コンピュータと教育 26－3
(1993. 4. 10)

日本語の学習を支援するシステム

榎本 圭孝　劉 軼　伊丹 誠　伊藤 紘二

東京理科大学 基礎工学部

〒278 千葉県野田市山崎 2641
東京理科大学野田校舎

TEL:0471-24-1501 (内線 4226)

あらまし　従来の日本語CAIは，あらかじめ用意された教授シナリオに沿って処理が行なわれることが多く，解答が主として多岐選択的であり，学習者は与えられた問題を解くだけで意欲を持ちにくい．

そこで我々は，入力される自然言語の語る「世界」を限定し，学習者は限定された世界の中で自由に文章を作ってシステムからの質問に答え，また，システムに問いかけることができる学習者主導の日本語CAIを開発している．世界を限定した対話により学習者の誤解・誤答の原因の推定が可能になり，学習者にとっても言語表現と意味の対応が把握しやすくなる．本稿は，このようなシステムを実現するためのインターフェイス部と格助詞の使い方における基本的な誤りを正す機能を持った入力文解析部について報告している．

和文キーワード　マイクロワールド，学習者主導，ヒューマンインターフェイス，格情報，ボトムアップ解析，語彙機能文法

A Computer Assistant for Learning Japanese as A Foreign Language

Yoshitaka ENOMOTO　Yi LIU　Makoto ITAMI　Kohji ITOH

Department of Applied Electronics, Science University of Tokyo

2641 Ymazaki, Noda, Chiba 278, Japan

TEL:0471-24-1501 (ext.4226)

Abstract　Most of the existing Computer-Assisted Instruction (CAI) systems for learning Japanese employ fixed teaching scenarios and the students have nothing to do but to select his answer from the given choices. As a result, the students are likely to be insufficiently motivated.

In this paper, we propose a student-initiative system for learning Japanese. The system sets up a micro world and the student is allowed to compose sentences with freedom as far as they concern the set-up world, in order to ask questions to the system as well as to answer questions posed by the system. Thus constraining the context enables the system to analyze the student's phrases and also facilitate for the student to relate linguistic expressions with their meaning.

As one of the core functions required for the system, we have implemented a mechanism for diagnosing usage of postpositions. The paper reports the user interface and the natural language processing exploited in implementation.

英文 key words　micro world, student initiative, human interface, case information, BUP, LFG

1 はじめに

近年，日本の経済成長や技術発展に伴って，日本の高度な諸技術を学ぶ目的で，日本に留学する外国人が非常に増えている．また，海外においても日本語を学ぶ人々が多い．このような状況の中で，多くの教育機関が日本語教育コース，日本語教科書，日本語CAIなどのような形で外国人のための日本語教育を提供している．特に日本語CAIは，日本語教育教師不足や日本語に対する学習ニーズの多様化などの問題から注目されている[1].

従来の日本語CAIシステムの多くは，あらかじめ用意された教授シナリオに沿って処理が行なわれ，解答が主として多岐選択的で，学習者は与えられた問題に対する自分の考えを自由に発話できないので意欲を持ちにくい．

そこで我々は，学習者に高い自由度を与える対話機構を持つシステムを実現するために，入力される自然言語の話る「世界」を限定し，その中で学習者はシステムからの質問に対して自由に文章を作って答えたり，逆にシステムが学習者の作った簡単な質問に答えるような日本語CAIシステムを目指している．このような「世界」を限定した対話により学習者の誤解・誤答の原因の推定が可能になり，一方，学習者にとっても言語表現と意味の対応が把握しやすくなると考えられる．

本稿では，支援項目として日本語の学習において習得が困難とされている助詞(格助詞)に着目し，上述のことを実現するためのインターフェイス部と基本的な誤りを正す機能を持った入力文解析部について述べる．ただし，自立語の意味の把握については，母国語あるいは画像による説明で行なうものとする．なお，本稿が想定している日本語の文法は，日本古来の伝統的な文法理論を受け継いだ時枝文法[2]に基づいている．

2 システムに必要な機能

まず我々は，学習者に高い自由度を与える双方主導対話を実現するにはどのような機能が必要であるかを検討した．その結果，以下にあげるような機能を考え，これらに基づいてシステムの構築を行なっている．

2.1 インターフェイス

インターフェイスは学習者とシステムの相互伝達の流れを処理する．どちらの方向においても，システムの内部的な表現と学習者が理解できる自然言語の間の翻訳を行なうものである．我々のシステムの場合，インターフェイスの構築にあたっては次の機能を重視する必要がある．

(1) 学習者が自由かつ簡単に文章を作ることができる．

(2) システムから学習者にわかりやすく誤りを指摘することができる．

(3) 学習者が入力文で表現しようとした意図を，何らかの形でシステムと共有する手段を提供する．

(3)の実現に際しては，システムから学習者に対して日本語だけでなく，母国語の利用，あるいは図や写真，ビデオ映像といったメディアを使った対話を提供することが必要である．特にメディアを使うことで，言葉のみで説明するよりも具体性の高いものを示して感性的に理解させることができ，学習効果も上がり能率的に進められる．

2.2 誤り診断・訂正機能

語学教育の場合は，数学のような答えを導くプロセスの誤りを同定する場合と同定手法が異なる．後者の場合は，学習者モデル(student model)を使って，学習者の理解状況をモデル化し，誤りの原因を推定する必要がある．しかし，語学教育の場合には，入力情報そのものに誤りが含まれているため，入力文を解析すれば誤りの原因を同定できるという違いがある．ただし，このような手法を実現するには，2.1節で述べた(3)の機能が重要であり，学習者との対話のなかで誤りを同定し訂正することが必要である．

2.3 理解と生成の統合

従来の多くの自然言語処理システムは，理解と生成を独立したシステムとして構成している．しかし，日本語CAIシステムにおいては，不完全なあるいは間違いを含む入力に対し，問い返しを行ない，正しい入力をガイドするだけの能力を持つ必要がある．このためには，理解と生成とが同じ枠組みで緊密に連絡し合うことのできる仕組みが不可欠である．さらに，開発コストの節約のために，基本的な部分を分野独立な構成にする必要がある．

3 インターフェイス部

3.1 インターフェイスの概要

学習者とシステムとを結ぶために，機能別に以下の3つのウィンドウを設ける．

- メッセージウィンドウ
- メインウィンドウ
- ヘルプウィンドウ

メッセージウィンドウは，学習者に対して次に行なうべき操作を学習者の母国語または日本語で提示する．メインウィンドウは，学習者が文章を作る場であり，また，システムからのメッセージも表示する．ヘルプウィンドウは，学習者がわからない単語があってシステムに問いかけてきたときに，学習者の母国語でその単語の意味を表示する．

メインウィンドウで学習者が文章を作る方法としては，キーボードを使った方式，単語メニューからマウスを使って選ぶ方式が考えられる．しかし，日本語をあまり知らない学習者にとって前者の方式では，キーの配置を知らないと一つの文を作るにも時間がかかってしまい，学習意欲を失わせてしまうことになる．一方，世界を限定するのであるから単語をメニュー化することができる．そこで本システムでは，後者のマウスによる入力方式を用いることにす

る．この方式だと，学習者の入力が楽になるだけでなく，システムにとっても入力文の語彙的なレベルでの誤りを防ぐことができるので，タイプ誤りによる処理エラーをなくすことができる．そのほかに，単語の削除や挿入，語順変換などもマウスでボタンを選ぶことによってできるようにし，これらの操作がしやすいように入力文は単語ごとに分かち書きの形で表示する．

また，文章を作るだけでなく，コンピュータに慣れていない人でも操作しやすいシステムを実現するために，基本操作はすべてマウスを使用する．

3.2 インターフェイスの実現

3.2.1 MMO(Media Metaphor Object)

システムは3.1節で述べたように，3種類のウィンドウを使うことによって，学習者に概念や知識内容を提示したり，学習者からの入力をその上で受け付けて解釈する．このような学習者との対話の媒体として，静的あるいは動的なメディア表現とその上における学習者の介入の受け付けを行なうオブジェクトをメディア・メタファ・オブジェクト(Media Metaphor Object - MMO：以下MMOとする)と名付ける[3]．本システムでは，各ウィンドウに対応した3つのメディア・メタファ・オブジェクトを定義する．

3.2.2 MIP(Media Interface Prolog)

入力文解析部ではプログラミング言語としてPrologを用いている．しかし，文字の入出力，メニューなど実際に視覚情報に訴える機能を扱うMMOはXView[1]のツール呼び出しを用いたC言語で記述するのが良いと判断し，C言語で記述されたオブジェクトモジュールにアクセスできる述語を組み込んだインタープリタを用いた．このインタープリタのことをメディア・インターフェイス・Prolog(Media Interface Prolog - MIP：以下MIPとする)と呼ぶ[4]．これにより，Prologからウィンドウの生成や抹殺などの制御ができる．これを使ってできたインターフェイスを図1に示す．

単語入力ボタンを押すと，その品詞に相当する単語のメニューが表示され(用言の場合は終止形で表示)，入力したい単語をマウスでクリックすると，入力文の部分にその単語が表示される．ただし，活用がある品詞の場合は単語を選ぶと語尾変形テキストウィンドウが表示され，キーボードにより活用を変えることができるようになっている．HELPボタンは，学習者がわからない単語があった場合，その単語の意味を母国語でヘルプウィンドウに表示する(開発中)．また，語順変換や一語削除ボタンを押すと，入力文が単語ごとに分かれたメニューとなって表示され(図2)，語順変換であれば2ヶ所，一語削除であれば削除したい単語をマウスでクリックするだけで各々の機能を果たすことができるようになっている．一語挿入も挿入したい単語を選んでから，挿入する位置(“↓”で表示)をマウスでクリックするだけでいいようになっている．

[1]OPEN LOOKに準拠したグラフィカル・ユーザ・インターフェイス・ツールキット

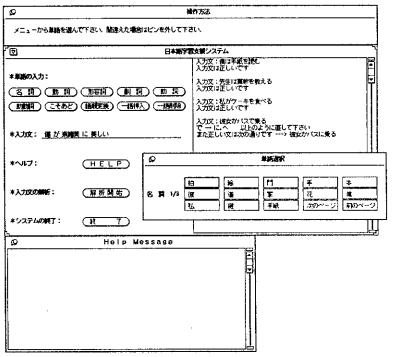

図1: 日本語学習支援のためのインターフェイス

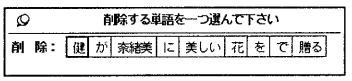

図2: 一語削除メニュー

このように作られた入力文は，解析開始ボタンを押すことで，それぞれの単語を要素としたPrologのリストの形で入力文解析部に送られる．そのため，解析部での形態素に関しての問題はない．

4 入力文解析部

4.1 解析部の概要

4.1.1 語彙機能文法

2.3節でシステムにとって必要な機能に日本語の理解と生成の統合を上げたが，これを実現するために，共通の意味表現を持ち，構文的拘束と意味拘束を結び付ける共通で汎用の仕組みとして，ユニフィケーション・ベースの文法である語彙機能文法(Lexical Functional Grammar - LFG：以下LFGとする)を用いる[5][6][7]．

LFGは主語・目的語などの文法機能を無定義要素(primitive)として扱い，統辞部門での変形操作なしに語彙機能部門で文法機能の操作を可能にする．表示のレベルとしてはF構造(機能構造：functional structure)とC構造(構成素構造：constituent structure)を持ち，F構造は辞書情報にその源泉を持ち，F構造の合成により文中の意味単位の相互限定関係を表わす．C構造は文法規則に構文的なF構造合成手続き(機能的注釈)を付加したものである．

他のユニフィケーション・ベースの文法に一般化句構造文法(Generalized Phrase Structure Grammar - GPSG：以下GPSGとする)[5]があるが，これは意味拘束を構文的構造にいれるため，語の意味や語の間の意味拘束が分野ごとに異なる場合，分野に依存した辞書情報さえ用意すれば，構文的な拘束をかける仕組みが汎用なものですむLFGの方がGPSGに比べモジュラリティの点で都合が良い．

4.1.2 不適格文の処理

学習者が入力した曖昧さおよび誤りを含んだ不適格な文を解析する方法としては，文中の用言を認識することで語(用言のみならず)を意味的に安定させ，用言と用言に挟まれた部分と右端の用言とからなるブロックにおいて，その部分が用言に対して果たす役割を解析することが必要である．そこで，これらのことを実現するために通常のBUP[8]を改良し，用言を中心にボトムアップな解析を行なう手法(以下用言中心BUPとする)を提案する．これは，日本語の構造が一般に「格助詞句[2](体言句[3]+格助詞)+用言」を単位とするつながりと考えられることから，入力文を文末から辞書引きして用言の格情報を取り出し，その用言から文頭へ遡る方向で格助詞句の単位でスタックに入れて格情報との照合を行ない，その句で使われている助詞が適当なものかどうかを判断する．これにより係り受けの方向性と係り受け非交叉の原理[4]を保持しつつ解析を進めることができる．

4.2 解析部の実現

4.2.1 日本語LFGへの拡張

LFGは英語を基礎にした文法であり，日本語の理解と生成のために改良する必要がある[9]．

(a) C構造

まず，文法規則に機能的注釈(F構造合成手続き)unifyを付加したC構造を定義する．一例として，「健が奈緒美に美しい花を贈る」を解析するのに必要な最小限のLFGのC構造規則の定義を図3にDCG[5]の形式で記す．

s，mv，a，pp･･･は構文範疇を表わし，それらが引数として持つのがF構造である．構文範疇はそれぞれ，s: 文，v: 動詞，a: 形容(動)詞，mv: 連用修飾句，pp: 連用修飾格助詞句，np: 体言句，p: 連用修飾格助詞，n: 体言，mn: 体言修飾句，mns: 文による連体修飾句を表わす．

F構造は，ペアリスト[⟨ 属性名 ⟩,⟨ 値 ⟩]を要素とするリストの形をしており，⟨ 値 ⟩は，またF構造でありうる．

unifyは解析においてF構造の合成に用いるが，詳細は後出の(c) unifyの定義の節を参照されたい．

(b) 辞書

語の単位の言語表現とF構造の源泉を結ぶための辞書が必要である．例として「健が奈緒美に美しい花を贈る」に使われている単語の辞書の記述をDCGの形式で図4に示す．辞書情報の中で，predの値の部分は意味を表わす

```
s(SF)→mv(MVF),v(VF),
  unify([VF,yougen_kaku_unify(MVF)],SF)
s(SF)→a(SF)
mv(MVFF)→mv(MVF),pp(PPF),
  unify([MVF,attr(case,PPF)],MVFF)
mv(MVF)→pp(PPF),
  unify([attr(case,PPF)],MVF)
pp(PPF)→np(NPF),p(PF),unify([PF,NPF],PPF)
np(NPF)→n(NPF)
np(NPF)→mn(MNF),n(NF),
  unify([NF,MNF,taigenkakari_slot(MNF)],NPF)
mn(MNF)→mns(MNF)
mns(MNSF)→s(SF),
  unify([check(活用,SF,連体),add(体言係り,SF)],MNSF)
```

図3: C構造規則

```
v([[pred,贈る(V,X,Y,Z)],
  [対象/[を],Z],[終点/[に],Y],
    [行為者/[が],X],[活用,終止]])   →  '贈る'
a([[pred,美しい(A,X)],
  [記述対象/[が],X],[活用,連体]])   →  '美しい'
n([[pred,健(N1)]])                 →  '健'
n([[pred,奈緒美(N2)]])             →  '奈緒美'
n([[pred,花(N3)]])                 →  '花'
p    →  'が' | 'に' | 'を'
```

図4: 辞書の記述

述語表現である．また，'対象'や'終点'，'行為者'は「贈る」がとる格であり，付記された格助詞(表層格)の意味的な格(深層格)の名前である(この情報を格情報と呼ぶことにする)．

深層格は[10]を参考に31種類を定義する．ただし，用言の辞書には必須格について，表層文中の語順および解析手順も考慮して記載されている．

(c) unifyの定義

unifyは前にも述べたようにF構造の合成に用いられる．LFGでは，辞書に付記されたF構造の源泉からC構造規則を用いて文要素の，そして最後に文全体のF構造に変換することができればその文法にかなっていることになる．F構造ができなければその文法にかなっていないのであり，その文は不適格文となる．また，F構造は実際の文と意味構造との橋渡しの役も担っており，F構造の導出は意味解析においてのプロセスでもある．unifyの定義は次のようになっている．

unify(⟨ リスト1 ⟩, ⟨ リスト2 ⟩)

これは，⟨ リスト1 ⟩の1要素について処理した結果のF構造は，次の要素の処理に用いられて，その結果がまた次の要素の処理に使われ，最後の要素の処理が済めばその結果を⟨ リスト2 ⟩に移して終了するようになっている．

[2] まとまった意味や機能を表わす語の並びを一様に「句」という言葉で表わす[2]

[3] 連体修飾句＋体言

[4] 語と語の係り受けの関係は，同一文中の他の語と語の係り受け関係と交叉しないという原理

[5] 「句構造規則を生成する左辺にはただ1つの変数が現れる」という約束で書かれた規則を文脈自由文法というが，これをPrologのマクロ記述にしたものをDCG(Definite Clause Grammar：確定節文法)という

〈リスト1〉の要素INFの処理については，1つ前の処理の結果をPREFとしたとき，次のようにして結果OUTFを得る．ただし，最初のPREFは空リストである．

- INFがF構造の場合
  その要素とPREFの要素を並べた(すなわちappendした)リストをOUTFとする．
- INF = add(NAME, XF)の場合
  リスト[NAME, XF]というペアをPREFの要素と並べたリストをOUTFとする．
- INF = replace(NAME, XF)の場合
  PREFにおける属性名NAMEの値を，XFで置き換えたものをOUTFとする．
- INF = check(NAME, XF, VAL)の場合
  XFにおける属性名NAMEの値がVALであることを確かめる．
- INF = attr(NAME, XF)の場合
  XFの[〈属性名〉, 〈値〉]のペアのうち，〈属性名〉= NAMEであるものの〈値〉を見いだし，この〈値〉を属性名とし，その値としてXF自身を入れたペア[〈値〉, XF]を作り，PREFの要素と並べたリストをOUTFとする．
- INF = taigenkakari_slot(XF)の場合
  XFの中で，値にF構造の入っていないスロットを見いだし，その値をPREFのpred値の述語の引数と一致させた上で，このPREFをOUTFとする．
- INF = yougen_kaku_unify(XF)の場合
  XFの[〈属性名〉, 〈値1〉]のペアのうち，〈属性名〉を共有するペア[〈属性名〉, 〈値2〉]がPREFの中に見いだされるものについては，〈値1〉の中のpred値の述語の引数と〈値2〉とを一致させ，〈値2〉を〈値1〉で置き換えた上で，このPREFをOUTFとする．ただし，同じ属性名に異なる値があってはならない．

そのほかにunifyの中で使う属性として，用言係り(用言に係るものを値とする属性)，体言係り(体言に係るものを値とする属性)，係り体言(「の」を介して体言に係るものを値とする属性)，活用(その語が持つ終止形や連体形などの条件を値とする属性)，case(いわゆる格で用言に対する体言の役割を値としてとる属性)の5つを定義する．

4.2.2 用言中心BUP

4.1.2節で不適格文の処理方法として用言中心BUPを提案したが，この節ではその具体的なプログラミングについて述べる．

(a) BUP節への変換

まず，図3の文法規則をBUP節に直すわけだが，入力文を文末から処理できるように図5のような変換規則でBUP節に直す．ただし，文は差分リストで

```
DCG規則 : a(A) → b(B), c(C).
BUP節   : c(G, C, D, X, Z):-
              goal(b, B, X, Y),
              a(G, A, D, Y, Z).
```

図5: BUP節への変換規則

```
dict(v,[[pred,贈る(V,X,Y,Z)],
    [対象/[を],Z],[終点/[に],Y],
      [行為者/[が],X],[活用,終止]],[贈る|U],U).
dict(a,[[pred,美しい(A,X)],
    [記述対象/[が],X],[活用,連体]],[美しい|U],U).
dict(n,[[pred,健(N1)]],[健|U],U).
dict(n,[[pred,奈緒美(N2)]],[奈緒美|U],U).
dict(n,[[pred,花(N3)]],[花|U],U).
dict(p,_,[が|U],U). dict(p,_,[に|U],U).
dict(p,_,[を|U],U).
```

図6: BUPで用いる辞書の記述

```
{X,Z}={[健,が,奈緒美,に,美しい,花,を,贈る|Z],Z}
```

のようなリストの対で表現され，各変数は，G: 最終的に求めたい範疇記号，{A,B,C,D}: 文または単語の意味構造，X: 文，Y: 文Xの最初の単語を除いた部分文のリスト，Z: 終了条件を表わしている．

また，辞書もBUPで解析できるように直す必要があり，図4の例を用いて書き換えると図6のようになる．助詞の辞書の"_"にはF構造の源泉である格属性名を入れるのが普通であるが，用言の辞書に格属性名と格助詞がペアで記述されており，解析の途中でそこから得ることができるので省いた．

(b) goal節の定義

次にgoal節であるが，入力文中のどの助詞が誤っているのかを判断する機構を持たせる．図7にgoal節の定義を示し，その中で用いられている変数について説明する．

- Pstruc
  格助詞およびその意味的な格が，用言の辞書に記述されている格情報をもとに，リスト[[case, 格属性名], 格助詞]の形で入る．
- AVstruc
  用言がとる格情報および名詞が，解析順(すなわち文

```
goal(G,Pstruc/NewPstruc,AVstruc/Err,
                RetAVstruc/RetErr,Fstruc,X,Z):-
  interpret(Pstruc/NewPstruc,AVstruc/Err,
       NewAVstruc/NewErr,X,SynCat,Leafstruc,Y),
  P=..[SynCat,G,NewAVstruc/NewErr,
       RetAVstruc/RetErr,Fstruc,Leafstruc,Y,Z],
  call(P).
```

図7: goal節の定義

末から) にリストの形で [[名詞],…[名詞],[用言, 用言の格情報, 活用]] のように積まれていく (これを格リストと呼ぶことにする).

- Err
  後述の格照合における誤り情報として, 助詞と名詞を要素としたスタック [p, 助詞, 名詞] の形で入る.

- Leafstruc
  単語を辞書引きしたときに得られる F 構造の源泉.

- Fstruc
  入力文の最終的な F 構造が入る.

また 'New' あるいは 'Ret' が前についた変数は, 前者が述語 interpret における解析の結果, 後者が最終的な結果を意味する.

述語 interpret の機能は辞書引きおよび格照合であり, 単語を辞書引きしたときに得られる品詞によって以下のように処理を変えている (ただし, 副詞および助動詞については今のところ扱っていない).

- 名詞の場合
  その後 (処理手順で言えばその前) の助詞の格がその名詞に合うかどうかを調べる.

- 形容 (動) 詞の場合
  活用が連体形ならば, その後の一番近い位置にある名詞を被修飾名詞と判断して, その名詞が形容 (動) 詞の持っている格に合うかどうかを調べる. 合った場合は一般に形容 (動) 詞の連体形は後ろにのみ係るので文頭にその格情報は流さない. それ以外の活用であれば, AVstruc にその格情報を加えて文頭に向かって流す.

- 動詞の場合
  形容詞の場合とほぼ同じだが, 動詞の場合は連体形でもそれより前に格を持つことが多いので, 他の活用形と同じように AVstruc にその格情報を加えて文頭に向かって流す.

- それ以外の品詞の場合
  AVstruc をそのまま文頭へ流す.

(c) 格情報との照合方法および訂正

照合は用言に最も近い格助詞句から始め, その句の助詞をキーに格リスト AVstruc からその用言の格情報を取り出して格の引き当てを行ない, その格が句の中の名詞をとることができるかを用言ごとに書かれた格・名詞対応辞書 (図 8参照) により調べる. そして, その格がその名詞をとるのであれば, その格助詞の使い方は合っているとし, 用言の格情報からその格を削除 (1 文 1 格の原理を適用[6]) したものを次の照合のために使う. もし, 格の引き当てに失敗したり, 格がその名詞をとらなかったりしたときは, 用言の格情報はそのまま次の照合に使い, 格助詞句は誤り情報 Err として保持する.

通常, 格助詞句は最も近い用言に係ることが多いので上記のような照合方法を用いるが, 場合によっては直後の用言を越えてより後方の用言に係るものがあり (飛び越し係り), このときは以下の方法を用いる [11].

(1) 格照合の過程で直後の用言とマッチしない場合は, それよりも後方の用言でその格に合うものがあるかどうかを調べる.

(2) 「〜は」という表現は文の主題などを提供し, 通常連体形の用言には係らない. したがって「〜は」という句が現れたとき, その直後の用言が連体形の場合には, それより後方の用言と格照合を行なう.

(1) の例としては「私は桜が咲いたので公園へ出かけた」という文があてはまる. この場合, 「私は」という句の直後の用言は「咲いたので」となってしまうので, それより後方の用言である「出かけた」と照合を行なう必要がある.

(2) の例としては「私は彼からもらった本を読んだ」という文があてはまる. 連体形用言である「もらった」に「私は」は係らないので, それより後方の用言である「読んだ」と照合を行なう. なお, 格情報に付記される助詞に「は」は書いておらず, 「は」についてはその都度「が」に読み変えて照合を行なう.

これらの方法で文頭に向かって入力文を処理していき, 最終的に格リスト AVstruc 中の用言の格情報が空リストであれば入力文は正しいとし, もし格情報の中に残った格があれば, 誤り情報 Err より格助詞句を取り出し, 句の中の名詞がその格をとることができるかどうかを調べてから, その句の助詞を格に付記されたものに変える. 格リスト AVstruc および誤り情報 Err は, 解析した順番 (文末から) に情報が入っているので (すなわちスタックと同じ働き), 訂正するときの誤り情報と格情報との対応は格助詞句中の名詞をキーとして容易に対応をとることができる.

用言が連体形の場合は前にも述べたように, その後の一番近い位置にある名詞を被修飾名詞と判断して格・名詞対応辞書を用いて照合を行なうが, このときは助詞の情報は用いない.

なお格・名詞対応辞書は, 自然言語の語る世界を限定しているためにできることであり, 述語 semantic という名前で図 8のような形式をしている.

```
semantic(用言(終止形),[格属性名/[体言リスト],…]).

例 : semantic(贈る,
          [行為者/[私,彼,僕,健],
                終点/[彼女,奈緒美,あなた],
                      対象/[プレゼント,花,絵]]).
     semantic(美しい, [記述対象/[花, 絵, 景色]]).
```

図 8: 格・名詞対応辞書

[6] 1 単文中に同一の格はたかだか 1 度しか現れないという経験則

5 システムの実行

システムを実行するには，コマンドラインでmipと打ち込んでMIPを立ち上げ，そこでPrologで書かれた入力文解析部をコンサルトし，goと打てば述語mio_createを実行することにより図1で示したウィンドウが立ち上がる．学習者はウィンドウ上で単語入力ボタンをマウスでクリックすることにより入力文を作成する(ただし入力する文は，学習者に何らかの動機づけをしたものに対する答えなどである)．そして作成し終わったら，解析開始ボタンを押すことで入力文を解析部に送り，解析を開始する．以下に「健が奈緒美で美しい花を贈る」という誤った文を例として，格リストAVstrucおよび誤り情報Errに着目して解析の流れを説明する．

- **Step1**
  まず，リストの形で解析部に送られた入力文を，Prologの組み込み述語reverseで文末が文頭になるようにし，川言から辞書引きできるようにする．すなわち以下のようになる．

  入力文　：[贈る, を, 花, 美しい, で, 奈緒美, が, 健]
  AVstruc：[]
  Err　　：[]

- **Step2**
  「贈る」の辞書引きを行ない，辞書(図6参照)より格情報を得てAVstrucに入れる．

  入力文　：[を, 花, 美しい, で, 奈緒美, が, 健]
  AVstruc：[[v, 贈る, [対象/[を], Z], [終点/[に], Y],
  　　[行為者/[が], X], [活用, 終止]]]
  Err　　：[]

- **Step3**
  「を」と「花」の辞書引きをしたあと格照合を行なう．「を」をキーとしてAVstrucより格を引き当てると‘対象’を得る．すると，「贈る」の格・名詞対応辞書(図8参照)の‘対象’に「花」があるので，AVstrucから‘対象’を除いて新たなAVstrucとする．

  入力文　：[美しい, で, 奈緒美, が, 健]
  AVstruc：[[n, 花], [v, 贈る, [終点/[に], Y],
  　　[行為者/[が], X], [活用, 終止]]]
  Err　　：[]

- **Step4**
  次に「美しい」の辞書引き行なう．例文の場合，「美しい」の言い切りの形(すなわち終止形)で終わっておらず，かつ，語尾の形から‘連体形’という活用情報を得る．したがって，その後(処理手順ではその前)の一番近い位置にある名詞「花」を被修飾名詞と判断し，「美しい」の格・名詞対応辞書と照合を行なう．すると，「美しい」の格である‘記述対象’に「花」があることから，「花」は「美しい」の被修飾名詞として適格であることがわかる．この場合AVstrucに「美しい」の格情報は加えず，そのまま文頭に流す．

  入力文　：[で, 奈緒美, が, 健]
  AVstruc：[[n, 花], [v, 贈る, [終点/[に], Y],
  　　[行為者/[が], X], [活用, 終止]]]
  Err　　：[]

- **Step5**
  「で」と「奈緒美」の辞書引きをしたあと格照合を行なう．「で」をキーとしてAVstrucより格の引き当てを行なうが，「贈る」の格情報に「で」を持つ格がないために失敗する．そのため，誤り情報Errにその格助詞句を入れる．

  入力文　：[が, 健]
  AVstruc：[[n, 奈緒美], [n, 花],
  　　[v, 贈る, [終点/[に], Y],
  　　[行為者/[が], X], [活用, 終止]]]
  Err　　：[[p, で, 奈緒美]]

- **Step6**
  「が」と「健」の辞書引きをしたあと格照合を行なう．「が」をキーとしてAVstrucより格を引き当てると‘行為者’を得る．すると「贈る」の格・名詞対応辞書の‘行為者’に「健」があるので，AVstrucから‘行為者’を除いて新たなAVstrucとする．

  入力文　：[]
  AVstruc：[[n, 健], [n, 奈緒美], [n, 花],
  　　[v, 贈る, [終点/[に], Y], [活用, 終止]]]
  Err　　：[[p, で, 奈緒美]]

- **Step7**
  入力文が空リストになったので誤り訂正を行なう．まず，AVstruc中の格情報に残っている格‘終点’を，Err中の名詞「奈緒美」がとることができるかを「贈る」の格・名詞対応辞書により調べてみる．すると，とることができるので格助詞を「で」から「に」に変えることで正しい文「健が奈緒美に美しい花を贈る」を得ることができる(図9参照)．

以上のような段階により入力文の解析，誤り訂正を行なっている．なお，F構造は誤りがある文を解析した場合，その部分の属性caseの値を‘未定’にしているので，正しい文を得てから再びgoal節に代入することで正しいF構造を求めることができる．図10に解析によって得られたF構造を示す．図のF構造において重要なことは，「贈る(V, X, Y, Z)」という属性値を持たせることで，「健」と「奈緒美」と「花」が関係付けられているということである．このような関係付けを行なうことで不適格な文を排除できる．

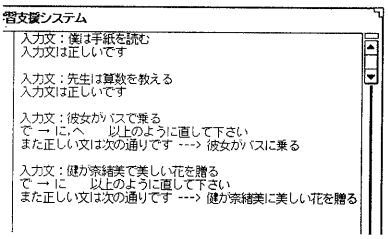

図 9: 解析結果を出力した画面

```
[[pred,V, 贈る (V,X,Y,Z)],[活用, 終止],
   [対象,[[体言係り,[[pred,A, 美しい (A,W)],
      [活用, 連体]]],[pred,Z, 花 (Z)],[case, 対象]]],
         [未定,[[pred,Y, 奈緒美 (Y)],[case, 未定]]],
            [行為者,[[pred,X, 健 (X)],[case, 行為者]]]]
```

(a) 誤りを含む文における F 構造

```
[[pred,V, 贈る (V,X,Y,Z)],[活用, 終止],
   [対象,[[体言係り,[[pred,A, 美しい (A,W)],
      [活用, 連体]]],[pred,Z, 花 (Z)],[case, 対象]]],
         [終点,[[pred,Y, 奈緒美 (Y)],[case, 終点]]],
            [行為者,[[pred,X, 健 (X)],[case, 行為者]]]]
```

(b) 訂正した文における F 構造

図 10: 解析によって得られた F 構造

6 おわりに

学習者にとって使いやすいインターフェイスおよび助詞の誤りを指摘・修正するための一手法として，用言中心 BUP の提案および構築を行なってきた．

インターフェイスについては，基本操作のほとんどをマウスによって行なうことができ，入力文解析部は，簡単な文について誤りを見つけて正しい文に直すことができる．これにより学習者は自分の考えを簡単に表現することが可能になり，システムと学習者の間に自由度の高い対話を実現する一歩になると考えられる．

問題点としては，学習者が誤った文章を入力した場合，本当は何が言いたかったのかがわからないことである．これは，飛び越し係りの処理と関係が深い．例えば，「私で彼をもらった本を読んだ」という誤った文の場合，「私」は「もらった」に係るのか (「私が」になる)，「読んだ」に係るのか (「私は」になる) がわからず，また「彼」の場合，「彼がもらった」のか「彼から (に) もらった」のかが明確にわからない．したがって，このような場合には誤りを見つけた段階で学習者に問うような機構 (静止画像あるいは母国語の表現などで選ばせる) が必要である．

今後の課題としては，インターフェイスにおいて，ヘルプウィンドウ機能の完成を目指す．また，入力文の作成において，誤りのたびに語順変換・一語挿入・一語削除ボタンを押すのは面倒なので，これらの機能を統合したウィンドウを作成していく．解析の方は今回は格助詞だけに絞ったが，それ以外の助詞の扱い，連用修飾句，連体助詞句および任意格についても取り扱っていきたい．また，学習者の指導においては，知識ベースの内容を伝えるために生成の方からも考える必要がある．今回は生成に関しては触れなかったが，部分的な F 構造と辞書情報を用いて，両者を C 構造規則によってつなぐプロセスで生成を行なうことは実現されており [9]，これらを用いた支援についても探っていきたい．さらに，限定された世界での対話の展開に基づく制約を用いた広い視点からの誤りの検出と矯正について考えていきたい．

参考文献

[1] 日本語教育支援システム研究会：“「日本語教育・支援システムの開発」日本語教育とコンピュータ” (1992), 文部省科学研究費補助金・国際学術研究 (共同研究) No.02044153.

[2] 時枝誠記：“日本文法 口語編”, 岩波全書 (1950).

[3] 伊藤紘二, 伊丹誠：“知識ベースに制御されたメディアインターフェイスによる学習支援”, 情報処理学会研究報告, 91-CE-19 (1991).

[4] 石田一郎, 小野哲, 加藤陽治, 猪瀬康二, 宮本健, 伊丹誠, 伊藤紘二：“学習支援のためのメディアインタフェースオブジェクトの試作”, 電子情報通信学会技術研究報告, ET92-18, Vol. 92, No. 22 (1992).

[5] 郡司隆男：“自然言語の文法理論”, 産業図書 (1987).

[6] Sells, P.: “*Lectures on Contemporary Syntactic Theories*”, CSLI Lecture Notes Series No.3 (1985).

[7] 淵一博監修, 古川康一, 溝口文雄共編：“自然言語の基礎理論”, 共立出版 (1986).

[8] 田中穂積：“自然言語解析の基礎”, 産業図書 (1989).

[9] 伊藤紘二, 伊丹誠, 松田秀平：“学習支援インターフェイスとしての日本語理解生成システムの試案と試作”, 第 34 回 CAI 学会研究報告 (1991).

[10] ICOT 第二研究室：“文生成ツール “jpnsgen” マニュアル” (1988).

[11] 大島義光, 阿部正博, 湯浦克彦, 武市宣之：“格文法による仮名漢字変換の多義解消”, 情報処理学会論文誌, Vol. 27, No. 7, pp. 679–687 (1986).