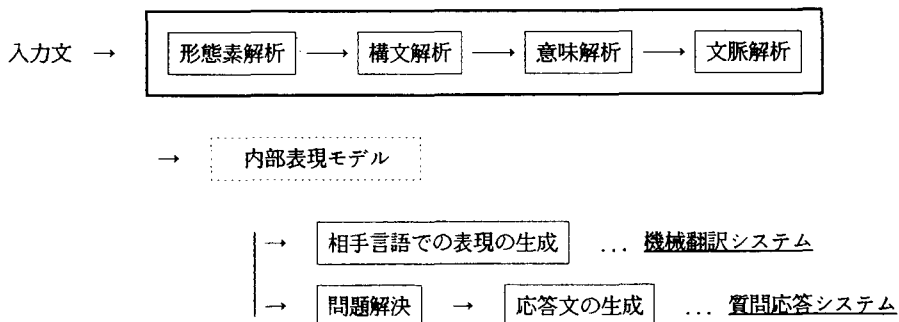


図 1.1 自然言語システム

お水 ⇒ 汚水／おみず

またここでは、辞書の見出しをどのように構成すれば照合処理にとって、また記憶容量の観点から効率的であるかといったことや、辞書に登録されていない未知語をどう処理するかといった研究課題がある。

② 構文解析

構文解析では、形態素解析によって得られた品詞情報などをもとに、文の構造を抽出する。

文の構造とは、N. Chomsky⁷⁾が提案した生成文法で使われている、句構造を指すのが普通である。そして、構文解析システムとしては、句構造文法のひとつである文脈自由文法をベースとして、規則の適用条件などを付加できるように拡張したモデルに基づくシステムが開発されている^{9),28)}。ATN²⁹⁾や LINGOL^{19),83)}などが良く知られている。形式言語においては、文脈自由文法に対する解析アルゴリズムとして確立された効率的な手法があるので^{1),8)}、それを利用することができる。また最近では、Chomskyの変形文法ではなく、句構造文法における新しい枠組みとして、GPSG¹⁰⁾ (Generalized Phrase Structure Grammar) や、LFG⁴⁾ (Lexical Functional Grammar) などの言語理論も出され、自然言語処理への適用も行われている。

しかし文の構造は、構文規則だけではなく、意味や文脈を考慮しないと決定できないことも多い。たとえば、

They are flying planes.

では、flying を形容詞とみるか、現在分詞とみるかで二通りの解釈がある。構文構造の曖昧さは、どのような構文規則を設定しておくかにもよる。実際の言語事象を、全般にわたって十分に記述する構文規則を求めるといふ点では、未だ非常に不十分である。また、種々の言語事実を細かく規則化しようとする、個々の単語の意味との関わりなど個別的な事項が多く、文脈自由文法の書き換え規則のような規則表示の形では例外処理が複雑になり、見通しが悪くなってしまふということが問題点として指摘されている。また、日本語文に対しては、英語に対する発想を源点とする句構造モデルが適当であるかどうかという問題もある。構文解析技術は、言語モデルの問題を含め、意味解析さらには文脈解析と統合した解析手法の研究など、未だ多様な試みが必要である⁸⁸⁾。

③ 意味解析

意味解析では、構文解析の結果をもとに、文が表現している意味内容について解析する。

構文解析の結果が複数あるときには、意味解析によって意味的に妥当でないものを除くことができる。たとえば、

「太郎と京都に行く」

には、構文的には次の2つの解釈が有り得る。

- a. (太郎ト京都) に行く
- b. 太郎トイッショニ京都に行く

しかし意味解析によって、b は意味的に解釈可能であるが、a は意味的に解釈困難であることがわ

かる。

このように、正しい構文解析を導くためにも、意味の扱いを避けて通ることはできない。また、言語理解のモデルに基づく質問応答システムや機械翻訳システムを実現するためには、意味解析が中心的な役割を果たす。しかし、意味とは何か、意味をどのように表現し、どのように取り扱えばよいのかということについては、未だ一般的な解はない。意味表現については、意味ネットワーク、格フレーム、概念依存図式、述語論理などの手法が提案されてきたが、未だ決定版はない⁹⁸⁾。

④ 文脈解析

文脈解析(あるいは、文章解析/談話解析)では、それ以前に入力された文との関連を調べ、指示語や省略語を同定し、文章としての構造を解析する。同定の問題は、指示語や省略の場合だけでなく、一般に、名詞が指し示しているものは何かという、言語表現と内部の記憶表象との対応の問題である。このような問題の解決には、これまでの言語学ではあまり研究されてこなかった、省略や指示語の使用に関する規則(談話文法)を明らかにしなければならない⁷⁹⁾。また、一般常識など言語外の知識とそれに基づく推論なども必要とする。たとえば、

「太郎は弟をなぐった。彼は泣きだした。」

において、「彼」が、「太郎」でなく「弟」を指すと判断するのは一般常識による。また、「弟」が、「太郎の弟」であるか「話者の弟」であるかは、語られている場面/文脈による。文脈解析ではまた、

①「窓を開けて、手を出さないでください」

②「窓を開けて、窒息しないようにしてください」

のような、否定辞の勢力範囲の問題も指摘されているが、この問題の解決にも知識に基づく推論を必要とする。

文脈解析については、やっと研究が始まったという状態であり、人工知能や認知科学の方面からの研究も含め、今後の取り組みが求められている⁶⁹⁾。

研究課題としては、この他、

- ・語彙や文法知識を、自動的ないし半自動的に構築していくための学習のメカニズムの研究、
- ・対象とする文章をうまく制約するための制限文法の研究など、自然言語処理を実用化するための技術^{92),108)}、

など、様々の課題がある。このように、自然言語処理の研究課題は未だ山積しており、多方面からの研究努力が必要である。

1.2 本研究の目的、動機

本研究は、日本語文の解析技術に関する研究であり、日本語文に対する構文・意味解析の手法を提案し、計算機上に具体化することを目的とする。日本語文に対しても、前節で述べたような句構

造モデルをベースとした構文解析がよく行われるが⁸³⁾、本研究は、欧米語とは異なった構造を持つ日本語文に対して、その構造を素直に反映した解析システムを構築することが目的である。本研究では、意味解析のレベルまでを扱い、構文解析と意味解析を統合した、決定論的な係り受け解析システムを提案する。

句構造文法では、文の部分構造を統語カテゴリとして階層的に分類し、どのように統語カテゴリが組み合さっていれば正常な文として認定できるかということ、統語カテゴリの書換え規則として表現する。構文解析の結果は、ノードに統語カテゴリを、リーフ(末端のノード)に単語を配置した木構造として表現される。このモデルの基調にあるのは、どのような単語の並びが文をなすのかという見方である。従って、英語のように、語順が統語関係に重要な役割を演ずる言語を分析する基本的な枠組みとしては自然なものであるといえよう。

ところが日本語は膠着語であって、語順が統語関係において果す役割は、英語とはかなり異なっている。日本語には文節という明確な構文的単位が存在し、文節内の語の並びには強い制約がある。一方、文節相互間の並びには、係りの文節が受けの文節に先行すること、および言語一般の制約としての非交叉条件のほかには、文体/文脈/語用論上の要因を除けば構文上の制約はない。(文節内の語の並びと文節の並びとは明確に区別すべきものであり、語順という言葉で統一することは適当でない。) 構文関係を表現するには、文節の並び順ではなく、文節中の機能語(格助詞)が重要な役割を果す。格助詞もなく、直接に意味上の関係から構文関係が決まることも多い。また英語では、たとえば主語を省略するなどのことは不可能であるが、日本語では、意味的に省略可能な文節は構文上も省略可能である。(英語ではそのために代名詞が発達しているが、日本語ではそれほど発達していない。) このように日本語は意味主導型の構造を持った言語であると言える。

たとえば、英語の場合には、句構造表現で最初に現れる名詞句〔図 1.2 では John, すなわち、 $S \rightarrow NP+VP$ における NP〕はその文の主語であるといったことが読みとれるが、日本語の場合にはそのようなことは言えない。図 1.2 で、「太郎は」と「りんごが」は順序が入れ替ってもよいし、また省略されてもよい。格助詞「が」がつけば主語である、ということも必ずしも言えないし、格助詞が使われない場合も多い。また、「主語」というような概念自身も、日本語にとって適切なものではない^{103),104)}。

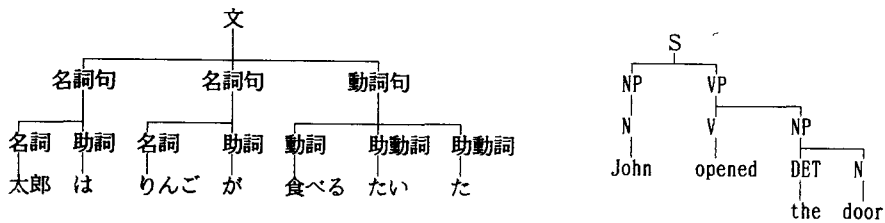
本研究は、このような日本語文の構造に適合した解析手法の提案を試みたものである。ここでは、句構造モデルではなく依存構造モデルを用いる(図 1.3)。日本語文における依存構造とは、伝統的な国文法という係り受けの構造にはかならない。このモデルでは構文解析の結果は、親ノードに受けの語を、子ノードに係りの語を配置した木構造である。これは、句構造がもつぱら統語関係を表現しているのと比べると、述語論理様の意味的な関係の表現となっており、意味主導型である日本語文を解析する枠組みとして適している。またここでは、構文解析と意味解析を統合した手法を提案する。前節で述べたように、一般には構文解析の後に意味解析を行って、可能な構文図式の中から意味的に妥当なものを選び出すというのが広く行われている方法である。しかし、意味上の

制約が強く働く日本語文の解析では、構文関係と意味関係を同時に扱っていくほうが有利である。また一般に心理モデルとしても、構文と意味を同時に処理するほうが自然である。

従来の日本語文の解析システムでは、意味解析の段階で動詞の格構造を用いることが広く行われてきた(64),78),83),89),90),99),100),105)。この格構造による解析は、名詞とそれを受ける動詞との間の係り受け関係を解析しているものとみなすことができる。しかし、動詞の格構造による係り受け関係以外の係り受け関係の扱いや、格構造の変形(表層格シフト)、連体埋め込み文の扱い、格構造の意味記述と完全には整合しない文の扱いなども含み、構文解析と一体化した統一的な係り受け解析方式の枠組みとしては、十分なものは示されてこなかったように思われる。

また、句構造モデルに基づく従来の方式では書き換え規則のような規則表示を用いるが、言語現

(a) 句構造モデル



(b) 依存構造モデル

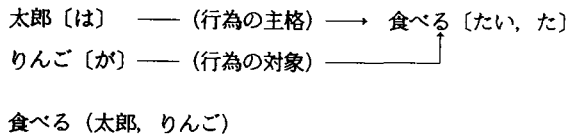


図 1.2 句構造モデルと依存構造モデル

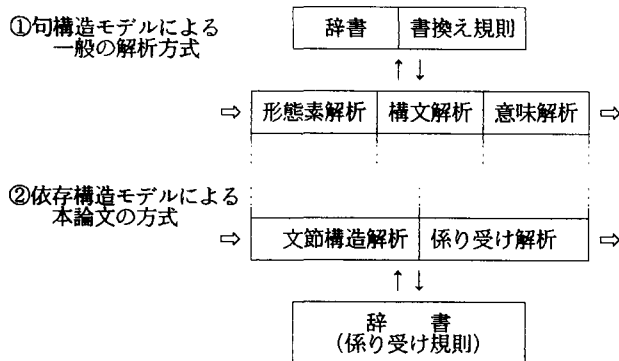


図 1.3 解析方式の比較

象には個別的な事項が多く、規則表示の形では例外処理などが複雑化する問題があった。本研究では、規則表示の形ではなく、すべての文法規則を内容語、機能語および文節カテゴリの属性として辞書上に記述する方式を提案する。これによって、個別的な言語事実も素直に表現することができるとともに、解析システム本体を簡潔に構成することができる。

本研究を開始する際、筆者が、人間の言語処理過程について抱いたイメージは以下のようなものであった⁴²⁾。本論文で提案する解析の手法は、このイメージが下敷となっている。

語はあたかも化学における反応基であり、語の系列はあたかもこの反応基がつぎつぎ反応していく過程である。文はあたかもこの反応による化学生成物である。幾つかの生成物のうち最も安定したものが、通常、文の意味としてイメージされる。この反応は”体験・知識”という溶液の中で進行する。溶液が異なれば反応も一般には異なる。

格構造は反応基の第1近似であり、係り受け解析におけるルールは反応方程式の第1近似である。

1.3 本研究の範囲と論文の構成

本研究は、前節で述べたように、日本語文の構文・意味解析の手法について考察し、具体的な解析システムを提案するものである。本研究ではまた、この構文・意味解析の結果を機械翻訳に応用する手法についても考察して、機械翻訳システムを構成した(図1.4)。これによって、本方式による構文・意味解析の有効性を確認した。

本研究で対象とする解析のレベルは、意味解析のレベルまでである。文脈解析についても若干の考察を述べたが、文脈解析の方式を確立し、構文・意味解析と統合したシステムを構成することは今後の課題である。

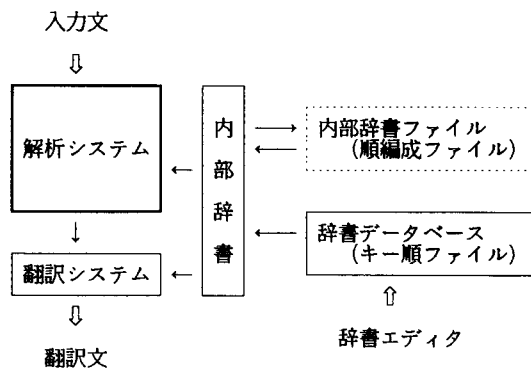


図1.4 構文・意味解析システムの構成

本論文は8章からなる。第2章から第5章までを使って、解析の手法および解析システムについて述べる^{12),30),42),61)}。第6章で翻訳システムについて^{54),58),59)}、また第7章で、本研究のために開発した辞書エディタなどシステムの支援環境について述べる^{57),60)}。

第2章では、本論文で提案する構文・意味解析システムの全体構成、基本的な方式について述べる。本システムは、文節構造を解析する部門と、文節間の係り受け関係を解析する部門からなる。係り受け解析の方式は shift-reduce-parsing を基本的な構図としている。

第3章～第5章で解析システムの各部分について述べる。

第3章では、文節構造の解析手法について述べる。自律語＋付属語の図式(表層の文節構造)から内容語＋機能語の図式(深層の文節構造)へ移行するための整形規則を導入した。これは、一貫した係り受け方式による解析システムを構築するのに貢献している。

第4章では、格構造規則の構成および格構造変形処理の手法について述べる。ここでは、格構造を用言だけでなく任意の内容語が持ち得る属性として一般化した。これによって、格構造を内容語の個別的な用法を記述する手段として位置づけることができ、体言や連体埋め込み文などの係り受け関係も統一的に処理することができる。用言については文型と対応づけて格役割を設定し、文が表現している内容と構文上の特徴との関連についても考察した⁵⁶⁾。使役や受身等による格構造の変形は、文型の変換ととらえることができる。

第5章では、係り受け解析について述べる。ある語句へ係るための、またはある語句を受けのための、構文・意味上の制約条件を記述する一連の係り受け規則を設定した。格構造も係り受け規則の一つとして位置づけ、統一的な枠組みを得た。文節カテゴリ、機能語、内容語の係り受け規則の記述例を示し、連体埋め込み構造や数量表現、種々の呼応関係などの解析も、この枠組みの中で統一的に処理できることを示す。

第6章では、前章までに述べた方式による解析の結果を、機械翻訳に応用する手法について述べ、解析・翻訳の実験結果を示す。これによって、本方式による構文・意味解析の有効性を確認した。翻訳手法としては、解析された係り受け関係から直接翻訳規則を導く手法を提案する。

第7章では、このシステムの実験環境について述べる。キー順ファイルに直接アクセスできるようにリスプの機能を拡張し、キー順ファイル上の辞書に対するエディタを作成した。プログラムや辞書データ、入力文の管理などを、すべてキー順ファイルをベースとして行うことにより、データやプログラムの修正や追加にも容易に対応できる、簡便な実験環境を整えることができた。

第8章では、本研究の結果をまとめ、文脈処理など今後の課題について述べる^{38),62)}。

第2章 構文・意味解析システムの構成

2.1 はじめに

本章では、構文・意味解析システムの全体構成と、解析手法の概要について述べる。本システムは、1.2で述べたように、日本語文の構造に即した手法として、係り受け解析の観点から一貫した構文・意味解析手法の提案を試みたものである。日本語文の解析システムとしては、これまでに格構造を用いた係り受け解析の手法によるものが提案されてきているが、句構造モデルをベースとしたATNやLINGOLなどのように統一的な簡潔な枠組みとして定着したものはない。これは、係り受け解析は基本的に意味主導型の解析法であるが、意味関係の制約を含む係り受け規則の設定が、十分に一般化された形で整理されていないからだと考えられる。本システムの係り受け規則については第5章で具体的に述べるが、本システムでは、係り受け規則を語および文節カテゴリの属性として整理している。解析システムは、この係り受け規則を順次に評価することによって隣接する部分構造間の係り受け関係を解析し、その積み重ねによって文全体の解析を行うという簡潔な枠組みとなっている。以下に本手法の特徴を要約する。

(I) 構文解析と意味解析を融合した手法である。

(II) 原理的な手順は、

① 文節構造を抽出する。

② プッシュダウンスタックを利用して、隣接する部分構造間の係り受け関係を調べる。

の繰り返しであり、極めて単純である。係り受け構造の解析は、shift-reduce-parsingを基本機構とする決定論的な解析手法となっている。

- 係り受け解析においては、係り受け関係の認定を0か1かではなく、“係り受け関係の良さ”として評価し“最も良い係り受け構造”を追及する処理を行っている。これによって格構造の意味記述と完全には整合しない文など、不確かな係り受け関係にも対処できる。
- 係り受け関係の認定が断定的に進めば、文の解析は文末の文節の処理の終了と同時に終了する。すなわち、実時間的である。

(III) 文法規則はすべて文節カテゴリ、自律語、または機能語の属性として表現しており、個別的な事項に対して柔軟に対処できる。また、外部の文法規則との照合を必要としないので、処理が高速である。

- 擬語を導入できるようにすることで、係り受け処理の一貫性を図った。
- 係り受け規則として係りの規則および受けの規則を設け、文節カテゴリ、自律語、あるいは機能語の属性として記述した。格構造は自律語の属性としての受けの規則と位置づけた。

- ・格構造は動詞のみならず任意の自律語が持ち得る属性として一般化した。
- ・使役や受身などの助動詞の働きは、格構造を変形するプログラムを助動詞の属性としてもたせることで表現した。

2.2で解析システムの基本機構について述べ、2.3で多義性への対処について述べる。

2.2 解析システムの全体構成と解析の基本的手法

図2.1に、構文・意味解析システムの構成を示す。

解析システムは、文節構造解析部と格構造処理部、および係り受け解析部からなる。

文節構造解析部は、入力文字列の文頭側から順次に文節を切り出す。文節構造解析部は、前処理部、文節構造抽出部、文節構造整形部からなる。

- ・前処理部は、数値の列や読点などで区切られた部分毎のかたまりに分割する。
- ・文節構造抽出部は、入力文字列の文頭側から順次に、自律語とそれに後接する付属語からなる文節を切り出す。
- ・文節構造抽出部で抽出される文節構造は、入力文中の自律語と付属語をそのまま切り出しただけのもの（いわば表層の文節構造；自律語＋{付属語}）である。文節構造整形部では、必要に

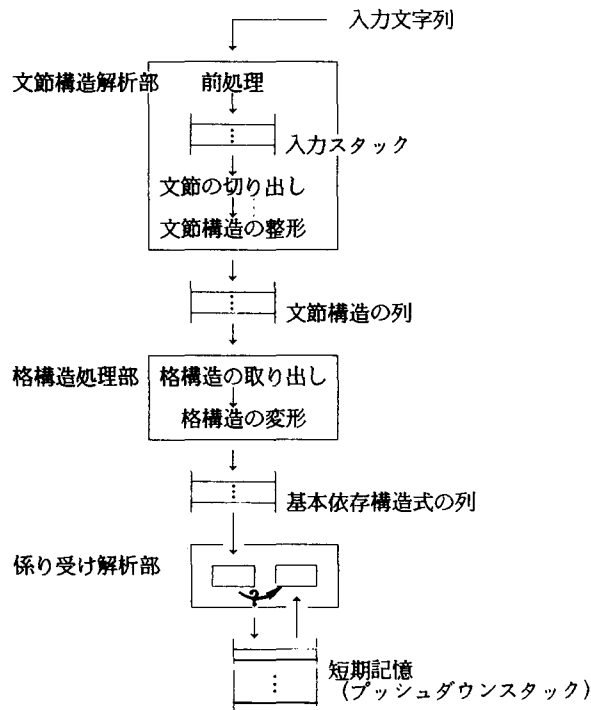


図2.1 構文・意味解析システムの構成

応じてこれに適当な変更を加え、係り受け解析に適合した形（いわば深層の文節構造：内容語＋{機能語}）に整形する。

たとえば、「食べたのは」という文節は、

((連用体言文節 は の た) (食べる))

という文節構造から、

((連用体言文節 は) (の*))

((連体用言文節 た) (食べる))

という2つの文節構造に分割する

格構造処理部は、その文節の内容語の格構造規則を辞書から取り出し、文節構造を基本依存構造式に展開する。さらに、その基本依存構造式に対して、その文節中の機能語に応じた変形処理（受身変形や使役変形など）を施す。

係り受け解析部では、係り受け解析が局所的に終了し、受けの依存構造式を待っている状態にある依存構造式を記憶しておく短期記憶として、プッシュダウンスタックを用いる。係り受け解析部は、格構造処理部から送られてくる格構造式（基本依存構造式）に対し、順次、次のようにして先行する語句との間の係り受け関係を調べる。これは、**shift-reduce-parsing**の構図である。

- D1. まず基本依存構造式を受けの項に置き、短期記憶上の依存構造式を係りの項に置いて、2項間の係り受けを調べる。
- D2. 係り受け関係が認められれば、その2項を新たな依存構造式として統合して受けの項に置き、短期記憶上の次の依存構造式を係りの項に置いて、同様に係り受け関係の処理を繰り返す。
- D3. 係り受け関係がなければ、係りの項の依存構造式を短期記憶に戻し、受けの項の依存構造式を短期記憶に積む。

このように文頭側から順に認識を進め、最後の文節の処理が終わったときに短期記憶上に唯1個の依存構造式が残されていれば、それが入力文に対する構文・意味構造である。図2.2にこの原理的な手順を、簡単な例で示す。

以上が、本解析方式の基本的な図式である。

このように全く単純な手順であるが、先に述べた日本語文の構造的特徴は、この中に自然に埋め込まれている。この手順は実時間的な手順であって、文の入力と同時進行的に解析処理が進行する。係り受け関係の判定が断定的に進めば、文の入力が終わった時点で解析処理は終了する。作業用の短期記憶として用いるプッシュダウンスタックは、ひとつの文節に係る係りの文節の個数程度の深さがあればよいことになる。

この方式では、文節を抽出するための規則群と、文節間の係り受け関係を調べるための規則群がある。

係り受け関係を記述する規則としては、係りの文節から導かれそれがどのような文節に係り得る

かを記述した係りの規則と、受けの文節から導かれそれがどのような文節を受け得るかを記述した受けの規則がある。これらの係り受け規則は、自律語あるいは機能語（助詞）また文節カテゴリの属性として記述してある。格構造は、受けの規則のひとつであり、動詞のみならず任意の自律語が持ち得る属性として一般化してある。たとえば、

「魚を焼く煙」

のように連体修飾される名詞が、修飾する側の文の格要素とならない場合（付加名詞連体修飾）⁶⁰には、その付加連体名詞（すなわち「煙」）の格構造として、連体型用言文節を格要素として持とうことを記述しておくことで処理することが出来る。

「れる」、「させる」等の機能語（助動詞）は格構造を変化させる機能を持つが（表層格シフト）、この機能も助動詞の属性として記述されており、文節構造を変形するプログラムとして評価される。

文節構造を抽出する規則（活用規則、接続規則、文節カテゴリ規則、文節構造整形規則）もすべて語の属性であり、本方式は辞書中の語彙規則主導型の方式となっている。

2.3 解釈の多義性の扱い

解析の基本的な図式は前節で述べたとうりであるが、実際に解析を進めていくに際しては、様々のあいまいさ、解釈の多義性の問題に対処しなければならない。文節の切出しにおいても、係り受け関係の判定においても複数の解釈が可能である場合がでてくる。

人間は、話されている状況や文脈の中で、言語外の膨大な知識をも駆使して文を理解している。

| | |
|--|--------------------|
| 入力文... | <昨日、太郎の弟は東京へ出かけた。> |
| 文節を切り出す... | <昨日、太郎の弟は東京へ出かけた。> |
| スタックに入れる... ((昨日、)) | <太郎の弟は東京へ出かけた。> |
| 文節を切り出す... ((昨日、)) | <太郎の弟は東京へ出かけた。> |
| 係り受けを調べる... | <弟は東京へ出かけた。> |
| スタックに入れる... ((昨日、) (太郎の)) | <弟は東京へ出かけた。> |
| 文節を切り出す... ((昨日、) (太郎の)) | <弟は東京へ出かけた。> |
| 係り受けを調べる... ((昨日、)) [弟は {太郎の}] | <東京へ出かけた。> |
| 係り受けを調べる... | <東京へ出かけた。> |
| スタックに入れる... ((昨日、) (弟は {太郎の})) | <東京へ出かけた。> |
| 文節を切り出す... ((昨日、) (弟は {太郎の})) | <東京へ出かけた。> |
| 係り受けを調べる... | <出かけた。> |
| スタックへ入れる... ((昨日、) (弟は {太郎の}) (東京へ)) | <出かけた。> |
| 文節を切り出す... ((昨日、) (弟は {太郎の}) (東京へ)) | <出かけた。> |
| 係り受けを調べる... ((昨日、) (弟は {太郎の})) [出かけた。 {東京へ}] <> | <> |
| 係り受けを調べる... ((昨日、)) [出かけた。 {東京へ} {弟は {太郎の}}] <> | <> |
| 係り受けを調べる... () [出かけた。 {東京へ} {弟は {太郎の}} {昨日、}] <> | <> |
| スタックへ入れる... ((出かけた。 {東京へ} {弟は {太郎の}} {昨日、})) <> | <> |

図 2.2 shift-reduce-parsing を基本構造とする係り受け解析

その結果、文節の認識も、係り受け関係の判定も、ほとんどの場合あいまいさなく一意的に行われて、文末の語が入力された時点で、直ちに、文全体の一意的な解釈を得ているものと思われる。

機械処理においては、設定する諸規則はいずれにせよ近似的なものであり、規則の粗さとシステムの規則解釈の仕方に応じて種々の複数の解釈の可能性がでてくる。以下に、本方式で採用した解釈の多義性に関する扱いについて述べる。

文節構造抽出処理における対処

- ① 自立語は最長一致で切り出されるものを正しいものとした。

これは、最長一致法としてよく用いられるヒューリスティクスである。

- ② 最長の文字列を切り出す文節を正しい文節であるとした。

これは、たとえば、「僕のもの」からは「僕の」を切り出すが、「僕のもの」からは「僕のもの」を切り出してしまうから一般には成立しない。しかし漢字仮名混じり文に関してはそのようなケースは稀であるから、個々に規則をたてて対処するか、そのような部分は分かち書きすることを想定してこのような対処のしかたをした。このような避難処置と比較して、この仮定を置かない場合の処理の負担は未登録後の処理法ともからんで大きい。なお、これは漢字仮名混じりの通常の表記での入力を前提としての仮定だが、カナ又はローマ字入力とする場合には、文節以下の単位に分かち書きしておけばよい。また仮に音声入力を想定したとしても、音声情報は文節分かち書きに相当する以上の情報を含んでいるはずである。

- ③ 文節の付属語部分は、通常一意的に解析できると前提した。

ただし、「泳ぎ、」に対して((体言)泳ぎ)と((連用用言)泳ぐ)があるように、一意的に解析できなかった場合には、整形処理、格構造処理を保留したまま係り受け解析部門に送り出し、係り受け解析部門で妥当な文節構造を選択した後、あらためて整形処理、格構造処理を経て、係り受け解析を行なう。

格構造処理における対処

- ① 切り出した1つの文節、即ち自律語+(付属語)は複数の解釈を持ち得る。即ち、自律語には同形異語が有り得るし、またそれぞれ複数の格構造を持ち得る。

この結果、係り受け解析部への入力は、複数の解釈に対応した格構造式(基本依存構造式)のリストとなる。係り受け解析は、2つの依存構造式のすべての対に対して行われる。

係り受け解析部における対処

- ① 係り受け解析部門では、係りの位置にある依存構造式のリストと、受けの位置にある依存構造式のリストとの間で、全ての組み合わせについて係り受け関係を調べる。係り受け関係は、係り受け規則との適合度がある値以上であるとき成立するとみなす。

係り受け関係があれば、2つの依存構造式を一つの依存構造式に統合し、係り受け関係の適合度をもとに、統合された依存構造式の評価値を算出する。このようにして生成された依存構造式の中で、評価値が相対的にある値より高いものだけを取り出す。

- ② いったん成立した係り受け関係は破棄しない。すなわち後戻り処理はしない。ただし、連体埋め込み構造に関しては、後戻り処理を含んだ別途の扱いとする。
- ③ 文末の文節の処理が終了した段階で、短期記憶上に唯1個の要素が積まれていれば解析成功であるが、複数の要素が積まれているなら解析は完結していないことになる。
- 本方式では、係り受け規則との適合度に関する閾値を使って、「最も良い係り受け関係」を追求する処理を行なう（弛緩分析）。

弛緩分析（図 2.3 参照）

- (a) 閾値 T を $T = T_s$ (たとえば $T_s = 0.9$) とし、前節で述べた D1~D3 の係り受け処理を行なう。

(この結果、短期記憶の要素が1個になっていれば終了。)

- (b) 短期記憶の内容を、係り受け解析部門の入力側に逆順に積み戻し、閾値を $T = T - d$ (たとえば $d = 0.1$) と下げて D1, D3 (D2 は省く) の係り受け処理を行なう。つまり、係り受けの成立した部分はそのままとし、成立しなかった隣接項間の係り受け関係を、閾値 $T - d$ で再度調べる。ただし、受けの項の最左端の係りとの間も調べる。

(この結果、短期記憶の要素が1個になっていれば終了。)

- (c) (b)の手順で新たな係り受け関係が成立した場合には、同様に短期記憶の内容を入力側に積み戻して、(a)の手順に戻り、係り受け関係を成長させる。

そうでなければ、(b)の手順に戻ってさらに閾値を下げる。(但し、閾値が負になったら解析不能として終了する。)

解析が成功した場合には、短期記憶には1個の要素が残されている。それは依存構造式のリストであり、複数の依存構造式を含んでいれば複数の解釈があったということになる。その場合、それらは、依存構造式の評価値によって順序づけることができる。

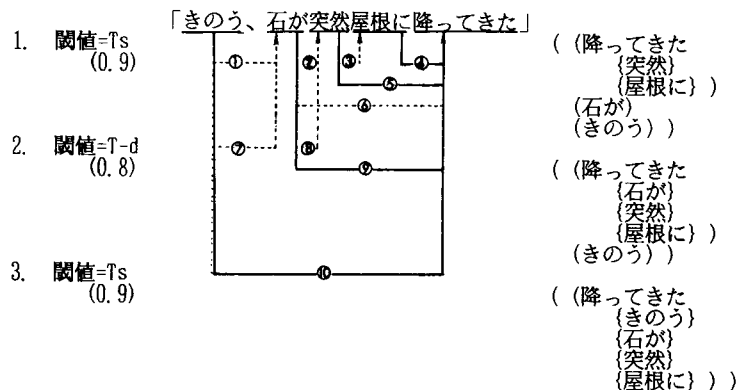


図 2.3 弛緩分析による係り受け関係の検査手順の例

- ④ 補助手段として、係り受けの範囲を指定する方法を提供する。すなわち、前編集として括弧を挿入しておけば、括弧で囲まれた部分を優先して一つの構造にまとめあげる。

係り受け構造の多義性(曖昧さ)には、次の2種がある。

- ① 係りと受けの対は同じであるが(すなわち、解析木の形は同じ)、その意味的な関係が異なる。

例:「麦は日本も輸出する」

「麦」、「日本」の何たるかが分っていなければ(すなわち、その意味特徴が適切に記述されていなければ)、

「麦が日本を輸出する」

「麦を日本が輸出する」

のいずれの意味であるか曖昧である。

- ② 係りと受けの対が異なる(解析木の形が異なる)。

例:イ.「計算機の社会への影響」

(a) (計算機の社会への) 影響

(b) 計算機の(社会への影響)

ロ.「チーズとネズミの尻尾」

(a) (チーズとネズミの) 尻尾

(b) チーズと(ネズミの尻尾)

ハ.「太郎が食べた青森のりんご」

(a) 太郎が(青森)を食べた、その青森のりんご

(b) 太郎が(青森のりんご)を食べた、そのりんご

いずれも、単語の意味が分っていなければ、多様な解釈が可能である。この他、

「太郎が食べたりんごで作ったジャム」

のように、相当程度の知識がなければ、解釈を一意に定めることができない例など、解釈が多義になる例は数多い。

上に述べた、係り受け解析部における多義性への対処法によれば、①の種類が多義性に関しては(ある評価値以上で)可能な解釈を同時に保持して解析を進め、解釈のもっともらしさを評価値で順序づける。②の種類が多義性に関しては、係り受け規則との適合度の閾値を少しずつ上げていくことによって、より高い適合度で係り受けが成立する部分を先に結合する。つまり、その時点で不確かと判断される係り受け関係は保留しておいて、後で見直しを行う。これによって、いったん成立した係りと受けの対を後で破棄すること(後戻り/バックトラック)はしないで、決定論的な解析を行っている。(決定論的な解析方式としては、PARSIFAL¹⁶⁾が知られているが、これは英文に対する句構造の構文解析を行うものである。)ただし、上の例のロは「と」の係りの規則で「チーズ」と「尻尾」の関係を見て判断しており、ハは連体埋め込み構造の処理アルゴリズムに

よって判断している(第5章参照)。イについては、「計算機の」と「社会」の係り受けの適合度、「社会への」と「影響」との係り受けの適合度の高い方が先に結合されることになる。適合度は係り受け規則の中に記述されている。この場合には、「影響」の格構造規則{(～への) → 影響}によって、「計算機の(社会への影響)」と解析することができる。しかしたとえば、「計算機の会社への影響」という場合には、この文だけからは、「(計算機の会社への)影響」という読みも「計算機の(会社への影響)」という読みも有り得、「会社」が「計算機の会社」を指すのか、別の今話題にしている会社を指すのかは判断できない。これは、文脈処理をも統合した解析システムによって解決しなければならない問題である。現在のシステムでは、係り受け規則との適合度が同じ程度である場合には、文頭側の係り受け関係が成立することになる。

弛緩分析法によれば、諸規則に完全には合致しない場合でも「もっともらしい解釈」をひきだすことができる。たとえば、

「昨日 彼 石 食べた」

のように助詞条件、意味条件が合わない文でも、

「昨日、彼は石を食べた」

と解釈することができる。また、たとえば、

「計算機の心」

において、「心」の受けの規則でも、「計算機」の係りの規則でも、この係り受け関係を認定できないとする。その場合でも、「の」の係りの規則として、任意の体言に(低い評価値で)係るという規則が記述してあれば、弛緩分析法によって、この係り受け関係を解析することができる。

弛緩分析で閾値を下げる回数、機械にとっての理解のしづらさの度合であると解釈することができる。

2.4 まとめ

この章では、構文・意味解析システムの全体構成と解析手法の概要について述べた。解析システムは、文節構造解析部と格構造処理部、および係り受け解析部からなる。係り受け解析は、構文関係と意味関係を同時に扱う係り受け規則を用いて、shift-reduce-parsingを基本的な構図として行う。これは、きわめて単純な図式であるが、膠着語であり意味主導型である日本語文の構造に即した解析手法となっている。不確かな係り受け関係の扱いについては、通常とられているバックトラックによる方法でなく、係り受け関係の良さについての閾値に関する弛緩法によって対処した。これによって、意味上の観点も含めた確かな規則をまず適用し、それで解析が完了しなければ、統語上の観点のみからの粗い規則を適用してみるというように、係り受け規則の記述に柔軟性を持たせることができ、不確かな係り受け関係も扱うことができるようになった。

また、文法規則はすべて、文節カテゴリおよび語の属性として辞書中に記述する方法をとった。

書き換え規則のような規則表示の方法をとる場合には、例外処理が複雑になってしまうことが問題点として指摘されているが、このような語彙規則主導型の方法をとることによって、個別的な事項の多い言語規則を直感的に素直に記述することができる。

日本語文の解析システムでは、従来から格構造がよく用いられてきたが、それは構文解析の後に、動詞中心の意味解析を行うものである。本システムでは、格構造を動詞だけでなく任意の自律語が持ち得る属性として一般化し、係り受け規則の一種として位置づけることによって、統一的な係り受け解析方式を確立した。

第3章 文節構造の解析

3.1 はじめに

本章では、文節構造の解析手法について述べる。

日本語の単語は、自律語と付属語に分けることができる。文節は自律語にいくつかの付属語が後続した単位である。自律語から付属語、付属語から付属語への接続には明確な制約がある。ある付属語が直前の語に接続するかどうかは、直前の語の種類およびその活用形で決まる。この接続関係については、既に多くの研究がある^{69),74),95),96)}。仮名またはローマ字のべた書き文の場合には、多くの曖昧さが生じるために、文全体を調べて整合のとれる単語の列を見出さなければならない。また、

「かれがくるまではしらなかった」→「彼が来るまで走らなかった」

「彼が来るまでは知らなかった」

のように、構文や意味あるいは文脈まで考慮しないと、完全には解決しない場合も多い。しかし、通常の漢字かな混じりの表記の場合には、ほとんどの場合、文頭から逐次に文節を切り出していくことができる。本システムでは、漢字かな混じりの表記を前提として、文節最長一致法によって文節を切り出している。文節最長一致法は、文頭から始めて、できるだけ長く文節を切り出していく方法である。本システムでは、このようにして切り出した文節、いわば表層の文節構造を、係り受け解析に適合した文節構造、いわば深層の文節構造に変換する整形処理を導入した。

文節構造解析部における処理の特徴をまとめると、次のとおりである。

- (I) すべての自律語は活用するとみなして処理を統一した。品詞の区別は活用形の区別として表現できる。
 - ・文節の構文上の種類を文節カテゴリとして定めた。文節カテゴリは、係り受け規則を属性としてもち、係り受け解析部でそれが起動される。
- (II) 入力文から文節を切り出した後、さらに、格構造処理部および係り受け解析部での処理に適合する形に整形する処理を行っている(整形処理部)。

文節は(自律語+付属語の列)であるが、自律語と付属語は、内容語と機能語という観点からとらえなおすことができる。自律語はすべて内容語であり、大部分の付属語は機能語であるが、付属語の中には、「猿は動物だ」における「だ」のように内容語ととらえるべきものがあり、また、「太郎のは」における「の」のように後続すべき内容語をも代行しているのとらえるべきものもある。

整形処理によって、自律語+(付属語)という図式(表層の文節構造)から、内容語+(機能語)という図式(深層の文節構造)に移行する。

- 入力文に現れる語とは別に、格構造処理および係り受けの処理を効果的におこなうために、活用規則あるいは整形規則において擬語を導入できる。

たとえば、「太郎のは」において「太郎の*モノは」のように隠されている体言*モノを導入したり、また、「読みは」において「読み*RY>Tは」のように用言を名詞化する機能を表現する機能語*RY>Tを導入するなど。

このように、擬語の導入、整形処理の導入によって、格構造処理部、係り受け解析部の処理方式の一貫性を図ったことが、ここでの大きな特徴となっている。

3.2 前処理

入力文の文字列は、前処理部門で、数値の列あるいは読点や空白、さらに括弧や引用符で区切られた部分毎のかたまりに分割して入力スタックに積む。入力スタックの要素(入力式と呼ぶ)は、かたまりの種類を示す指示子とかたまりの対であり、次のようなものがある。

- (*NUM* 数値列)
- (*WORD* 文字列)
- (*QSTC* 入力式の列) ... 引用符「と」で囲まれた部分
- (*PSTC* 入力式の列) ... 2.2.2で述べた係り受け解析の範囲を指定するための補助手段である括弧(と)で囲まれた部分

図3.1に簡単な例を示す。

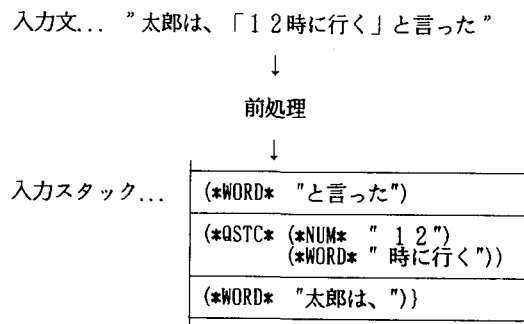


図3.1 前処理の例

入力スタックには、上述のような要素の他に、後の処理によって積み戻されてくるものもあり、入力スタックから取り出され以下のように処理される。

- (*WORD* XXX) ... XXX は文字列であり、文節構造抽出部に引き渡される。
- (*DWORD* XXX) ... XXX は先頭の文節が切り出された残りの文字列であり、*WORD* の場合と同様に文節構造抽出部に引き渡される。
- (*PP* XXX) ... XXX は文節構造式であり、文節構造整形部に引き渡される。
- (*PP.X* XXX) ... XXX は整形処理済みの文節構造式であり、格構造処理部に引き渡される。
- (*CHFL* XXX) ... XXX は格構造式であり、依存構造解析部に引き渡される。
- (*NUM* XXX) ... XXX は数値列である。
- (*QSTC* XXX) ... XXX は引用符で囲まれた部分に対する入力式の列である。
- (*PSTC* XXX) ... XXX は、括弧を用いた前編集によって、係り受けの範囲を指定された部分に対する入力式の列である。この入力文を対象として解析を完了し、結果を依存構造解析部門に引き渡す。

3.3 文節構造

文節は、自律語とそれに後接するいくつかの付属語からなる。文節構造式は、これを次のように表現したものである。

((文節カテゴリ 付属語 n ... 付属語 1) 自立語 1 ... 自立語 m)

ここで、文節カテゴリは、この文節の統語的なカテゴリを示すもので、**図 3.2** の 14 種を設けてい

| | 文節カテゴリ | 例 |
|----|-----------|--------------|
| 1 | \$T>Y | 連用型体言文節 太郎が |
| 2 | \$T>T | 連体型体言文節 太郎の |
| 3 | \$T> | 体言文節 太郎、 |
| 4 | \$RENYOU | 連用型用言文節 歩き、 |
| 5 | \$RENTAI | 連体型用言文節 歩く |
| 6 | \$KATEI | 仮定型用言文節 歩けば |
| 7 | \$SYUSI | 終止型用言文節 歩く |
| 8 | \$MEIREI | 命令型用言文節 歩け |
| 9 | \$TOIKAKE | 質問型用言文節 歩くか |
| 10 | \$Y>Y | 接続型用言文節 歩くので |
| 11 | \$F>T | 連体型副言文節 その |
| 12 | \$F>Y | 連用型副言文節 ゆっくり |
| 13 | \$F> | 副言文節 もっと |
| 14 | \$S>Y | 接言文節 しかし |

図 3.2 文節カテゴリの種類

る。付属語 n は最後尾の付属語，付属語 1 は自立語の直後の付属語である。この付属語 i は，内部コードであって，たとえば格助詞の「が」と接続助詞の「が」は別のものとして区別されている。自立語 $1 \sim$ 自立語 m は自律語に対する同形異語である。

たとえば，「行ったが」という文節の文節構造式は，

$$(\$Y > Y \text{ が} Y \text{ タ}) \text{イク オコナウ}$$

となる。が Y は接続助詞の「が」である。

3.4 文節構造の抽出

3.4.1 文節構造の抽出手順

文節構造の抽出手順を，図 3.3 に示す。

文字列 W に対する文節構造抽出部の処理手順は，次の通りである。

① W の長さを N とする。

② $N < 0$ になっていれば，照合する自律語がなかったか，正しく接続する付属語列が見出せなかった場合である。この場合には， W に対して未登録語処理を行い，得られた文節構造式を文節構造整形部にひきわたして終了。(未登録語処理に関しては後述する。)

③ W の $0 \sim N$ の部分文字列と一致する自立語語幹があるかどうか調べる。

そのような自立語語幹があれば，後続部分に対して接続処理を行い文節構造式を作りあげる。(接続処理に関しては後述する。)

但し， $N = 0$ の場合には，*@ を自律語語幹とみなしてこの処理を行う。*@ は，「する」や「くる」など，語幹を持たない自立語を処理するために導入した擬語幹である。

そのような自立語語幹がないか，あっても後続部分に対する接続処理が成功しなければ，

$$N := N - 1$$

として②に戻る。

自律語の語幹を切り出す

--> 活用語尾の処理

| | |
|-----|-------------|
| --> | 付属語の語幹を切り出す |
| | --> 活用語尾の処理 |
| | --> 接続関係の確認 |
| | {この処理を繰り返す} |

--> 文節カテゴリの決定

図 3.3 文節構造の抽出手順

- ④ 得られた文節構造式を文節構造整形部に引き渡し、さらにその結果を格構造処理部に引き渡す。そこまでの処理が成功すれば、格構造処理部の結果である格構造式（基本依存構造式）を係り受け解析部に引き渡して終了。文節構造整形部、または格構造処理部で不都合があつて成功しなければ、

$$N := N - 1$$

として②に戻る。

3.4.2 接続処理

接続処理は、

- ① 残りの文字列を調べて、切り出されている自律語の語幹に活用語尾が正しく接続しているかどうかを確認する、
- ② さらに残りの文字列から、接続条件を確認しながら後接する付属語の列を切り出す、
- ③ 付属語列を切り出し終えたら、最後の付属語の活用形から文節カテゴリを決定する、という処理を行う。

接続処理に関連する規則は、**活用規則**と**接続規則**、および**文節カテゴリ規則**である。

活用規則は、本システムでは、動詞や形容詞、助動詞など、通常、活用語といわれているものだけでなく、名詞や助詞なども含め、すべての語（自律語および付属語）が必ず備えている属性として一般化した。活用規則は、通例、次のようなりストである。

（活用語尾 1 活用形 1）（活用語尾 2 活用形 2）...

名詞などの不活用語の場合は、活用語尾を NIL としておけばよい。その場合、通常の品詞あるいはその細分に相当するレベルの語のカテゴリを、活用形として表現する。図 3.4 に、活用規則の例を示す。

活用規則には、活用形の後にいくつかの付属語を列記しておくこともある。たとえば、図 3.4 の「ま行 5 段活用」の中の、ウ、***RY>T**、***D>K** は付属語（の内部コード）である。

たとえば、

（み -T *RY>T）

は、「この漢字の読みは ...」のように連用形が名詞化する場合を処理するためのもので、

***RY>T** は、用言連用形を名詞化する機能を表現するために導入した擬機能語である。「読み」の文節構造は次のようになる。

（連用型体言文節 は *RY>T）読む

このように、活用形の後に記されている付属語は、文節構造式の中に取り込む。***RY>T** は、用言としての格構造を名詞としての格構造に変形するプログラム（格構造変形規則）を属性としてもっており、格構造処理部でその働きが起動される。同様に ***D>K** は、「読める」のように可能動詞化する場合を扱うために導入した擬機能語である。ウ の場合は、単に簡単のために、「読もう」

ま行5段活用（たとえば「読む」）

((ま -M, A)
 (み -RY)
 (む -SR)
 (め -M)
 (もう -S ウ)
 (めば -KT)
 (ん -RY, D)
 (み -T *RY>T)
 (め -MR, K *D>K)
 (める -SR *D>K)
 (めれば -KT *D>K)

体言一般

((N I L -T))

連用格助詞（「が」、「を」、「に」など）

((N I L -JTYO))

図 3.4 活用規則の例

の活用語尾を「う」まで含めて一括して扱ったために、助動詞「う」をここに残したものである。

このように、活用規則の中で適当に付属語を導入できるようにすることで、活用規則を記述する際の柔軟性を増すとともに、格構造処理を含め、係り受け解析手法の一貫性を図ることができる。

接続規則はすべての付属語が備えている属性であり、その付属語がどのような語の後にくることができるかという接続条件を記述するものである。接続条件は、前接する語はどのような活用形をしていなければならないかという制約として表現する。すなわち、接続規則は、その付属語が接続し得る直前の語の活用形のリストである。図 3.5 に接続規則の例を示す。

自律語から付属語、付属語から付属語への接続条件については、詳しい研究があるが、ここでは以上のように、前の語の活用形に関する制約だけを考慮した。ただし、ここでいう活用形は、前述したように、助詞の種類等をも表現できる一般的なものである。もちろん、この制約条件だけでは文法規則として万全ではなく、正しくない文が入力されたとき、その誤りを指摘することはできない。たとえば、図 3.5 の「させる」の接続規則では、

「食べられさせる」

という誤った文でも、そのままに正しいものとして解析してしまう。しかし、正常な文を正しく解

助動詞「させる」... (-MR -M, KO)

; 1段活用の未然形、カ変活用の未然形など

連体格助詞「の」... (-T -JT02)

; 体言、「だけ」、「など」、「ばかり」など

図 3.5 接続規則の例

析するための接続関係の確認としてはこれで十分であり、たとえば「行くそうだ」と「行きそうだ」における「そうだ」の区別、「太郎の」と「太郎のは」における「の」の区別などもこの接続規則で処理することができる。

文節カテゴリ規則は、切り出された文節の文節カテゴリを決定する規則であり、活用形の属性として記述してある。すなわち文節が、

((活用形 n 付属語 n ... 付属語 1) 自律語 1 ... 自律語 m)

と切り出されたとき、この文節の文節カテゴリは、活用形 n から決定する。活用形 n は、最後尾の付属語である付属語 n の活用形である。

図 3.4 の活用規則に現れる活用形についての文節カテゴリ規則を、図 3.6 に示す。

文節カテゴリ規則が NIL ということは、そのような活用形で終る文節は有り得ないということである。図 3.6 では、未然形で終る文節、音便形のままで終る文節は有り得ないことが表現されている。また、文節カテゴリ規則はリスプシステムで評価 (EVAL) された結果がカテゴリ名を与えるようになっており、図 3.6 の -SR の場合のように、手続きを書いておくこともできる。

文節構造抽出部では、以上に述べた活用規則と接続規則、および文節カテゴリ規則を用いて、次のようにして接続処理を行い、文節構造を抽出する。

- ① 切り出されている自律語語幹に対応する自律語は一般には複数ある。その各々に対して活用処理、付属語処理を行い文節を切り出す。
- ② 付属語処理は、自律語を切り出す場合と全く同様にして、付属語語幹を切り出し活用処理を行うさらに、接続規則により接続関係が正しいものであるかどうか確認する。そしてさらに、後続部文に対して付属語処理を繰り返す。
- ③ このようにして付属語の列を切り出し終えたら、文節カテゴリ規則によってそのような文節が有り得ることを確認して文節カテゴリを決定し、文節構造式を出力する。
- ④ このようにして得られる文節構造式は一般には複数個あるが、その中で入力文字列から最も長い部分を切り出すものを正しい文節構造と見なす。

このような意味で正しい文節構造は、通常一意に定まると前提しているが、一意的に定まらな

| | |
|--|---|
| ま行5段活用 (たとえば「読む」) ((ま -M, A) (み -RY) (む -SR) (め -M) (もう -S ウ) (めば -KT) (ん -RY, D) (み -T *RY>T) (め -MR, K *D>K) (める -SR *D>K) (めれば -KT *D>K) | 文節カテゴリ規則 -M, A => NIL -RY => 連用形用言文節 -SR => 終止形用言文節 ... 文が終っているとき 連体形用言文節 ... 文が続いているとき -M => 命令形用言文節 -S => 終止形用言文節 -KT => 仮定形用言文節 -RY, D => NIL -MR, K => 連用形用言文節 -T => 体言文節 -JTY0 => 連用性体言文節 |
| 体言一般 ((N I L -T)) | |
| 連用格助詞 (「が」、「を」、「に」など) ((N I L -JTY0)) | |

図 3.6 図 3.4 の活用規則 (再掲) 中の活用形についての文節カテゴリ規則

「太郎は行きたくなかったようだ」
 ((\$T>Y ハ) タロウ)
 ((\$SYUSI ヨウダ タ ナイ タイ) イク)

「太郎は歩き、二郎は走った」
 ((\$T>Y ハ) タロウ)
 (*PPL* ((\$RENYOU *COMMA) アルク)
 ((\$T> *COMMA *RY, T) アルク))
 ((\$T>Y ハ) ジロウ)
 ((\$SYUSI タ) ハシル)

図 3.7 文節構造式の抽出例

かった場合には、文節構造解析部での以後の処理を保留して、

(*PPL* 文節構造式 1 ... 文節構造式 n)

を係り受け解析部に送りだす。係り受け解析部では、妥当な文節構造式を選択した後、あらためて文節構造整形部に送り出す。

3.4.3 未登録語の処理⁵³⁾

文字列から辞書に符合する自立語が切り出せなかった場合には、未登録語があったものとして、次のような仮定のもとに、未登録語を含む文節構造を抽出する。

- ① 未登録語は、体言／サ変動詞／形容動詞の型の活用をする自律語である。
- ② 未登録語の語幹となる部分は、最短のものが妥当である。
- ③ 未登録語には（未登録語が文末にまで至る場合を除いて）1個以上の付属語が後続していなければならない。

（後続する付属語部分は、通常の場合と同様に最長のものを妥当なものとする。）

未登録語処理のために、未登録語を代表する自律語として、*UNDW を導入し辞書に登録しておく。*UNDW の活用規則は、体言型の活用とサ変動詞型の活用、および形容動詞型の活用を合せ持つものとする。具体的な処理手順は次のようである（図 3.8）。

- ① 文字列の最初の 1 文字を未登録語とみなし *UNDW に置きかえる。
- ② 文節構造抽出処理を行う。

残りの文字列がないか、または残りの文字列があって且つ 1 個以上の付属語を含む文節構造が抽出できていたら終了。

- ③ 文字列の次の 1 文字までを未登録語とみなして *UNDW に置きかえ、②に戻る。

このような処理では、一般の活用語や付属語が未登録である場合に対しては正しく文節を切り出すことはできないが、通常、未登録語として対処する必要がある対象は体言であり、この方式で多くの場合対処できる。

「鹿児島からは遠い」において鹿児島が未登録であった場合；

鹿+? ... 付属語が続かない
 鹿児+? ... 付属語が続かない
鹿児島+から+は ... 成功

「穴に押し込まれた」において「押し込む」が未登録であった場合；

押し込まれた 全体が未登録体言として抽出されてしまい、最終的に解析は不成功となる。

図 3.8 未登録語の処理例

3.5 文節構造の整形

自立語は、通常、何らかの観念を指し示す内容語であり、付属語は文を構成するための機能を担う機能語である。(接続詞、指示詞などは高次の機能語というべきであるが、ここでは便宜上一括して内容語とよぶ。)

ところで付属語の中には、内容語の役割を併せもつとみなすべきものがあり、構文・意味解析の立場からは、適当な内容語で置き換えるか、あるいは適当な観念語を前後に補って処理を進めると都合がよいものがある。たとえば、

- ・「これは本だ」の「だ」は、付属語であって、名詞文を構成する助動詞であるが、構文・意味解析の立場からは、「これ」と「本」を係りとして文を構成する用言とみなしたほうが都合がよい。
- ・「これは本らしい」や「これは本か」の、「らしい」や「か」の直前には、この用言「だ」が省略されているとみなすべきである。
- ・「彼のはつまらない」の「の」の直前には、体言「モノ」が省略されているとみなすべきである。
- ・「行くのはつまらない」の「の」は体言とみなすべきである。

このように、入力文の文字列を語彙解析することによって得られた文節構造(いわば、表層の文節構造)を、付属語の用法・機能の見地から再構成(いわば、深層の文節構造)するために、文節構造整形部を置いた。文節構造抽出部での**自立語+付属語**の図式(表層の文節構造)は、文節構造整形処理によって、**内容語+機能語**の図式(深層の文節構造)に移行する。

文節構造を再構成するための規則(整形規則)は付属語の属性であり、手続き型の規則(リスプのプログラム)である。文節構造整形部では、文節構造式中の付属語を出現順に走査し、整形規則を持っていればそれを評価(EVAL)することによって整形処理が行われる。

整形規則を記述するために、次のような関数が用意してある。

[対象となっている文節構造式を、

((C Z1 ... Z* ... Zn) H1 ... Hm)

とする。現在対象としている付属語をZ*とする。]

- ・(RF-A a1 a2 ... an)

その付属語の後に、付属語の列 a1 ... an を追加する。

((C Z1 ... a1 a2 ... an Z* ... Zn) H1 ... Hm)

- ・(RF-R a1 a2 ... an)

その付属語を、付属語の列 a1 ... an で置き換える。

(RF-R)なら、その付属語を削除することになる。

((C Z1 ... a1 a2 ... an ... Zn) H1 ... Hm)

• (RF-D h c a1 a2 ... an)

その付属語を境に、文節構造式を次のように2つに分割する。

① ((C Z1 ...) h)

② ((c a1 a2 ... an) H1 ... Hm)

①は、入力スタックに積み戻し、②を新たな処理対象とする。

たとえば、

「猿は動物だ」

において、下線の部分は文節構造抽出部で、

((SYUSI ダ) 動物)

と1個の文節構造式に分析されるが、「ダ」の整形規則；

(RF-D *デアル \$T>Y ダ*)

によって、次のような2個の文節構造式に分割される。

① ... ((\$T>Y ダ*) 動物)

② ... ((SYUSI) *デアル)

②は、指示子*PP*をつけて、すなわち、

(*PP* ((SYUSI) *デアル))

として、入力スタックに積み戻す。①は、文節構造解析部の出力として格構造処理部に引き渡す。

ここで、「ダ*」は、「ダ」の痕跡を残すために導入した擬機能語であり、「*デアル」は、

「～は～だ」

という、いわゆる名詞文を処理するために導入した擬用言である。

また、たとえば、

「彼が食べたのはチーズだった」

において下線の部分は、文節構造抽出部で；

((\$T>Y ハ ノ タ) 食べる)

のような1個の文節構造式に分析されるが、付属語「ノ」の整形規則；

(RF-D (*モノ *コト) \$RENTAI)

によって、次のような2個の文節構造式に分割される。

① ... ((\$RENTAI タ) 食べる)

② ... ((\$T>Y ハ) *モノ *コト)

②は、「ダ」の場合と同様に、指示子*PP*をつけて入力スタックに積み戻し、①を文節構造解析部の出力として格構造処理部に引き渡す。

ここで、「*モノ」、「*コト」は、いわゆる準体助詞の「ノ」を処理するために導入した擬体言である。「*モノ」は、同一名詞連体修飾の場合に対応する擬体言であり、「*コト」は、同格名詞連体修飾の場合に対応する擬体言である。

3.6 まとめ

この章では、文節構造の解析手法について述べた。文節の切り出しは、文節最長一致法によって、文頭から逐次に切り出す方式をとった。文節の種類として、14種の文節カテゴリを設けた。これは構文上のカテゴリであり、句構造モデルの句標識に対応する。文節カテゴリは、係り受け規則を属性としてもち、係り受け解析部で評価される。

従来、文節構造の解析は、表層の字面上の文節を切り出すことだけが処理内容とされていたが、ここではさらに、整形処理部を設け、表層の文節構造(自律語+付属語の列)から、深層の文節構造(内容語+機能語の列)へ移行する処理を導入した。これによって、格構造処理部、係り受け解析部を、一貫した方式によって構成することができる。

第4章 格構造の処理

4.1 はじめに

本章では、格構造規則の構成および格構造変形処理の手法について述べる。

格構造処理部は、文節構造解析部から入力される文節構造式に、その内容語の格構造を付加する。さらに、助動詞「られる」や「させる」など、受身化や使役化など、格構造の変形に関する処理を行う。

格構造は、従来から日本語文の解析システムではよく用いられてきた。それは、格構造が動詞を中心とした意味構造を表現することから、構文解析後の意味解析において格構造を利用しようとするものであった。本システムでは、構文解析と意味解析を融合しており、格構造は係り受け規則の一種としてとらえ一般化して、係り受け解析の中で統一的に扱っている。

ここでの特徴をまとめると、次のとおりである。

(I) 格構造を、動詞だけでなくどの内容語でも持つことができる属性として一般化した。

また、格構造規則は係り受け規則の一種として位置づけている。

これにより、「太郎の弟」、「魚が焼けるにおい」、「鼻が長い象」等の体言に関する係り受け関係も統一的に処理することができる。

(II) 深層の格関係でなく表層の格関係に注目し、格構造を内容語の基本的な用法を表現するものととらえた。用言の格構造は、文の分類と対応づけ、文型毎に格役割を設定した。

つまり、格文法のように、格構造を深層の意味構造を表示する枠組みとして考えるのではなく、語の用法を表現し、それと照合することによって文を解析するための枠組みとしてとらえた。

(III) 「られる」、「させる」あるいは「*RY>T(動詞連用形の名詞化に際して導入した擬機能語)」など、ある種の機能語の働きを、文型を変換し、格構造を変形する働きとして規則化した。これは、その機能語の属性として記述してある。

4.2 格構造処理の手順

格構造処理の手順を、図4.1に示す。

文節構造解析部からの入力である文節構造式は、一般には複数の内容語(同形異語)を含んでいる。格構造処理部では、それぞれの内容語について格構造を取り出し、必要な変形処理を行う。各内容語も、一般には複数の格構造を持つから、変形処理は各々の格構造式に対して行われる。変形

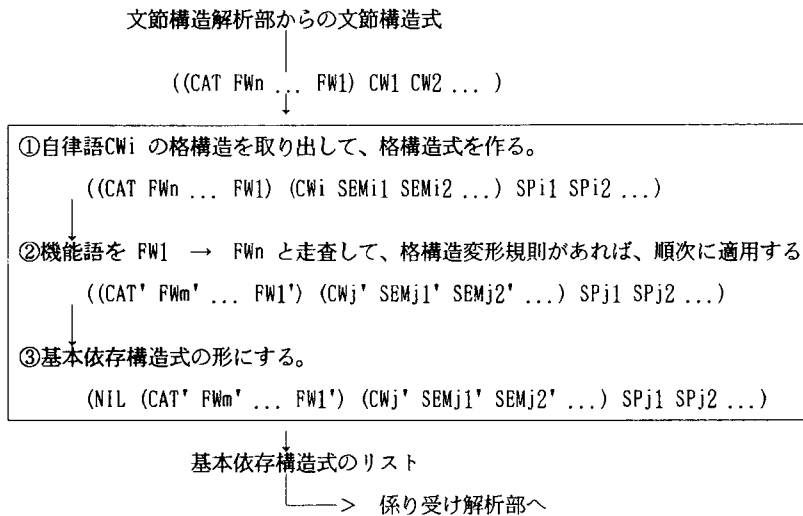


図 4.1 格構造処理の手順

処理もまた、ひとつの格構造式に対して、複数の格構造式を出力する場合もある。このようにして格構造処理部は、1個の文節構造式に対して、一般には複数の格構造式（基本依存構造式）を出力する。

図 4.1 で、基本依存構造式は格構造式の第 1 項として NIL を挿入したものであるが、これは係り受け解析部で係り受けの履歴を記録しておく項であり、初期値として NIL を与えている。

4.3 格構造規則

4.3.1 格構造

格構造 (case frame) は、従来、動詞を中心に、それを取りまく格 (case) としてどのようなものがあるかをとりまとめた枠組みとして扱われてきた。この場合の格は、主格や目的格といった統語上の格ではなく、動詞との意味的關係を表わす深層格 (deep case) を指している。格文法の提唱者である Fillmore³²⁾によれば、たとえば、

- (1) The door opens.
- (2) I open the door with the key.
- (3) The key opens the door.

に現れる door は、統語上の格は異なるが、意味上はいずれも open に対する対象格であり、key も統語上の格は異なるが、意味上はいずれも道具格である。格文法では、同一の深層意味構造から、どのようにして(たとえば、何を主語化するかによって)、様々の表層の文が生成されると

いったことが研究されてきた。

格構造は、特に日本語文の解析システムでは、従来からよく用いられてきた。意味主導型の言語である日本語文の解析では、意味解析が必須であることから、その意味表現の形式として用いられてきたものである。そこでは、格構造を利用して、動詞を中心とした意味解析を行い、構文解析結果の中から意味的に整合するものを選びだして、さらに、一意的な深層の意味構造を得ることを目的としている。この格構造による解析は、名詞とそれを受ける動詞との間の係り受け関係を解析しているとみなすこともできる。しかし、動詞と名詞との係り受け関係以外の扱いなども含めた、全体的な係り受け解析の手法として一般化したものではない。また、“一意的な深層の意味構造”を表現するという点では、適当な深層格の集合を設定する必要があるが、どのような深層格を設定すればよいかについての解答がないことが問題とされてきた⁸⁷⁾。

本システムでは、格構造をもっと表層に近いレベルの構文的および意味的關係を表現する枠組みとしてとらえ、一意的な深層の意味構造表示という立場はとらない。つまり、上記の(1)~(3)をopenに対する同一の格構造で意味表示しようとするのではなく、openには(1)~(3)のような異なった基本的用法があるのであって、このような用法を表現する枠組みとして格構造を考える。“一意的な深層の意味表現”を必要とする場合には、さらに何らかの解釈操作を経て得るべきものとする。

また、格構造は、従来のように動詞だけでなく、任意の内容語が持ち得る属性であって、個々の内容語の基本的な用法を表現する枠組みとして一般化した。さらに、格構造を係り受け規則の一種として位置づけた。すなわち、ここでは格構造は、内容語が通常どのような係りの語句を受けるかという、内容語の用法を記述する係り受け規則の一種として位置づけた。(係り受け規則の全体については次章で述べる。)これによって、係り受け規則による統一的な視点からの係り受け解析システムを構成することができた。

格構造の記述形式を図4.2に、格構造の記述例を図4.3に示す。

図4.2で、ptypeは格構造の種類を表現する。格構造の種類については次節で述べる。siは、

```
((ptype s1 s2 ..) SP1 SP2 ..)
  ptype ... 格構造のタイプを示す.
  s      ... その格構造にまつわる意味特徴.
  SP     ... ある依存構造式が係りとして認定されるための条件を記述した式. (格条件式)
  SP := (rtype EC SC )
  rtype ... 格役割名
  EC     ... 特記条件
  SC     ... 意味条件
```

図4.2 格構造の記述形式

- ① ((<L) (:L 0 !ドウブツ) (.LO 0 !モノ))
 ... 「子供が窓を割る」
- ② ((<L) (:L 0 !ヒト) (.LO 0 !エキタイ) (.LTP 0 !エキタイ))
 ... 「ウイスキーを水で割る」
- ③ ((<L) (.DO 0 !スウチ) (.DTP 0 !スウチ))
 ... 「10を3で割る」

図 4.3 「割る」の3つの用法に対応する格構造の記述例

ptype の細目として、その格構造の特徴を表示しておきたい場合に記述する。

格条件式は、どのような係りの語句をとるのかという、係りの語句に関する条件を記述する。

格役割名は、係りの語句の構文・意味上の位置ないし役割を表示する記号である。また同時に、係りの語句に付随しているべき格助詞等に関する条件 (機能語条件という) を記述する。具体的には格役割名は、助詞または文節カテゴリのリストを値としてもつリスプのアトムであり、この値は高い適合度をもつものから低い適合度をもつものへと3段階に別けてある。たとえば、図 4.3 で :L は、高い適合度 (1.0) の {ガ}, 次に高い適合度 (0.7) の {ハ, モ}, 低い適合度 (0.5) の {コソ, サエ, ...} の3種のリストを値としてもつ。また、.LO は {ヲ}, .LTP は {デ}, .DO は {ヲ}, .DTP は {デ} を高い適合度の機能語条件としてもつ。これらの格役割の種類についても次節で述べる。

意味条件は、係りの語句 (依存構造式) が備えているべき意味特徴について記述する。普通は意味特徴のリスト (S1 S2 ...) であり、

《係りの語句は意味特徴として S1 または S2 または ... をもっていないなければならない》

ということを表現する。一般には、意味条件 SC は、以下に示すような形式で、意味特徴の和、積、否定の形で記述でき、また、意味条件に適合度を付随させることもできる。

SC := (NS1 NS2 ...)

NS := (g S1 S2 ...)

g ... “適合度” を示す実数 (通常, $0.0 \leq g \leq 1.0$)

Si := s | (s1 s2 ...)

si := sem | -sem

但し、SC = ((1.0 s1 s2 ...)) の場合には SC = (s1 s2 ...) と略記してよい。

この意味条件 SC は、係りの語句がもつ意味特徴 L に対して、NS1, NS2, ... の順に条件を検査し、条件が適合すれば適合度として g を与えることを表現している。条件が適合するとは、

《L が s11 及び s12 および ... を含むか、または、

s21 及び s22 および ... を含むか、または、

.....

...を含む。

(但し-sem に対しては、sem を含まないとよみかえる。)》

ということである。つまり、NSの連鎖はORを表し、Sの連鎖はANDを表している。

但し、空の意味条件は、どんな場合でも意味条件は満足され、適合度1.0であることを表す。

たとえば、図4.3の①の格構造を、次のように、より詳細に書くこともできる。

①' ((<L) (:L 0 (1.0 !ヒト) (0.7 ドウブツ) (0.3 モノ))
(.LO 0 (1.0 (!ブツタイ !アツミ)) (0.7 !ブツタイ) (0.3 !モノ)))

なお、格構造規則全体としての適合度は、

(機能語条件に対する適合度)*0.6+(意味条件に対する適合度)*0.4

として計算している。

特記条件としては、その格の省略のされにくさ、連体修飾のされにくさなどを必要に応じて記述することができる。特になければ空(0)とする。特記条件を記述するための関数を下に示す。

(A a1 a2 ...)

機能語条件としてa1, a2, ...を追加する。つまり、この格を支える機能語として、a1, a2, ...を加える。

(P n)

この格が省略されたときのペナルティ値をnとする。このペナルティ値によって、この格が欠いた係り受け関係は、評価値がnだけ下げられる。

(既定のペナルティ値は0.05としてある。)

たとえば、図4.3の②の格構造の例で、「水で」を省略するのは極めて不自然であり、

「ウイスキーを割る」

は

「ウイスキー (の入れたびんを) を割る」

のように、①の用法として解釈するほうが自然であろう。このように、省略されると不自然となる格の特記条件として(P 0.9)のようにペナルティを課しておくことができる。

(RP n)

この格を連体埋め込み文で修飾するときのペナルティ値をnとする。このペナルティ値によって、この格を連体修飾した係り受け構造は、評価値がnだけ下げられる。

たとえば、

「AがBからCへ行く」

という「行く」の用法において、

「AがCへ行ったB」

のように「Bから」の格を連体修飾するのは極めて不自然である。

「Aが行ったX」

の X は、B ではなく、C と解釈するべきである。このように B は連体修飾され難い格であるが、このようなとき、B に関する特記条件で (RP 1.0) のようにペナルティを課しておくことができる。(既定のペナルティ値は 0.0 である。)

4.3.2 文の種類と格構造

4.3.2.1 用言の格構造⁵⁶⁾

格構造は、前節で述べたように、一般には深層格に関して表現するものであるが、どのような深層格をいくつ設定するのかという点については確立した解答はない。格文法家の数だけ、あるいは格を用いた自然言語システムの数だけ格の体系があるといわれるほど、格の概念はさまざまである^{5),87),106)}。これは、格構造によって“深層の意味”を表現しようとするわけであるが、“深層の意味”のとらえかたが依然として困難であることを反映している。普通よくとりあげられる深層格には次のようなものがある。

- (1) 動作主格 ... 動作を行う主体
- (2) 目標格 ... 動作の目標
- (3) 対象格 ... 動作の対象
- (4) 起点格 ... 動作の原動力、感情の源など
- (5) 原因格 ... 動作や状態の原因
- (6) 道具格 ... 動作のための道具など
- (7) 場所格 ... 動作や現象が起った場所
- (8) 時間格 ... 動作や現象が起った時間

本システムでは、前節で述べたように、このような深層格ではなく表層の言語表現に近い格関係を扱い、格構造は内容語の基本的な用法を表現するものとしてとらえている。したがって、用言の格構造の種類および格役割の種類は、文の種類、すなわち文型と対応することになる。ここでは、意味上および構文上の観点から、文(単文)を表 4.1 に示すように分類し(これが格構造の種類ということになる)、それぞれに対して表 4.2 に示すような格役割を設定した。

用言の格構造規則を記述する際には、記述しようとする用法がどの文型に属するものであるかを判断し、その文型に応じた格役割を用いて記述すればよい。全体の格役割の数は多いが、各文型毎にみれば少数である。

表 4.1 に示した文型は、意味内容と表現形式との関連を考慮した分類となっており、以下に述べるような特徴が観察される。

- F 文の述語は形容詞である。A 文、Z 文の述語の殆どは形容詞か形容動詞である。
- F 文、A 文は『～ハ ～ガ』の構文を、Z 文は『～ニハ ～ガ』の構文をとり、いずれも『ヲ』格を含まない。
- P 文は現象の原因とみなされるものが無意志のものである場合『～ガ ～デ』の構文をとり、有

表 4.1 文の種類

| | | |
|---|-----------|---------------------------------------|
| 1. 事実の知覚に関して述べる文 | | |
| ① 行為に関して述べる文 | | |
| イ. 精神上の行為に関して述べる文 | ... (W 文) | 私は彼を憎む |
| ロ. 観念上の行為に関して述べる文 | ... (D 文) | 点の集まりを線と呼ぶ 2を3でわる |
| ハ. 往来に関して述べる文 | ... (T 文) | 首相が全国を遊説する |
| ニ. 相手を伴う共同の行為に関して述べる文 | ... (C 文) | 不動産業者に土地を売る |
| ホ. 動作・労働などイ～ニ以外の一般の行為 に関して述べる文 | ... (L 文) | 窓を壊す, 歌を歌う 文書を翻訳する |
| ② 現象に関して述べる文 | ... (P 文) | 台風で雨が降る 病気で死ぬ |
| ③ 状態に関して述べる文 | | |
| イ. 感情・感覚などに関して述べる文 | ... (F 文) | 私は本を／本が読みたい |
| ロ. ものの属性に関して述べる文 | ... (A 文) | 電子は質量が小さい 蠣は広島が本場だ 彼は英語が出来る／得意だ |
| ハ. 可能性・必要性などに関して述べる文 | ... (Z 文) | 植物には水が必要だ／要る 大学が卒業出来ない |
| ニ. 存在に関して述べる文 | ... (E 文) | ブラックホールがある |
| ホ. 存在の状態を述べる文 | ... (Y 文) | 山がそびえている 水が流れている 扉が開けてある |
| 2. 対象間の関係などの認識に関して述べる文 ... (R 文) | | |
| 分子は原子を含む 猿は動物だ | | |
| 3. 事実の評価・解釈などについて述べる文 | | |
| イ. 事実の影響を被るものについて述べる文 | ... (S 文) | 彼は雨に降られた |
| ロ. 事実の原因・発動者について述べる文 | ... (M 文) | 台風が雨を降らせる |
| ハ. 行為を受け取るものについて述べる文 | ... (V 文) | 子が親に本を読んでもらう |
| ニ. 行為を受けるものについて述べる文 | ... (B 文) | 親が子に本を読んであげる |

表 4.2 用言の格役割の種類

| 文 型 | 格役割 | 主な機能語条件 | 例 | |
|-----|------|---------------|-------------|---------------------------------|
| W 文 | :W | 精神上の行為の主体 | ハ | <u>私は</u> 考える |
| | .WQ | 行為の内容 (引用) | ト | 彼は <u>・・・</u> と考えた |
| | .WP | 行為の内容 (用言性体言) | ヲ | <u>成功を</u> 期待する |
| | .WO | 行為の向けられる対象 | ヲ | <u>彼を</u> 憎む |
| | .WN | 行為の向けられる対象 | ニ | <u>彼に</u> 期待する |
| D 文 | :D | 観念上の行為をする主体 | ハ | <u>我々は</u> プログラミングを...と定義する |
| | .DO | 行為の対象 | ヲ | <u>点を線と</u> みなす |
| | .DO | 行為の目標 (引用) | ト | 点を <u>線と</u> 思う |
| | .DT | 行為の相手 | ニ | <u>2に</u> 3を加える |
| | .DW | 行為の相手 | ト | <u>2と</u> 3を加える |
| | .DF | 行為の相手 | カラ | <u>2から</u> 3をひく |
| | .DTP | 行為の手段・道具 | デ | <u>2で</u> 3をわる |
| T 文 | :T | 往来する主体 | ガ | <u>首相が</u> アメリカに行く |
| | .TP | 従来する場所 | ヲ | <u>野原を</u> 歩く |
| | | | | <u>大学を</u> 卒業する |
| | .TF | 出発点 | カラ, ヲ | <u>日本を</u> 出発する |
| | .TT | 到着点 | ニ, ヘ | <u>日本に</u> 到着する |
| C 文 | :C | やりとりする主体 | ガ | <u>議員が</u> 演説する |
| | .CO | やりとりする対象 | ヲ | <u>資料を</u> 渡す, <u>話を</u> 聞く |
| | .CA | やりとりする対象 | ニツイテ, ニカンシテ | <u>天候について</u> 話す |
| | .CT | やりとりする相手 | ニ | <u>彼に</u> 話す |
| | .CW | やりとりする相手 | ト | <u>彼と</u> 話す <u>彼女と</u> 結婚する |
| | .CF | やりとりする相手 | カラ, ニ | <u>彼から</u> 貰う |
| | .CQ | やりとりする内容 | ト, ヨウニ | <u>行くように</u> 話す |
| L 文 | :L | 行為する主体 | ガ | <u>犬が</u> 吠える |
| | .LO | 行為の対象 | ヲ | <u>窓を</u> 壊す <u>歌を</u> 歌う |
| | .LT | 行為の目標 | ニ, トシテ | <u>絵に</u> 描く |
| | .LCT | 行為の相手 | ニ | <u>台風</u> に備える |

| | | | | |
|-----|------|------------|--------|--|
| | .LCW | 行為の相手 | ト | 水と油を混ぜる |
| | .LCF | 行為の相手 | カラ | 攻撃から守る |
| | .LSF | 対象の始状態 | カラ | 日本語から翻訳する |
| | .LST | 対象の終状態 | ニ、ヘ | 日本語に翻訳する |
| | .LM | 目標物の材料 | カラ、デ | 米から作る |
| | .LTP | 行為の手段・道具 | デ、ニヨッテ | ノコギリで切る 勾配法で求める |
| P 文 | .P | 現象の対象 | ガ | 水が流れる |
| | .PCP | 現象の原因 | デ | 台風で雨が降る 病気で死ぬ 多数決で決まる |
| | .PCH | 現象の原因 | ニ、ニヨッテ | 警察につかまる／つかまえられる |
| | .PF | 始状態 | カラ | 液体から気体になる |
| | .PT | 終状態 | ニ | 液体から気体になる |
| | .PO | 目標 | ニ | 試験に落ちた |
| F 文 | .F | 感情を述べる主体 | ハ | 私はさみしい |
| | .FO | 感情の源 | ガ | 頭が痛い |
| A 文 | .A | 属性を述べる対象 | ハ | 水は密度が大きい |
| | .AO | 属性を述べる対象 | ガ | 役場は屋根が新しい 彼は妻が病気だ 春はあけぼのが良い 雪は色が白い 彼は資質が優れている 蠣は広島が本場だ ガマは筑波が元祖だ 彼は英会話が出来／得意だ |
| | .AN | 属性 | ニ | この水は飲料に適する 日本は資源に富む |
| | .AC | 比較の対象 | ト | 彼は体格が父親と似ている |
| Z 文 | .Z | 属性を述べる対象 | ハ、ハニ | タバコにはタールが多い 僕には歌えない |
| | .ZR | 属性を述べる為の対象 | ガ | 植物には水が必要だ 僕には歌が聞える |

| | | | | |
|-----|------|------------|------------|--------------------------------|
| E 文 | :E | 存在を述べる対象 | ガ | 人がいる 計算機がある |
| | .EP | 存在する場所 | ニ | 外に人がいる 2階に計算機がある |
| Y 文 | :Y | 存在状態を述べる対象 | ガ | 山がそびえている 犬が歩いている 扉が開けてある |
| R 文 | :R | 関係などを述べる対象 | ハ | 法律は相続権を認める |
| | .RI | 関係の相手 | NIL | 猿は動物だ |
| | .RST | 関係の相手 | ヲ | 分子は原子を含む |
| | .RSB | 関係の相手 | カラ | 分子は原子から成る |
| | .RSN | 関係の相手 | ニ | 相続権は法律に由来する |
| | .RSC | 関係の相手 | ト | 彼は彼女と同じだ |
| S 文 | :S | 影響を被るもの | ハ | 彼は雨に降られた |
| | .S | 事象の主体 | ニ | 彼に知恵を盗まれた |
| | .SO | 事象の対象 | ヲ, ガ | 彼は子供がほめられた |
| M 文 | :M | 使役するもの | ガ | 先生が生徒に話をさせる |
| | .MP | 使役されるもの | ニ, ヲシテ | 計算機に計算させる |
| | .MO | 使役されるもの | ヲ | 台風が雨を降らせる |
| | .MH | 使役されるもの | カラ, ヲシテ, ニ | 担当者から回答をさせる |
| V 文 | :V | 行為を授けるもの | ガ | 先生が生徒に本を買ってあげる |
| | .V | 行為を受けるもの | ニ | 議員に演説してもらう |
| B 文 | :B | 行為を受けるもの | ガ | 生徒が先生に本を買ってもらった |
| | .B | 行為を授けるもの | ニ, カラ | アメリカに防衛してもらう |

意志のものである場合『～ガ ～ニ』の構文をとって『ヲ』をとらない。

いわゆる受身文は P 文に分類される。

- ・『ヲ』格が、L 文、C 文では行為を被って変化・生成・移動されるものを示し、T 文では場所（経過する場所、出発地点）を示す。W 文、D 文、R 文では精神的・観念的行為に関連する不変

化の対象を指す。

- S文, M文, B文, V文, Y文は基底の文としては存在せず, 他の文型からの変形として得られる。

S文はいわゆる被害受身文である。

- F文, D文, W文では, 現在形の場合, 主格は通常1人称であり省略されることが多い。
- F文, W文, A文, Z文, R文, D文では, 主格は既知のもの・主題化されているものを前提としており基底的には『ハ』で示される。『ガ』をとるばあいには対比・強調の意味がこめられる。

他の文では主格は基底的には『ガ』で示されるが, 既知のもの・主題化されている場合には『ハ』で示される。

日本語の動詞を分類した研究としては, 時間的に見た動作・作用の種類による分類, すなわちアスペクトの観点から見た動詞の分類の研究がある(金田一:70)。そこでは日本語動詞を, 次の4種に分類している。

第1種 状態動詞...『テイル』をつけることが出来ない。状態の不変化を表す。

第2種 継続動詞...『テイル』がつくと進行中のことを表す。状態の一時的変化を表す。

第3種 瞬間動詞...『テイル』がつくと結果の状態を表す。状態の永続的变化を表す。

第4種 特殊動詞...『テイル』を常につけて用いる。状態を帯びること, あるいは状態の発端を表す。

表4.1の分類は, この4分類とおおむね次のような関係にある。

- 「状態に関して述べる文」のうち, A文, Z文, E文の動詞は, 『テイル』がつかないか, または常につけて用いられる。つかない動詞が第1種の「状態動詞」, 他は第4種の「特殊動詞」である。
- Y文の動詞は常に『テイル』をつけて用いられ, 「どのような状態をもって存在しているか」を述べる文であり, このうちでY文としてのみ使われる動詞, すなわち『テイル』をつけない形では用いられない動詞が第4種の「特殊動詞」である。
- R文は事態の動的な部分には関心がなく, ある事態にあることを述べる文であって, 『テイル』がついても意味上の変化が見受けられない。従ってR文の動詞は第3種の「瞬間動詞」と分類される。

{なお, (金田一:70)では, 助動詞『タ』はすべての類の動詞につくとされているが, R文の動詞には直接にはつかず, 『テイタ』の形でしかつかない。}

- W文, D文は時間的な始点・終点が言い難い内面的な精神的行為・観念的行為を述べる文であり, 第3種の「瞬間動詞」に分類される。ただし, 『考える』{今, 考えている}のように, 『テイル』がついて進行中の状態を表わす第2種の「継続動詞」とみなすべき場合もある。
- 残りのT文, C文, L文, P文の動詞は, その行為または現象の時間的な性質から, 第2種の

「継続動詞」であるか第3種の「瞬間動詞」であるかのいずれかに分類される。

文型に関するこのような考察は、表現の形式(文型)と表現される内容との間に見られる規則を見出そうとするものである。上に述べた考察は未だ部分的なものであるが、より詳細に、網羅的に考察を深めることは、より適切な格役割や意味特徴の集合を設定することに役立つ、テンス・アスペクトや文脈の解析とも統合した、より高度の解析規則を設定していく上で重要であると考えられる。

4.3.2.2 体言の格構造

体言の中には、「太郎の父」,「壁をたたく音」のように、ある修飾語を伴って意味的に安定するものがある。このように、基本的にある種の係りの語句を伴う体言の修飾関係を、その体言の格構造としてとらえることができる。このような体言への係りとしては、「体言+の」の形で係るものと、連体埋め込み文の形をとるものがある。これらについて、表4.3のように格役割を設定した。

連体修飾構造の分類は(奥津:66)によった。奥津は、連体修飾構造を、同一名詞連体修飾と付加名詞連体修飾の2種に分けている。

同一名詞連体修飾は、

「太郎が食べたリンゴ」... (「太郎がリンゴを食べた。そのリンゴ」)

のように、連体修飾文中に被修飾名詞と同一の名詞をもつものである。

付加名詞連体修飾は、連体修飾文中に被修飾名詞と同一の名詞を持たないものであり、表4.3に掲げた連体修飾の格役割は付加名詞連体修飾によるものである。付加名詞連体修飾は、さらに相対

表 4.3 体言の格構造における格役割

| 格役割 | 機能語条件 | 例 |
|----------------|----------|------------------------------|
| @S あるものの部分 | ノ | <u>彼女の目</u> |
| @R あるものとの関係 | ノ | <u>彼女の弟</u> |
| @A あるものの属性 | ノ | <u>彼女の身長</u> |
| &GA 連用ガ格の連体化 | ノ | <u>太郎の手紙</u> (太郎が書いた手紙) |
| &WO 連用ヲ格の連体化 | ノ | <u>ダムの破壊</u> (ダムを破壊する) |
| &HE 連用ヘ格の連体化 | ヘノ | <u>太郎への手紙</u> (太郎へ出した手紙) |
| &KR 連用カラ格の連体化 | カラノ | <u>太郎からの手紙</u> (太郎から来た手紙) |
| &MD 連用マデ格の連体化 | マデノ | <u>東京までの距離</u> (東京まで行く距離) |
| &TO 連用ト格の連用化 | トノ | <u>太郎との喧嘩</u> (太郎と喧嘩する) |
| =A 相対名詞連体修飾 | \$RENTAI | <u>仕事にでかける前に</u> |
| =E 同格名詞連体修飾 | \$RENTAI | <u>技術が発達すること</u> / ようす |
| =Q 部分的同格名詞連体修飾 | \$RENTAI | <u>壁をたたく音</u> , <u>魚を焼く煙</u> |

名詞連体修飾と同格名詞連体修飾にわけられる。相対名詞連体修飾は、「前」、「後」、「右」などのように相対的な内容を表す名詞に対するものであり、同格名詞連体修飾は、「事」、「現象」、「様子」のように、連体修飾文全体を受けるものである。部分的同格名詞連体修飾は、同格名詞連体修飾と同種のものではあるが、「魚を焼く煙」、「魚が焦げるにおい」のように、連体修飾文の全体を指すのではなく、それから派生するある物事を指すものである。

このように名詞に格構造を持たせることによって、付加名詞連体修飾構造は、名詞の格構造によって解析することができる。同一名詞連体修飾については、用言の格構造によって解析することができる(第5章参照)。

4.3.3 意味特徴

格構造規則をはじめとする言語規則を記述する場合に、品詞レベルの構文的な制約条件だけでなく、意味的な制約条件をも記述しなければならない。このために、語の意味的な分類・体系化を行っておく必要がある。語の意味を表現することは、究極的には、語がどのように記憶されているかという、記憶のありようを表現することであり、極めて困難な課題である。現状では、粗い近似的なものにならざるをえない。

ここでは、語の意味的な特徴を表示するコード(意味特徴/意味マーカ)を、語の属性としてもたせることで語の意味を表現した。意味的制約を意味特徴によって記述することは良く行われている方法であり、様々の意味特徴の体系が提案されている。しかし、確立した意味特徴の体系というものは未だない。広範な語彙について、比較的詳細な意味分類を行ったものに、国立国語研究所の分類語彙表⁷⁹⁾がある。分類語彙表は約3万語を、同義・類義の関係によって約800項目に分類したものである。本システムでは、分類語彙表をも参考にしながら、表4.4に示すように、より簡易化した意味特徴の体系を設定した。

語の意味特徴は、語の属性の一つとして辞書に記述するが、それは意味特徴のリストであって、その語の意味を特徴づける意味特徴のコードを任意個書き並べてよい。表4.1の意味特徴は、簡単な階層関係も表現している。たとえば、!OOF(液体)という意味特徴を付与された語は、自動的に、

{!OOX(物質), !OO(物理的モノ), !O(モノ), !(体言一般)}

という意味特徴をも持つものとみなされる。また、語それ自身も、自動的にその語の意味特徴の一つとみなされる。また、表4.4の意味特徴は、意味表現のための一つの基本的な体系を用意したものであるが、これ以外の任意の記号を意味コードとして書き加えてよい。この語の意味特徴のリストは、言語規則(たとえば格構造規則における意味条件)と照合される以上の役割は担わない。

意味表現の方法としては、意味特徴によるものの他に、意味ネットワークによって語の間の階層的な関連や連想的な関連を表現する方法や²⁴⁾、意味素に分解する方法が考えられているが¹³⁾、広範な語に渡って体系化されたものはない。

表 4.4 意味特徴

| | | | |
|---|------|----------|--|
| ! | 体言一般 | | |
| | !IO | モノ | |
| | | !OO | 物理的モノ 物, ... |
| | | !OOX | 物質 (カタチのないモノ) 物質, ... |
| | | !OOF | 液体 液体, 水, 油, ... |
| | | !OOS | 固体 固体, 金属, 石, 塩, ... |
| | | !OOG | 気体 気体, 空気, 蒸気, ... |
| | | !OOO | 物体 (カタチのあるモノ) 物体, 机, 計算機, 瓶, 星, ... |
| | | !OOP | 生産物 |
| | | !OON | 自然物 |
| | !IOA | 人間以外の動物 | 犬, 猿, ... |
| | !IOV | 植物 | 野菜, 桜, ... |
| | !IOH | 人間活動の主体 | |
| | | !IOHH | 人間 人, 老人, 男, ... |
| | | !IOHO | 組織・集団 国会, 学校, ... |
| | | !IOHM | ♂ 男, ... |
| | | !IOHF | ♀ 女, ... |
| | !IOP | コト的モノ・現象 | |
| | | !OPC | 自然現象... 雨, 雪, 光, 音, 匂, ... |
| | | !OPB | 生理・心理現象... 心, 気持, ... |
| | | !OPH | 人間活動またはその所産... 政治, 経済, ... 学問, 理論, 法則, ... 字, 記号, 小説, ... |
| | | !OPA | その他の描象的実体 力, エネルギー, ... |
| | !IOF | モノを指す語 | 実体, 本体, あれ, これ, なに, |
| | !IV | コト | |
| | | !IVF | コトを指す語 事, 事柄, 事項, 現象, |
| | | !< | 用言の体言化 <W~<B (文型) ... |
| | !IA | サマ (様相) | |
| | | !IAF | サマを指す語 様相, 属性, 状況, ... |
| | | !IAFO | モノの... 色, 形, 大きさ, ... |
| | | !IAFV | コトの... 理由, 原因, 目的, ... |
| | !IAV | サマの値 | |
| | | !IAVO | モノの... 赤, 三角形, ... |

| | | | | |
|------|--------|---------|--------|--------------------------|
| | | !AVV | コトの... | |
| !T | トキ | | | |
| | !TF | トキを指す語 | | 時, 季節, 前, 後, いつ, ... |
| | !TP | 周期的トキ | | 春, 夏, 日曜, ... |
| | !TD | 過去 | | 過去, 昔, ... |
| | !TN | 現在 | | 現在, 今, ... |
| | !TA | 未来 | | 未来, 将来, ... |
| | !TC | 場合 | | 場合, 際, ... |
| !P | トコロ | | | |
| | !PF | トコロを指す語 | | 所, 前, 後, あたり, どこ, ... |
| | !PG | 地理的... | | 日本, 東京, 山, 川, ... |
| | !PP | 部位 | | 上面, 側面, ... |
| | !PS | 空間 | | 空間, 宇宙, 空, ... |
| | !PC | 機能的... | | 道路, 屋根, 台所, ... |
| !N | カサ (量) | | | |
| | !NQ | 数量 | | 一つ, たくさん, すこし, どれだけ, ... |
| | !NF | 数を指す語 | | 数, 数量, 分量, ... |
| | !NN | 数値 | | 100, 200, ... |
| | !NU | 数の単位 | | キロ, ミリ, ... |
| | !NP | 単位 | | メートル, グラム, ... |
| | !NC | 割合等 | | 倍, パーセント, ... |
| !MT | 手段・道具 | | | |
| :~ | ~の部分 | | | 目 (:カラダ), ... |
| !UNQ | 唯一 | | | 日本, 池田, ... |
| !DEG | 程度 | | | 昔, 上, ... |
| !SNG | 単数 | | | |
| !PRL | 複数 | | | |
| !WHQ | (疑問詞) | | | いつ, だれ, なに, どの, ... |
| !IND | (指示詞) | | | あれ, それ, あの, その, ... |

最初に述べたように、このような意味の体系化はいずれにしても粗い近似であって、本質的な意味の表現については、今後の長期的な研究課題といわなければならない。

4.4 格構造の変形

機能語の中には、受身文を作る助動詞「レル」のように、文の構造を変化させる働きをもつものがある。このような機能語を表 4.5 に示す。

本システムでは、このような機能語の働きを、もとの文型から別の文型に変換し、格構造を変形する機能としてとらえる。この機能は、格構造を変形するプログラム(格構造変形規則)を機能語の属性として記述しておくことによって、解析システムの上を実現することができる。

格構造変形規則について、機能語「テモラウ」を例にとって説明する。

機能語「テモラウ」には、次のように格構造を変形する機能がある。

「花子が本を買った」

→ 「太郎が花子に本を買ってもらった」

{L 文から B 文への変換

L 文のガ格が B 文のニ格にvari, 新たに B 文のガ格が加わる}

「住民が議員に演説させた」

→ 「議員が住民に演説させてもらった」

{M 文から B 文への変換

M 文のガ格が B 文のニ格にvari, M 文のニ格が B 文のガ格にvaru}

「太郎が花子に資料を渡した」

→ 「二郎が太郎から花子に資料を渡してもらった」

{C 文から B 文への変換

C 文のガ格が B 文のカラ格にvari, 新たに B 文のガ格が加わる}

→ 「花子が太郎に資料を渡してもらった」

{C 文から B 文への変換

C 文のガ格が B 文のニ格にvari, C 文のニ格が B 文のガ格にvaru}

図 4.4 は、「テモラウ」の格構造変形規則であり、以上のような格構造の変形についての言語事実をプログラムしたものである。格構造変形規則の一般的な記述形式は、図 4.5 に示した。

変形規則を記述するための関数として、次のような関数が用意してある。これらに適当な条件文をかぶせて、変形を引き起こす条件を記述することは容易である。

(XA case)

新たな格条件 case を付加する。

(XX (r1 . r1') (r2 . r2') ...)

格役割 ri を ri' に置き変える。

(XD r1 r2 ...)

表 4.5 格構造を変形する機能語

| 新文型 | 旧文型 | 変形を指示する機能語 | 例文 |
|-----|---------------------|-------------------------------|--------------------------------|
| F | W D T C L E S M V B | タイ | タバコがすいたい |
| W | F | ガル | タバコをすいたがる |
| P | L | アガル | 論文がしあがる |
| P | W C L M | レル, ラレル | 土地が売られる 行かされる |
| Z | W D T C L E M V B | デキル, *D>K | 彼が憎めない 大学が卒業できない |
| Y | L | テアル | 窓が開けてある |
| Y | W D T C L E M V B | テイル | 日本に来ている 雨が降っている 雨に降られている |
| R | D R | レル, ラレル | 原子は分子に含まれる 点の集りは線と呼ばれる |
| S | W T C L E M | レル, ラレル | タバコをすわれて困る |
| M | W T C L E | セル, サセル | 彼を日本に行かせた |
| V | T C L E | テアゲル | 窓を開けてあげた |
| B | W T C L E M | テモラウ | 彼は日本に行かせてもらった |
| 体言 | ... | *RY>T, *SH>T, *KD>T, *KY>T | 泳ぎ, 処置 静かさ, 騒がしさ |

<B ((<W <T <C <L <E)

(XA (:B !ヒト))

(XX (:W . .B) (:T . .B) (:C . .B) (:L . .B) (:E . .B))

<M

(XX (.MW . .B) (.MC . .B) (:M . .B))

<B <C

(XX (.CT . .B) (.CW . .B) (:C . .B))

(変形されて出来る新文型名

(現文型の並び リスブ関数の並び)

(現文型の並び リスブ関数の並び)

...

図 4.4 機能語「テモラウ」の格構造変形規則

図 4.5 格構造変形規則の記述形式

格役割が *ri* である格を削除する。格が削除される変形の例としては、

「～が論文を書く」

が、

「論文が書きあがる」

に変形されるように、機能語「シアガル」によって、L文の「ガ格」が削除される例がある。

図4.4の変形規則によって、たとえば、「買う」の格構造、

((<L) (:L 0 !ヒト) (.LO 0 !モノ))

は、

((<B) (:B 0 !ヒト) (.B 0 !ヒト) (.LO 0 !モノ))

と変形される。これによって、たとえば、

「ソ連はアメリカに穀物を買ってもらった」

という文は、

:B=ソ連, :L=.B=アメリカ, .LO=穀物

というように分析されることになる。

なお、表4.5で、*D>K, *RY>T, *SH>T, *KD>T, *KY>Tは擬機能語であって、活用処理の段階で導入されるものである。

*D>Kは、動詞を可能動詞にする機能語である。(行ける、書ける、起きれる、…)

*RY>T, *SH>T, *KD>T, *KY>Tは用言を体言化する機能語(それぞれ、一般の動詞、サ変動詞、形容動詞、形容詞の体言化)である。これらにおいては、格役割名に結合している機能語のなかで『ガ』, 『ハ』, 『ヲ』を『ノ』に変更し、それ以外のものについては『ノ』が後続していることを条件として付加している。

格構造の変形は、一つの文節中で連続して引き起されることがある。たとえば、

「食べさせられた」

においては、①「させる」による使役変形、②「られる」による受身変形の二つの変形が行われる。変形処理部は、機能語の出現順に(つまり、「させる」→「られる」の順に)、機能語を走査して、変形処理を重畳する。

4.5 まとめ

この章では、文節構造を抽出した後、係り受け解析を行うための準備として、①内容語の格構造を取り出し、さらに、②必要な変形処理を行う、格構造処理について述べた。

格構造を、動詞だけでなく任意の内容語が持つ属性として一般化し、内容語の基本的な用法を表現する枠組みとしてとらえる方式を提案した。格構造は、第5章で、係り受け規則の一種として位

置づけられる。これによって、統一的な係り受け解析の方式を確立することができる。

また、格構造の種類、格役割の種類について提案し、文型との関連について考察した。さらに、格構造などの言語規則の記述において、意味的な制約条件を記述する際の基礎となる、意味特徴の体系についても述べた。

受身化や使役化などの助動詞の働きについては、格構造を変形する機能としてとらえた。そして、機能語の属性として格構造変形規則を設けることによって、これを処理する手法について提案した。

第5章 係り受け解析

5.1 はじめに

係り受け解析の基本的な手法・特徴については第2章で述べた。本システムでは、文節カテゴリ、内容語および機能語の属性である5種類の係りの規則と、文節カテゴリおよび内容語の属性である4種類の受けの規則を、係り受け規則として設定している。格構造規則は、内容語の受けの規則の一種として位置づけられている。隣接する文節（依存構造式）間で、そこに現れる文節カテゴリや機能語、あるいは内容語の係り受け規則を評価することによって、その係り受け関係が解析される。このように一連の係り受け規則によって、構文関係と意味関係を同時に処理していく統一的な解析方式は、本研究で提案する独自の方式である。

本章では、これらの係り受け規則の構成と適用手順、係り受け規則の記述法および具体例について述べる。個別的なあるいは一般的な、種々のタイプの係り受けに関する規則が、これらの係り受け規則によって記述でき、意味主導型である日本語文の解析に有効であることを示す。係り受けの方向にいくつかの異なったタイプがある連体埋め込み構造についても、これらの係り受け規則を用いて解析できることを示す。

5.2 係り受け解析の手順

5.2.1 係り受け規則の種類と適用手順

係り受け解析部では、(2.3)で述べたように、格構造処理部からの入力（依存構造式 Df）と、前段の係り受け解析部の出力（すなわち、直前の部分の解析結果で、プッシュダウンスタック上にスタックされている依存構造式 Dm）との間の係り受け関係を、次のようにして解析する（図 5.1）。

- ① Df と Dm との間の係り受け関係の有無を調べる。

{ 弛緩分析で、適合度の閾値を下げて再度調べるという場合には、Df 内の左端の依存構造式との間の係り受け関係も調べる。 }

- ② 設定されている閾値以上の適合度で係り受け関係が認められれば、Dm を Df に統合する。
これを新たな Df として①へ。

そうでなければ、Dm をプッシュダウンスタックに戻す。さらに、Df をプッシュダウンスタックに積んで終了。

ここで係り受け関係の有無を調べる規則が係り受け規則である。係り受け規則、すなわち係り受け関係についての制約条件には種々のタイプのものがあるが、本システムではそれらを、文節カテ

ゴリの属性、内容語の属性または機能語の属性として記述した(表 5.1)。それぞれに、係りの規則と受けの規則がある。係り受け解析部は、これらの係り受け規則を、図 5.1 に示す順序で評価(EVAL)することによって、係り受け関係が成立するかどうかを調べる。

係りの規則は、その文節カテゴリ、内容語または機能語を含む語句が、どのような語句に係っていくかということについての制約条件を記述する。受けの規則は、同じように、どのような語句を受けるとかということについての制約条件を記述する。

格構造規則(\$FRULE)と係り構造規則(\$MRULE)は宣言型の規則であり、その他の規則は手続き型の規則(リスプのプログラム)である。格構造規則と係り構造規則は図 5.1 の適用順序の中に現れていないが、これらは手続き型の規則の中で呼び出されて評価される。つまり、格構造規則

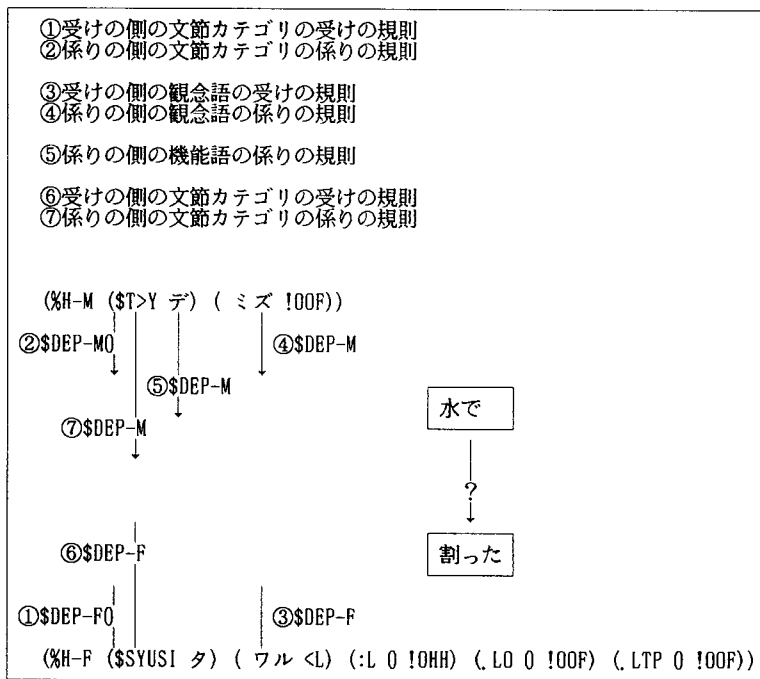


図 5.1 係り受け規則の適用順序

表 5.1 係り受け規則の種類

| | 係りの規則 | 受けの規則 |
|--------|-----------------------------|----------------------------|
| 文節カテゴリ | \$DEP-M0 \$DEP-M | \$DEP-F0 \$DEP-F |
| 内容語 | \$DEP-M \$MRULE (係り構造規則) | \$DEP-F \$FRULE (格構造規則) |
| 機能語 | \$DEP-M | |

と係り構造規則については、それをどのような場合に評価するかを手続き型の規則の中に記述しておく。

係り構造規則は、格構造規則と同様に内容語の属性であって、格構造規則が内容語の受けの条件を記述する規則であったのと対照的に、内容語の係りの条件を記述する規則である。係り構造規則は、格構造の格条件式と同様の形式；

(rtype EC SC)

で記述される。rtype は格役割名とまったく同様に、係りの側に対する機能語条件であり、EC は特記条件である。SC は、格構造規則の場合には、係りの側に対する意味条件であったが、係り構造規則では逆に、受けの側に対する意味条件である。係り受け関係の適合度の計算も格構造の場合とまったく同様である。格構造規則が、受ける側から係りの語句に関する条件を記述する規則であったのに対して、係り構造規則は、係る側から受けの語句に関する条件を記述する規則であり、副詞や連体詞あるいは連体性体言文節などを記述するのに有用である。たとえば、副詞「とても」は、

〈属性を述べる文 (F 文, A 文, Z 文) に、その程度を表現するものとして係る。〉

といったことを、係り構造規則によって記述することができる。

文節カテゴリの係り受け規則は、たとえば連用型体言文節は、

〈受けの側が用言文節ならば、係る可能性があるから、格構造規則および係り構造規則を評価せよ。意味的に解釈できなかつた場合でも、最終的に (すなわち、低い評価値で) 用言文節に係る。〉

といった類のことを記述することができる。

係り受け規則の具体例については (5.4) で述べる。

図 5.5 の ⑤ でいう係りの側の機能語は、たとえば、

| | | |
|----------|---|------|
| 「東京からは」 | → | 「から」 |
| 「東京は」 | → | 「は」 |
| 「行ったので」 | → | 「ので」 |
| 「きのう行った」 | → | なし |

のように、その文節の機能語の中で、もっとも強く結合に関与する機能語を指す。これは、格助詞があれば格助詞、接続助詞があれば接続助詞、そうでなければ最も文末側の副助詞としている。

手続き型の規則は、評価した結果その値が NIL でないならば、係り受け関係が確立したか、または係り受け関係が成立し得ないことが判明したものと解釈する。つまり、値が NIL でなかつた場合には、それ以降の係り受け規則の評価は行わない。文節カテゴリの係り受け規則には、最初に評価されるものと最後に評価されるものとの 2 種類がある。(5.5) の例にみるように、図 5.1 の ①～⑤ の規則は意味的な解釈を含む強い規則、⑥～⑦ の規則は意味的な解釈ができなかつた場合に、統語的な関係だけから判断する弱い規則とすることができる。

〔文節構造が確定していない場合の処理〕

以上に述べたのは、格構造解析部をへて依存構造式が入力される通常の場合の処理手順である。文節構造が一意的に定まっていない場合には、文節構造解析部から文節構造のリストがそのまま係り受け解析部に入力される。このようなケースは稀であると前提しているが、

「太郎は泳ぎ、二郎は歩いた」

のように、「泳ぎ、」に関して連用形か連用形が体言化したものか複数の解釈があり得るというようなことがある。このような場合、文節カテゴリの属性である文節構造選択規則を評価して、文節構造の候補をしばらくこむことを試みる。係りの側の文節カテゴリの文節構造選択規則には、たとえば連用型の文節カテゴリなら用言性の文節、連体型の文節カテゴリなら体言性の文節が期待される文節のリストとして書いてあり、それに沿う文節構造が選択される。これによって、「泳ぎ、」の直前にたとえば「彼は」があれば連用型用言文節と解釈され、「彼の」とあれば体言文節と解釈される。

文節構造選択規則は、このように直前の文節から判断してしまう単純なものであり、全体の文法構造を反映していないから不十分なものである。たとえば、

「太郎はおどり、歌、スポーツが好きだ」

では、「おどり」を連用型用言文節と誤って解釈してしまう。

文節構造選択規則を評価しても一意に定まらない場合は、現在は、システムがユーザに問い合わせようになっている。

5.2.2 連体埋め込み構造の解析

(1) 連体埋め込み構造の解析手順

係りの文節が連体型用言文節の場合、これまでに述べた後戻りなしの解析法では、埋め込み文の範囲を誤ってしまい、正しく解析できない場合がある。たとえば、

「犬がネズミを殺した猫を殺した」

という文では、「犬がネズミを殺した」という部分が「猫」を修飾する部分となってしまう、解析不能となる。そこで、連体埋め込み構造の場合には、次のような、後戻り（バックトラック）を含んだ解析手法をとる。

① 連体型用言文節 (Df とする) に対して通常に係り受け解析を行う (結果を Df' とする)。

② 後続部分から体言文節を解析する (Dn とする)。

③ Df' と Dn との係り受けを調べる。

係り受けが成立すれば終了。

④ 成立しなければ、Df' の最左端の依存構造式 (つまり、Df' への係りで最も文頭側のもの) との係り受け関係を解除したものを、あらためて Df' として ③ へ。

これによって、さきほどの例文は、図 5.2 に示すように解析される。

- 手順① --> Df' = [犬がネズミを殺した]
 手順② --> Dn = [猫を]
 手順③ --> Df' と Dn との係り受けが成立しない
 手順④ --> Df' = [ネズミを殺した]
 手順⑤ --> Dn が Df' の主格として係る... 終了
 犬が [猫: 猫がネズミを殺した] を殺した。

図 5.2 連体埋め込み構造の解析手順例
 「犬がネズミを殺した猫を殺した」

上記の手順では、修飾される体言の側 (Dn) の範囲は最初にくる体言であると仮定しているが、これは必ずしも正しくない。たとえば、

- イ. 太郎が食べた花子のリンゴ
 ロ. 太郎が食べたリンゴで作ったジャム
 ハ. リンゴを食べた太郎の妹

イ では、「花子を食べる」よりも「リンゴを食べる」と解釈するほうが常識的である。

ロ では、“食べてしまったリンゴではジャムは作れないこと”を知っていなければ正しく解析できない。

ハ では、「太郎がリンゴを食べた」という解釈も、「太郎の妹がリンゴを食べた」という解釈も両方ありえる。

このように、埋め込み文の範囲を正しく解析することは、一般常識や文脈の解析を本質的に必要とする、困難な課題のひとつである。

本システムでは、“修飾される体言は最初に現れる体言”とする仮定をいくらか緩めるために、上記①～④の手順終了後、

- ⑤ Dn が連体型体言文節であって、後続の体言 Dn' に係る場合には、Df' と Dn' との係り受け関係も調べ、評価値がより良くなるならば、Df' が係る範囲をその体言にまで広げる。

という処理を追加している。これによって、上の例イに対しては、

{太郎が花子のリンゴを食べた：そのリンゴ}

という解釈を導くことができる。

(2) 連体修飾関係を調べる関数

連体修飾構造には次の 2 種があることは先に述べた (4.3.2.2)。

① 同一名詞連体修飾

連体修飾文中に被修飾名詞と同一の名詞をもつもの。

「ネズミが食べたチーズ」... {ネズミがチーズを食べた}

「蓋が壊れた鍋」... {鍋の蓋が壊れた}

② 付加名詞連体修飾

連体修飾文中に被修飾名詞と同一の名詞をもたないもの

「魚が焼けるにおい」

「太郎がリンゴを食べた時,」

付加名詞連体修飾の構造をとる名詞の場合は, (4.3.2.2) で述べたように, 連体修飾文節を係りとしてとるということを, その名詞の格構造に記述しておく。たとえば, 「におい」の格構造を,

(N<) (=Q 0 <L <P))

のように記述しておく。

(1) で述べた手順 ③:

“Df と Dn との係り受け関係を調べる”

で, Df を係り, Dn を受けとする通常の係り受け規則の適用を行なえば, この格構造規則が評価される。すなわち, 付加名詞連体修飾の場合は, その付加連体名詞の格構造によって解析することができる。

同一名詞連体修飾は, 連体修飾文中の係りの要素が, 被修飾名詞として外に押し出された構造である。したがって, 表層の係り受けの方向とは逆に, Dn を係り, Df を受けとする係り受け関係を調べることによって, その構造を解析することができる。ところで, 同一名詞連体修飾には, 上の例にみるように, さらに 2 つの場合がある。

① 被修飾名詞が, 連体修飾文への直接の係りとなる場合

「ネズミが食べたチーズ」... {チーズを → 食べる}

② 被修飾名詞が, 連体修飾文への係りの語句への係りとなる場合

「蓋が壊れた鍋」... {(鍋の → 蓋が) ... 壊れる}

「目がきれいな少女」... {(少女の → 目が) ... きれいだ}

「値段が高い商品」... {(商品の → 値段が) ... 高い}

したがって, 同一名詞連体修飾の場合には, 通常の係り受けの方向とは逆方向の係り受け関係を, このような 2 つの場合について調べなければならない。関数 (RRULE) は, このための次のような関数である。

(a) Dn が Df への係りとなり得るかどうか調べる。係り受け関係が成立すれば, Dn を Df に統合し, さらにその Df を Dn に統合する。

(b) 成立しないなら, Dn が Df に既に係っている依存構造式への係りとなるかどうかを調べる。係り受け関係が成立すれば, 同様に Dn と Df を統合する。

いずれの場合も, Dn との係り受け関係を評価する際, Dn に対する機能語条件は無視する。

(RRULE) は, (5.4.1) で示すように, 連体型用言文節カテゴリの係りの規則の中で評価される。

図 5.3 に連体修飾構造の解析例を示す。

```

魚が焼ける匂い
($T JN0039) ニオイ
  (=Q JN0038) ヤケル
    (:P JN0037) サカナ
      (/ .PCP 0 !00Z !OPA)

蓋が壊れた鍋
($T JN0033) ナハ
  (=R JN0032) コワレル
    (:P JN0031) フタ
      (@ RENTAI-M) ナハ
        (/ .PCP 0 !OPP)

猫が殺した鼠が食べたチーズは腐っていた。
($SYUSHI JN3275) クサル
  (:P JN3274) チ ス
    (=R JN3273) タヘル
      (.LO RENTAI-M) チ ス
        (:L JN3272) ネスミ
          (=R JN3271) コロス
            (.LO RENTAI-M) ネスミ
              (:L JN3270) ネコ

```

図 5.3 連体修飾構造の解析例

5.2.3 後処理

(1) 依存構造式の評価

係り受け解析の結果、Df に D1, D2, …, Dn が係って Df となったとする。このとき、Df なる解釈のもっともらしさの目安として、係り受け関係の適合度をもとに、その評価値 h を次のように算出する。

“Di の評価値を hi とし、Di と Df との間の係り受け関係の適合度を gi とすると、

$$h = \sum (hi + gi) / 2 * n$$

ただし、Df に係るものがない場合 (Df が格構造をもたない場合) には、h=1.0 とする。

また、文全体の解析が終わった時点では、解析結果の構造中の省略項のペナルティ値の合計を差し引く。

たとえば、

「彼は石も食べた」

という文では、「食べる」の格構造が、

((<L) (:L 0 !OA !OHH) (.LO 0 !OO))

となっていれば、

{彼ガ石ヲ食べる}

という解釈は、「彼」が !OHH の意味特徴を含み、「石」が !OOS の意味特徴を含むから意味条件の適合度はいずれも 1.0 である。また、機能語条件の適合度はいずれも 0.7 であるから (4.3.1 参照)、全体としての適合度はいずれも、 $0.7 * 0.6 + 1.0 * 0.4 = 0.82$ と計算される。したがって評価値は、

$$h = ((1.0 + 0.82) + (1.0 + 0.82)) / 2 * 2 = 0.91$$

となる。また、

{彼ヲ石ガ食べる}

という解釈は、意味条件がいずれも合わないから、格構造の適合度はいずれも $0.7 * 0.6 = 0.42$ と計算され、したがって評価値は、

$$h = ((1.0 + 0.42) + (1.0 + 0.42)) / 2 * 2 = 0.71$$

となる。この2つの評価値を比較することによって、{彼ガ石ヲ食べる} という解釈の方がもっともらしいと判断することになる。

係り受け解析部では、生成された依存構造式の中で、評価値が最も高いものからある範囲の中にあるものだけを取り出して、可能性の低い解釈をとりのぞいておくことができる。

(2) テンス・アスペクトの抽出

構文・意味解析終了後、テンス・アスペクトの抽出を行っている。

- ・アスペクトの抽出は、(井上：65)に従った。

ここでは、アスペクト素性として、

状態 (STATE±), 動作 (ACT±), 継続 (CONT±), 完結 (COMP±),

主観 (SUB±), 反復 (REPEAT+), 始動 (START+), 結果 (RESULT+)

の8種を設け、用言をアスペクト素性の観点から8類に分類している(図5.4)。本システムでは、この分類も用言の意味特徴として付加している。アスペクトの解析としては、次の①、②の処理を行った。

- ① 用言に、「～ている」、「～てしまう」、「～し終る」、...などの機能語(アスペクト形式素)が続いた場合の、全体としてのアスペクト素性の集合を求める(井上：65)。
- ② その結果を、表5.2に示す解釈ルールによって解釈し、アスペクトとして、

状態相, 起動相, 進行相, 完結相, 実現相, 結果状態相

のいずれであるかを求める。これらは、英語への翻訳実験において、進行形/完了形を選択

| | | | |
|--------|-------|-------|---------------------|
| STATE+ | CONT+ | ... | ある、あたる、いる |
| | CONT- | ... | そびえる、ただよう、似る |
| STATE- | ACT+ | CONT+ | COMP+ ... 読む、作る、散る |
| | | COMP- | ... 降る、荒れる、待つ |
| | ACT- | CONT- | ... 死ぬ、結婚する、曲げる |
| | | CONT+ | SUB+ ... 信じる、感じる、思う |
| | | SUB- | ... 悲しむ、まよう、じれる |
| | | CONT- | ... 知る、わかる、わきまえる |

図5.4 述語のアスペクト素性(井上：65)

するために求めたものである。

- テンスの抽出は、(草薙：71) のアルゴリズムに従って、

絶対テンス ... 現象にあたっている焦点と発話時点との時間的前後関係

焦点が発話時点より前 ... 「過去」

焦点と発話時点が同じ ... 「現在」

焦点が発話時点より後 ... 「未来」

相対テンス ... 焦点があたっている現象と第二の現象との時間的前後関係

第二の現象が焦点のあたっている現象より前 ... 「前」

第二の現象と焦点のあたっている現象が同じ ... 「同」

第二の現象が焦点のあたっている現象より後 ... 「後」

を抽出した。

表 5.2 アスペクトの解釈ルール

| アスペクト | アスペクト素性の集合 |
|-------|--------------------------------------|
| 状態相 | STATE+, CONT+ |
| 完結相 | STATE-, ACT+, CONT-, COMP+ |
| 進行相 | CONT+, ACT+, RESULT-, COMP- |
| 起動相 | STATE-, ACT+, CONT-, START+, RESULT- |
| 実現相 | STATE-, ACT+, CONT- |
| 結果状態相 | STATE+, CONT+, RESULT+, |

5.3 係り受け規則のプログラミング

5.3.1 プログラミングの環境

(1) 係りおよび受けの項を参照する変数

手続き型の係り受け規則は、リスプのプログラムである。このプログラムでは、問題となっている係りの依存構造式と、受けの依存構造式の内容を、次のような変数を通じて参照することができる。

%HM...係りの依存構造式

%H-M%P-M に既に係っている依存構造式に関する履歴

%H 1-M%H-M に最後に係った依存構造式に関する情報 (%H-M の先頭)

%CLIST-M%H-M に既に係っている依存構造式の格役割名ないし係りの標識のリスト

%P-M 依存構造式%HMの本体
 %CAP-M (%C-M, %AP-M)
 %C-M %P-Mの文節カテゴリ
 %AP-M %P-Mの機能語列
 %BA-M ... %AP-Mの中で結合に最も強く関与する機能語
 %KSP-M (%K-M, %SP-M)
 %K-M %P-Mの意味特徴のリスト
 %KW-M ... %P-Mの内容語
 %SP-M %P-Mに係っている依存構造式, および格構造のうち未だ満たされて
 いない格記述のリスト

%HF...受けの依存構造式

%H-F %P-Fに既に係っている依存構造式に関する履歴
 %H 1-F %H-Fに最後に係った依存構造式に関する情報 (%H-Fの先頭)
 %CLIST-F %H-Fに既に係っている依存構造式の格役割名ないし係りの標識
 のリスト
 %P-F 依存構造式 %HFの本体
 %CAP-F (%C-F, %AP-F)
 %AP-F %P-Fの機能語列
 %KSP-F (%K-F, %SP-F)
 %K-F %P-Fの意味特徴のリスト
 %KW-F ... %P-Fの内容語
 %SP-F %P-Fに係っている依存構造式, および格構造のうち未だ満たさ
 れていない格記述のリスト

以上の変数の他に, 次の変数も参照できる。

%%....その係り受け規則を対応づけている辞書項目それ自身。

すなわち, %C-F, %C-M, %KW-F, %KW-M, %BA-Mのいずれか。

(2) 係り受け規則を記述するための関数

係り受け規則は, 2つの依存構造式の間係り受け関係を調べて, 係り受け関係の適合度がその時点で設定されている閾値 (*CUT. B) よりも大きければ, 係りの依存構造式を受けの依存構造式の中に取り込んで, 1つの依存構造式に統合するプログラムである。このプログラムのために次のような関数が用意してある。

① (GRULE condition role index)

- condition は, 係り受けの条件を記述するリスブ関数であり, その値は, 係り受けが成立しなければ NIL, 成立すれば適合度を与える実数 (通常 0.0~1.0) である。

condition の記述の中で (1) で述べた大域変数を参照することができる。

GRULE は condition を評価し、その値が現在設定されている閾値 (*CUT. B) よりも大きいならば、2つの依存構造式を統合する。

- role は、係り受け関係の種類を示す標識である。係りの依存構造式を受けの依存構造式の中に取り込む際に、係りの文節カテゴリが role で置き換えられる。
- index は、どの規則で係り受け関係が成立したのか (すなわち、この GRULE 自身) を、係り受けの履歴情報の中に記録しておくための識別記号である。履歴情報の中には、この係り受け規則を記述している辞書項目と index との対が記録される。

この記録は、6章で述べる機械翻訳への応用の際重要となる。

② (FRULE)

- 格構造規則による係り受け関係を調べる関数。係り受け関係が閾値 (*CUT. B) 以上で成立すれば、2つの依存構造式を統合する。

③ (MRULE)

- 係り構造規則による係り受け関係を調べる関数。係り受け関係が閾値 (*CUT. B) 以上で成立すれば、2つの依存構造式を統合する。

[(FRULE) と (MRULE) を統合した関数 (FRULE!MRULE) もある。]

5.3.2 係り受け関係の種類

係り受け関係には、格構造で表現される関係の他に多くのものがある。それらを以下のように分類した。それぞれのグループごとに、係りの標識の先頭の記号を区別している。

これらの係り受け関係の標識は、係り構造規則 (rtype EC SC), あるいは、前節で述べた関数 (GRULE condition rtype index) の rtype として記述されるものであって、係り受け関係の種類を表示する。係り受け関係が成立することが確認されて、係りの項を受けの項に統合する際、係りの項の文節カテゴリの位置が、この rtype によって置き換えられる。

なお、同じ標識 (格役割名) を持つ係り受け関係が既に存在すれば、重ねて同じ標識の係り受けを成立させることはできない (1文1格の原理)。

① 体言→用言の係り受け関係

| | | | |
|-----|----------|--------|-------------|
| _TF | 時の範囲の始まり | カラ | 朝から晩まで働く |
| _TT | 時の範囲の終り | マデ | 朝から晩まで遊ぶ |
| _TD | 期間 | NIL | 5年間 |
| _T | 時 | NIL, ニ | 夕方, 夕方に |
| _T. | 時 | デ | 5時で帰る |
| _PF | 所の範囲の始まり | カラ | 鹿児島から東京まで行く |
| _PT | 所の範囲の終り | マデ | 鹿児島から東京まで歩く |

| | | | |
|----|----|-----|---------|
| _P | 所 | デ | 工場で作る |
| _W | 同伴 | ト | 彼と帰る |
| _C | 比較 | ヨリ | 彼より速い |
| _N | 量 | NIL | 5グラム入れる |

この他、*ヲノゾイテ、*ヲツレテ、*カドウカ等、格助詞相当の補助用言による任意的な格がある。

② 体言→体言の係り受け関係

| | | | |
|------|----------------|---------|----------------|
| @NUM | 量 | ノ | 5グラムの水 |
| @ | ノで結ばれる任意的な関係一般 | ノ | 太郎の本 首相の中曽根 |
| & | 並立 | ト, ヤ, カ | 太郎と(や, か)二郎 |

この他、#_TF, #_TT, #*ヲノゾイテ, # \$シテカラ等、①あるいは③の関係の体言化(～カラノ, ～マデノ, ～ヲノゾイテノ, シテカラノ等)に伴うものがある。

③ 用言→用言の係り受け関係

\$シナガラ, \$シテカラ, \$シタカラ等、接続助詞による係り受け関係

④ 用言→体言の係り受け関係

すべて、同一名詞連体修飾または付加名詞連体修飾として、修飾する用言の格構造、あるいは修飾される体言の格構造で把握するものとした。

⑤ 副言、接言と用言、体言、副言の間の係り受け関係

| | | |
|-----|-------|-----------|
| Y | 副言→用言 | ゆっくり歩く |
| - T | 副言→体言 | あの本 |
| / F | 副言→副言 | もっとゆっくり歩く |
| + Y | 接言→用言 | しかし行かなかった |

5.4 係り受け規則の記述例

5.4.1 文節カテゴリの係り受け規則

14個の文節カテゴリに対する係り受け規則の記述例を、図5.5に示す。

図5.5で、たとえば連用型体言文節(\$T>Y)の規則は、次のようなことを表現している。

\$DEP-M0 =>

◀① 受けの側が連体型または仮定型の用言文節で、かつ係りの側が「～は」という形ならば、係り受け関係は成立しない。

② 受けの側が連用型または接続型の用言文節で、かつ係りの側が「～は」あるいは「～、」という形ならば、機能語条件を調べる際の評価値を低く算出するようにしておいて、格構

図 5.5 文節カテゴリーの係り受け規則の記述例

```

$T>Y
$DEP-MO ... (COND((AND (MEMQ %C-F '$RENTAI $KATBI)) (EQ (CAR %AP-M) 'ハ)) T)
              ((AND (MEMQ %C-F '$RENYOU $Y>Y)) (MEMQ (CAR %AP-M) '(ハ *COMMA)))
              (LETS ((%A-MATCH 0.4)) (FRULE!MRULE)))
              ((MEMQ %C-F &YOUGEN) (FRULE!MRULE))
              ((AND (LISTP %K-F) (EQ (CAR %K-F) '*NUM) (EQ %BA-M 'ガ)) NIL)
              (T T))
$DEP-M ... (GRULE (AND (MEMQ %C-F &YOUGEN) 0.05) *? 1)

$T>T
$DEP-MO ... (COND ((EQ %C-F '$RENTAI) (OR %H-F (NEQ %BA-M 'ノ)))
                  ((MEMQ %C-F &TAIGEN) (FRULE!MRULE))
                  (T T))
$DEP-M ... (GRULE (AND (MEMQ %C-F &TAIGEN) 0.01) @? 1)

$T>
$DEP-MO ... (FRULE!MRULE)

$RENTAI
$DEP-MO ... (COND ((MEMQ %C-F &TAIGEN) (FRULE!MRULE) (RRULE))
                  (T T))
$DEP-M ... NIL

$RENYOU
$DEP-MO ... (COND ((MEMQ %C-F &YOUGEN) (FRULE!MRULE))
                  (T T))
$DEP-M ... (OR (GRULE (SEMQ %K-M
                      ((<A <F <Z)
                      (AND (NOT (INTERSEQ '(:A .AO :F .FO :Z .ZR) %CLIST-M))
                          (SEMQ %K-F ((<A <F <Z) NIL) (T T))
                          2.0)))

```

```

      |REYOU 1)
(GRULE (SELECTQ %C-F
        (($SYUSI $TOIKAKE $MEIREI $REYOU $Y>Y) 0.6)
        ($KATEI
          (COND ((MEMQ '$KATEI %CLIST-M) -1.0)
                (T 0.6)))
        ($RENTAI 0.5)
      $REYOU 2))

```

\$KATEI

```

$DEP-FO ... (MEMQ '$KATEI %C-M)
$DEP-MO ... (COND ((MEMQ %C-F &YOUGEN) (FRULE!MRULE))
              (T T))
$DEP-M ... (GRULE (SELECTQ %C-F
                  (($SYUSI $TOIKAKE $MEIREI) 0.6)
                  (($REYOU $Y>Y) 0.4)
                  ($RENTAI 0.2)
                $KATEI 1)

```

\$Y>Y

```

$DEP-MO ... (COND ((MEMQ %C-F &YOUGEN) (FRULE!MRULE))
              (T T))
$DEP-M ... (GRULE (SELECTQ %C-F
                  (($SYUSI $TOIKAKE $MEIREI) 0.6)
                  ($REYOU 0.4)
                  (($Y>Y $RENTAI) 0.2)
                $Y>Y 1)

```

\$SYUSI

```

$DEP-MO ... (COND ((EQ %K-F '*QUO) (FRULE))
              (T T))

```

```

$MEIREI
  $DEP-MO ... (COND ((EQ %K-F '*QUO) (FRULE))
                (T T))

$TOIKAKE
  $DEP-MO ... (COND ((EQ %K-F '*QUO) (FRULE))
                (T T))

$F>Y
  $DEP-FO ... (COND ((MEMQ %C-M '$F>) (MRULE))
                    (T T))
  $DEP-MO ... (COND ((MEMQ %C-F &YOUGEN) (FRULE!MRULE))
                    (T T))
  $DEP-M ... (GRULE (AND (MEMQ %C-F &YOUGEN) 0.4) !? 1)

$F>T
  $DEP-FO ... T
  $DEP-MO ... (COND ((MEMQ %C-F &TAIGEN) (FRULE!MRULE))
                    (T T))
  $DEP-M ... (GRULE (AND (MEMQ %C-M &TAIGEN) 0.4) -? 1)

$F>
  $DEP-FO ... T
  $DEP-MO ... (COND ((OR (MEMQ %C-F &YOUGEN) (MEMQ %C-F '$F>Y $F>))) (FRULE!MRULE))
                (T T))

$DEP-M ... (GRULE (AND (OR (MEMQ %C-F &YOUGEN) (MEMQ %C-F '$F>Y $F>))) 0.4) /? 1)

$$>Y
  $DEP-FO ... T
  $DEP-MO ... (COND ((MEMQ %C-F '$SYUSI $TOIKAKE $MEIREI)) (MRULE))
                (T T))
  $DEP-M ... (GRULE (AND (MEMQ %C-F '$SYUSI $TOIKAKE $MEIREI)) 0.4) +? 1)

```

図 5.5 文節カテゴリの係り受け規則の記述例

造規則および係り構造規則を評価せよ。

{%A-MATCH は機能語条件に関する重みであり、既定値は、%A-MATCH=0.6、意味条件に関する重み%S-MATCH=0.4、となっている。}

- ③ 「温度が5度の...」のような数値表現の場合を除いて、受けの側がそれ以外の用言文節ならば、格構造規則および係り構造規則を評価せよ。
- ④ それら以外の場合には、係り受け関係は成立しない。》

\$DEP-M =>

《⑤ 受けの側が用言文節ならば、低い評価値 (0.05) で係り受け関係を成立させよ。》

\$DEP-M0 の ① は、「～ハ」という形の連用型体言文節は、連体形文節および假定形文節には係らないという規則を表現している。たとえば、

「花子はきのう作ったケーキを食べた」

における「花子は」は、「作った」ではなく「食べた」に係るように分析される。なお、

「花子 きんのう作ったケーキ 食べた」

という不完全な文の場合には、

(花子(ガ)きのう作ったケーキ) (ヲ)食べた

と分析される。

② は、「～ハ」は、連用形文節や接続助詞を伴った連用性文節には係りにくい、すなわち「～ハ」は、通常、文末の用言に係るということを表現している。

③ は、数値表現の場合を除けば、連用型体言文節は用言文節に係ることを表現している。数値表現の場合には、「温度が5度の水」のように、連用型体言文節「温度が」が、体言「5度」に係っているように見える現象がある。数値表現の解析については後述する。

⑤ は最後に適用される規則であり、意味的に解釈できなかった場合でも、この規則によって最終的には(つまり、他に係り受けを成立させるものがなければ)、係り受け関係が成立すると判断される。

5.4.2 機能語の係り受け規則

機能語の係り受け規則の記述例を、図 5.6 に示す。

たとえば、「カラ」の係りの規則は、

《・〔時間〕を意味特徴に持つ内容語に伴って用言文節 (T 文, C 文, L 文, P 文) に係り、時間の開始点を示す。その場合、受けの側が「歩きはじめた」のように、機能語「ハジメル」を伴っていれば、その係り受け関係は確実である。そうでなければ、やや不確かである。

・〔場所〕を意味特徴にもつ内容語に伴って用言文節 (T 文) に係り、空間的な移動の開始点を示す。その場合先と同様に、受けの側が機能語「ハジメル」を伴っていれば、そ


```

カラ ... (SEMQ, %K-M (!T (COND ((MEMQ 'ハジメル %AP-F) (GRULE 1.0 _TF 1))
                                (T (SEMQ, %K-F ((<T <C <L <P) (GRULE 0.8 _TF 1))))))
          (!P (COND ((MEMQ 'ハジメル %AP-F) (GRULE 1.0 _PF 2))
                   (T (SEMQ, %K-F (<T (GRULE 0.8 _PF 3))))))
          (T (COND ((MEMQ 'ハジメル %AP-F) (GRULE 1.0 _S 1))
                (T (SEMQ, %K-F (<L (GRULE 0.4 _S 4)))))))

ガ1 {接続助詞の「が」}
... (GRULE (AND (EQ %C-F 'SSYUSI) 1.0) *ケレドモ 1)

デ ... (SEMQ, %K-M (!P (SEMQ, %K-F ((<T <C <L <P <F <W) (GRULE 0.8 _P 1))))
        (!O (SEMQ, %K-F (<L (GRULE 0.4 .LTP 1))))))

ト ... (OR (HRULE &AND 1)
          (GRULE (SEMQ, %K-M (!OH (SEMQ, %K-F ((<T <C <L) 0.8)))) _W 2))

```

図 5.6 機能語の係りの規則 (\$DEP-M) の記述例

の係り受け関係は確実である。

- ・そうでなければ、L 文に係って、「チーズから食べる」のように、行為の開始物を示す。この場合も同様に「ハジメル」を伴っていれば、この係り受け関係は確実である。➤

ということ表現している。

「デ」の規則は、

- ◀・[場所] を意味特徴にもつ内容語に伴って、用言文節 (T 文, C 文, L 文, P 文, F 文, W 文) に係り、その行為などの行われる場所を示す。
- ・[物] を意味特徴にもつ内容語に伴って、用言文節 (L 文) に係り、道具・手段・材料を示す。➤

ということ表現している。もちろん、「デ」の用法はこれだけではない。それらは、格構造規則など他の規則によって規則化される。

「ト」の規則中の (HRULE ...) は、並立関係を調べる関数である。HRULE は、依存構造式中の履歴情報を参照して、直前に成立した係り受け関係と同様の係り受け関係が、その文節 (依存構造式) に関して成立するかどうかを調べる。

- ① 「太郎は花子とアメリカに行った」
- ② 「太郎はイギリスとアメリカに行った」

この例では、「行く」に対して成立した直前の係り受け関係〔アメリカに→行く〕と同じ関係で

①〔花子に→行く〕

②〔イギリスに→行く〕

が成立するかどうか調べる。②の場合は成立するが、①の場合は成立しない。②の場合は、「イギリス」は「アメリカ」に並立するものと判断して、この2つを統合する。

①の場合は、「ト」のもう一つの規則 (GRULE ... _W ...) によって、随伴格_W で係り受けが成立することになる。

ただし、(HRULE ...) は、「日本とアメリカ。」のように受けの側が文末である場合には、無条件に並立関係が成立するものと判断している。

5.4.3 内容語の係り受け規則

内容語の係り受け規則の記述例を、図 5.7 に示す。

内容語の係り受け規則には、手続き型の規則として、受けの規則 (\$DEP-F) および係りの規則 (\$DEP-M) があり、宣言型の規則として、受けの規則 (格構造規則; \$FRULE) および係りの規則 (係り構造規則; \$MRULE) がある。格構造規則については第 4 章で述べた。図 5.7 では、\$DEP-F, \$DEP-M, \$MRULE の例を示した。

*QUO は、引用文を処理するために文節構造解析部で導入した擬語である。たとえば、

『『日本へ行く』と言った』

という文は、

『『日本へ行く』 *QUO と言った』

のように扱われる。『...』の部分は、優先して一つの構造にまとめられ、それが図 5.7 に示した *QUO の受けの規則 \$DEP-F によって、*QUO に係る。%H-F は、受けの側に既に係っている項に関する履歴情報である。この場合には、まだ何も係っていない (%H-F=NIL)、つまり *QUO の直前にあるものでなければならないということが条件となっている。この例文は、次のように解析される。

```
($SYUSI) イウ
  (.CQ ト) *QUO
    ($SYUSI) イク
      (.TT へ) ニホン
        (:T 0 !OHH)
          (:C 0 !OHH)
```

*NUM は、数値表現を処理するために文節構造解析部で導入した擬語である。数値表現は次のようにして解析される。

\$DEP-F (受けの規則) の例

*QUO (引用文を処理する擬語)

```
(COND ((NULL %H-F) (GRULE 1.0 #QUO 1))
      (T T))
```

*NUM (数値表現を処理する擬語)

```
(COND ((AND (MEMQ %BA-M '(ガ NIL)) (UNIT-MATCH %KW-M %AP-F))
      (LET ((%KSP-M %KSP-F) (%K-F %K-M) (%SP-F %SP-M) (%CLIST-F %CLIST-M)
            (%C-M %C-F) (%H-M %H-F) (%H-M %H-M) (%HM (MODUS-PCR %HF)))
          (GRULE 1.0 @NUM 1)))
```

```
オンド... (COND ((AND (EQ %BA-M 'ノ) (UNIT-MATCH %KW-F %AP-M))
                 (GRULE 1.0 @NUM 1)))
```

\$DEP-M (係りの規則) の例

```
キノウ ... (GRULE (AND (MEMQ %BA-M '(ハ モ NIL)) (MEMQ 'タ %AP-F) 1.0) _T 1)
ナゼ ... (GRULE (AND (MEMQ 'カ %AP-F) 1.0) |Q 1)
マツク... (GRULE (AND (MEMQ 'ナイ %AP-M) 1.0) |D 1)
モシ ... (GRULE (AND (EQ %C-F '$KATEI) 1.0) |D 1)
```

\$MRULE (係り構造規則) の例

```
ユックリ... ((Y ((A ト)) <T <C <L <P)
トテモ ... ((Y 0 <F <A <Z)
              (/F 0 *YM))
トキ ... (T 0 <)
*NUM ... (@NUM 0 !0)
```

図 5.7 内容語の係り受け規則の記述例

①たとえば,

「温度が5度の水」

という文は、文節構造解析部で、図 5.8 のように解析する。前処理部で数値部分を分離して、数値そのものは擬語 *NUM* で置き換え、さらに数値表現を処理するための擬語 *NUM を導入する。

| |
|-------------------|
| |
| ((\$T>)水) |
| ((\$T>Tの度)*NUM) |
| ((\$T>T)(*NUM*5)) |
| ((\$T>Yが)温度) |
| |

図 5.8 「温度が5度の水」の文節構造

② *NUM の格構造で [*NUM*] を受ける。

*NUM の格構造=>((>N)(&NUM 0 *NUM*))

③ *NUM の受けの規則 (図 5.7) で, [温度が] を受ける。

この規則の中の (UNIT-MATCH %KW-M %AP-F) は, 「温度」と「度」の間の適合性を調べる関数である。

また, (LET X (GRULE ...)) の X の部分は, 係り [温度] と受け [*NUM] の位置を逆転して統合するための処置である。

以上によって, 「温度が5度の水」は, 次のように解析される。

(\$T>)ミズ
 (@)オンド
 (@NUM) *NUM
 (&NUM) (*NUM* 5)

「温度が5度の水」における「温度が」は, 「体言+が」であるから連用型体言文節であって, 一般には, 用言文節に係るべきものである。この場合には, もともと, 「温度が5度である水」における連体型用言文節「5度である」に係っていたものであるが, 「5度である」が, 連体型体言文節「5度の」に転化してしまったために, 体言文節に係る形になっているものと観察される。この連用型体言文節における例外的事項は, 連用型体言文節 (\$T>Y) の係りの規則 (\$DEP-M0; 図 5.5) に記述してある。

一般に数値表現には, まったく同じ内容を表現するのにいくつかの表現法がある。

イ, 「温度が5度の水」

ロ, 「温度5度の水」

ハ, 「5度の温度の水」

イとロは, 上述の手順で解析される。ハは, 「温度」の受けの規則 (図 5.7) によって, イ, ロと同じ形に解析することができる。

この他内容語の係り受け規則では, 図 5.7 にみるように, 係りの規則 (\$DEP-M) で,

「きのう・・・～した」

「なぜ・・・～したか」

「まったく・・・～しない」

「もし・・・～すれば」

のような、種々の呼応関係を容易に記述することができる。

5.5 まとめ

この章では、係り受け規則の種類とその適用手順、係り受け規則の記述法について述べ、具体例を示した。係り受け規則は、文節カテゴリ、機能語および内容語の属性として記述してある。格構造規則も係り受け規則の一種として位置づけてある。これらの係り受け規則によって、一般的なあるいは個別的な種々のタイプの係り受けの規則を記述できることを示した。句構造文法における書き換え規則のような規則表示の形ではなく、その規則をもたらす主たる契機となっている言語対象の属性として記述したことによって、個別的な事項も素直に表現できた。また、解析システムは、規則を取り出して評価するという形で簡潔に構成することができた。連体埋め込み構造についても、これらの係り受け規則で解析できることを示した。

第6章 機械翻訳への応用

6.1 はじめに

これまでに述べた方式による構文・意味解析の有効性を確認するために、その解析結果を応用した機械翻訳システムを作成した。本章では、この機械翻訳の手法について述べ、解析・翻訳の実験結果を示す。

この機械翻訳の手法は、以下のような特徴をもっている。

- ① 翻訳規則は、係り受け規則と1対1に対応させた。

たとえば、「割る」の格構造、

((<L) (:L 0 !ヒト) (.LO 0 !エキタイ) (.LTP 0 !エキタイ))

は、[:L が, .LO を, .LTP で, 割る]

{たとえば、「花子が、ウイスキーを、水で、割る」}

という「割る」の用法を表現しているが、これに対応する翻訳規則を、

(<V :L MIX .LO (<PP WITH .LTP))

と表現する。

このように、係り受け規則すなわち日本語の個々の用法と、その目標言語での表現を直接、自然な形で対応させている。

翻訳規則は係り受け解析の結果から直接得られるから、外部の規則等と照合することなく高速に処理することができ、翻訳システムも簡潔に構成することができる。

- ② 翻訳規則は手続き型の規則（リスプのプログラム）であり、条件文によって容易に訳し分けを記述することができる。
- ③ 英語、独語、仏語など目標言語毎に翻訳規則を用意しておけば、同じ解析結果から、同じ翻訳システムで多言語への翻訳ができる。

6.2 翻訳システム

6.2.1 翻訳システムの構成

翻訳は、図6.1に示すように、①解析、②変換、③生成の3つのフェーズからなるトランスファ方式である。翻訳システムは、構文・意味解析の結果を翻訳の表現式に変換し、次いで表現式を展開して翻訳文を生成する（図6.2）。

以下に翻訳処理手順の概要を述べる。（次節に処理例を示す。）

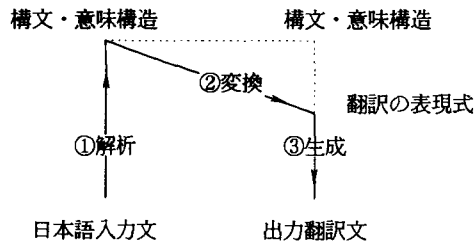


図 6.1 翻訳方式

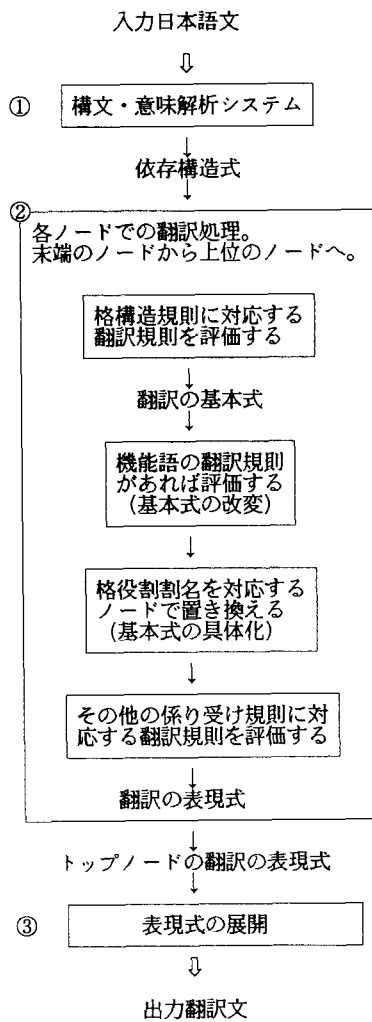


図 6.2 翻訳システムの構成

① 解析

前章までに述べた構文・意味解析システムによって、日本語入力文を解析し、その出力である依存構造木を翻訳システムに入力する。依存構造木は文節をノードとする木であり、その文節の文節カテゴリ、内容語、意味特徴のリスト、上位ノード、下位ノードのリスト、上位ノードに対する係りの関係（解析の際に採用された係り受け規則の識別子）、用言ノードであればテンスやアスペクトなどを属性としてもっている。

② 変換

依存構造木の末端のノードから上位のノードへ順次に、対応する表現構造（翻訳の表現式）への変換を積み上げる。

- (a) そのノードの分析に採用された格構造規則に対応する翻訳規則を評価する。この結果を翻訳の基本式という。
- (b) そのノードに付属している機能語を、出現順に走査する。翻訳式改変規則をもつものがあれば、それを評価して翻訳の基本式を修正する。
- (c) 翻訳の基本式を具体化する。すなわち、
 - ・基本式中に現れる格役割名を該当するノードで置き換える。
 - ・基本式中の翻訳文生成関数の第1引数として、当該ノードを挿入する。これは、生成関数を評価する際に、時制や数など各種の情報を参照するのに用いられる。
- (d) 格構造規則以外の係り受け規則で係っている下位ノードについて、対応する翻訳規則を評価する。

③ 生成

トップノードの翻訳の表現式を展開して翻訳文を得る。

翻訳の表現式は解析木の各ノードに対応する目標言語の表現構造を与えるものである。これは次の(a)~(d)で表現されるものであり、それぞれ(a)~(d)のように目標言語の翻訳文に展開される。

- (a) 目標言語の語彙
 - (a)それ自身が翻訳文となる。
- (b) 解析木のノード
 - (b)そのノードの翻訳の表現式を展開したものが翻訳文となる。
- (c) 翻訳の表現式を引数とする関数（翻訳文生成関数という）。
 - (c)その翻訳文生成関数を評価したものが翻訳文となる。
- (d) 翻訳の表現式の列
 - (d)各表現式を展開して、そのままの順序でつなぎ合せたものが翻訳文となる。

翻訳の表現式には、次の5種類のものがある。

- (1) そのノードに対応する翻訳の中核部分を表現する式。

(格構造規則に対応する部分である。)

- (2) 中核部分の翻訳に後置すべき語句を表現する式。
- (3) 中核部分の翻訳に前置すべき語句を表現する式。
- (4) 挿入する副詞句を表現する式。
- (5) 挿入する形容詞を表現する式。

翻訳文への展開は、(3), (1), (2)の順に展開してつなぎ合わせることになる。(4)と(5)は、述部を生成する生成関数によって取り込まれる。

翻訳処理に関する規則には、次の3種がある。

- ① 係り受け規則に対応する翻訳規則，すなわち；

- ・ 格構造規則に対応する翻訳規則
- ・ 係り構造規則に対応する翻訳規則
- ・ GRULE, RRULE, HRULE などに対応する翻訳規則

これらは、係り受け規則と同様に、文節カテゴリや機能語、内容語の属性であり、係り受け規則の記述と対応して記述される。

- ② 翻訳基本式改変規則

機能語の属性であって、受身化や名詞化などを処理する規則

- ③ 形態素生成規則

名詞の複数形や動詞の過去形などにおける不規則変化を記述する規則。これは、目標言語の単語の属性である。

6.2.2 翻訳処理の例

簡単な例文で、前節で述べた翻訳処理の手順を説明する。

入力文 ... 「花子が学校を休んだので、太郎は悲しかった」

- ① 構文・意味解析の結果は、次のような依存構造木になる。

```

NODE1 ... 悲しい {タ}
NODE2 ... :F => 太郎
NODE3 ... $ノデ=> 休む {ノデ タ}
NODE4 ... :L => 花子
NODE5 ... .LO => 学校

```

- ② 末端のノード，NODE 4 と NODE 5 の翻訳の表現式を得る。

この場合には、「花子」も「学校」も格構造を持たないので、<空の格構造規則>に対応する翻訳規則が評価される。翻訳規則はそれぞれ、

「花子」 ... '(<N HANAKO)

「学校」 ... '(<N SCHOOL)

である。また、翻訳規則をとまなう機能語も含んでいないので、翻訳規則を評価した結果がそのまま翻訳の表現式となる。翻訳の表現式はそれぞれ、

NODE4<TL> ... (<N NODE4 HANAKO)

NODE5<TL> ... (<N NODE5 SCHOOL)

となる。(<N NODEi ...) は、名詞句を生成する関数である。第1引数の NODEi は、システムが挿入しているもので、生成関数を評価する際（翻訳の表現式の展開）に、生成関数が情報を参照するためのものである。

{以下で、生成関数の第1引数には、常にシステムが当該ノードを挿入する。}

③ NODE3 の翻訳の表現式を得る。

「休む」の格構造；

((<L) (:L 0 !OHH) (.LO 0 !HO))

に対応する翻訳規則は；

'(<VB :L BE ABSENT (<PP FROM .LO))

である。これを評価して、次の翻訳の基本式を得る。

(<VB :L BE ABSENT (<PP FROM .LO))

格役割名 :L, .LO をノード NODE4, NODE5 で置き換えて、次の翻訳の表現式を得る。

NODE3<TL> ... (<VB NODE3 NODE4 BE ABSENT
(<PP NODE3 FROM NODE5))

(<VB ...), (<PP ...) はそれぞれ、BE 動詞による文、前置詞句を生成する関数である。

④ NODE2 の翻訳の翻訳の表現式を得る。

NODE4, NODE5 の場合と同様に、NODE2 の翻訳の表現式を得る。

NODE2<TL> ... (<N NODE2 TAROU)

⑤ NODE1 の翻訳の表現式を得る。

(1) 「悲しい」の格構造；

((<F) (:F 0 !OHH))

に対応する翻訳規則は、

'(<VB :F BE SAD)

である。これを評価して、次の翻訳の基本式を得る。

(<VB :F BE SAD)

格役割 :F を、対応するノードで置き換え、基本式を具体化する。

NODE1<TL> ... (<VB NODE1 NODE2 BE SAD)

(2) NODE1 への, その他の係りの係り受け規則に対応する翻訳規則を評価する。

すなわち, NODE1 へは, NODE3 が \$ノデの関係で係っているから, その係りの規則;

(GRULE (AND (MEMQ %CF &YOUGEN) 0.8) \$ノデ 1)

に対応する翻訳規則;

(1 (TL.A (LIST “, BECAUSE” %NODE)))

を評価する。これによって, 次のような翻訳の表現式が NODE1 に付加される。

NODE1 <TL.A> ... (, BECAUSE NODE3)

翻訳の表現式には, <TL>, <TL.B>, <TL.A>, <ADV>, <ADJ>の5種類がある。<TL>が中核部分の表現式であり, <TL.B>はそれに前置する表現式, <TL.A>は後置する表現式, <ADV>は<TL>に埋め込む副詞の表現式, <ADJ>は<TL>に埋め込む形容詞の表現式である。

⑥ トップノード NODE1 の翻訳の表現式を展開する。

NODE1 <TL> := (<VB NODE1 NODE2 BE SAD)

NODE2 <TL> := (<N NODE2 TAROU)

=> (TAROU)

=> (TAROU WAS SAD)

NODE1 <TL. A> := (, BECAUSE NODE3)

NODE3 <TL> := (<VB NODE3 NODE4 BE ABSENT (<PP FROM NODE5))

NODE4 <TL> := (<N NODE4 HANAKO)

=> (HANAKO)

(<PP FROM NODE5)

NODE5 <TL> := (<N NODE5 SCHOOL)

=> (A SCHOOL)

=> (FROM A SCHOOL)

=> (HANAKO WAS ABSENT FROM A SCHOOL)

=> (, BECAUSE HANAKO WAS ABSENT FROM A SCHOOL)

=> (TAROU WAS SAD, BECAUSE HANAKO WAS ABSENT FROM A SCHOOL)

6.3 解析・翻訳実験

6.3.1 翻訳規則の記述例

図 6.3～図 6.5 に, 内容語, 機能語, 文節カテゴリの辞書記述の例を示す。関連する翻訳実験例を, 図 6.6 に示す。

図 6.3 は、内容語「重い」の辞書記述の例である。格構造規則では、「～には荷が重い」という用法を他の「重い」の用法と区別して記述している。格構造規則に対応する翻訳規則 (\$TL-FRULE) では、訳し分け (図 6.6 参照) の記述例を示した。翻訳規則中の関数 (SEMQ ...) は、意味特徴によって訳し分けするための関数である (後述)。[:~] は、[~の部分] を表す意味特徴である。また、@~ は、分類語彙表の分類コードを意味特徴として用いたものである。関数 (SEMQ ...) は、意味特徴によって訳し分けするための次のような関数である。

(SEMQ X (Y1 Y11 ... Y1i) ... (Yn Yn1 ... Ynj))

X は NIL/T/格役割名/格役割名のリストであって、

NIL なら自身のノードの、

T なら上位ノードの、

格役割名なら、その格役割名をもってそのノードに係っている下位のノードの、

格役割名のリストなら、順次にそのようなノードをたどっていった下位のノードの、

意味特徴のリストを参照する。

Yi は意味特徴に対する条件式。

Yij は任意の S 式。

関数 SEMQ は、X で指示されるノードの意味特徴リストと整合する Yi を選び、それに後続する S 式 ; Yi1, Yi2, ... を評価して、最後のものの値を返す。

ENTRY =オモイ

ICODE =オモイ

```

$STEM      ( "重" )          ... {語幹}
$KRULE     +KY              ... {活用規則}
$FRULE     (((<A) (:A 0 !0) (.AO ((P 0.3)) !0 -二))          ... {格構造規則}
             (<Z) (:Z ((P 0.4)) !0H) (.ZR ((P 0.4)) -二)))
$TL-FRULE  ((SEMQ .AO              ... {格構造規則に対応する翻訳規則}
             (:カラダ (SEMQ :A
                       (!0A !0H)
                       '<V :A FEEL (<ADJ HEAVY) IN (<POSS (<PRONX :A) .AO)))
                       (T '<VB (<POSS :A .AO) BE HEAVY))))
             (!0HP '<VB (<POSS :A .AO) BE IMPORTANT))
             (!0PB '<VB (<POSS :A .AO) BE SERIOUS))
             (@1.3046 '<VB (<POSS :A .AO) BE GRAVE))
             (T '<VB (<POSS :A .AO) BE HEAVY)))
             '<VB ((THE) BURDEN) BE HEAVY (<PP FOR :Z)))

```

図 6.3 内容語「重い」に対する辞書記述例

図 6.4 には、機能語「～化する」の辞書記述例を示した。

格構造変形規則 *VERBALIZE は、名詞の空の格構造を、次のような L 文の格構造に変形する。

((<L) (:L 0 !O) (.LO 0 !O))

翻訳式改変規則中の関数 (>#: A X) は、記号列 X に記号列 A を接合する関数である。この改変規則によって、次のように翻訳されることになる (図 6.6 参照)。

「ロボット化する」 → [:L ROBOTIZE .LO]

「ロボット化」 → [ROBOTIZATION OF .LO BY :L]

図 6.5 は、文節カテゴリ \$RENTAI に対する辞書記述例である。

\$TL-GRULE は (RRULE) に対応する翻訳規則であり、同一名詞連体修飾構造の場合の翻訳法を示している。付加名詞連体修飾の場合は、格構造規則に対応する翻訳規則で処理される。\$TL-GRULE の中の条件文は、たとえば「美しい花」を、

① A BEAUTIFUL FLOWER.

② A FLOWER WHICH IS BEAUTIFUL.

のいずれの型に訳出するかを選択するものである。ここでは、<埋め込み文の翻訳が BE 動詞による文に対応し、連体修飾名詞以外に係りがなく、「美しいのは～だ」のような文でない>場合にのみ、①のような形容詞修飾の型に訳出するように記述している。(<ANT X Y) は、関係代名詞

```
ENTRY =..カスル
ICOD=..カスル
$STEM      ( "化" )          ... {語幹}
$KRULE     +SH              ... {活用規則}
$FLINK     TO              ... {接続規則}
$TFORM     (*VERBALIZE)     ... {格構造変形規則}
$MOD       (MOD, A)        ... {機能語のサブカテゴリ}
$TL-MOD    (COND ((EQ (CAR %TLRULE) ' <N)          ... {翻訳基本式の改変規則}
                (COND ((MEMQ (_PTYPE) &TAIGEN)
                        (LIST ' <N
                              (># :IZATION (SECOND %TLRULE))
                              ' (<PP OF .LO)
                              ' (<PP BY :L)))
                ((MEMQ (_PTYPE) &YOUGEN)
                 (LIST ' <V ' :L (>#: IZE (SECOND %TLRULE)) ' .LO))
                (T %TLRULE)))
            (T %TLRULE))
```

図 6.4 機能語「...化する」に対する辞書記述例

を決定して先行詞に後置する関数である。X は被修飾名詞（先行詞）のノード、Y は連体修飾文中におけるその名詞のノードである。

翻訳文生成関数によって、目標言語での種々の表現を翻訳規則中に指示することができる。

以下に、英語への翻訳のために現在設定しているいくつかの生成関数の例について簡単に説明する。生成関数には、翻訳処理の過程で当該ノードをシステムが挿入するが、ここでは挿入される以前の形で述べる。また、格役割名は、該当するノードで置き換えられるが、ここでは、ノードに置き換えられたものとして説明する。

① 名詞句への翻訳を記述する翻訳文生成関数

- ・ (<N . noun)

```
noun:= (noun NA1 NA2 ... ) |
        ((D NB1 NB2 ... ) noun NA1 NA2 ... )
```

noun が主名詞である。NB/NA は任意の表現式であって、主名詞に前置／後置すべき語句がある場合に記述する。

D は、以下のように冠詞に関する指示を表す。

- NIL ... 冠詞をつけない。
- T ... 冠詞は<N 内のアルゴリズムで決定する。
- その他 ... D を冠詞とする。

(この他、常に単数形／複数形で用いられる名詞を表現するための類似の関数<NS/<NP を設けている。)

例：義理の母 ... (<N MOTHER IN LAW)

お湯 ... (<N (NIL (<ADJ HOT)) WATER)

```
ENTRY = $RENTAI
$DEP-MO      (COND ((MEMQ %C-F &TAIGEN) (FRULE!MRULE) (RRULE)) (T T))
$TL-GRULE    ((1 (LETS ((NODE, SUP-NOUN (SUP-NODE %NODE))
                        (NODE, EMB-NOUN (NODE, EMB-NOUN %NODE)))
                    (COND ((AND (EQ (_GFNAME) '<VB)
                                (EVERY (_SPEC) (FUNCTION (LAMBDA (X)
                                                            (OR (GET X 'ELLIP) (GET X 'RENTAI-M))))
                                (NEQ (_KWORD NODE, SUP-NOUN) '*Y/))
                            (NOUN, B (LIST '<AD (NTH 4 (_TL))))))
                    (T (TL, A (LIST (LIST '<ANT NODE, SUP-NOUN NODE, EMB-NOUN
                                        %NODE)))))))))
```

図 6.5 文節カテゴリ \$RENTAI の辞書記述例

- (<PRON pronoun)
代名詞を表現する関数。代名詞の格はノードの情報を参照して決定する。
- (<PRONX node)
node の代名詞表現を与える関数。
- (<POSS pnode nnode)
“pnode の nnode” に対応する英語の所有表現を与える関数。
例：～の顔 ... (<POSS @ FACE)
@ は「顔」に対する「～の」の格役割名である。
「～」が「彼」/「太郎」/「日本」に応じて、
HIS FACE / TAROU'S FACE / A FACE OF JAPAN となる。

② 動詞への翻訳，すなわち文への翻訳を記述する翻訳文生成関数

- (<V snode verbg . X)
一般の動詞による文を記述する。第1引数は主語，第2引数は述語である。
verbg:=verb | (verb VA1 VA2 ...) |
((VB1 VB2 ...) verb VA1 VA2 ...)
verb が主動詞であり，VB/VA は，主動詞に前置／後置すべき語句がある場合に記述する。
否定文，命令文，疑問文等の場合には語順変更等の処理を行う。
当該ノードに (ADV adverb) があれば，verbg に adverb を付加する。
これらの処理の後，各引数を評価し翻訳文を生成する。主動詞には，時制と数に関する形態素処理がなされる。
この他，BE 動詞による文を記述する関数と，その他の助動詞による文を記述する関数
(<VB ...)
(<VA ...)

がある。

③ 従属文への翻訳を記述する翻訳文生成関数

- (<THAT node) ... node を that 節に訳出する。
- (<TO|FOR node) ... node の主語に for を前置し，述部を to 不定詞句で訳出する。
- (<TO|OF node) ... node の主語に of を前置し，述部を to 不定詞句で訳出する。
- (<ING|POSS node) ... node の主語を所有格とし，述部を動名詞句で訳出する。

6.3.2 解析・翻訳実験例

図 6.6～図 6.14 に実験例を示す。翻訳実験は、英語への翻訳を中心として行い、ドイツ語と論理式風の形式言語への翻訳についても試みた。〔ドイツ語への翻訳については、中嶋による（池田，中嶋：59）。〕

図 6.6 は、図 6.3～図 6.5 に示した翻訳規則に関連した翻訳実験例である。「重い」に関する訳し分けと、ROBOT から ROBOTIZE, ROBOTIZATION への改変がなされている。

図 6.7 では、英語およびドイツ語への翻訳実験例を示した。図 6.7 の翻訳文中で、(..) とあるのは、日本語文では省略されているが、英語（ドイツ語）では省略できない要素である。省略語の回復処理は文脈処理の課題であり、本システムでは扱っていない。4 番めの例文では、前編集として括弧を挿入しているが、これを行わないと、1 番めの例文にみるように、「基礎としての」が、「基数，負数の表現法，整数の表現法を」の全体に係らないで、「基数，」のみに係ってしまう。人名は未登録語として処理されているが、その翻訳は原文中のものをそのままの形で訳出している。「松井」については、「松」が辞書に登録されていたために、「松」と未登録語「井」の複合語と解釈されている。図 6.7 の 3 文を解析・翻訳するのに要する CPU 時間は、1 秒程度である。

図 6.8 は機能語の改変規則による翻訳例、図 6.9 は準体助詞「の」に関する解析・翻訳の例である。

図 6.10 は連体埋め込み文の例であるが、日本語と英語での埋め込み文の構造の違いを無視した直訳であるために、1 番めの例文は理解困難な英語に翻訳されている。（日本語の埋め込み文は「先行詞」が後にくる左枝分かれの構造をなし、英語では先行詞が前にくる右枝分かれの構造をなす。）

入力文： 彼は頭が重かった。
 翻訳*ENG HE FELT HEAVY IN HIS HEAD .

入力文： この机は足が重い。
 翻訳*ENG A LEG OF DESK IS HEAVY .

入力文： 太郎の病気は重い。
 翻訳*ENG TAROU'S DISEASE IS SERIOUS .

入力文： 彼の地位は重い。
 翻訳*ENG HIS POSITION IS HEAVY .

入力文： 重い罪。
 翻訳*ENG THE GRAVE CRIME .

入力文： 彼には荷が重い。
 翻訳*ENG A BURDEN IS HEAVY FOR HIM .

入力文： 病気が重い太郎。
 翻訳*ENG TAROU WHOSE DISEASE IS SERIOUS .

入力文： 重い彼の病気。
 翻訳*ENG HIS SERIOUS DISEASE .

入力文： 経営者による工場のロボット化。
 翻訳*ENG A ROBOTIZATION OF THE FACTORY BY A 経営者 .

入力文： 工場をロボット化する。
 翻訳*ENG (..) ROBOTIZES THE FACTORY .

図 6.6 図 6.3～図 6.5 の翻訳規則に関連する翻訳実験例

入力文： 数値データの表現法を述べる。
 翻訳*ENG (..) PRESENTS A METHOD OF A REPRESENTATION OF NUMERAL - DATA .
 翻訳*DEU (..) SAGT EINE METHODE VON EINEM AUSDRUCK VON ZAHLENWERT - DATEN .

入力文： 基礎としての基数、負数の表現法、整数の表現法を述べたあと、固定小数点表現、有理数表現を述べる。
 翻訳*ENG AFTER (..) PRESENTED CARDINAL NUMBER AS A FOUNDATION AND A METHOD OF A REPRESENTATION OF NEGATIVE NUMBER AND A METHOD OF A REPRESENTATION OF AN INTEGER , (..) PRESENTS A REPRESENTATION OF FIXED POINT NUMBER AND A REPRESENTATION OF RATIONAL NUMBER .
 翻訳*DEU NACHDEM (..) GRUND ZAHL ALS EINE GRUNDLAGE UND EINE METHODE VON EINEM AUSDRUCK VON NEGATIVE ZAHL UND EINE METHODE VON EINEM AUSDRUCK VON GANZE ZAHL SAGTE , SAGT (..) EINEN AUSDRUCK VON DEM FIXEN DEZIMALBRUCH UND EINEN AUSDRUCK VON RATIONALE ZAHL .

入力文： 浮動小数点表現については、最新の研究成果である松井・伊理による表現、浜田による表現にも言及する。
 翻訳*ENG ON A REPRESENTATION OF FLOATING POINT NUMBER , (..) MENTIONS OF A REPRESENTATION BY PINE - 井 AND 伊理S AND A REPRESENTATION BY A 浜田 THOSE ARE THE RESULT OF THE NEWEST INVESTIGATION .
 翻訳*DEU VON EIN AUSDRUCK VON SCHWANK DEZIMALBRUCH , BERUSHREN (..) EINEN AUSDRUCK VON 伊理E UND AUSDRUCK VON 浜田 , DIE DAS RESULTAT VON AM NEUEST FORSCHUNG SIND .

入力文： 基礎としての（基数、負数の表現法、整数の表現法を）述べたあと、固定小数点表現、有理数表現を述べる。
 翻訳*ENG AFTER (..) PRESENTED A METHOD OF A REPRESENTATION OF CARDINAL NUMBER AND NEGATIVE NUMBER AND A METHOD OF A REPRESENTATION OF AN INTEGER AS A FOUNDATION , (..) PRESENTS A REPRESENTATION OF FIXED POINT NUMBER AND A REPRESENTATION OF RATIONAL NUMBER .
 翻訳*DEU NACHDEM (..) EINE METHODE VON EINEM AUSDRUCK VON GRUND ZAHL UND NEGATIVE ZAHL UND EINE METHODE VON EINEM AUSDRUCK VON GANZE ZAHL ALS EINE GRUNDLAGE SAGTE , SAGT (..) EINEN AUSDRUCK VON DEM FIXEN DEZIMALBRUCH UND EINEN AUSDRUCK VON RATIONALE ZAHL .

図 6.7 英語およびドイツ語への翻訳実験例
 (入力文は、「情報処理」vol. 24, No. 4, pp 349 からとった)

入力文： 鼠はチーズが食べたかった。
 (\$SYUSHI JN3022) ヲハル
 (:F JN3020) ネス"ミ
 (.FO JN3021) チ ス"
 翻訳*ENG A RAT HOPED TO EAT CHEESE .

入力文： 鼠はチーズを食べそこなった。
 (\$SYUSHI JN3025) ヲハル
 (:L JN3023) ネス"ミ
 (.LO JN3024) チ ス"
 翻訳*ENG A RAT HAD FAILED TO EAT CHEESE .

入力文： 鼠がチーズを食べかけた。
 (\$SYUSHI JN3028) ヲハル
 (:L JN3026) ネス"ミ
 (.LO JN3027) チ ス"
 翻訳*ENG A RAT WAS ABOUT TO EAT CHEESE .

図 6.8 機能語の改変規則に関する翻訳実験例

```

入力文： 鼠が食べたのはチーズだった。
($SYUSHI JN3057) *チ`アル
  (:R JN3055) *Yノ
    (=R JN3054) タ`ル
      (.LO RENTAI-M) *Yノ
        (:L JN3052) ネス`ミ
          (.RI JN3056) チ ス`
翻訳*ENG IT WAS CHEESE THAT A RAT HAD EATEN .

入力文： 猫が殺したのを食べたのは鼠だ。
($SYUSHI JN3238) *チ`アル
  (:R JN3237) *Yノ
    (=R JN3235) タ`ル
      (.LO JN3233) *Yノ
        (=R JN3225) コロス
          (.LO RENTAI-M) *Yノ
            (:L JN3223) ネコ
              (:L RENTAI-M) *Yノ
                (.RI JN3230) ネス`ミ
翻訳*ENG IT IS A RAT THAT ATE WHAT A CAT KILLED .

```

図 6.9 準体助詞「の」に関する解析・翻訳実験例

```

入力文： 猫が殺した鼠が食べたチーズは腐っていた。
($SYUSHI JN3275) クサル
  (:P JN3274) チ ス`
    (=R JN3273) タ`ル
      (.LO RENTAI-M) チ ス`
        (:L JN3272) ネス`ミ
          (=R JN3271) コロス
            (.LO RENTAI-M) ネス`ミ
              (:L JN3270) ネコ
翻訳*ENG CHEESE THAT A RAT THAT A CAT KILLED ATE WAS ROTTEN .

入力文： 腐ったチーズを食べた鼠を猫が殺した。
($SYUSHI JN3283) コロス
  (.LO JN3281) ネス`ミ
    (=R JN3280) タ`ル
      (:L RENTAI-M) ネス`ミ
        (.LO JN3279) チ ス`
          (=R JN3278) クサル
            (:P RENTAI-M) チ ス`
              (:L JN3282) ネコ
翻訳*ENG A CAT KILLED A RAT THAT ATE ROTTEN CHEESE .

```

図 6.10 連体埋め込み構造に関する解析・翻訳実験例

図 6.11 は複合文の解析・翻訳の例である。図 6.12 には、解析木のノードの内容の例として、図 6.11 の最初の例文の解析木のノード (JN0059～JN0065) の内容を示した。

図 6.13 は、エキスパートシステムのルール (機械装置類の故障診断システム PRODOG の故障診断ルール) を、日本語文から翻訳・生成してみたものである⁵⁵⁾。たとえば、「吸い込み口での空気の温度」は、故障診断システムのパラメータ TIN に翻訳されている。図 6.14 に「温度」の辞書記述を示した。図 6.14 中の ;~, .~, は、~がそれぞれ内容語、機能語の内部辞書ベクトル (第 7 章参照) 中のシンボルであることを記述するものである。

入力文： 鼠がチーズを食べたから猫がその鼠を殺した。
 (\$SYUSHI JN0065) コロス
 (*タカラ JN0062) タハル
 (:L JN0059) ネスミ
 (.LO JN0060) チス
 (:L JN0063) ネコ
 (.LO JN0064) ネスミ
 (-ソノ JN0061) ソノ
 翻訳*ENG A CAT KILLED THIS RAT , BECAUSE A RAT HAD EATEN CHEESE .

入力文： 鼠がチーズを食べてから猫がその鼠を殺した。
 (\$SYUSHI JN0072) コロス
 (*シテカラ JN0069) タハル
 (:L JN0066) ネスミ
 (.LO JN0067) チス
 (:L JN0070) ネコ
 (.LO JN0071) ネスミ
 (-ソノ JN0068) ソノ
 翻訳*ENG A CAT KILLED THIS RAT , AFTER A RAT HAD EATEN CHEESE .

図 6.11 複合文に関する解析・翻訳実験例

```

NODE=JN0065
TL!      (A CAT KILLED THIS RAT , BECAUSE A RAT HAD EATEN CHEESE)
TENSE    PAST
NUMBER   SINGLE
PERSON   P3
TL.A     ((, BECAUSE JN0062))
TL.      (<V JN0065 JN0063 KILL JN0064)
GFNAME   <V
TL        T
ASPECT.E NIL
TENSE.E  PAST
TENSE.J  (KAKO)
ASPECTS  (STATE- ACT+ CONT-)
TIME*    (タ)
SPEC     (JN0062 JN0063 JN0064)
RTYPE    $SYUSHI
SEM      (コロス I.E STATE- ACT+ CONT- < <L)
NO.FRULE 0
KWORLD   コロス
FWORDS   (タ)
GNESS    +0.992 +00
MOD.T     タ
BRULE    NIL
PTYPE    $SYUSHI
STYPE    <L
TFORM    NIL
BSTYPE   <L

NODE=JN0063
NUMBER   SINGLE
PERSON   P3
TL!      (A CAT KILLED THIS RAT)
TL.MG    T
CASE     SUBJ
TL.      (<N JN0063 CAT)
TL        T
SPEC     NIL
SUP      (JN0065)

```

図 6.12 解析木のノードの内容 (図 6.11 の最初の例文の解析木のノード)

```

RTYPE      :L
SEM        (ㇿㇿ !OA !O ! <?)
NO.FRULE   0
KEYWORD    ㇿㇿ
FWORDS     (か")
GNESS      +0.100 +01
BRULE      FRULE
PTYPE      $T>Y
STYPE      <?
TFORM      NIL
BSTYPE     <?

NODE=JN0064
TL!        (THIS RAT)
TL.MG      T
DET        THIS
TL.        (<N JN0064 RAT)
TL         T
SPEC       (JN0061)
SUP        (JN0065)
RTYPE      .LO
SEM        (ㇿㇿㇿㇿ !OA !O ! <?)
NO.FRULE   0
KEYWORD    ㇿㇿㇿㇿ
FWORDS     ( )
GNESS      +0.950 +00
BRULE      FRULE
PTYPE      $T>Y
STYPE      <?
TFORM      NIL
BSTYPE     <?

NODE=JN0061
TL.MG      T
TL.        NIL
TL         T
SPEC       NIL
SUP        (JN0064)
RTYPE      -ツノ
SEM        (ツノ <?)
NO.FRULE   0
KEYWORD    ツノ
FWORDS     NIL
GNESS      +0.100 +01
BRULE      (GRULE ツノ . 1)
PTYPE      $F>T
STYPE      <?
TFORM      NIL
BSTYPE     <?

NODE=JN0062
TL!        (A RAT HAD EATEN CHEESE)
TENSE      PAST
NUMBER     SINGLE
PERSON     P3
TL.MG      T
TL.        (<V JN0062 JN0059 EAT JN0060)
GFNAME     <VA
TL         T

```

図 6.12 解析木のノードの内容 (続き)

```

TENSE.E   PAST
ASPECT.E  PERFECT
TENSE.J   (NIL . MAE)
ASPECTS   (STATE- ACT+ CONT+ COMP+)
TIME*     (タ . タ)
SPEC      (JN0059 JN0060)
SUP       (JN0065)
RTYPE     *タ"カラ
SEM       (タ^"ル I.C STATE- ACT+ CONT+ COMP+ < <L)
NO.FRULE  0
KWORLD    タ^"ル
FWORDS    (タ カラ1)
GNESS     +0.100 +01
MOD.C     カラ1
MOD.T     タ
BRULE     (GRULE カラ1 . 1)
PTYPE     $Y>Y
STYPE     <L
TFORM     NIL
BSTYPE    <L

```

```

NODE=JN0059
NUMBER    SINGLE
PERSON    P3
TL!       (A RAT HAD EATEN CHEESE)
TL.MG     T
CASE      SUBJ
TL.       (<N JN0059 RAT)
TL        T
SPEC      NIL
SUP       (JN0062)
RTYPE     :L
SEM       (ヲス"ニ !0A !0 ! <?)
NO.FRULE  0
KWORLD    ヲス"ニ
FWORDS    (カ")
GNESS     +0.100 +01
BRULE     FRULE
PTYPE     $T>Y
STYPE     <?
TFORM     NIL
BSTYPE    <?

```

```

NODE=JN0060
TL!       (CHEESE)
TL.MG     T
TL.       (<NS JN0060 (NIL) CHEESE)
TL        T
SPEC      NIL
SUP       (JN0062)
RTYPE     .LO
SEM       (ヲ ス" !00P !00 !0 ! !000 <?)
NO.FRULE  0
KWORLD    ヲ ス"
FWORDS    ( )
GNESS     +0.100 +01
BRULE     FRULE
PTYPE     $T>Y
STYPE     <?
TFORM     NIL
BSTYPE    <?

```

図 6.12 解析木のノードの内容 (続き)

入力文： 吸い込み口での空気の温度をXとし、吹き出し口での空気の温度をYとすれば、温度落差はXからYを引いたものである。
 翻訳*LOG (IF (AND (BE TIN *X) (BE TOUT *Y)) (BE TDROP (- *X *Y))) .

入力文： 装置が全く動かないわけではなく、送風系が異常でもなく温度差が7度以下なら熱ポンプ系異常である。
 翻訳*LOG (IF (AND (ONWORK SYS LITTLE) (NOT (ABN FSS)) (LE TDROP 7)) (BE (..) (#ABN HSS))) .

入力文： 装置が全く動かないわけではなくコンプレッサの作動音がして吹き出し風がなければ送風系異常である。
 翻訳*LOG (IF (AND (ONWORK SYS LITTLE) (ZZ MMS) (NOT (EXIST AIR))) (BE (..) (#ABN FSS))) .

図 6.13 論理式風の形式言語への翻訳例

```
ENTRY= オント`
ICODE= オント`
$STEM (温度)
$KRULE /+T
$SEM (!AFO)
$FRULE (((<?) (#_P O !P) (@A O !O)))
$DEP-F (COND ((AND (EQ %BA-M 'ノ)) (UNIT-MATCH %KW-F %AP-M)) (GRULE +0.100 +01 @NUM 1)))
$TL-FRULE ('<N (THE) TEMPRETURE (<PP OF @A) (<PP AT #_P)))
$TL-GRULE (<1 (TL.A (LIST (" %NODE ""))))
$TL-FRULE*L ((SEMQ #_P (ノスイコミク`チ 'TIN) (ノフキタ`シク`チ 'TOUT) (<ノカ`イキ `ソト) 'TAIR) (T 'TEMP)))
```

図 6.14 「温度」の辞書記述

6.4 まとめ

構文・意味解析の結果を機械翻訳に応用し、前章までに述べた解析システムの有効性を確認した。本章では、この機械翻訳の方式について述べ、解析・翻訳の実験結果を示した。

機械翻訳の手法は、目標言語での表現構造（翻訳の表現式）へ変換するフェーズと、表現式を展開して翻訳文を生成するフェーズからなるトランスファー方式である。従来のトランスファー方式では、構造の変換は同じレベルの構造への変換を行っているが、本システムの翻訳の表現式は、目標言語の表層の表現に近い構造のものである。これは、係り受け規則と対応して記述してある翻訳規則から得られるもので、係り受け解析の結果から、外部規則と照合することなく直接に得られる。翻訳規則を個々の語および文節カテゴリの属性として、解析の諸規則と対応づけて設定したことによって、日本語の用法と目標言語の用法を直感的に対応づけることができ、さまざまな表現への訳し分けも随意に行うことができた。また、翻訳文生成関数を設計し、翻訳規則を書き分ければ、種々の目標言語に適用できることを確認した。

翻訳処理は、外部規則と照合することなく、解析結果の依存構造木から得られる翻訳規則を評価することによって行われるから、翻訳システムは簡潔に構成でき、処理も高速である。翻訳部のプログラムは、リスプ (UTILISP) で約 400 行、翻訳文生成関数と翻訳規則を記述するための諸関数

(英語版の場合)で、現在約600行程度である。処理速度は、解析と翻訳を含めて、1文1秒のオーダーである。

この翻訳実験をつうじて、構文・意味解析の結果の有効性を確認することができた。翻訳システムとしては、

- ① 基本的に、日本語の構造を保ったままの直訳であるために、たとえば、

「激しい雨が降っている」

を、

「激しく雨が降っている」... “It is raining heavily”

のように翻訳するためには、英語的発想に基づく構造への変換規則を個々にプログラムしなければならない。翻訳システムとしては、発想の違いによる表現構造の違いに、組織的に対処する方法の開発が重要である。

- ② 文脈処理まで含んだ完全な解析でないために、省略語の補完を行っておらず、また冠詞の扱いも不十分である。
- ③ 目標言語での省略や代名詞化などの処理がなされていない。

などの問題を残している。従来の機械翻訳システムにおいても、①については若干の研究があるが^{81),82),94)}、②、③については解決されていない。これらは、自然言語理解研究における今後の課題である。

第7章 辞書データベースと実験環境

7.1 はじめに

自然言語処理の研究・開発においては、辞書や文法規則などのデータをあらかじめ完全な形で与えておくことは非常に困難であり、データの追加・修正と実験による検証を積み重ねることが必要である。そこで、辞書データベースのエディタや簡単に実験が行えるシステム環境が重要となる^{68),76)}。

本システムはリスプ (UTILisp⁶⁹⁾) で記述しているが、従来のリスプは順編成ファイルを扱うのみで、ランダムにファイルにアクセスできず、データベースを扱うには不便があった。そこで本システムでは、UTILisp からキー順ファイルに直接アクセスできるように機能を拡張し、次のような実験環境を整えた (図 7.1)。

① リスプ (UTILisp) から直接キー順ファイルにアクセスできるようにした。

VSAM キー順ファイルでは、キー (見出し) を指定することで、ランダムに、あるいは部分的にシーケンシャルに、高速にデータにアクセスすることができる。

これによって、リスプでも大規模なデータベースを容易に扱うことができるようになった。

② キー順ファイル上の S 式に対するエディタおよび辞書エディタを作成し、プログラムおよび辞書データ・規則類の簡便な管理を実現した。

すべての辞書データ・規則類は、1 個のキー順ファイル上で 1 元的に管理できる。

③ エディタ上で解析システムや翻訳システムを起動でき、容易に実験が行える。

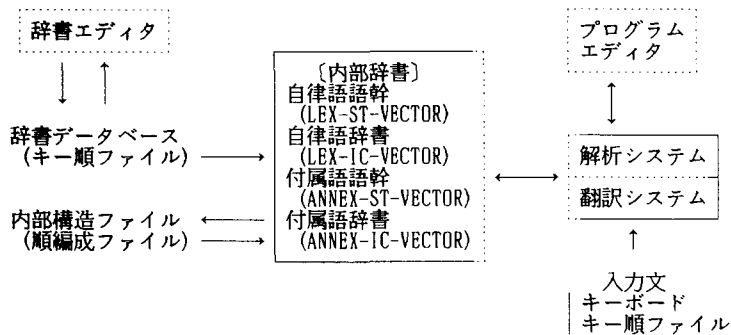


図 7.1 辞書とシステムの構成

7.2 キー順ファイルへのアクセス

VSAM(Virtual Storage Access Method) は、ディスク装置と主記憶装置間のデータの転送を行う OS のサービス群であるが¹⁰²⁾、VSAM が扱うファイルのひとつにキー順ファイルがある。キー順ファイルのレコードはキーをもち、各レコードはキーの昇順に配列されている。VSAM は、キー順ファイルに対して、キーを指定することによって直接そのレコードにアクセスしたり、指定されたキーを始点として順次にアクセスするなど種々のアクセス法を提供している。キー順ファイルは、辞書データベースのような大規模なデータベースを処理するのに極めて有用である。

本システムでは、キー順ファイルにリスプから直接アクセスできるように、一連のリスプの組込関数を追加した。この機能拡張には、電子技術総合研究所で COSMO 700 上に開発されたリスプ系言語 PETL(塚本：85) を参考にした。追加した関数を表 7.1 に示す。

アクセス・モードには直接アクセス、スキップ順アクセス、順アクセスの 3 種がある。また、代替インデクスを作成して、代替キーによってアクセスすることもできる。直接アクセスおよびスキップ順アクセスの場合には、次のようにキーとエリア番号を指定してレコードにアクセスする。

(PUTDB stream key area value) ... 書き込み

(GETDB stream key area) ... 読み込み

表 7.1 キー順ファイルにアクセスするための関数

| ファイルのオープン、クローズなど | |
|------------------|--------------------------------|
| STREAMDB | VSAM用ストリームの作成 |
| STREAMDBP | VSAM用ストリームの判定 |
| INOPENDB | オープン (入力モード) |
| INOUTOPENDB | オープン (入出力モード) |
| OUTOPENDB | オープン (出力モード) |
| CLOSEDB | クローズ |
| MODEDB | モードの設定 |
| 直接アクセス、スキップ順アクセス | |
| PUTDB | S 式の出力 |
| GETDB | S 式の入力 |
| RENOB | キーの削除 |
| KEYDB | キーの確認 |
| GETDBA | S 式の入力 (代替キーの場合) |
| GETDBN | 同上 (同じ代替キーをもつものが複数あるとき次のものを入力) |
| KEYDBA | キーの確認 (代替キーの場合) |
| 順アクセス | |
| PRINTDB | S 式の出力 |
| READDB | S 式の入力 |
| DELOB | キーの削除 |
| その他 | |
| POINTDB | 順アクセス、スキップ順アクセスの場合の開始キーの位置付け |
| POINTDBA | 同上 (代替キーの場合) |
| ENDDB | 順アクセス、スキップ順アクセスの終了 |

key は数値/ストリング/シンボル (漢字も可) であり、area は 0~255 の整数である。VSAM がアクセスする際の実際のキーは、エリア番号 (1 バイト) と key のタイプ (1 バイト) および key から合成される。キー長はファイル作成時に決まる。

7.3 辞書データ・ベースの構造

7.3.1 キー順ファイル上の構造

解析・翻訳システムで用いる辞書データの種類を、表 7.2 にまとめた。これらのデータは、すべて見出しをキーとして、

((属性1 属性値1)(属性2 属性値2) ...)

の形の属性-属性値の集合として、辞書ファイルに記述されている。各辞書毎に使用するエリアを定めており、いくつかの辞書属性毎に異なるエリアに記述しておくこともできる (表 7.3)。

辞書には語彙型の辞書と規則型の辞書がある。語彙型の辞書は、同形異語を区別する必要から設

表 7.2 辞書データベース

| 辞書名 | 見出し | 属性 |
|----------|---------|-------------------------------|
| DEF-DICT | 辞書名 | 辞書タイプ、辞書属性、重複禁止エリア |
| CONST | 定数名 | 定数 |
| SUPSEM | 意味特徴名 | 上位の意味特徴 |
| KRULE | 活用型名 | 活用表 |
| LRULE | 接続型名 | 接続表 |
| CRULE | 活用形名 | 文節カテゴリ規則 |
| PRULE | 文節カテゴリ名 | 係り受け規則 (...)、翻訳規則 (...) |
| TRULE | 変形規則名 | 格構造変形規則 |
| ARULE | 格役割名 | 機能語条件 |
| ANNEX | 付属語 | 語幹、活用型、接続型、意味特徴、整形規則 |
| TAIGEN | 体言 | 変形規則、係り受け規則 (...)、翻訳規則 (....) |
| YOUGEN | 用言 | 語幹、活用型、意味特徴、 |
| SONOTA | その他の自立語 | 係り受け規則 (...)、翻訳規則 (...) |
| EVERB | 英語動詞 | 同上 |
| ENOUN | 英語名詞 | 過去、過去分詞、... (不規則変化) |
| EPRONOUN | 英語代名詞 | 複数形、... (不規則変化) |
| DVERB | 独語動詞 | 目的格、所有格、... (不規則変化) |
| ... | ... | 過去、過去分詞、... (不規則変化) |

表 7.3 キー順ファイル上のデータ

| エリア | キー | 内容 |
|-----|-----|--|
| 50 | イク | ((ICODES イク)) |
| 51 | イク | ((STEM "行" "い")) |
| 52 | イク | ((KRULE +IKU) (\$SEM <T ...) (\$PRULE (<<T) (:T ...) (TT ...)))) |
| 53 | イク | ((MRULE ...) (\$DEP-M ...)) |
| 60 | イク | ((STL-FRULE*ENG .. GO..) (\$TL-MRULE*ENG ...)) |
| 61 | イク | ((STL-FRULE*GER .. FAHREN..) (\$TL-MRULE*GER ...)) |
| ... | ... | ... |

けたもので、同類の語を代表する外部見出しと、内部で区別するための内部見出しをもつ。外部見出しには、

((ICODES IC1 IC2 ...))

の形で、その類に属する語の内部見出し IC1, IC2, ... を、属性 ICODE の値として記述してある。語の諸属性は内部見出しの属性として記述される。

エリア 0 には、辞書ファイルを管理するための情報が記述されており、後述する辞書エディタを通じて辞書ファイルは管理される。

このようにして、すべての辞書属性を1個のキー順ファイル上で1元的に管理することができる。内容的には、基本的な規則類と付属語を内容とする核辞書、基本的な自立語を内容とする基本語辞書、特別な分野のための辞書、実験用の一時的な辞書などが区別されるから、これらは別々のファイルに分けておくのが实际的である。

7.3.2 内部記憶上の構造

自立語および付属語の語幹と内部見出しは、通常のオブジェクト・ベクトルとは別のオブジェクト・ベクトル (LEX-ST-VECTOR, LEX-IC-VECTOR, および ANNEX-ST-VECTOR, ANNEX-IC-VECTOR) に登録してある (図 7.2)。

語幹には、その値として、その語幹をもつ語の内部見出しのリストが設定してある。

内部見出しには、辞書の内容がベクトル値として、あるいは属性リスト上に表現してある。ほとんどの見出しに共通する属性はベクトル値として、そうでないものは属性リストで表現しメモリの効率的使用を図っている。ベクトル値として表現するか属性リストとして表現するかは、辞書読み込み関数の引数で簡単に指定できる。

内部見出しに結合される辞書属性は、一般にいくつかのエリアに分散して記述されているので、キー順ファイルからの辞書の読み込みは、直接アクセス・モードで行なっている。このために、大きな規模の辞書の場合には読み込みに時間がかかる。そこで、キー順ファイルから読み込んだ辞書の内部記憶上の構造を、順編成ファイルにはき出しておいて (内部構造ファイルとよぶ)、そこから読み込むモードも用意してある (図 7.1)。

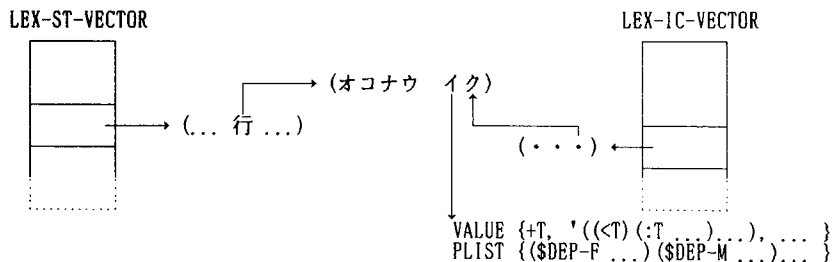


図 7.2 辞書情報の内部構造

7.4 辞書エディタと実験環境

キー順ファイルのデータを編集・管理するために、S 式を編集する一般的なエディタ (KEDIT) と、辞書を編集するエディタ (DEDIT) を開発した。いずれも S 式を編集するレベルでは、UTILisp のライブラリとして開発されている、S 式に対する構造エディタ AMUSE¹⁸⁾を用いている。

(1) KEDIT は、キーを指定して AMUSE を呼び出す、キーを削除する、エリア間の移動をする、あるいは、キー順ファイルを確保する、2 つのキー順ファイルをマージする、順編成ファイルに清書出力するなどのコマンド群をとりそろえたものである。

プログラムの開発には KEDIT を用いた。通常、関数名をキーとして、エリア毎にひとまとまりのプログラムを格納している。同名の関数で、版の異なるものを別のエリアに保存しておくことができるなど、プログラムの開発に便利である。

(2) DEDIT は、エリア 0 に記述してある辞書管理情報を用いて辞書を編集する。

図 7.3 は、この辞書管理情報の例である。キーは辞書名である。

属性 ¥DTYPE の属性値は、

(LEX/RULE area)

であり、LEX/RULE で、語彙型の辞書であるか、規則型の辞書であるかを記述する。area は、その辞書のどのエンタリも持っている属性が入っているエリアである。(エンタリが既出のものであるかをみたり、エンタリをリストイングする際に必要となる。)

属性 ¥PROP の属性値は、

(prop area condition)

のリストであり、prop は、その辞書のエンタリが持つ辞書属性の名前である。area は、その辞書属性を格納するエリアであり、condition は、その辞書属性の記述が満足すべき条件を表現する関数である。DEDIT は、編集された辞書記述をファイルに格納する際に condition を評価する。値が NIL になるなら、記述の形式に誤りがあるとみなして警告を出し、編集モードに再入する。この機能によって、辞書記述の形式的な誤りを防ぐことができる。

属性 ¥AREAS-R の属性値はエリア番号のリストである。これは、新しくエンタリを登録する場合に、エンタリが重複してはならない他の辞書のエリア番号である。DEDIT は、重複していれば警告を出す。

属性 ¥COMMENT には、任意の S 式をコメントして記述できる。

エリア 0 内の記述自身も規則型の辞書の 1 つ (DEF-DICT, 図 7.4) である。従って、DEF-DICT がエリア 0 に最初に記述してあれば、DEDIT を使ってエリア 0 の内容を編集することによって、任意の辞書および辞書属性を辞書ファイル上に容易に登録することができ、辞書属性の追

加・変更にも簡単に対処することができる。

DEDIT では、編集する辞書ファイルの指定、親ファイルの指定、辞書名の指定の後、エントリ (外部見出し) を指定するコマンド、語彙型の辞書の場合は内部見出しを指定するコマンド、辞書属性を指定するコマンドを用いて編集するレコードを定め、AMUSE を用いて編集する。親ファイルの参照・コピー、編集対象とする辞書属性の指定、記述内容の一括表示、辞書名やエントリのリスティング、削除、改名、他のエントリの記述の表示やコピー等のコマンドも用意してある。

DEDIT では編集後、そのままエディタ上でそのエントリの記述を内部構造に読み込み、解析システムや翻訳システムを起動して実験することができる。

```

キー: YOUGEN
内容: ((¥DTYPE LEX 51)
      (¥PROPS ($STEM 51 {ストリングのリスト})
              ($KRULE 52 {+ で始まるアトム})
              ($TL-FRULE*ENG 60 {...})
              ($TL-FRULE*GER 61 {...})
              ...))
      (¥AREAS-R 101 151)
      (¥COMMENT ...))

```

図 7.3 用言辞書に対する管理情報

```

キー: DEF-DICT
内容: ((¥DTYPE RTYPE 0)
      (¥PROPS (¥DTYPE 0 {第1引数はLTYPE またはRTYPE }
                  {第2引数は 1~256 の整数})
              (¥PROPS 0 {第1項がアトム、第2項が 1~256 の整数、
                          第3項が関数のbodyであるもののリスト}
              (¥AREAS-R 0 { 1~256 の整数のリスト})
              (¥COMMENT 0 T))
      (¥AREAS-R)
      (¥COMMENT 辞書管理情報の辞書))

```

図 7.4 辞書管理辞書に対する管理情報

7.5 まとめ

本章では、辞書データベースの扱いなど、本システムで整備した解析・翻訳の実験環境について述べた。

リスプからキー順ファイルにアクセスできるように、機能を拡張した。さらに、キー順ファイル上のプログラムや、辞書データベースに対するエディタを作成した。プログラムの管理や辞書データの管理、入力文の管理などを、すべてキー順ファイルをベースとして行うことにより、簡易で適応性に富む実験環境が実現できた。

システムは、FACOM-M 380 上で稼働している。システムのサイズは、

解析部本体 (約 140 KBYTE), 辞書入出力等ユーティリティ (約 30 KBYTE)

翻訳部本体 (約 40 KBYTE), 生成関数など (約 60 KBYTE)

エディタ (約 150 KBYTE)

辞書 (約 1200 KBYTE) ... 機能語約 400 語, 自律語約 2000 語,
その他文節カテゴリ, 意味特徴など

である。

第8章 結 論

8.1 研究結果の概要

本研究では、係り受け解析による日本語文の構文・意味解析の手法について提案して、計算機上に具体化し実験した。また解析結果を応用した機械翻訳システムを作成し、この係り受け解析の有効性を確かめた。

日本語は、英語などと比べると意味主導性の強い構造をしている。したがって、専ら統語関係を表現する句構造モデルよりも、意味関係をも表現する依存構造モデル(係り受け構造モデル)によって、構文関係と意味関係を同時に処理していくほうが日本語の姿に合っている。日本語文の解析システムでは、従来から格構造を用いた意味解析がよく行われてきたが、構文解析と意味解析を一体化した統一的な係り受け解析方式の枠組みには至っていなかった。本研究は、このような観点から、統一的な係り受け方式による日本語文の構文・意味解析システムの提案を行ったものである。

解析システムは、文節構造を解析する部門と、係り受けの構造を解析する部門から構成した。係り受け解析は、*shift-reduce-parsing*を基本的な構図として、係り受け規則の適合度に基づいて“最も良い係り受け関係”を迫及する方式(弛緩分析)をとった。係り受け規則は、構文上および意味上の係り受けの制約条件を記述するもので、文節カテゴリ、機能語および内容語の属性として一連の係り受け規則を設けた。格構造は用言だけでなく任意の自律語がもつ属性として一般化した上で、係り受け規則の一種として位置づけた。文節構造部では整形規則を設け、擬語の導入などによって入力文の表層の文節構造(自律語+付属語)から係り受け解析に適合した深層の文節構造(内容語+機能語)に整形する手法を導入した。

このようにして、係り受け解析方式によって一貫した、簡潔な構文・意味解析システムを構成することができた。構文関係と意味関係を融合した係り受け規則によって、日本語における言語規則を直感的な自然な形で記述することができた。第5章では係り受け規則の記述例を示し、連体埋め込み構造や数量表現、種々の呼応関係などの解析も、この枠組みの中で統一的に処理できることを示した。また、弛緩分析によって、意味記述と完全には整合しない文の解析も可能となり、同時に係り受け規則の記述に柔軟性をもたせることができた。

本システムではまた、すべての規則を文節カテゴリおよび語の属性として記述する方式をとった。書き換え規則のような規則表示を用いる方式では例外処理が複雑化するが、この方式によって、個別的な言語事実も素直に記述することができた。また、解析システムは、入力文そのものから発生してくる諸規則を適宜引き出して評価するという形で、コンパクトに構成することができ

た。同時に、多数の外部規則と照合するというようなフェーズを含まないために、高速な処理を実現することができた。(システムはリスプ (UTLisp) で記述している。解析システムの本体は約 140 Kbyte。解析・翻訳に要する CPU 時間は 1 文 1 秒の程度である。)

また、解析結果を機械翻訳に応用し、この係り受け解析の有効性を確かめた。第 6 章では、この機械翻訳の手法について述べ、解析・翻訳の実験例を示した。翻訳方式としては、係り受け規則と対応した翻訳規則による方式を開発し、簡潔な翻訳システムを構成した。

以上のような、解析・翻訳システムを構成し、実験するために、キー順ファイルをベースとした簡便な支援環境を整備した。第 7 章で、リスプからのキー順ファイルへのアクセス、辞書の構造、辞書エディタなどについて述べた。

8.2 今後の課題

本システムでは、意味特徴を用いて意味上の制約を記述したが必ずしも十分な記述手段とはなっていない。意味関係の記述が不十分であると、構文的には有り得ても意味的には間違っただけの解析結果を出してしまうことが多い。本システムでは構文と意味を同時に解析する方式をとったが、このことは、構文解析をした後で、その解析結果の中から意味的に整合したものを取り出すという方式においても事情は全く同様である。広範な領域に渡っての一般的な意味の記述は、意味特徴ではなく意味ネットワークやフレーム等の手段を用いたとしても依然として困難な問題であり、今後の研究が待たれる⁹⁸⁾。

また、今後の大きな課題としては文脈解析の問題がある⁹⁹⁾。本解析方式では、隣接している部分間の関係だけを見て、それより以前に発話されている部分、すなわちスタック内の奥の要素との関連を見ていないという問題があるが、この問題はさらに、以前に発話された文との関連の解析、すなわち文脈解析のありかたとも関連して解決すべきものと考えられる。構文・意味解析と統合した文脈解析の方式を確立しなければならない。係り受け規則は文脈情報との関連をも記述するものでなければならず、それは文脈情報による制御のもとに評価されるべきものであろう。それによって、解釈の多義性の問題も小さな問題となり、より簡潔なシステムを構成できるはずだと考える。

文脈解析には次の 2 つの問題がからみあっている。

① 文に現れる語が指しているものは何かという同定処理の問題。

これには、「この」、「その」、「これ」、「それ」などの指示語が指しているものは何かという明らかな参照の問題だけでなく、「犬が走っている」における「犬」と「犬は動物だ」における「犬」が指しているものは違うというように、普通名詞が指している対象を同定する問題、0 代名詞すなわち省略されているものが指している対象を同定する問題がある。これは、自然言語『理解』における本質的な問題である。

② 文と文の関連 (coherency) を見出し、話題の展開の構造すなわち文章構造を明らかにする。複

合文の中の単文間の関連の問題もこの一部である。

これには、主題や焦点は何か、話者の視点はどこにあるかといった問題が含まれる。

文脈理解の問題は、世界に関して人間がもっている様々の知識を使い、推論し、文の意味を読み取るということである。定理証明、曖昧論理、非単調論理など、推論処理の研究も重要である。文脈理解の問題は、本質的には人間の記憶の構造そのものの研究とつながる深い問題であり、今後の大きな課題である。概念依存構造や MOPs などによる R. Schank 等の研究が注目される^{22),23)}。

文脈解析の研究としては、言語外の知識のありように比重をおいた研究の他に、1文の範囲を超えた言語テキスト上に観察される文法規則（談話文法）の研究がある⁷⁹⁾。談話文法については、言語学の側でも未だ研究は少ないが、今後の重要な研究課題である。{(池田：62)は、日本語の名詞文「AはBだ」の解釈に関連する談話規則について考察している。}

自然言語『理解』への挑戦には、このような文脈解析の問題の他、言語学習の問題など、難問が山積している。今後の課題としては、また、自然言語処理を実用化するための技術的な面での研究も重要であり、本システムで導入した括弧による前編集のような補助手段や制限文法の研究など^{92),108)}、人間と機械が協調して処理してゆく方向での研究も重要である。

謝 辞

本研究は、筆者が電子技術総合研究所において行った「日本語文の構文・意味解析システムに関する研究」をまとめたものである。

本研究をまとめるにあたっては、手塚慶一・大阪大学教授から懇切なる御指導と御鞭撻を賜った。謹んで深謝の意を表します。また、種々の御指導を賜った同大学・中西義郎教授、倉菌貞夫教授、北橋忠宏教授に深謝の意を表します。

本研究を遂行するにあたって、電子技術総合研究所・佐藤孝平所長(元制御部長)には、研究の初期から終始変らぬ励ましと暖かい御支援を戴いた。白井良明・制御部長、若松清司・前制御部長(現新エネルギー総合開発機構)には、適切な御指導と励ましを戴いた。これらの方々に深く感謝致します。

本研究は後期においては、同所特別研究「自然言語情報処理技術に関する研究」において行ったものであり、同テーマを指導された中島隆之・パターン情報部長、また石崎俊・同部推論システム研究室長、田中穂積・前推論システム研究室長(現東京工業大学)には種々の御支援を戴いた。また、柏木寛・電子計算機部長には、「日英科学技術文献の速報システムに関する研究」との関連で種々の御支援を戴いた。ここに感謝の意を表します。

また、柿倉正義・同所制御部情報制御研究室長、山崎正人主任研究官をはじめ、本研究の中期において利用したリスプ系言語 PETL を開発された塚本亨治主任研究官ほか、制御部の諸兄から戴いた日頃の御助力と御討論、御支援に感謝致します。また、井上博允・東京大学教授(元同所制御部)には、本研究の出発点において、御指導、御激励を戴いた。白川洋尤・立命館大学教授(元同所制御部)には、本研究をまとめるに際して御鞭撻を戴いた。これらの方々、および御支援を戴いた多くの諸先輩に感謝致します。

本研究では、その初期に同所情報部門共用計算機 TOSBAC 5600 を、中期には制御部共用計算機 COSMO 700 を、そして後期には工業技術院情報計算センター・RIPS を利用した。これらのシステムの整備と運用に尽力された方々に感謝の意を表します。

最後に、本研究を遂行するにあたって御指導、御助力を戴きながら名前を挙げられなかった多くの方々に感謝申し上げます。

参 考 文 献

- 1) Aho, A. V., Ullman, J. D, "The Theory of Parsing, Translation, and Compiling", Prentice-Hall, (1972)
- 2) ALPAC, "Language and Machines, Computers in Translation and Linguistics", National Academy of Sciences, (1966)
- 3) Bar-Hillel, Y., "A Demonstration of the Nonfeasibility of Fully Automatic High Quality Translation", in Bar-Hillel "Language and Information, Selected Essays on their Theory and Application", Addison-Wesley, (1964)
- 4) Bresnan, J. (ed.), "The Mental Representation of Grammatical Relations", MIT Press, (1982)
- 5) Bruce, B., "Case Systems for Natural Language", Artificial Intelligence, 6, (1975)
- 6) Chikayama, T., "UTILisp Manual", METR 81-6, University of Tokyo, (1981)
- 7) Chomsky, N., "Aspects of the Theory of Syntax", MIT Press, (1965)
- 8) Early, J., "An Efficient Context-Free Parsing Algorithm", Communication of the ACM., 13, (1970)
- 9) Grishman, R., "A Survey of Syntactic Analysis Procedures for Natural Language", American Journal of Computational Linguistics, 47 (microfiches), (1976)
- 10) Gazdar, G, "Phrase Structure Grammar", in Jacobson, P and Pullum, G (eds.), "The Nature of Syntactic Representation", Reidel, (1982)
- 11) Hutchins, W. J., "Progress in Documentation-Machine Translation and Machine Aided Translation", Journal of Documentaion, 34-2, (1978)
- 12) Ikeda, T., "J-Analyser : Analyser of Japanese Sentences based on Binding Structure Representation of Words", Journal of Information Processing, 3-4, (1981)
- 13) Katz, J, "Semantic Theory", Harper & Row, (1972)
- 14) Kuno, S., "Predictive Analyser and a Path Elimination Technique", Communication of ACM., 8, (1965)
- 15) Locke, W. N., and Booth, A. D. (eds.), "Machine Translation of Languages", Greenwood Press, (1955)
- 16) Marcus, M., "A Theory of Syntactic Recognition for Natural Language", MIT Press, (1980)
- 17) Nagao, M. and Tsujii, J., "Analysis of Japanese Sentences Using Semantic and Concep

- tual Information” , *American Journal of Computational Linguistics*, 41 (microfiches), (1976)
- 18) Nakashima, H., Tomura, S., “Introduction to AMUSE. A Multi-Use Structure Editor”, METR83-3, University of Tokyo, (1983)
 - 19) Prat, V. R., “A Linguistic Oriented Programming Language”, 3rd International Joint Conference of Artificial Intelligence, PP272-381, (1973)
 - 20) Rustin, R., (ed.), “Natural Language Understanding”, Algorithmic Press, (1971)
 - 21) Schank, R. C., and Colby, K. M., “Computer Models of Thought and Language”, Freeman & Co., (1973)
 - 22) Schank, R. C., “Conceptual Information Processing”, North-Holland, (1975)
 - 23) Schank, R. C., “Dynamic Memory”, Cambridge University Press, (1982)
 - 24) Simmons, R. F., “Semantic Networks : Their Computation and Use for Understanding English Sentences”, in Schank and Colby (eds.), “Computer Models of Thought and Languages”, Freeman & Co., (1973)
 - 25) Tennant, H., “Natural Language Processing”, Petrocelli Books, (1981)
 - 26) Wilks, Y., “Grammar, Meaning and Machine Analysis of Language”, Routledge & Kegan Paul, (1972)
 - 27) Winograd, T., “Understanding Natural Language”, Academic Press, (1975)
 - 28) Winograd, T., “Language as a Cognitive Process (Syntax)”, Addison Wesley, (1983)
 - 29) Woods, W. A., “Transition Network Grammars for Natural Language Analysis”, Communication of the ACM., 13, (1970)
 - 30) Икеда, Т., “Я-АНАЛИЗАТОР: АНАЛИЗ ПЛЕДЛОЖЕНИЙ НА ЯПОНСКОМ ЯЗЫКЕ”, ВОПРОСЫ РОСЫ РАЗРАБОТКИ ПРИКЛАДНЫХ СИСТЕМ, ВЫЧИСЛИТЕЛЬНЫ ЦЕНТР СО АН СССР, pp 129-146 , (1979), { ノボシビルスク計算センター論文集 }
 - 31) ヒル (Hill, A .A ed.), (宮部菊男, 監訳), “現代言語学—紹介と展望”, 研究社, (1971)
 - 32) フィルモア (Fillmore, C. J.), (田中春美, 舟城道雄 訳), “格文法の原理”, 三省堂, (1975)
 - 33) ヘルビヒ (Helbig, G), (岩崎, 他 共訳), “近代言語学史”, 白水社, (1973)
 - 34) 育野准治, 池田尚志, “格構造による意味表現と ATN モデルによる英語文の parsing と gene-rating, 昭和 52 年電子通信学会総合全国大会, 1101, (1977)
 - 35) 池田尚志, “ある種の動詞の構成と課題解決法”, 電子通信学会オートマトンと言語研究会, AL 73-44, (1973)
 - 36) 池田尚志, “式の変形による定理の自動証明について”, 情報処理学会プログラミングシンポジウム委員会記号処理シンポジウム報告集, PP 100-106, (1974)

- 37) 池田尚志, “等式証明問題について”, 昭和 50 年電気学会全国大会, 1400, (1975)
- 38) 池田尚志, “定理証明における等号関係の扱いについて”, 電子通信学会論文誌 (技術談話室), 58 D-4, (1975)
- 39) 池田尚志, “格構造による意味表現を用いた日本語文の構文解析”, 昭和 52 年電子通信学会総合全国大会, 1100, (1977)
- 40) 池田尚志, “日本語文の構文解析—特に辞の辞書構成について—”, 情報処理学会第 18 回全国大会, 98, (1977)
- 41) 池田尚志, “語の機能 意味構造の記述に基づく日本語文の分析”, 電子通信学会オートマトンと言語研究会, AL 77-53, (1977)
- 42) 池田尚志, “一般化された格構造による意味表現を用いた日本語文の構文解析法について”, 電子通信学会論文誌, 60 D-10, (1977)
- 43) 池田尚志, “語の構造・意味記述と文の構造・意味分析”, 電子通信学会オートマトンと言語研究会, AL 78-32, (1978)
- 44) 池田尚志, “語の構造・意味記述と文の構造・意味解析”, 情報処理学会第 19 回全国大会, 5 E-10, (1978)
- 45) 池田尚志, “日本語文の構文および意味の分析法について”, 電総研彙報, 42-6, (1978)
- 46) 池田尚志, “日本語文の分析と構造翻訳”, 昭和 54 年電子通信学会情報・システム部門全国大会, 43, (1979)
- 47) 池田尚志, “日本語文における機能語の働きについて”, 情報処理学会第 21 回全国大会, 41-2, (1980)
- 48) 池田尚志, “PETL 上のテキストエディタとその清書機能について”, 昭和 56 年電子通信学会情報・システム部門全国大会, 38, (1981)
- 49) 池田尚志, “機械翻訳システム”, {田中穂積, 他, “自然言語処理技術と言語理論”, 電総研調査報告, 第 8 章}, (1981)
- 50) 池田尚志, “機械処理の側から見た日本語文法”, 視聴覚情報研究会, 17-3, (1981)
- 51) 池田尚志, “日本語文における文脈処理の枠組みについて”, 情報処理学会第 25 回全国大会, 3 K-6, (1982)
- 52) 池田尚志, “語法規則の翻訳に基づく日英機械翻訳”, 情報処理学会第 26 回全国大会, 4 L-2, (1983)
- 53) 池田尚志, “複合語・未定義語を含む日本語文の構文・意味解析およびその機械翻訳への応用について”, 情報処理学会第 27 回全国大会, 5 F-9, (1983)
- 54) 池田尚志, “語法規則の翻訳に基づく日本語からの機械翻訳”, 情報処理学会自然言語処理研究会, NL 35-6, (1983)
- 55) 池田尚志, 平井成興, “故障診断システム PRODOG のための日本語インタフェース—日本語

- による診断知識表現からのルール抽出”，情報処理学会第 27 回全国大会，6 D-1，(1983)
- 56) 池田尚志，“日本語文における格の種類についての考察”，情報処理学会自然言語処理研究会，NL 41-2，(1984)
- 57) 池田尚志，“リスプによるキー順データセットへのアクセス機能”，電総研彙報，48-8，(1984)
- 58) 池田尚志，“語法規則変換方式による機械翻訳—翻訳の表現式と翻訳文生成関数—”，情報処理学会自然言語処理技術シンポジウム論文集，PP 107-114，(1984)
- 59) 池田尚志，中嶋正夫，“語法規則変換方式による日独機械翻訳”，情報処理学会第 30 回全国大会，7 G-7，(1985)
- 60) 池田尚志，“キー順ファイルをベースとした解析・翻訳の実験環境”，情報処理学会自然言語処理研究会，NL 50-4，(1985)
- 61) 池田尚志，“語法規則方式による日本語文の構文・意味解析”，情報処理学会論文誌，26-6，(1985)
- 62) 池田尚志，“名詞文の解釈について”，電子通信学会論文誌（技術談話室），69 D-11，(1986)
- 63) 石崎 俊，伊佐原均，“文脈処理技術”，情報処理，27-8，(1986)
- 64) 市山俊治，村木一至，“日本語処理の観点から見たプッシュダウンオートマタの再評価”，情報処理学会「自然言語処理」シンポジウム論文集，(1983)
- 65) 井上和子，“変形文法と日本語（上・下）”，大修館，(1976)
- 66) 奥津敬一郎，“生成日本文法論”，大修館，(1974)
- 67) 小野修一，安藤 茂，池田尚志，“Melcom 7500 シリーズのリスト処理言語について”，情報処理学会プログラミングシンポジウム報告集，PP 35-43，(1974)
- 68) 片桐恭弘，中村順一，辻井潤一，長尾 真，“Mu プロジェクトにおける翻訳実験支援環境”，情報処理学会自然言語処理研究会，47-9，(1985)
- 69) 北原保雄，“助動詞の相互承接についての構文論的考察”，国語学，83，(1970)
- 70) 金田一春彦，“国語動詞の一分類，{金田一春彦（編），“日本語動詞のアスペクト”，むぎ書房，(1976)}
- 71) 草薙 裕，“日本語文解析におけるテンス・アスペクトの問題”，情報処理学会自然言語処理研究会，34-11，(1982)
- 72) 久野 暲，“日本文法研究”，大修館，(1973)
- 73) 久野 暲，“談話の文法”，大修館，(1978)
- 74) 国立国語研究所，“現代語の助詞・助動詞—用法と実例—”，秀英出版，(1951)
- 75) 国立国語研究所，“分類語彙表”，秀英出版，(1964)
- 76) 小暮 潔，野村浩郷，“翻訳実験のためのインタラクティブな支援環境”，情報処理学会自然言語処理研究会，51-5，(1985)

- 77) 佐久間鼎, “日本の表現の言語科学”, 厚生閣, (1967)
- 78) 島津 明, 内藤昭三, 野村浩郷, “格構造モデルに基づいた日本語文の分析と解析”, 情報処理学会自然言語処理研究会, 29-1, (1982)
- 79) 島津 明, 内藤昭三, 野村浩郷, “機械翻訳システム LUTE の日本語意味解析”, 情報処理学会自然言語処理研究会, 33-6, (1982)
- 80) 島津 明, 内藤昭三, 野村浩郷, “構造予測を用いた日本語文の意味解析法”, 情報処理学会論文誌, 27-2, (1986)
- 81) 首藤公昭, “専門分野を対象とした日英機械翻訳について”, 情報処理, 14-9, (1973)
- 82) 高松, 藤田, 谷, 西田, “日英機械翻訳における内部表現の部分的トランスファと英文の生成”, 情報処理学会論文誌, 26-5, (1985)
- 83) 田中穂積, “電子計算機による自然言語の意味処理に関する研究”, 電子技術総合研究所研究報告, 797, (1979)
- 84) 田中穂積, 他, “自然言語処理技術と言語理論”, 電総研調査報告, 205, (1981)
- 85) 塚本亨治, “Petl システム説明書”, 電子技術総合研究所, (1982)
- 86) 辻井潤一, “日本語構文解析”, 情報処理学会自然言語処理研究会, 38-5, (1983)
- 87) 辻井潤一, 山梨正明, “格とその認定基準”, 情報処理学会自然言語処理研究会, 52-3, (1985)
- 88) 辻井潤一, “文解析方式”, 情報処理, 27-8, (1986)
- 89) 内藤昭三, 島津 明, 野村浩郷, “格構造モデルに基づく日本語意味構造の表現について”, 情報処理学会自然言語処理研究会, 36-2, (1983)
- 90) 長尾 真, 辻井潤一, 田中一敏, “意味および文脈情報を用いた日本語文の分析—名詞句・単文の処理”, 情報処理, 17-1, (1976)
- 91) 長尾 真, 辻井潤一, 田中一敏, “意味および文脈情報を用いた日本語文の分析—文脈を考慮した処理”, 情報処理, 17-1, (1976)
- 92) 長尾 真, “制限言語の試み”, 情報処理学会「自然言語処理」シンポジウム論文集, (1983)
- 93) 長尾 真 (監修), “日本語情報処理”, 電子通信学会, (1984)
- 94) 西田豊明, “変換処理過程の基本設計”, 情報処理学会自然言語処理研究会, 38-6, (1983)
- 95) 西村恕彦, 水谷静夫, 尾上圭介, 田中幸子, “日本語基本文法 (単文篇)”, 電子技術総合研究所研究報告, 783, (1978)
- 96) 西村恕彦, 水谷静夫, 尾上圭介, 大野美江子, “日本語基本文法 (複文篇)”, 電子技術総合研究所研究報告, 784, (1978)
- 97) 溝口文雄 (編), “機械翻訳特集”, 情報処理, 27-8, (1986)
- 98) 野村浩郷, 内藤昭三, “自然言語理解における意味表現”, 情報処理, 27-8, (1986)

- 99) 日高 達, 吉田 将, “格文法による日本語の構文解析”, 情報処理学会「自然言語処理技術」シンポジウム論文集, (1983)
- 100) 平井 誠, “格構造の変換規則と簡単な事象の分類を用いた日本語文解析”, 情報処理学会自然言語処理研究会, 33-3, (1982)
- 101) 平井 誠, 北橋忠宏, “言語理解システム MARION-IIIにおける意味表現について”, 情報処理学会自然言語処理研究会, 40-1, (1983)
- 102) 富士通, “FACOM OS IV/F4 VSAM 機能説明書”
- 103) 三上 章, “象は鼻が長い”, くろしお出版, (1960)
- 104) 三上 章, “現代語法新説”, くろしお出版, (1972)
- 105) 村木一至, 市山俊治, “日英機械翻訳システム VENUS の解析メカニズム”, 情報処理学会自然言語処理研究会, 40-7, (1983)
- 106) 村木進次郎, 青山文啓, 六条範俊, 村田賢一, “辞書における格情報の記述”, 情報処理学会自然言語処理研究会, 46-3, (1984)
- 107) 吉田 将, “二文節間の係り受けを基礎とした日本語文の構文解析”, 電子通信学会論文誌, D 55-4, (1972)
- 108) 吉田 将, 田中武美, “自然言語文書のための日本語文法の制限について—付属語的表現の語彙および語義の制限を中心として”, 情報処理学会自然言語処理研究会, 37-4, (1983)