

## Connectionist Model を用いた自然言語処理システム†

田 村 淳†† 安 西 祐一郎†††

自然言語処理システムの研究では、構文解析、意味解析を経て、文脈解析の研究が進みつつある。また、高度な言語情報処理は連想によるところがあることも、生理学や心理学の進歩によって明らかにされつつある。一方、ハードウェア技術においては、従来の逐次型計算機だけでなく、並列型計算機の研究が発展しつつある。本論文の目的は、こうした背景に基づいて筆者らが開発した、Connectionist Model を用いた自然言語処理システムの内容を述べることである。Connectionist Model とは、多数の処理ユニットを興奮性リンクと抑制性リンクでつないだもので、リンクを通じた値の受け渡しでそれぞれのユニットの活性化レベルが変化することによって、情報処理を行う計算モデルである。本研究では、Connectionist Model に基づいた、多義語を含む日本語文の意味処理を行う自然言語処理システム CMCP を設計し、インプリメントした。CMCP は、日本語と英語に対して、構文解析と意味解析を擬似的に並列処理し、連想により文脈に応じて多義文を処理することができる。その特徴は、筆者らが開発したオブジェクト指向型言語 OPHELIA で書かれているために、ユニットの数値的な活性化をユニット間通信によって実現し、同時に文法解析のような記号処理をも行えるようになってきていることである。

### 1. はじめに

自然言語処理システムの研究は、構文解析の時代から意味解析の時代を経て、現在は文脈解析が問題になりつつある<sup>1)</sup>。特に、質問応答や会話処理においては、構文や意味の解析さえ、文脈が大きな影響を与えることが知られている<sup>2)</sup>。文脈の処理にあたっては、いろいろな方法が研究されているが、計算量や曖昧さの解消などの点で、それぞれ問題があり、システム構築における決め手がないのが現状である。このような状況のもとでは、文脈解析のためのいろいろな方法を模索し、その中からよい方法を固めてゆく必要がある。

このような問題に対処するため、筆者らは、これまで日本語処理システムの研究ではほとんど取り上げられてこなかった、連想記憶に基づく日本語処理システムを設計し、インプリメントしてきた。本論文では、その概要について報告するとともに、日本語文の文脈処理における本システムの特徴と限界について述べる。

### 2. Connectionist Model による日本語処理システム

逐次処理をする CPU と受動的な記憶装置を用いて演算を行うノイマン型計算機に対して、相互に結合し

た多数のプロセッサがたがいに並列に影響を及ぼし合いながら演算を行うようなシステムを一般に **Connectionist Model** (以下 **CM** と略す) という<sup>3)</sup>。CM の基本的構造はきわめて単純で、多数の処理ユニット間を興奮性 (excitatory) リンクあるいは抑制性 (inhibitory) リンクで結合したネットワーク構造をしている。各ユニットの演算は、基本的にはいわゆる McCulloch-Pitts の神経回路網モデル<sup>4)</sup> と類似のものを考えればよい。すなわち、モデルでは、細胞の膜電位がある閾値以上になるとパルスが発生し、その細胞に結合した細胞にこれが伝えられると考える。パルス入力には、結合している細胞の膜電位を高めるように働く興奮性入力と、低めるように働く抑制性入力がある。パルス入力を受け取った細胞の膜電位は、それまでの膜電位と入力パルスの値の線形結合として表される。CM では、各ユニットについて、膜電位に相当する値を活性化レベル (activation level) とよぶ。このとき、CM による「演算」とは、その CM の各ユニットの活性化レベルを変化させることであり、演算の「結果」とは、CM を構成しているユニットの集合の上に分布した、ある時点での活性化レベルの値のことである。つまり、神経回路網モデルに基づく連想記憶システムのように、CM においては、ユニット間では記号列を用いた大量の情報が伝達されるわけではなく、数値のような簡単なメッセージが多数のユニットに並列に送られることによって演算が実行される。

ただし、神経回路網に似ているからといって CM が直接神経回路網の構造モデルになっているというわけではない。CM は、演算を明示的な記号表現、たと

† Natural Language Processing System Based on Connectionist Models by ATSUSHI TAMURA (C&C Information Technology Research Laboratories, NEC Corporation) and YUICHIRO ANZAI (Department of Behavioral Science, Hokkaido University).

†† 日本電気(株) C&C 情報研究所

††† 北海道大学文学部行動科学科

えば、プロダクション・ルールや述語論理のレベルで特徴づけるのではなく、神経生理学的な構造に似たモデルで特徴づけようとするものである。

CM を自然言語処理システムの構築に応用した研究としては Waltz and Pollack<sup>6)</sup> があり、そこでは英語における多義的な単文の意味解釈を行うシステムがハードウェアで実現されている。またほかに、記号処理的な方法と CM を組み合わせたシステムがいくつか提案されている<sup>6),7)</sup>。しかしながら、CM の神経回路網的な構造とこれまでの自然言語処理システム研究における記号処理的なアプローチがどのような知識表現構造の上で自然な形で融合できるのか、また日本語処理において、CM、あるいは CM と記号処理システムの融合システムがどのような形で実現しうるのかは、未だ明らかでない。

そこで本研究では、日本語処理システム、特に日本語多義文の意味解釈システムを、CM の形で実現した。このシステム (Connectionist-Model-based Context Parser; 以下 **CMCP** と呼ぶ) は、簡単な文法規則と CM のメカニズムを融合したものであり、OPHELIA とよばれるオブジェクト指向型知識表現言語<sup>8)</sup>で書かれている。このことは、記号処理的な推論と CM による演算とを組み合わせた自然言語処理システムのプロトタイプに、オブジェクト指向型の知識表現言語が適していることを示唆している。また、CMCP は、構文解析部およびごく一部の制御機構を除いて英語文にも使用できる。このことは、CM をベースとする自然言語処理システムの構文解析部、辞書部、意味・文脈解析部をモジュール化すれば、多言語の処理に利用できるシステムの構築が比較的容易にできることを示唆している。

### 3. システムの概要

#### 3.1 インプリメンテーション

CMCP は、OPHELIA で書かれ、UNIX (Berkeley 4.2 bsd) 上で稼動している。OPHELIA は Franz Lisp で書かれたオブジェクト指向型言語である。

CMCP では、CM の各ユニットを OPHELIA のオブジェクトとして表現し、興奮性および抑制性リンクによる他ユニットへの数値の伝達をメッセージ・パッシングによって行っている。また、各ユニットにおける活性化レベルの計算は、その上位のオブジェクトの中に書かれたメソッドを継承して行っている。以下に、システムの細かい内容を述べる。

#### 3.2 CMCP におけるユニットの表現と活性化

CMCP においては、CM のユニットを OPHELIA のオブジェクトに対応させ、ユニット間のリンクはそのオブジェクトの中に書いた変数を用いて表現する。OPHELIA ではオブジェクトの変数は、

( <変数名> <値> )

のように書くので、第  $i$  ユニットの興奮性リンクで他の  $n$  個のユニットがつながっていることは、

(Excitatory (( $k_{i1}$   $x_{i1}$ ) ( $k_{i2}$   $x_{i2}$ )...( $k_{in}$   $x_{in}$ )))

のように表現する。ただしここで、'Excitatory' は変数名、対リストのリスト (( $k_{i1}$   $x_{i1}$ )( $k_{i2}$   $x_{i2}$ )...( $k_{in}$   $x_{in}$ )) はその値であり、 $x_{ij}$  は結合先のユニット名、 $k_{ij}$  はその結合の強さを表す ( $j=1, \dots, n$ )。抑制性リンクは、変数名を 'Inhibitory' として同様に表現される。なお、興奮性リンクの場合は、結合の強さを示す数値は正の値であるが、抑制性リンクの場合は負の値になっている。

各ユニットの活性化レベルもそのオブジェクトの変数を用いて次のように表される：

(ActivationLevel <活性化レベル>)

ここで 'ActivationLevel' は変数名であり、<活性化レベル>は活性化レベルを表し、-10 から 10 までの実数値をとる。各ユニットに対して、その活性化レベルにリンクの結合の強さ (神経回路網でいえばシナプス荷重) を乗じた値が、そのリンクで結合したユニットに送られる。ただし、CMCP では、活性化しているユニットだけが他のユニットに値を送ることができる。いいかえると、活性化レベルの閾値は 0 である。

一般に、活性化レベルが閾値より大であれば、そのユニットは活性化しているという。なお、上の表現では、あるユニットから「出て」いる有向リンクを表し

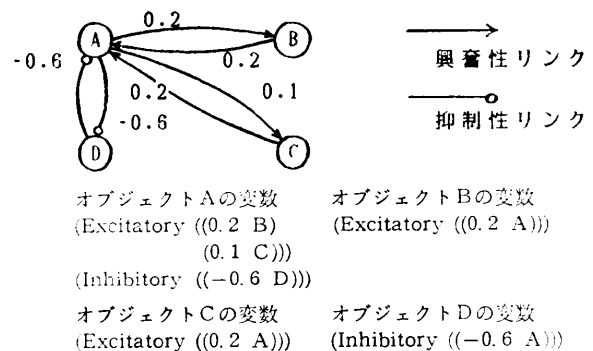


図 1 リンクの表現の例

Fig. 1 Example of representation for links.

ており、双方向のリンクを表現するためには、その両端のユニットを示すオブジェクトそれぞれの変数に他方のユニット名とリンクの結合強さを記述しなければならない。

図1に、OPHELIAによるリンクの表現の例を示す。

### 3.3 内部表現

#### 3.3.1 辞書

CMCPの辞書部は、各単語ごとに一つのクラス・オブジェクトで表される。辞書には、その単語の品詞と単語が表す概念が記述されている。これらは、変数名をそれぞれ 'SynCat', 'concepts' とする変数で表現される。図2にその一般形を示し、図3に具体例を示す。ただし、図2の中の<概念>には3.3.2項で説明する概念クラス名が入る。

単語を読み込むとその単語オブジェクトのインスタンスが生成される。これは、1回の実行中に同じ単語が複数回現れてもそれらを区別できるようにするためである。インスタンス名は 'tako-1', 'tako-2' のように、クラス名の後にハイフンと数字が続いたものになる。

#### 3.3.2 意味表現

##### 3.3.2.1 ユニットの分類

CMCPにおいては、意味とは概念の活性化状態全体であると考えられる。たとえば、温度はマクロ的に見た分子運動の全体で定まり、ミクロ的に見た個々の分子の温度というものは存在しないと同様に、意味もまた、あまりに分析的になっても解明できないという考え方に立っている。

また、CMCPでは一つの概念をいくつかの特徴で表現し、概念そのものとその特徴とを区別する。ここで、概念を表すオブジェクトを概念ユニット、特徴を表すオブジェクトを特徴ユニットとよぶ。たとえば、「蛸(OCTOPUS)」の特徴ユニットとしては、

```
(SynCat (<品詞 1><品詞 2>...<品詞 m>))
(concepts (<概念 1><概念 2>...<概念 n>))
```

図2 辞書の表現方法

Fig. 2 Representation of dictionary items.

```
***** CLASS NAME: ageru
SUPER: (S: word)
VARIABLES: (SynCat (V))
             (concepts (RAISE GIVE FRY))
```

図3 辞書オブジェクトの例(「あげる」)  
(必要な箇所だけを示した)

Fig. 3 Example of dictionary objects (part).

ANIMALness:1, ANIMALness:2, EDIBLEness:1, EDIBLEness:2 など、適当に名前をつけて用いる。CMでは、特徴ユニットの数が重要なファクタであり、またこの表現方法によって、上位概念の自動継承が可能になる。つまり、特徴 A, B, C が活性化されることによってそれらからなる概念 X が表現されれば、同時に特徴 A, B からなる X の上位概念 Y も表現されると考えるのである。

概念ユニットがもつ特徴ユニットの名前は、概念ユニットを表すオブジェクトの変数 'pattern' の値として書かれる。先の例では、オブジェクト OCTOPUS の変数とその値として、

```
(pattern (ANIMALness:1 ANIMALness:2
          EDIBLEness:1 EDIBLEness:2...))
```

のように表現される。

概念ユニットの活性化レベルは、その特徴ユニットの活性化レベルの和で与えられる。なお、全特徴ユニット数のうちの一定割合以上の数の特徴ユニットが活性化していないと、概念ユニットが活性化しないようになっている。ただし、格ユニット(概念ユニットのうち格を示すもの)など一部のものは、下部ユニットとしての特徴ユニットをもたず、それ自体で表現される。

また、CMCPでは一般的な概念と単語が直接示す概念とを区別し、OPHELIA上で前者をクラス、後者をインスタンスで表している。ここではそれぞれを概念クラス、概念インスタンスとよぶ。このように区別するのは、異なった単語が同じ概念を指す場合に対応するためである。図4に概念クラスの例を示す。

##### 3.3.2.2 リンクの分類

ユニット間のリンクは、二つの次元で分類できる。一つは、前述したような、結合先のユニットに対する影響の正負による興奮性リンクと抑制性リンクの区別である。CMCPでは、興奮性リンクはそのユニット

```
***** CLASS NAME: KITE
SUPER: (S: concepts)
VARIABLES: (SynCat N)
            (Excitatory ((0.1 NEW_YEAR)
                         (0.1 FlyKite)
                         ...))
            (Inhibitory ((-0.4 OCTOPUS)))
            (pattern (TOYness:1 TOYness:2
                      FLYABLEness:1...))
            (SumUp 0)
            (ActivationLevel 0)
```

図4 概念クラスの例('KITE')(必要な箇所のみ)  
Fig. 4 Example of conceptual class objects (part).

から連想しうるものに結合されているが、そのような結合は、概念ユニット間だけでなく、特徴ユニット間や、概念ユニットと特徴ユニットとの間にもある。また、抑制性リンクは主に多義語の概念ユニット間につながれている。

もう一つの分類次元は、**長期的リンク**と**短期的リンク**の区別である。長期的リンクとはクラスユニットとその間のリンクであり、これらはつねに存在する。これに対して、短期的リンクとはインスタンスユニットにつながったリンクを指す。クラスユニットや長期的リンクと違って、インスタンスユニットと短期的リンクは実行が終了すると消えてしまう。ただし、システム内部では、短期的リンクは長期的リンクと同様にオブジェクトの変数で表され、結合先ユニット名と結合強さを表す変数名としては、興奮性リンクおよび抑制性リンクそれぞれに対して 'Excitatory\*', 'Inhibitory\*' を用いて、長期的リンクのそれと区別する。これは、Excitatory\* および Inhibitory\* リンクの場合、インスタンスユニットが発生しなければリンクを変数として表現できないためである。

### 3.3.3 構文の表現

N (名詞), V (動詞), S (文) などの構文要素をそれぞれユニットで表し、**構文ユニット**とよぶ。また、このうち品詞を表すものをとくに**品詞ユニット**とよぶ。構文ユニットにもクラスとインスタンスがあるが、構文クラスは構文インスタンスを生成するためにしかはたらかない。構文木の枝は、双方向の興奮性リンクで示され、文法規則に合致した構文ユニットが活性化される。多品詞語の品詞ユニット間には抑制性リンクがつながれる。なお、後に3.7節で述べるように、構文ユニットを監視する**構文モニタ**があり、文法規則はその中のプロダクション・ルールで表現される。

### 3.4 実行サイクル

CMCP の基本的な実行は、

- (1) 単語  $w$  の読み込み
- (2)  $w$  のインスタンスの生成

を行った後、

- (3)  $w$  の概念クラスの活性化  
 $w$  の概念インスタンスの生成、活性化
- (3')  $w$  の特徴クラスの活性化  
 $w$  の特徴インスタンスの生成、活性化
- (4)  $w$  の品詞インスタンスの生成、活性化
- (5) 構文処理

### (6) 拡散活性化 (spreading activation)

を行うことからなる。ただし、(3') は特徴ユニットをもつものについてだけ行う。(5)については3.7節を参照されたい。(6)は、

- (A) 活性化している各ユニットから興奮性リンクあるいは抑制性リンクで結合されたユニットに値を送る
- (B) 各ユニットの値  $w$  を計算する

の二つからなり、これを**1サイクル**とよぶ。(A)は、モニタ(一つのオブジェクトで表現されている)が、活性化しているユニットに 'send' というメッセージを渡して、各ユニットに値の送付を行わせることによって実行される。同様に、(B)は 'fire' というメッセージによって実行される。実行はユニットの上位クラスが持っているメソッドによって行われる。(B)の計算は次式を用いる:

$$a_j(t+1) = 0.9 a_j(t) + \sum_{i=1}^n k_{ij} a_i(t)$$

ここで、 $a_j(t)$  は時刻  $t$  における第  $j$  番目のユニットの活性化レベル、 $k_{ij}$  は第  $i$  番目のユニットから第  $j$  番目のユニットへのリンクの結合強さである。ただし、

興奮性リンクで結合していれば  $k_{ij} > 0$

抑制性リンクで結合していれば  $k_{ij} < 0$

結合していなければ  $k_{ij} = 0$

である。

なお、上の計算式の右辺第1項は、時間とともに活性化レベルが減衰していくことを示している。

### 3.5 文脈処理

各時点でのユニットの活性化状態全体が、そのときの文脈を表しているとみなすことにする。すると、新しい文脈の計算は、上にあげた実行サイクルによって行われると考えてよい。すなわち、まずユニットの活性化は、各ユニットからの興奮性リンクを通じた値の伝播によって行われる。興奮性リンクは、その概念から連想しうる概念につながられているので、このリンクを通じた拡散活性化によって連想が行われると考えることができる。つまり、連想はクラス・オブジェクトのレベルで行われ、すでに読み込んだ単語から始めて、それらが直接意味しない概念のクラスをも活性化する。そして、新たな単語がその活性化された概念を指す場合には、活性化されていた概念クラスの活性化レベルがその単語の概念インスタンスの活性化レベルに引き継がれる(ただし、日本語の助詞の場合は、格クラスの値は格インスタンスに引き継がれない)。こ

の機構によって、以前に読み込んだ単語列から連想される概念が、そうでないものより活性化されやすくなり、連想による文脈処理が実現できる。多義語の場合には、それ以前の文脈からその単語の一つの概念クラスが活性化していれば、その他の概念クラスの活性化は抑制性リンクによって抑えられるから、その効果は大きくなる。

### 3.6 意味処理

CMCP では、概念ユニットと特徴ユニットの活性化が意味処理に相当する。意味処理において重要なことは多義語の意味解釈であるが、これは、

(1) 多義語の概念インスタンス生成時に、それらの間を抑制性リンクで結合する

(2) 他の概念ユニットと興奮性あるいは抑制性リンクで結合し、実行を重ねて、多義語の概念インスタンスの活性化レベルを変化させる

ことによって行う。その結果、一つ概念インスタンスだけが低い値を示せば多義性が解消されたことになり、複数の概念インスタンスが同等程度に活性化していればその語の多義性は解消されず曖昧なままであることになる。

なお、(1)において、常に同じ強さで抑制性リンクをつなぐと、多くの概念をもつ単語ほどその概念インスタンスの活性化レベルが低くなるので、概念インスタンスが  $n$  個あるときは  $-0.6/(n-1)$  でその強さを与えている ( $n \geq 2$ )。

### 3.7 構文処理

構文規則は、OPHELIA のオブジェクトの一つである構文モニタに書かれたプロダクション・ルールで表されている。単語を読み込むと、その単語がとりうる品詞のインスタンスが作られ、構文モニタのワーキング・メモリに入る。その状況で構文規則が適用できるかどうかはそのプロダクション・ルールを用いて調べられ、構文規則に合ったものがリンクで結ばれる。ただし、CMCP における構文規則は、規則における上位の構文単位の候補を挙げるために使われ、解析そのものを行うわけではない。構文処理は、他の意味処理や文脈処理と並列に、リンクを通じた値の受け渡しによってなされる。つまり、どのユニットが活性化しているかによって、以下のように解析が行われる：

(1) 複数の品詞になりうる単語の品詞ユニット間を抑制性リンクでつなぐ。

(2) プロダクション・ルールとして表現された構文規則によって、上位の構文構造が形成され

うるかどうかを調べる。

(3) 上位の構文構造が形成されうる場合には、下位のユニットと上位のユニット間を興奮性リンクでつなぐ。

なお、(3)において、2個の構文単位から1個の構文単位ができる場合と、1個の構文単位から1個の構文単位ができる場合とがある。どちらも同じ強さで結合すると、前者の上位の構文単位の方が後者の上位の構文単位と比べて下位から2倍に強められることになる。そこで、前者の場合は0.1、後者の場合は0.2の強さで結合することになっている。

日本語の構文規則としては、

$S \rightarrow NP^* + V$

$NP \rightarrow N + P$

を用いた。ここで、N は名詞、P は助詞、V は動詞、S は文を表す。また、NP は名詞句、NP\* は NP が1回以上任意個繰り返したものを示す。多品詞語の具体的な処理例については、4.1 節例2に述べる。

## 4. 例

### 4.1 単一文の処理例

本研究は、形態素解析などを直接の対象とはしていないので、CMCP の現在の版では、日本語文はローマ字で入力し、単語間は空白によって区切るようにしている。

[例1] 正月にたこをあげる。

この例文では、「に」と「たこ」と「あげる」が多義語である。ここでは、それぞれ次の意味をもつと仮定する。括弧内は、その意味を表す概念ユニットの名称である：

- |     |                           |
|-----|---------------------------|
| に   | 1. 時を示す (CaseTime)        |
|     | 2. 受け手を示す (CaseRecipient) |
|     | 3. 場所を示す (CaseLocP)       |
|     | 4. 目的地を示す (CaseLocD)      |
| たこ  | 1. 空にあげるおもちゃの凧 (KITE)     |
|     | 2. 8本足の軟体動物の蛸 (OCTOPUS)   |
| あげる | 1. 高いところに位置させる (RAISE)    |
|     | 2. 与える (GIVE)             |
|     | 3. 熱い油で煮る (FRY)           |

例1では、はじめに「正月」という単語があるため、概念クラス間で、

正月 → 凧あげ → 凧

の連想がはたらき、「たこ」は凧の意味になり、さらに凧から「あげる」の1.の意味が活性化される。ま

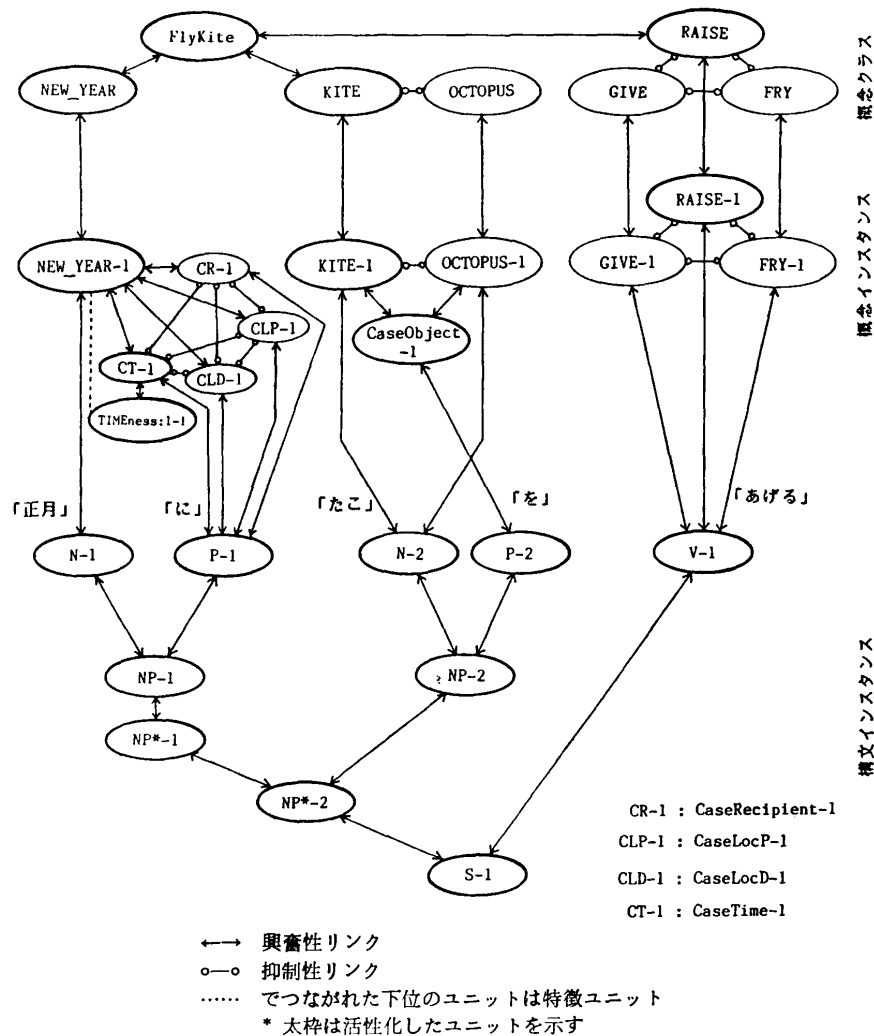


図 5 例 1 の処理における主なユニットとリンク  
 Fig. 5 Some units and links used in processing of the sentence in Example 1.

た、「に」は「正月」と文節をなしているが、「正月」は時を表すので特徴インスタンス 'TIMEness:1-1' をもち、それが「に」の格インスタンスのうち 'Case Time-1' とだけ興奮性リンクで結ばれて、これが選択される。単語「たこ」を読むと、多義語であるから、その概念インスタンス 'OCTOPUS-1' と 'KITE-1' が抑制性リンクで結合される。概念インスタンスが生成されるときには、その概念クラスの値を引き継ぐので、'KITE' がすでに活性化されていた分だけ、'KITE-1' の方が 'OCTOPUS-1' よりも高い値となる。そして、これらは全体として不適切な箇所はなく安定した状態にあるので、そのまま確定される。全体の概略を図 5 に示す。

【例 2】 おもち屋さん が たこ を 食べる。

この例では、はじめ、概念クラス間の連想

おもちゃ屋 → 尻

によって、概念インスタンス 'KITE-1' が 'OCTOPUS-1' よりも高い値になるが、これは「食べる」ことができないので、後に「蛸」に訂正される。

なお、「おもちゃ屋さん」は、いずれも名詞として、

1. 店としてのおもちゃ屋さん (TOY-SHOP)
2. 店員としてのおもちゃ屋さん (TOY-SHOP-MAN)

の意味をもつものとする。

まず、単語「おもちゃ屋さん」を読むとその概念クラスと概念インスタンス、および品詞インスタンス 'N-1' が活性化され、「KITE」と抑制性リンクで結合された 'OCTOPUS' は抑制される。

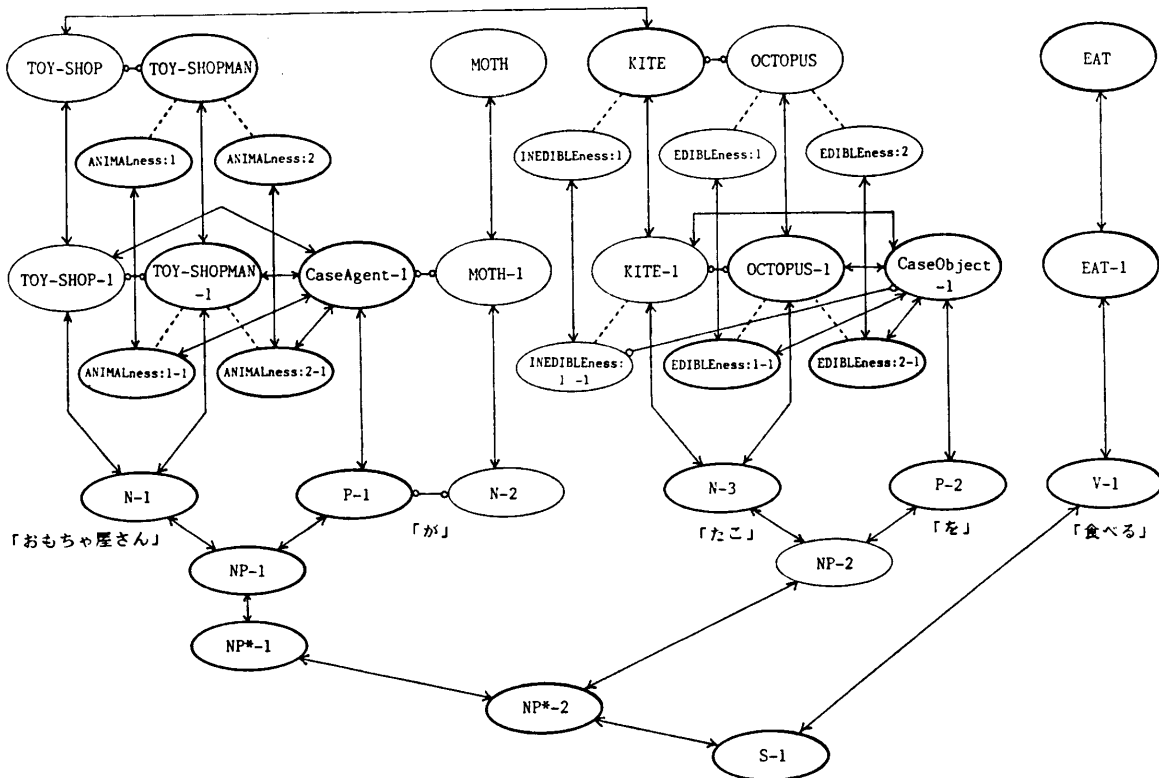


図 6 例 2 の処理における主なユニットとリンク (記号は図 5 に準ずる)

Fig. 6 Some units and links used in processing of the sentence in Example 2. (notation follows Fig. 5)

単語「が」には、

1. (助詞) 主格を示す (CaseAgent)
2. (名詞) 蝶に似た昆虫の蛾 (MOTH)

の二つの意味があると仮定すると、多品詞語であるため、助詞の 'P-1' と名詞の 'N-2' が抑制性リンクで結ばれる。'N-2' は構文規則に合致しないので他の構文ユニットと興奮性リンクでつながれず、'P-1' だけが上位の構文ユニット 'NP-1' と興奮性リンクでつながれて強められる。以後 'N-2' と 'MOTH-1' の活性化レベルは減少していき、'CaseAgent-1' と 'P-1' が残ることになる。このように、CMCP では、構文解析も意味処理と同様にユニットの活性化という形でなされるので意味処理と並行して行われる。

ここで単語「たこ」を読むと、その概念インスタンスについては、概念クラス 'OCTOPUS' が抑制されたままであるので 'KITE-1' の方が 'OCTOPUS-1' よりも高い値になる。つまり、この時点では「凧」の意味に解釈される。

また、単語「食べる」を読むと、その主格が動物で、目的格が食べられるものでなければならないので、'ANIMALness:1-1' と 'CaseAgent-1'、'ANIMAL-

ness:2-1' と 'CaseAgent-1'、'EDIBLEness:1-1' と 'CaseObject-1'、'EDIBLEness:2-1' と 'CaseObject-1' がそれぞれ興奮性リンクで結合され、'INEDIBLEness:1-1' と 'CaseObject-1' が抑制性リンクで結合される。'ANIMALness:1-1'、'ANIMALness:2-1' は 'TOY-SHOPMAN-1' の特徴インスタンスであり、'EDIBLEness:1-1'、'EDIBLEness:2-1' は 'OCTOPUS-1' の特徴インスタンスであり、'INEDIBLEness:1-1' は 'KITE-1' の特徴インスタンスである。この結果、'INEDIBLEness:1-1' が抑制されるので、'KITE-1' もその特徴ユニットの一つを失って抑制され、これ以後「たこ」の意味は「蛸 (OCTOPUS-1)」だけになる。

全体の概略を図 6 に示す。

なお、たとえば

「魚屋さんがたこを食べる」

という文では、CMCP は単語「たこ」の意味をはじめから「蛸」ととらえるので、上のようなことは生じない。

#### 4.2 複数文の処理例

文が複数になっても同様に処理することができる。

**[例 3] おもちゃ屋さん に 行く. たこ を 買う.**

この例でも、「おもちゃ屋さん」から「凧」が連想されて、第2文の「たこ」は「凧」の意味になる。

ここでは、「おもちゃ屋さん」は、「店としてのおもちゃ屋さん」の意味になるが、これは動詞「行く」によって、「店としてのおもちゃ屋さん」の特徴のうち 'PLACEness:1-1' と格インスタンス 'CaseLocD-1' が興奮性リンクで結合されるためである。また、これによって助詞「に」の意味が 'CaseLocD-1' になる。

なお、第1文が

「魚屋さん に 行く」

であれば、「魚屋さん」から「蛸」が連想されて、第2文の「たこ」は「蛸」の意味になる。

**[例 4] 贈り物 を する. たこ を あげる.**

単語「贈り物」の概念クラス 'PRESENT' から、概念クラス 'GIVE' が活性化され、そこから抑制性リンクでつながっている概念クラス 'RAISE' と 'FRY' は抑制される。そのため、第2文で単語「あげる」を読んだときには、「GIVE」だけが活性化される。なお、ここで「たこ」の意味は曖昧であり、CMCP の出力も「凧」と「蛸」の活性化レベルがほぼ同じ値になっている。

このほか、「The astronomer married a star.<sup>5)</sup>」などの英語文も文法規則等の変更だけで処理することができる。

## 5. 結 論

本論文では、CM をベースとする自然言語処理システム CMCP, およびその多義文の意味解釈への応用について述べた。CMCP の特徴をまとめると次のようになる：

- (1) 興奮性および抑制性リンクで結合されたユニットの活性化レベルの変化によって、ある程度の文脈処理が可能である。つまり、従来の方法を用いるならば、たとえば例3では、単語「たこ」には「蛸」と「凧」の二つの意味があるが、「おもちゃ屋さんには凧が売られている」という形の明示的な知識から推論して「たこ」の意味を決定するであろうが、CMCP ではある程度までなら連想による概念の活性化だけでもこのような文脈処理が可能である。
- (2) 構文処理と意味処理を並列に実行できる。これによって、構文と意味の両方の情報を同時に用

いて、どちらについても整合性のある解釈がなされることになる。

- (3) (1)と(2)を同時に行うための、CM と記号処理を融合したシステムを表現する方法として、オブジェクト指向型知識表現言語を利用することができる。つまり、ユニットの活性化レベルの変化についての並列的な「アナログ」処理とプロダクション・ルールによる文法規則のような直列的な「記号」処理を組み合わせたシステムの構築において、少なくともプロトタイプ段階では、オブジェクト指向型の知識表現が使えることを示している。

以上のとおり、CMCP は日本語の文脈・意味処理システム作成における一つの新しい方向を示すものと考えられる。しかしながら、実用化レベルの自然言語処理システムを構築するには、さらに今後の検討が必要である。特に重要な問題は、実行の制御方法を改良することである。つまり、CMCP では、一単語読み込むと一定サイクルの実行が行われるが、本来何サイクルになるかは単語によって違うはずである。さらに、同じ単語であっても常に同様に制御されるとは限らないであろう。今後は、こうした動的な制御を研究する必要があると思われる。また、現在、実行は単語の読み込みが終わった後一定サイクルで終了しているが、この終了条件も検討の余地がある。

また、本研究では簡単な例文だけを扱ったが、今後より複雑な多数の例文に対応できるように検討していく予定である。ただし、CMCP を拡張する際に、複雑な処理まで CM で対処することは現在の技術では困難であり、記号レベルでの処理をも拡張するのが適当であると思われる。

これに関係して、本論文に述べたような value-passing だけでなく、いわゆる marker-passing<sup>9)</sup> をも含めた複合的なシステムを構築することも考えられる。

このほか、並列計算、省略と照応、学習、類推など、困難ではあるが興味深い問題も重要であり、そのいくつかについてはすでに研究が進められている。

謝辞 本論文の作成にあたり協力していただいた近藤公久、中村久肇の両氏に感謝いたします。

## 参 考 文 献

- 1) Reichman, R.: *Getting Computers to Talk Like You and Me*, The MIT Press, Cambridge, Mass. (1985).
- 2) Brady, M. and Berwick, R. C. (eds.): *Compu-*



*tational Models of Discourse*, The MIT Press, Cambridge, Mass. (1983).

- 3) Feldman, J.A. and Ballard, D.H. : Connectionist Models and Their Properties, *Cognitive Sci.*, Vol. 6, No. 3, pp. 205-254 (1982).
- 4) McCulloch, W. S. and Pitts, W. H. : A Logical Calculus of Ideas Immanent in Neural Nets, *Bull. Math. Biophys.*, Vol. 5, pp. 115-133 (1943).
- 5) Waltz, D.L. and Pollack, J.B. : Massively Parallel Parsing: A Strongly Interactive Model of Natural Language Interpretation, *Cognitive Sci.*, Vol. 9, No. 1, pp. 51-74 (1985).
- 6) Cottrell, G. W. : Connectionist Parsing, *Proc. of the 7th Annual Conference of Cognitive Science Society*, pp. 201-211 (1985).
- 7) Selman, B. and Hirst, G. : A Rule-Based Connectionist Parsing System, *Proc. of the 7th Annual Conference of Cognitive Science Society*, pp. 212-221 (1985).
- 8) 近藤公久, 神岡太郎, 細矢貴茂, 安西祐一郎: OPHELIA マニュアル (1986).
- 9) Fahlman, S. E. : Representing Implicit Knowledge, in Hinton, G. E. and Anderson, J. A. (eds.), *Parallel Models of Associative Memory*, pp. 145-159, Lawrence Erlbaum Associates, Hillsdale, N. J. (1981).

(昭和 61 年 5 月 19 日受付)

(昭和 61 年 11 月 5 日採録)



田村 淳 (正会員)

昭和 35 年生。昭和 59 年慶応義塾大学工学部管理工学科卒業。昭和 61 年同大学院工学研究科管理工学専攻修士課程修了。同年日本電気(株)入社。現在、同社 C & C 情報研究所勤務。人工知能, 認知心理学, 言語学, 生体の情報処理等に興味を持っている。日本認知科学会, 人工知能学会, 日本ソフトウェア科学会各会員。



安西祐一郎 (正会員)

昭和 21 年生。昭和 49 年慶応義塾大学大学院博士課程修了。工学博士。慶大講師を経て昭和 60 年より北海道大学文学部行動科学科助教授。昭和 56~57 年カーネギーメロン大学客員助教授。計算機科学, 人工知能, 認識の情報処理過程の研究に従事。著書「知識と表象」(産業図書)ほか。日本ソフトウェア科学会, ACM などの会員。