自然言語処理 51－5
（1985. 9. 13）

翻訳実験のためのインタラクティブな支援環境

小暮 潔　　野村 浩郷
(NTT 武蔵野電気通信研究所)

1. はじめに

自然言語処理、その応用としての機械翻訳を考えるとき、高度な機能を実現するには、文法規則や語彙などに関する多様かつ相互に複雑に関連し合った知識の表現･操作を必要とする。これらの知識をあらかじめすべて用意しておくことは非常に困難であり、実験や運用を通じて段階的に追加･修正して組上げていくことになる。これは、"知識の追加･修正"と"実験による検証"というサイクルの繰り返しを示している。このようなサイクルを効率化するには、言語処理、機械翻訳に適した研究開発支援環境が重要になってくる。

一方、AI研究の分野においては、プログラミング環境、特にデバッギング･ツールなどが重要視されれているが、両者を統合したような高度で、フレンドリな支援環境が重要である。

我々は、言語理解の研究を進めているが、研究により得られたものを検証するために、機械翻訳実験システムLUTE (Language Understander, Translator & Editor)を作成してきた。[1]、[2]、[3]、[4]、[5]、[6]このシステムは、日英、英日双方向の機械翻訳処理の実験を目的としたもので、システムにおける知識表現としては、辞書や意味構造表現などすべてのデータを統一的にフレーム形式で表現する方法をとっている。

このシステムの中で、
(1) 実験サイクルの短縮
(2) 言語作業の操作性の向上
を重視した支援環境について検討を行い、これに基づき、Symbolics Lispマシン上に実験システムを作成した。このシステムは、「実験者の思いつき」を速やかにテストできることを目標に、マン･マシン･インタラクションを重視して設計している。システムは、フレーム･エディタFRED[1]とZMACSエディタ[7]をベースに、自然言語処理用の機能を付け加えることによって実現され、ディスプレイの画面表示はマルチ･ウィンド形式であり、システムはメニューからの選択によって操作する。このシステムは、現在実際に使用されている。

2. 望まれる実験環境

既に述べたように、翻訳などにおける高度な機能の実現は、実験と知識の追加･修正のサイクルによって進められる。これを効率的に行うためには、個々のサイクルのターン･アラウンド･タイムを縮小させることが重要である。そのためには、その要素である実験による問題点の発見･テストとそれに基づく新しい知識の追加･修正の双方を効率的にすることが必要である。

2.1 結果の確認において望まれる環境

機械翻訳は、解析、変換、生成のように多段階の処理によって構成されるから、問題点の原因を発見する場合、まず、どの段階で不都合な問題が発生したのかを切りわけなければならない。次に、その段階の中で、何に関係しているのか、例えば、複合文、単位文、句、複合語、単語のレベルの内で、どのレベルと関係しているのかを切りわけなければならない。これは、一般のプログラム処理において入力データのどの部分が問題点の原因になっているのかを調べることと密接に関係している。

まず、問題がどの段階で発生したのかを調べるためには、中間結果やトレース情報を提示することが考えられる。この場合、中間結果、すなわち、解析結果や変換結果は、木などのような構造を持っているから、これを人間に理解しやすい形式で表示してやることができれば、問題点の発見のための大きな手助けになる。

ここで、計算機の内部表現に近い形式、例えば、LispのS-式をプリント･アウトしても、人間がそれを解釈するのに時間がかかる。また、労力がそのS-式の括弧の解釈に向けられ、内容の把握に専念できなくなる。すなわち、この方法では非効率的で、特に、Lispに不慣れなものにとっては、大きな負担となる。したがって、内容に専念できる表示形式が必要である。例えば、我々が言語学の教科書で見るような木表示などのように、我々のメンタル･モデルに整合する形式で表示することが望まれる。

しかし、木を図形的に表示する場合でも問題は残る。それは、
(1) 木の複雑さの度合い
(2) 理解しやすい表示の制約
に起因する問題である。我々が実験で問題にするような文の解析結果などは、多くの場合、教科書に見られるような単純なものではなく、複雑なものである。木のノード数は、比較的単純な文でも20以上あり、少し複雑な文になれば、すぐに100を超してしまう。一方、理解しやすい木の表示には、二つの制約がある。
(1) 物理的な制約
(2) 認知的な制約
である。前者は、画面の大きさ、解像度などである。後者は、複雑な木の全ての情報を表示しても、

全体像や詳細に知りたい情報を一度に把握することが困難であることによる制約である。したがって、その時々の目的に照らし合わせて重要度の低い情報の省略を行い、複雑性を抑制し、理解しやすい形式で提示することが重要となる。

次に、処理のどのレベルで、あるいは、どの入力に対して問題が発生したかを速やかに発見する方法であるが、処理をブラックボックスと見なして、入力データの一部に処理を部分的に適用してみたり、語順の変更や単語の置換などを行ったデータに対して、部分的な処理を適用することによって、調べていくことが考えられる。これを効率的に行うためには、テキスト・エディタなどと組み合わせて、編集を容易に、かつ、編集した文字列について速やかに実験できる機能が望まれる。

さらに、処理の途中段階で期待していたものが得られなかったことが判明した場合には、文法規則や辞書項目の修正などによって対処することになるが、その結果の確認を速やかに行うためには、中間結果の修正や中間状態から処理が再起動できる機能が望まれる。翻訳のように多段的な処理の場合、この効果は大きい。

2.2 知識の編集において望まれる環境

次に、知識の追加・修正の効率化の問題であるが、特に、語彙に関する知識、すなわち、辞書に関するものが、その量の多さから問題の中心となるであろう。また、システムに精通していない人間の参加などによって生じる問題なども考慮しなければならない。

ここでは、辞書の編集に関する問題を
(1) 辞書項目内での問題
(2) 辞書項目間にわたる問題
(3) 辞書間にわたる問題
に分類し、望まれる環境について検討してみる。

(1) 辞書項目内での問題
これは、さらに、表現形式上の問題と内容に関する問題とに分けて考えることができる。

まず、表現形式上の問題であるが、これは、ユーザへの辞書内容の提示法に関係する。フレームで表現された辞書項目の場合、その内部表現は、nested listであるため、S-式を編集する場合、括弧の過不足によるミスマッチなどを生じやすい。したがって、内部表現とは別の人間の直感と合った理解しやすい表示を提供することが望まれる。

同様のケアレス・ミスとして、タイプ・ミスなどがあるが、これは、可能なかぎりタイプしないですむようにすることによって減らすことができる。例えば、入力されるべきデータが特定の集合によって決まっている場合は、メニューの中から選択させることによって、タイプをなくすことができる。

以上で述べたようなデータの誤入力は、発生しやすく、発見しにくい。また、括弧の不整合などによる表現形式上の誤りは、プログラムのフェータル・エラーを誘発させることもあり、実験の効率を低下させる大きな要因である。したがって、これらを減少させることはとても重要である。

次に、内容に関する問題であるが、まず、入力された個々の情報が正しいデータであるかの判定が問題になる。これは、後でチェックのためのプログラムで確認するのではなく、システムによって入力直後に判定され、システムからの訂正の要求などがあると便利であろう。

また、入力された情報が全体として適正かどうかの判定が問題となる。例えば、現在、編集されている辞書項目が必須情報を十分に持っているかどうかの判定などは重要である。辞書項目として必要な情報の中には、他の情報の内容と相互に関係して必要になるものがある。例えば、活用のある語の場合だと、活用型を必要としているなどである。この場合も品詞の入力と同時に活用型についての情報も同時に必要としていることがシステムから自動的に知らされると便利である。このような適合性・十分性の判定を行いながら、インタラクティブに辞書が編集できると便利である。

(2) 辞書項目間にわたる問題
複数の人間が辞書の編集に参加する場合、辞書項目間で記述上の均質性を保つことが重要な問題とな

フレームの基本構造

```
(frame
 (slot (facet (datum...)
              (datum...)...)
       (facet (datum...)...)...)
 (slot (facet (datum...)...)...)...)
```

辞書項目の内部構造の例

```
(MI-V-1
 (*AKO       ($VALUE (ENTRY)))
 (*ROMAN     ($VALUE ("mi")))
 (*YOMI      ($VALUE ("み")))
 (*KANJI     ($VALUE ("見")))
 (*ROOT-FORM($VALUE ("見")))
 (*POS       ($VALUE (VERB)))
 (*INFLECTION-TYPE
             ($VALUE (1)))
 (*RENSETSU ($VALUE (V-1)))
 (*MEANING  ($VALUE (OR (MI-V-1-1)
                        (MI-V-1-2))))
 )
```

図-1 辞書項目の構造

る。これは、意味カテゴリや意味関係の判定などにおいて特に問題となる。これに対しては、判断の基準となるような類例を提示することが有効であろう。したがって、必要な時に類例を提示できる機能が望まれる。

また、同様な問題として、関係する辞書項目間の整合性も考慮しなければならない。例えば、動詞などの格フレームの辞書記述における格要素の意味カテゴリなどの束縛条件と、その格要素となりえる名詞などの意味カテゴリの対応などである。このような場合、関係する辞書項目の記述がその場で、同時に参照できたり、編集できる機能が望まれる。

(3) 辞書間にわたる問題

翻訳の場合、解析のための辞書、変換のための辞書、生成のための辞書が必要となる。したがって、特定の単語が自分の訳したい単語に翻訳されるかどうかを確認するためには、それぞれの辞書の内容を同時に見る機能が望まれる。それを用いることにより、不足している辞書項目を追加したり、編集したりすることが同時にできれば便利である。

3. 翻訳実験支援システム

インタラクティブな実験環境として、2.1での検討に基づき、翻訳実験のための支援システムを作成した。

3.1 システムの構成

システムは、大きく分けて、ZMACSエディタ、フレーム・エディタFREDなどのマン・マシン・インタフェースの部分と、解析、変換、生成の言語処理プログラムからなる。

3.1.1 マン・マシ・インタフェースの部分

(1) ZMACSエディタ

これはSymbolics Lispマシンに組み込まれたエディタである。このエディタは、入力文の編集、入力、および、生成された文章の出力に用いられる。ユーザが自由にコマンドを拡張することができ、キーに割り付けることができる。この機能を用いて、言語処理コマンドが割り付けられている。

(2) フレーム・エディタFRED[1]

中間結果の表示、編集、言語処理プログラムの起動に用いる。このエディタでは、主に、木表示の

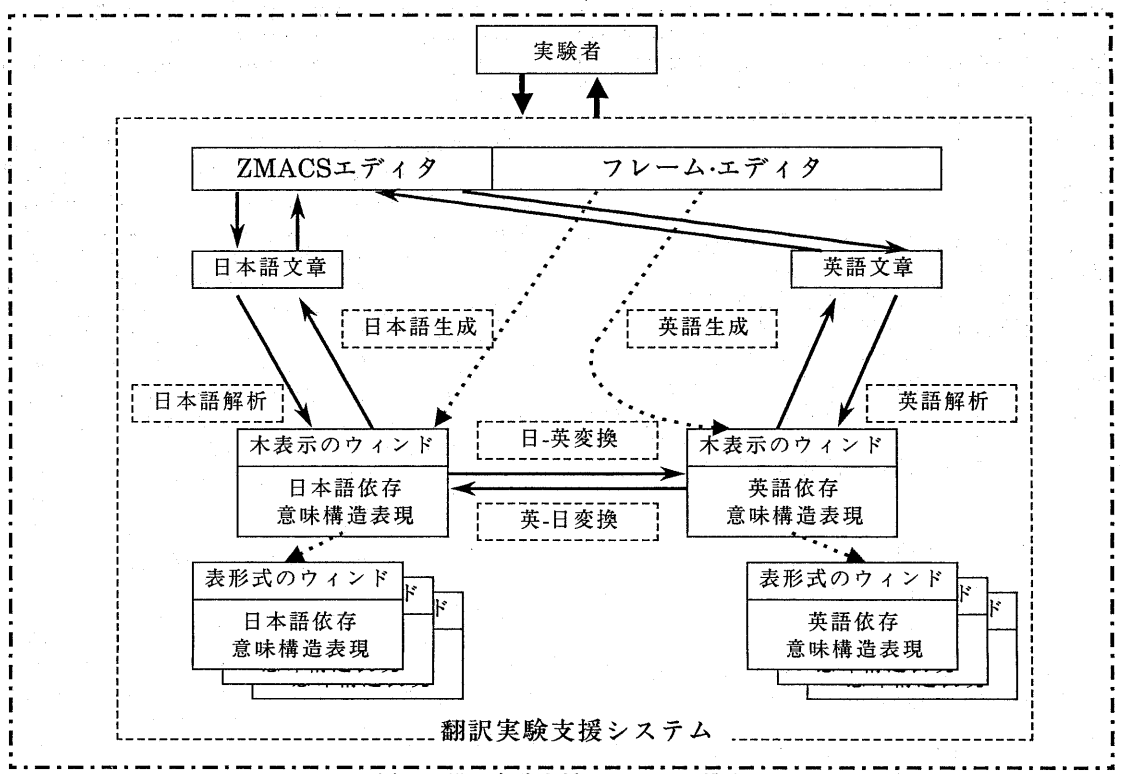

図-2 翻訳実験支援システムの構成

ウィンドと表の表示のウィンドを使って、フレーム型で表現されているデータを編集する。

(2-1) 木表示のウィンド
木の上のノードをマウスで選択することにより、メニューを表示させ、その中のコマンドを実行させることができる。コマンドとしては、以下のような種類のものがある。
(a) 表示を変更するコマンド:
そのノード以下の構造の表示、そのノード以下の構造の画面からの消去、ノードを木の根とした表示、上位構造の表示、省略の指定などのコマンドがある。
(b) フレームの内容を変更するコマンド:
これは、表形式のウィンドを呼び出して行う。
(c) 拡張コマンド
メニューに自由にコマンドを追加することができる。この機能を用いて、言語処理プログラムを呼び出す。

(2-2) 表形式のウィンド
フレームのデータを、表の形式で表示し、マウスでデータを選択し、追加、削除、修正することができる。また、下位構造としてフレームを含んでいる場合は、それを再帰的に表形式のウィンドで表示させることができる。

(3) その他のウィンド
トレース情報を表示するウィンドと処理の状態を表示するウィンドがある。トレースを表示するウィンドは、言語処理プログラムの中での規則が適用されている過程などを表示するために使われ、規則のデバッグのときなどに役にたつ。状態を表示するウィンドは、どのプログラムが実行されているかを表示する。

3.1.2 言語処理プログラム

言語処理プログラムとしては、LUTEシステムを構成しているモジュールの中から
日本語解析　日英変換　英語生成
英語解析　英日変換　日本語生成
の各モジュールを取り出し、単独で、あるいは、組み合わせて、ZMACSエディタ、フレーム・エディタのコマンドとして定義している。したがって、エディティング・コマンドと同レベルのコマンドでこれらのいずれの言語処理プログラムも起動できる。

3.2 システムの機能

このシステムを使用することによって、次のようなことが可能となった。

(1) テキスト・エディタ画面上の任意の文字列に対して、言語処理が適用できる。

これは、以下のようにして、行われる。まず、ZMACSエディタの機能を用いて、マウス、あるいは、カーサ移動用コマンドによって、処理対象の文字列を指定する。次に、この文字列に対して処理を起動する。起動方法には、2通りある。一方は、直接、コマンドで起動する方法で、例えば、<SUPER>とJを同時にタイプすることにより、日本語解析を起動する。他方は、言語処理のメニューを表示させ、その中から、マウスにより選択する方法である。これらにより、日本語ワードプロセッサにおけるカナ漢字変換操作と同程度の容易性で、解析や翻訳などの実験が開始できる。この容易性から、単語の置換、語順の変更などの入力結果の変更に伴う解析・翻訳結果の違いなどが容易に比較できる。

各処理は、最初に起動されるときに文法規則の読み込みや辞書の初期化などを行うが、この過程で、規則や辞書のファイルを指定することができる。したがって、規則や辞書を複数個用意しておいてそれらを入れ替えて解析・翻訳の実験をすることができる。また、処理が起動されると、状態表示用のウィンドに、処理の進行状態が表示される。

このようにして起動された処理の中間結果と最終結果は、速やかに画面上に反映される。解析の場合は、結果がフレーム・エディタのウィンドに木で出力される。また、翻訳の場合には、解析された結果と変換された結果がそれぞれ別のウィンドに木で表示され、生成された文字列、すなわち、訳文は、エディタの入力文字列の次の行に挿入され、表示される。処理の途中で複数の結果が得られた場合は、それぞれが個別のウィンドに表示され、比較することができる。これら処理結果の表示場所の指示・移動の機能も用意されている。

出力された木は、初期状態では、全体を一目で把握できる程度の複雑さに抑制するために、骨格情報が表示されている。すなわち、木全体の概略構成を表示する。これは、重要な情報を深さ2~3で表示したものである。木の各ノードはフレームに対応しているが、このフレームが下位構造の中で何が重要な情報であるかの情報を持っていて、それに基づいて、下位構造を表示している。この情報を検索する際、フレームの持っているインヘリタンスの機能などを用いる。深さ2とは、複合文の場合、主文の格構造までと従文の単位文までの構造に対応し、また、単位文の場合には、格要素の構造までに対応し、ノード数にして、10~20程度である。したがって、複合文の主文と従文および、その間の関係をとらえるのに適している。また、単位文の場合、格の数などを簡単に確認するにも便利である。

従文の構造や格の中身の細部を確認したい場合、フレーム・エディタのコマンドを用いて、木の細部の枝を新しく表示させたり、人間が視点を移動させるように、指定したノードを根とした木に再表示することができる。これは、マウスでノードを選択し、コマンド・メニューの中から対応するコマンドを選択することによって行うことができる。ま

た、木のノードの属性的な情報を参照したい場合は、表形式のウィンドをノードの上に重ねて表示することができる。また、ウィンドの属性を変えるコマンドにより、木全体を表示させることもできる。

(2) 中間結果の修正、中間結果に対する処理が可能となる。

フレーム・エディタの機能を用いて、ウィンドに表示された中間結果を修正したり、中間結果に対して、次の処理、すなわち、解析結果に対する変換、変換結果に対する生成などが可能となっている。変換の結果は、新たな木表現のウィンドに表示され、生成結果は、中間結果のウィンドの下部に表示される。これらの機能は、主に次の4通りに使われる。

(a) 変換、生成で失敗した場合の原因の発見
解析、変換の結果は、ウィンド上に木で表現されているが、木のノードを選択し、コマンドを適用することによって、その下位構造だけを処理することができる。例えば、解析された結果の埋め込み文だけを変換することができる。これにより、失敗の原因となる可能性のある文法規則や辞書項目の範囲を狭めることができる。

(b) 文法規則や辞書項目の修正と再起動
問題となる文法規則や辞書項目を修正し、再起動することによって、修正結果の確認を効率的に行うことができる。

(c) 中間結果の修正と処理の再起動
これは、更に二つの場合に分けて考えることができる。一方は、中間結果が誤っていたときに、これを修正し、後段の処理を再起動する場合である。これにより、前段の処理が完全に成功していなくても次の段の処理をテストすることができる。他方は、後段の処理が辞書項目の記述内容の欠如などで失敗した場合に、とりあえず、既存の単語などと置換して、その部分以外が正しく処理できるかどうか確認するために用いられる。

(d) 解析から生成への逆戻り
解析、生成の部分だけをそれぞれテストするために、例えば、日本語解析を行った結果に対して、日本語生成を適用してみる場合である。生成された文を見る方法は、解析結果を確認する手段としては、誰にでもわかる最も簡単な方法である。また、生成させることによって、入力文の省略された内容が正しく補完されているかの確認もできる。

(3) 結果の管理

以上で述べた実験の中間結果、最終結果は、トレース情報などとともに保存され、文法規則、辞書項目の修正、変更による結果の比較などを行うために使われる。

(4) その他のユーティリティ

表-1 木表示のウィンドのコマンド

コマンド名	機能
Show Table Window	表形式のウィンドを表示する。
Expose under this frame	木のノードの下位構造を画面に表示する。
Deexpose this frame	木のノードとその下位構造を画面から消去する。
Move this frame	木のノードとその下位構造を画面上で移動する。
Make this frame the root	選択したノードが木の根になるように再表示する。
Expose superior frame	もしあれば、木の上位構造を表示する。
Show words	木に属している辞書項目を表示する。
Transfer Japanese to English	日英変換を行い、その結果を新しいウィンドに表示する。
Transfer English to Japanese	英日変換を行い、その結果を新しいウィンドに表示する。
Generate Japanese Sentence	日本語生成を行い、そのウィンドに表示する。
Generate English Sentence	英語生成を行い、そのウィンドに表示する。
Attributes	ノードの属性を変更する。
Window's Attributes	ウィンドの属性を変更する。
Reset Input Buffer	バッファに蓄積されたコマンドをリセットする。

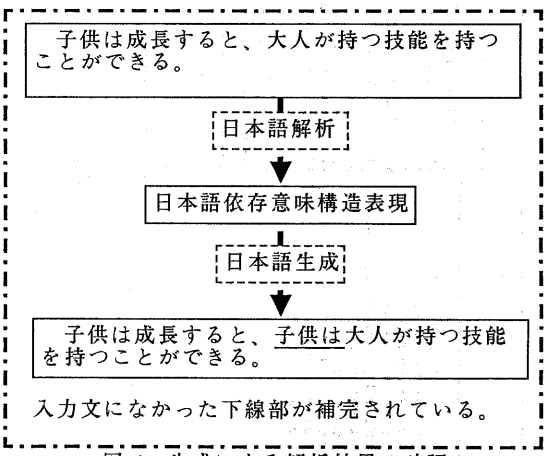

図-4 生成による解析結果の確認

Symbolics Lisp マシンでは、漢字コードがJISコードと異なるため、他の計算機で作られたデータを使用するために、双方向のコード変換のコマンドが用意されている。

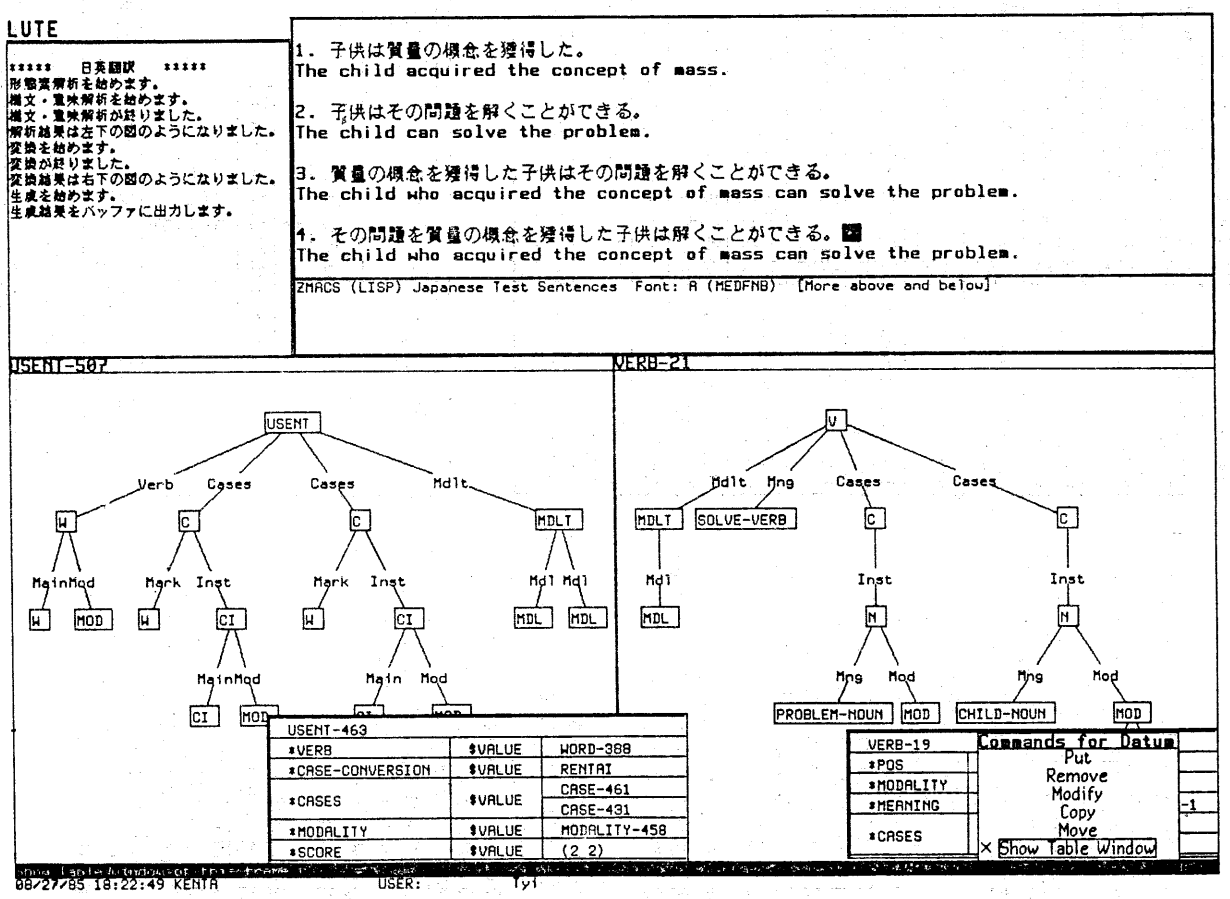

図-3 翻訳実験支援システムの画面
(左上) 状態表示のウィンド (右上) 入力文と訳文を表示するウィンド
(左下) 日本語解析の結果のウィンド (右下) 日英変換の結果のウィンド

「質量の概念を獲得した子供はその問題を解くことができる。」という文について、1番目と2番目の文では、文の一部を取り出し、実験している。3番目と4番目の文では、語順を変えて実験しているところ。ノードの内容について詳細情報が知りたいときは、そのノードをマウスでクリックすることにより、その内容が表表示される。

4. 辞書編集システム

辞書編集システムとして、2.2での検討に基づき、次の特徴を持ったシステムを実現した。

(1) 辞書の各項目内のフレーム、スロット、ファセット、データと呼ばれる内部構造(図1参照)を表現する方法として、図形的な表を用い、構造エディタ風に編集する。LispのS-式に見られるような括弧を、この目的で使用しない。

(2) 辞書項目の内容として何が要求されるかの情報を辞書編集システムが持ち、システムの内部状態に応じて、必要な情報をユーザに提示する。

(3) データの入力毎に、データの適合性をシステムが確認し、不適当な場合には、ユーザに対して訂正を要求する。

4.1 辞書編集システムの構成

辞書編集システムは、辞書項目の持つべき情報に関する情報を記述したフレームと、マン・マシン・インタフェースとなるウィンドとからなる。

4.1.1 辞書項目の情報に関するフレーム

辞書項目の情報に関する情報を記述するために、このようなメタ情報を記述するためのフレームが用意されている。このフレームに記述されている情報には、データ構成に関する情報と、ウィンド表示に関する情報とがある。

データ構成に関する情報としては、
(1) 各スロットに入るデータの条件
(2) 各スロットに入るデータの個数
(3) 各スロットにデータが入力されたとき起動される手続き(デーモン・プログラム)
(4) 各スロットにおけるデータの入力法
などがある。

ウィンド表示に関する情報としては、
(1) 表示するスロットのリスト
(2) 各データの表示法
などがある。

4.1.2 辞書編集のためのウィンド

辞書編集のときに使用するウィンドとしては、専用の辞書ウィンドと辞書ヘルプ・ウィンド、および、入力用の補助的なウィンドとメニュー・ウィンドからなる。これらのウィンドは、フレーム・エディタで実現されたウィンドに、特別なコマンドなどを組み込み、特殊化することによって、実現されている。

(1) 辞書ウィンド
辞書項目を編集するために用いるウィンドで、フレームに対応して、作られる。

一般に、普通の辞書編集者は、値を入れるファセットしか用いない場合が多い。そこで、このウィンドでは、編集するファセットをこれだけに限定し、フレーム名、スロット名、スロット値だけを表示することによって、表示・編集を簡単化してある。

ウィンド内の各項目をマウスで選択することによって、コマンドを実行することができる。コマンドとして、基本的には、データの追加と削除がある。ここで、特徴的なのは、データの追加のコマンドで、同じ"Put Data"コマンドでも、そのデータに関する情報によって、データの入力方法が異なることである。例えば、入力するデータが、品詞や意味カテゴリのように限られた集合の中から選び出される場合、選択用のメニュー、あるいは、次で述べる辞書ヘルプ・ウィンドで選択することによって、入力される。このことによって、タイプ量を削減でき、タイプ誤りを減らすことができる。

次に、データの適合性の確認であるが、例えば、値として、特定のクラスのフレームが要求される場合には、フレーム以外の値を入力すると、タイプされた名前のフレームを作成するか、あるいは、入力を取り消すかをシステムが質問してくる。

新しく入力されたフレーム、および、下位のフレームについての辞書ウィンドを現在の辞書ウィンドから呼び出して、編集することができる。

(2) 辞書ヘルプ・ウィンド

このウィンドは、辞書ウィンドにおいて、入力できる値が有限の集合の要素であるスロットに値を入力する場合に、呼び出される特殊なメニューである。このウィンドには、入力可能な値の集合と、各値についての日本語の注釈、および、例が表として表示される。この表で値の一つをマウスで選択すると、呼び出した辞書ウィンドのスロット値として、それが入力される。また、例をマウスで選択すると、その辞書項目の内容が辞書ウィンドで表示される。

4.2 辞書編集システムの機能

辞書編集システムの機能について、辞書入力の手順に従って、述べる。

(1) 辞書ウィンドを呼び出す:
新しく作る辞書エントリを指定して、新しい辞書項目フレームを作成し、辞書ウィンドに表示させる。

(2) 辞書にデータを入力する:
マウスで入力したいスロットを指定し、コマンドのメニューの中から"Put data"を選択する。この

とき、そのスロットによって、入力用のウィンドへタイプしたり、メニューの中から選択することによって、入力される。

(3) 入力データの適合性をシステムが確認する:
例えば、格フレームの編集の場合、格パターンの入力の過程で、格関係が重複してないかどうかの確認などを行う。

(4) 上位のフレームやフレーム内の情報によって、取り得る値の集合が変化する:
例えば、意味フレームは、その上位フレームである辞書項目フレームの品詞によって、意味カテゴリとして取る値のセットを変える。

KOUKUUKI-N-1	
*ROMAN	koukuuki
*YOMI	こうくうき
*KANJI	航空機
*POS	NOUN
*RENSETSU-CATEGORY	NOUN
*MEANING	KOUKUUKI-N-1

(a) 辞書ウィンドの例
ユーザが*POSスロットにNOUNを入れると、デーモン・プログラムが起動され、*RENSETSU-CATEGORYスロットにNOUNが入る。

動詞の素性		
Category	意味カテゴリ	例
PASSIVE	受身	与える
AFFECTED-PASSIVE	被害の受身	降る
POSSIBLE	可能	歩く
SPONTANEOUS	自発	思う
CAUSATIVE	使役	働く
PERFECTIVE	完結状態	置く
MOMENTAL	瞬間	点く
CONTINUAL	継続	食べる
CONTINUAL%+TEIRU	+ている	歩く
CONTINUAL%-TEIRU	-ている	ある
MOVEMENTAL	移動	歩く
TRANSITIONAL	変化	変わる
EMOTINAL	感情	感じる
THINKING	思考	思う
PERCEPTUAL	知覚	見る
EXISTENTIAL	存在	ある
JUDGEMENTAL	判断	決める
NON-WILLING	無意思	降る

(b) 辞書ヘルプ・ウィンドの例

図-5 辞書編集システムのためのウィンド

(5) 下位構造のフレームを編集する:
上位のフレームを編集中に作ったフレームは、内容のない形だけのフレームなので、上位フレームを編集し終わって、抜け出るときに、下位のフレームを編集するかどうかシステムが質問してくる。それに応じて、未完成のフレームを順に編集していくことになる。これは、再帰的に行われる。

5. おわりに

本報告では、まず、翻訳実験環境について検討し、次に、LUTEシステムにおける翻訳実験支援環境について、具体的な応用例として翻訳実験支援システムと辞書編集システムを取り上げ、これらを中心に述べた。これらは既に使用している。翻訳実験支援システムは、実験データの入力や結果の確認を容易にし、中間結果の確認や編集を可能にした。このシステムは、実験者が思いつきを速やかに試すことを可能にしたシステムで、これを用いることにより、実験の効率を大幅に上げることができる。

辞書編集システムは、辞書項目を編集する際に、ユーザの入力をガイドし、入力されたデータに誤りがないか確認するシステムである。このシステムを用いることにより、辞書への誤入力を削減できる。

参考文献

[1] 小暮、横尾、島津、野村「辞書編集用フレームエディタ」、自然言語処理研究会 NL45-1、1984

[2] Nomura,H., "Towards the High Ability Machine Translatation", Eurotra Joint Japanese-European workshop, Nov, 1983

[3] Shimazu,A., Naito,S., Nomura,H., "Japanese Language Semantic Analyzer based on an Extended Case Frame Model", Proceedings of the 8th IJCAI, pp.717-720, Aug, 1983

[4] Iida,H., Ogura,K., Nomura,H., "A Case Analysis Method Cooperating with ATNG and its Application to Machine Translation", COLING-84, pp.153-158, 1984

[5] 片桐、野村「機械翻訳システムLUTEにおける意味構造換処理」、自然言語処理研究会 NL35-5、1983

[6] Naito,S., Shimazu,A., Nomura,H., "Classification of Modality function and Its Application to Japanese Language Analysis", Proceedings of the Conference, 23rd Annual Meeting of the Association for Computational Linguistics, pp.27-34, 1985

[7] Lisp Machine Manual Vol.4.."Program Development Tools", ZMACS manual