

## PIVOT：ソフトウェアの構成と機能

4B-1

浜田和彦・落合尚良・小林正博・西村雄二・村木一至・小関正彦

・日本電気技術情報システム開発㈱ 〃㈱日本システムアプリケーション 〃 日本電氣㈱

## 1 はじめに

多言語間機械翻訳システムPIVOT<sup>1)</sup>のソフトウェアに  
関し、その機能、構成、データ構造について報告する。

## 2 システムの構成

P I V O T は形態素解析、構文意味解析、概念抽出、構文生成、形態素生成の5つのフェーズで処理が行われる。本システムでは、5つのフェーズ各々に異なったルールが使用されるため、ルールの資源管理は重要な問題である。そこで、ルールの記述形式を全てのフェーズで共通にし、ルールを解釈実行するルールインタプリタを1つにすることで、ルール資源を一元的に管理できるようにした。

PIVOTのソフトウェアは、図1に示すように、各フェーズごとに固有のモニタと全てのフェーズのモニタが共通に呼び出すルールインタプリタと呼ばれるモジュールから成る。

## ルールインタプリタ

ルールは図2に示すように弁別木の構造をもたせ、ルールの選択速度の向上を図っている。その弁別木上のノードはコンディション部とアクション部に分かれており、ルール記述の自由度を増すため、アクションを弁別木上のリーフだけでなく全てのノード上に書くことができるようにして、更に次回のルール適用の適用開始ノードを指定できるようにした。

ルールインタプリタは弁別木をトラバースし、コンディション部を評価していく。そしてリーフのひとつに到達した時点で、適用開始ノードからそのリーフノードまでのパス上のアクション部を実行する。

モニタ

モニタはフェーズごとに異なる対象データに対して、ルールインタプリタがルールを解釈実行するインターフェイ

ス管理を行うものである。ルールからは、基本的に4つの異なる位置のデータが参照できるようになっており、それを「ウインドウ」呼ぶ（以下A,B,C,Dと表す）。ウインドウに対応する対象データとのマッピングを行う動作をバインドという。モニタはバケットラック用のスタックを制御し、ウインドウに実データをバインドして、ルールインタプリタを呼び出す。

### 3 各フェーズにおけるモニタの機能

ここでは、5つのフェーズにおける対象データの構造と、それぞれのモニタが行うデータとウインドウのバインド方法について述べる。

### 3. 1 形態素解析

入力文の形態素構成を認定し、部分解析木を持つ単語リストを出力する。

形態素解析は

① 形態素分解  
入力文の形態素構成を認定し、形態素のリスト構造を出力する。このフェーズはソフトだけで動作する。

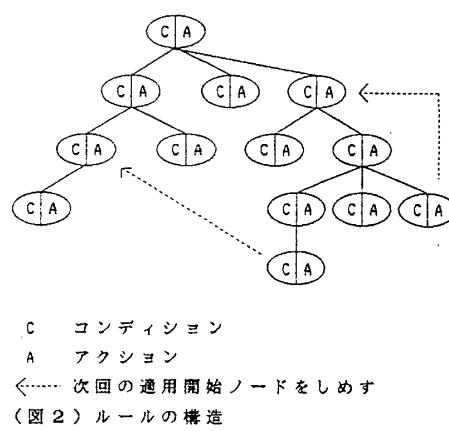
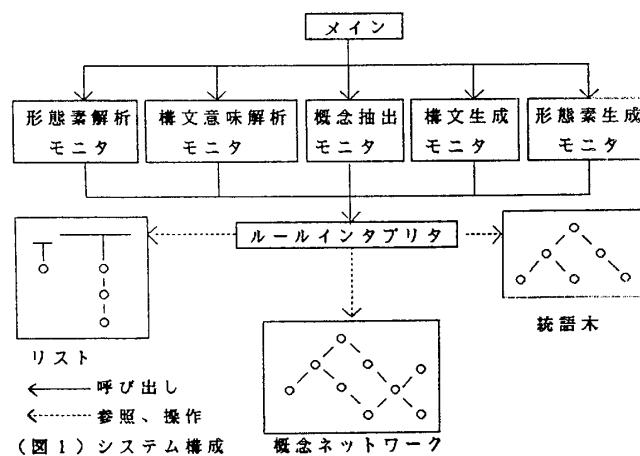
## ② 辞書ロード

リスト要素ごとに、辞書情報をロードする。統語、意味情報の多義性は、図3のw1,w1'のようにリストの要素を複数個作成することで管理する。このページをソフトでは

- ルールの適用

形態素解析において、ウインドウのバインドは次の原則を持つ。

  - (1) 初回バインドは左からA,B,C,Dの順で4つが横に並んで隣接するようにバインドされる。
  - (2) ウインドウB,Cはその順で常に横に隣接している。
  - (3) 一度ルール適用が行われるとそのルールにより次



## **PIVOT: Organization and Function of Software**

**PIVOT: organization and function of software**  
**Kazuhiko hamada<sup>1</sup>, Takayoshi Ochiai<sup>1</sup>, Masahiro Kobayashi<sup>2</sup>, Yuuji Nishimura<sup>2</sup>,**

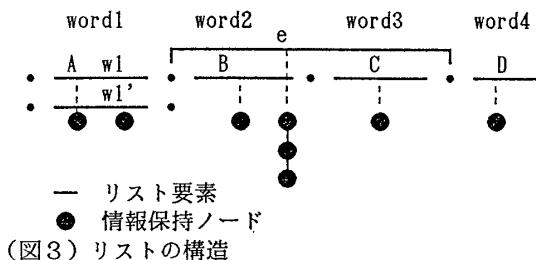
Kazuhiko Namada<sup>1</sup>, Takayoshi Ochiai<sup>2</sup>,  
Kazuyuki Munekata<sup>3</sup>, Masahiko Ozeki<sup>4</sup>

\*NEC Scientific Information System Dept.

NEC Scientific Information System Development, Ltd. Japan System Application, Ltd. NEC CORPORATION

のウインドウのバインド位置が指示される。(2回目以降のバインドは(1)に示した順にならなくてよい。)  
このフェーズでは、おもに次のオペレーションが可能である。

- (A) リスト要素の持つ情報の参照、操作
- (B) リスト要素の連結 (コンストラクション)  
隣接する2つウンドウB,Cの要素を連結し長いものをつくる。その際、木構造を作ることもできる。  
(図3の例では、ウンドウB,Cの要素をコンストラクションし、eを作成、その下に木構造を作ったものである。)



形態素解析においてはバックトラックをサポートしていない。このフェーズの終了はルールからの指示で成立する。

### 3.2 構文意味解析

形態素解析が出力したりストに対し、係り受けとその格関係を認定し、解析木を出力する。

このフェーズでは、形態素解析と同様にリスト上でルールを適用していく。使用できるオペレーションは形態素解析と同じである。ただし、このフェーズでは、次の原則を持つ。

- (1) ウンドウはA,B,C,Dの順に常に隣接して4つ横に並ぶ。(3.1の(3)と異なる)
- (2) ウンドウうち、内側2つのB,Cでのみ操作が可能で、外側2つのA,Dは条件(left context, right context)判定にのみ用いられるウンドウと定められている。

また、このフェーズにおいては、バックトラックをサポートしており、ルール適用時にルールの適用状況(ウンドウ位置、成功したルールパスなど)と、情報ノードの内容を記憶しておく。

このフェーズは、依存構造が1つにまとまった時点で終了する。

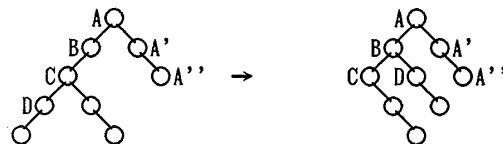
### 3.3 概念抽出

構文意味解析が出力した解析木に対し、より汎言語的な概念ネットワークに解析木を書き換える。

このフェーズではその解析木上のノードのデータ変更の他に、新しいノードのアロケートやノード間リンクの付け替えができる。4つのウンドウは、常に深さ方向に隣接して置かれ、デブスファーストにその位置を1つずつずらしていく。ただし、ルールから参照位置を強制的にいくつか戻したり、トップまで戻したりも指示できる。このフェーズではバックトラックはサポートされておらず、直接、木を書き換えていく。このフェーズでは4つの基本ウンドウに加えて、それぞれの基本ウンドウにバインドされたノードから、親または子、更にその子といった構造を指定するために、n個のウンドウが用意してある。

(図4の例では基本ウンドウがA,B,C,Dのように置かれ、

ウンドウAより、その子A'、更にその子A''が指定されたものである。またC,D間のリンクを切り、Bの下につけ替えたことを示している。)



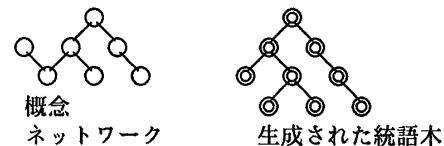
(図4) 構文木の書き換え

### 3.4 構文生成

概念ネットワーク上の情報をキーとして対象言語の生成側の辞書をロードし、統語情報を保持した統語木を生成する。さらに、その統語ノード間の並びを決定する。

#### ① 統語木の生成

ここでは、4つのウンドウは概念抽出フェーズと同様に、概念ネットワーク上に置かれ、それに対応する統語木を新たに作っていく。ルールによって統語ノードの生成を行ったり、生成した統語ノード間のリンクづけを指示できる。このフェーズではバックトラックして、木をつくり直すことも可能である。



(図5) 統語木の生成

#### ② 語順

前フェーズで生成された統語木をデブスファーストにトラバースし、そのノードの並びを決定する。4つのウンドウのうち並びを決定するためのウンドウは決められた2つで、その2つのあいだで前後を決めていく。

### 3.5 形態素生成

前モジュールで決定された並びにしたがって統語木上の注目ノード移動し、そのノードごとに表層文字列を出力する。

ウンドウは4つのうち2つが使用できる。統語木上の注目ノードに1つがバインドされ、1回前のルール適用時の注目ノードに他の1つがバインドされる。

### 4. おわりに

おおのののフェーズにおけるの処理データや機能の差異を、モニタの違いとすることで、ルールインタプリタの変更することなく容易に吸収できた。しかし、プログラムサイズが非常に大きくなっていることや、必要メモリサイズ、処理速度などにおいて満足できるものとはいはず、今後これらの問題の解決を図っていきたい。

### 参考文献

- 1) 村木、市山：「PIVOT：多言語間翻訳システム概要」昭和63年電子情報通信学会全国大会