

## 人間と機械における情報の表現と意味に関する研究

## ——一般化弁別ネットワークの提案——

田 中 穂 積, 奥 村 学

## 1. はじめに

弁別ネットワークは決定木の一般化であり、問題解決に広く用いられている<sup>3)</sup>。特に、自然言語処理においては、複数の語義を圧縮して表現するのに用いられている<sup>4,8,6,1)</sup>。弁別ネットワークは有向非循環グラフであり、一つの根ノードと複数の葉ノードをもつ。そして葉ノードに解が位置する。枝にはラベルとして制約が付与されている。ある制約下での問題解決プロセスは、ネットワークを下向きに、根ノードから葉ノードに向けて一段ずつたどる過程に対応する。この際、与えられた制約を満足する枝をたどることになる。

弁別ネットワークには次のような利点がある。第一に、(複数の可能な解が存在する)曖昧性を表現する際に、複数の候補を持つのではなく、複数の候補をすべて下位ノードとして包含する一つのノードで代表することができる。そうすると、ネットワークを下向きに一段ずつたどる過程は、制約プログラミングが目的とする解の漸進的洗練過程と考えられる。第二は、線形探索と比較して、弁別ネットワーク上での探索アルゴリズムは効率が良いことである。これは、解の探索が葉ノードに位置する解に向けて、根ノードからネットワークを下向きにたどる操作になり、この場合、枝に記述されている制約を索引として探索するので、探索空間が徐々に小さくなるからである。第三は、規則集合を圧縮して表現できることである。なぜなら、異なる規則中の同じ前提条件(制約)をネットワーク中の一つの枝としてまとめて表現できるからである。これにより、制約が満足されるかどうかの検査は、一つの制約につき一回しか行われ

ないので、冗長な再計算を回避することができる。

弁別ネットワークにはこのように優れた特徴があるが、同時に二つの問題点がある。第一は、弁別ネットワークは予め決められた順序で制約が入力されないと素直にたどることができない点である。制約は一般に決まった順序で得られるとは限らないので、ネットワークを下向きにたどるプロセスは、決められた順序で次に入力されるはずの制約が得られるのをしばしば待つ必要がある。弁別ネットワークは根ノードから下向きにたどるので、制約は根ノードに近いものから順々に入力されなければならない。第二の問題点は深刻である。ネットワークを下向きにたどる際、次に入力されるはずの制約が得られない場合には、問題解決プロセスはもはやネットワークを下向きにたどることができず、デッドロックに陥ってしまう。

このような問題点を解決するため、任意の順序で制約が得られてもたどることができる、一般化弁別ネットワーク (Generalized Discrimination Network; GDN) を我々は提案している<sup>9)</sup>。問題解決プロセスは、制約がどんな順序で得られても即座に GDN をたどり、解の空間を縮小することができる。

本稿では、GDN の日本語解析への応用について述べる。GDN を用いることにより、語義の曖昧性解消、省略の補完処理がどのように行われるかについて述べる。

## 2. 弁別ネットワークと 一般化弁別ネットワーク

### 2.1 弁別ネットワークの特徴

図1に弁別ネットワークの例を示す<sup>\*1</sup>。枝にはラベルとして制約が付与されている。葉ノードには解がそれぞれ存在する。葉以外のノードは、その下にある葉ノードに対応する解をすべて含む可能な解の集合を表す。なぜなら、そのノードから枝に沿ってネットワークを下向きにたどりそれらの葉ノードに到達することが可能だからである。根ノードはすべての解を含む集合に対応する。

弁別ネットワークを用いた問題解決プロセスは、与えられた制約を満足する枝に沿ってネットワークを根ノードから葉ノードに向けて下向きに一段ずつたどる過程であると考えられる。この過程で不適当な選択肢は排除され妥当な解が弁別（選択）される。葉ノードに到達することが解が得られたことに対応する。

1節で述べたように、弁別ネットワークには三つの利点があるが、二つの重大な問題がある。これに対し、いくつかの解決策が提案されているが<sup>1,12)</sup>、これらはどれもそれぞれ問題があり、十分な解ではない<sup>9)</sup>。

### 2.2 一般化弁別ネットワークの原理

図1の弁別ネットワークを考える。一般化弁別ネットワークでは、ネットワークを表形式で表現する。以下ではまず、そのための準備について述べる。それぞれのノードにユニークな識別子として数字列を割り当てる。根ノードには1を割り当てる。根ノードの1レベル下位の子ノードにはそ

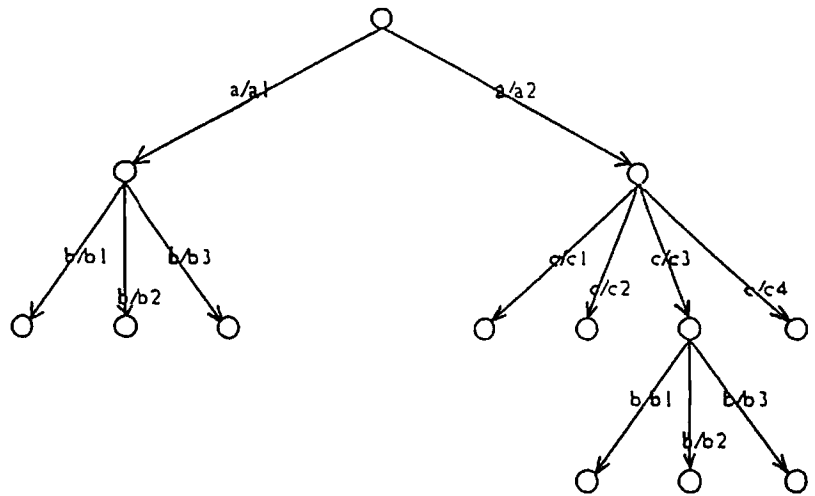


図1 弁別ネットワークの例

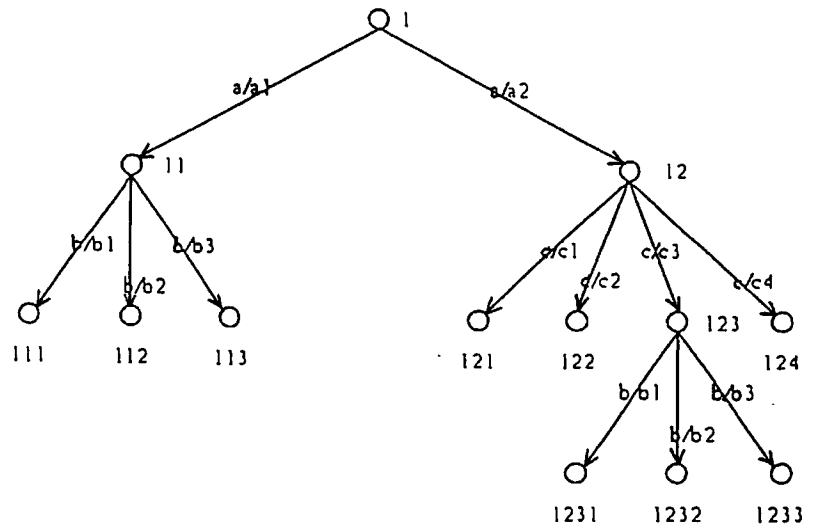


図2 ノードに識別子を付与した弁別ネットワーク

れぞれ2桁の識別子  $1i$  (ここで、 $i$  は1から  $n$  までの整数、 $n$  は子ノードの個数) を割り当てる。以下順に、ノード  $1i_1i_2 \dots i_m$  に対して、その子ノードには  $1i_1i_2 \dots i_m$  を割り当てる。図1のノードには図2のように識別子が付けられる。

次に、ネットワークから、枝のラベル（制約）と、その枝と直接つながった下位ノードの識別子の組を抜き出す。この制約と識別子の組は、表中の制約を満足するなら、弁別ネットワーク上で対応する識別子のノードに到達できることを意味する。たとえば、制約  $c/c_2$  が満足された場合、対応する識別子  $122$  のノードまで下向きに弁別ネットワークをたどれる。ここで  $c/c_2$  は、属性名が  $c$  で、その値が  $c_2$  であることを意味する。

\*1 説明を簡単にするため、本稿では例としてネットワークではなく弁別木を用いる。

表 枝のラベル (制約) と対応する識別子

$a/a_1$	11
$a/a_2$	12
$b/b_1$	{111 if $a/a_1$ , 1231 if $a/a_2$ and $c/c_3$ }
$b/b_2$	{112 if $a/a_1$ , 1232 if $a/a_2$ and $c/c_3$ }
$b/b_3$	{113 if $a/a_1$ , 1233 if $a/a_2$ and $c/c_3$ }
$c/c_1$	121 if $a/a_2$
$c/c_2$	122 if $a/a_2$
$c/c_3$	123 if $a/a_2$
$c/c_4$	124 if $a/a_2$

この際、根ノードから識別子のノードに到達するのに必要な、他の制約に注意する必要がある。上の例の識別子 122 の場合、制約  $a/a_2$  が満足されていないことがわかるので、この場合、「制約  $a/a_2$  が満足されたら、ノード 122 までたどることができる」という「条件付き」の可到達性を表すことになる。

したがって、図 2 のネットワークからは、表のような、制約と「条件付き識別子」(条件付きで到達できるノードの識別子) の組が得られる。条件付き識別子は、到達できるノードの識別子と、満足されていない制約のリストを表す「if 節」からなる。if 節のない識別子は、対応するノードが無条件に到達できることを意味する。属性名  $b$  の制約については組が複数存在するため、それらをまとめた {111 if  $a/a_1$ , 1231 if  $a/a_2$  and  $c/c_3$ } のような集合表現を対応させる。組が複数存在する場合には、それらの複数のノードに到達可能であることを表す。

図 2 の弁別ネットワーク上を識別子 1231 のノードへ到達するための制約の自然な順序は  $a/a_2$ ,  $c/c_3$ ,  $b/b_1$  であるが、それとは異なり、 $b/b_1$ ,  $c/c_3$  の順に制約が得られた時に、本手法でどのように弁別が進むか、その過程について以下では説明する。この場合、必要な制約である  $a/a_2$  がまだ得られておらず、得られた制約についても順序が自然のものとは異なっていることに注意して頂きたい。図 3 に弁別過程を表す「状態」遷移を示す。状態は条件付き識別子の形で表す。初期状態 (制約が何も得られていない状態) は、根ノードの識別子である 1 で表される。制約  $b/b_1$  が得られた後の状態は、制約に対応する条件付き識別子の集合 (表から得られる: {111 if  $a/a_1$ , 1231 if  $a/a_2$

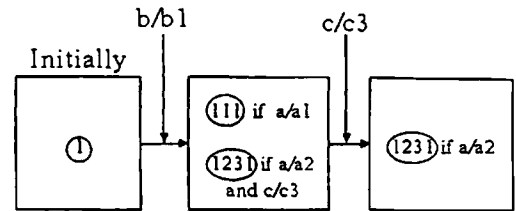


図 3 GDN を用いた弁別過程

and  $c/c_3$ )), および現在の状態 (初期状態 1) を用いて、次のように計算される。

111, 1231 はどちらも初期状態 1 を接頭部分文字列 (数字列) として含むので、より下位のノードに対応する 111, 1231 を返す。現在状態には if 節がないので、次状態の if 節は制約に対応する条件付き識別子の if 節と等しくなる。これは、現在のノード (根ノード) から識別子のノードに到達するのに必要な制約のリストから、得られた制約を除いたものであり、if  $a/a_1$ , if  $a/a_2$  and  $c/c_3$  となる。よって、次状態は、{111 if  $a/a_1$ , 1231 if  $a/a_2$  and  $c/c_3$ } である。

識別子間の演算では、弁別ネットワーク上で一方のノードからもう一方のノードへ到達可能であるかどうかを調べる。これは、図 2 から明らかのように、あるノードから弁別ネットワーク上で到達可能なノードには、識別子が部分文字列関係になるように割り当てられていることを利用している。そして一方から他方へ到達可能な場合、より下位にあるノード (文字列として長い方) を結果として返す。これは、得られた制約により弁別ネットワークを下向きにたどることに相当する。このことから、識別子に関して一方がもう一方を接頭部分文字列として含まない場合に失敗し、下向きにたどることができなくなる。

次に、制約  $c/c_3$  は条件付き識別子 123 if  $a/a_2$  と対応する。現在状態では複数の条件付き識別子 {111 if  $a/a_1$ , 1231 if  $a/a_2$  and  $c/c_3$ } が存在するので、それぞれについて制約に対応する条件付き識別子との間で演算を行う。

111 if  $a/a_1$  の方は、識別子同士が接頭部分文字列関係にないので解析に失敗する。したがって、演算結果は、1231 if  $a/a_2$

$a_2$  and  $c/c_3$ に対するものだけを考えればよい。具体的には、識別子 123 と 1231 から 1231 が得られる。また、現在状態の if 節中の制約  $c/c_3$  があるのでこれを満たし、if 節から取り除き、if 節は  $if a/a_2$  となる。よって、最終結果は 1231  $if a/a_2$  となり、これは「制約  $a/a_2$  が得られる場合にノード 1231 に到達できる」ことを意味する。

if 節は、制約の得られる順序が非決定的であることや、必要な制約が得られない場合に対処するためのものである。if 節は、目標ノードに到達するために得なければならないが、まだ得られていない制約を保存しておく。制約が本来の順序とは異なる順序で得られた場合、逸脱した順序で得られた制約は if 節中に保存されているので、それを if 節から取り除くことで解析を進めればよい。

### 3. GDN を用いた日本語解析

図 4 に動詞「直す」の語義および（表層）格フレームを表現した弁別ネットワークの一部を示す。弁別ネットワークの枝には、動詞の取りうる表層格とその格を満たしうる名詞句に関する選択制限の組が記述される。図 4 では、「直す」が「が」、「を」などの格を取りうるものが記述されている。弁別ネットワークの葉は、枝に記述された制約により弁別（選択）された曖昧性のない 1 つの語義を表す。葉ノードに付与されたラベル（下線を付したものはその語義を表すものとする。葉ノ

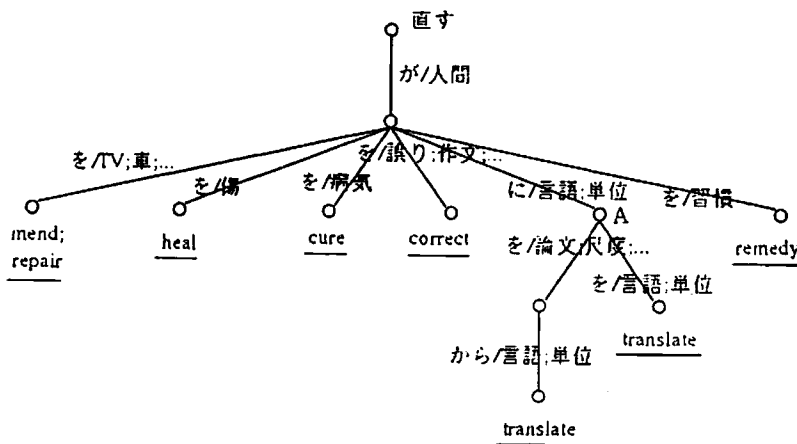


図 4 動詞「直す」の語義弁別ネットワーク（一部）

ド以外のノードは、下向きにさらに枝をたどることで、複数のノード（語義）に到達しうることから、到達しうる語義をすべて含んだ曖昧な意味を表現していると考えられる。根ノードは、すべての語義を含んだ表現であり、曖昧性がまったく解消されていない状態を意味する。根ノードから葉ノードまでのパス中の制約の集合が、その葉ノードの語義に対応する格フレームを表現する。

弁別ネットワークを用いた意味的曖昧性解消は、与えられた制約を満たす枝に沿って弁別ネットワークを根ノードから葉ノード（語義）に向けて下向きに一段ずつたどる過程であると考えられる。この過程で、文中の単語から得られる制約（動詞が取る格要素の情報）から、不適当な選択肢は排除され妥当な動詞の語義が弁別（選択）される。葉ノードに到達することが曖昧性が完全に解消されたことを意味する。

意味的曖昧性解消過程を、弁別ネットワークを下向きにたどる過程として実現する研究として、文献 1), 4), 6), 8) がある。これらの研究では、弁別ネットワークを意味的曖昧性解消に用いることの利点として、1 節で挙げたものが述べられている。

弁別ネットワークの代わりに GDN を用いることにより、意味的曖昧性解消を漸進的に行うことができるという、新たな利点が付加される。文を解析する過程で得られる制約を利用して、できるかぎり早期の段階で曖昧性を段階的に解消していかなければ、曖昧性の数（探索空間の大きさ）は組合せ的に爆発してしまうことを考えると、漸進的曖昧性解消手法<sup>7)</sup>は曖昧性解消のために望ましい方法である。

しかし、文の解析中に漸進的に得られる制約の順序はあらかじめ決められているわけではなく、また、必要な制約が得られない場合もあるため、解析過程で弁別ネットワークを上から下に順々にたどれる保証は一般にない。GDN は、漸進的曖昧性解消手法を阻害している弁別ネットワークのこれらの問題点を解決し、解析過程で漸進的に得られる制約を用いて、ネッ

トワークを下向きにたどることを可能にする。英語の関係節や受動態の例から明らかなように、先行詞や主語となるものが欠けているといった弁別ネットワークにとって情報の欠如によるデッドロックを引き起こす状況がしばしば生じる。しかし、日本語の場合事態はもっと悪い。日本語は語順が比較的自由であるため、あらかじめ定められた順序で制約が得られるとは限らない。さらに、日本語では、文脈から容易に推測可能な句が省略されることが多く、それらが得られないことが問題である。したがって、日本語解析でGDNの果たす役割は重要であると考えられる。

一例として、GDNによりどのように省略の補完処理を行うかについて述べる。GDNは格フレーム集合の表現形式の一つであるから、GDNを用いた省略補完処理は、従来の格フレームに基づく手法<sup>21)</sup>と類似している。以下に述べるように、省略されている句の存在を検出したり、選択制限を用いて補完句の候補の妥当性を検査したりすることは容易に行えるが、補完句の候補の間の優先性を文脈から決定したりするには他の機構が必要である<sup>10),5)</sup>。本稿では、この機構に関しては特に言及せず、「省略されている句に位置的に近いものを優先する」といった素朴なヒューリスティックスを仮定することにする。

省略された句の存在はGDNでは2つの方法で検出する。2節で述べたif節中の制約はいずれ得られるはずであることから、その存在を示していることになる。また、文全体の解析が終了した時点で、葉ノードに到達していない場合、最後に到達したノードと葉ノードの間にある制約は省略されているものとする。この場合、省略の補完処理を行うことにより、語義の曖昧性をさらに解消することが可能になる<sup>2)</sup>。補完句の候補として、省略された制約を利用して、補完句の選択が可能になる。

最後に、GDNを用いてどのように漸進的曖昧

<sup>2)</sup> 構文的妥当性を保証する機構が英語などの言語の場合には更に必要なことは言うまでもない。'I repair.'は目的語が省略された文法的な文であるとは言えないからである。しかし、日本語にはそういった機構は必要がないように思える。

<sup>3)</sup> 2つの語義に異なる格フレームが対応することから、2つの語義には(微妙ではあるが)違いがあると我々は考える。

性解消と省略補完処理が同時に実行されるかを例を用いて示す。次の文を考える。

太郎が日本語で論文を書き、  
英語に花子が直した。

例文中の「直す」の語義は図4に示したように曖昧である。また、「直す」に関する制約の得られる順序は、[に/英語, が/花子]であり、ネットワークの本来の順序と異なる。さらに、「直す」の対象格の要素が省略されている。

最初の文が解析された時点で、「直す」に対して省略された句の候補として、[論文, 日本語, 太郎]が得られる。二番目の文は、次のように解析される。

「英語」は言語であり、図4の「に」格の制約を満足する。よって、ノードAに到達できる。「直す」の語義の曖昧性は、7個から2個に減少する<sup>3)</sup>。次に、「花子」が「が」格の制約を満足し、その結果、ノードAが無条件に到達可能になる。文の解析が終了したが、到達したノードが葉ノードではないので、葉ノードに至るパス中の制約として、{を/言語; 単位}, あるいは{を/論文; スピーチ; 属性, から/言語; 単位}を得る。この制約を用いて、補完句の候補として、「論文」、「日本語」を選択し「直す」の語義を1つに決定する。二番目の文は最終的に、「花子が(太郎が書いた)論文を日本語から英語に翻訳した」を意味していると解釈される。

#### 4. おわりに

弁別ネットワークを一般化したGDNを提案し、それを用いた自然言語処理の方法を論じた。最近の研究から、本方法は音声認識において重要な役割を果たす予測情報の計算法として優れていることが、明らかにされている。これは、3節で述べた。省略語の補完が、一種の予測情報を利用していることから類推できるだろう。今後、この方向の研究を更に進めたいと考えている。

※ 参 考 文 献 ※

- 1) G. Adriaens and S.L. Small: Word expert parsing revisited in a cognitive science perspective. In S.L. Small, G.W. Cottrell and M.K. Tanenhaus, eds., *Lexical Ambiguity Resolution: Perspectives from Psycholinguistics, Neuropsychology, and Artificial Intelligence*, pp. 13-43, Morgan Kaufmann Publishers, 1988.
- 2) J.G. Carbonell: Discourse pragmatics and ellipsis resolution in task-oriented natural language interfaces. In *Proc. of the 21st Annual Meeting of the Association for Computational Linguistics*, pp. 164-168, 1983.
- 3) E. Charniak, C.K. Riesbeck and D.V. McDermott: *Artificial Intelligence Programming*, Lawrence Erlbaum Associates, 1980.
- 4) P.S. Jacobs: Concretion: Assumption-based understanding. In *Proc. of the 12th International Conference on Computational Linguistics*, pp. 270-274, 1988.
- 5) M. Kameyama: *Zero Anaphora: The Case of Japanese*, Ph.D. thesis, Stanford University, 1985.
- 6) S.L. Lytinen: Are vague words ambiguous? In S.L. Small, G.W. Cottrell and M.K. Tanenhaus, eds., *Lexical Ambiguity Resolution: Perspectives from Psycholinguistics, Neuropsychology, and Artificial Intelligence*, pp. 109-128, Morgan Kaufmann Publishers, 1988.
- 7) C.S. Mellish: *Computer Interpretation of Natural Language Descriptions*, Ellis Horwood, 1985.
- 8) G.D. Moerdler and K.R. McKeown: Beyond semantic ambiguity. In *Proc. of the 7th National Conference on Artificial Intelligence*, pp. 751-755, 1988.
- 9) M. Okumura and H. Tanaka: Towards incremental disambiguation with a generalized discrimination network. In *Proc. of the 8th National Conference on Artificial Intelligence*, pp. 990-995, 1990.
- 10) C.L. Sidner: Focusing in the comprehension of definite anaphora. In M. Brady and R.C. Berwick, eds., *Computational Models of Discourse*, pp. 267-330, MIT Press, 1983.
- 11) A. Stolcke: Gapping and frame semantics: A fresh look from a cognitive perspective. In *Proc. of the 13th International Conference on Computational Linguistics*, Vol. 2, pp. 341-346, 1990.
- 12) W.A. Woods: Taxonomic lattice structures for situation recognition. In *Theoretical Issues in Natural Language Processing 2*, pp. 33-41, 1978.

田 中 穂 積 (たなか・ほずみ, 1941年生)

東京工業大学工学部 教授, 東京工業大学大学院  
修士課程修了, 工学博士。

研究課題: 自然言語処理/音声認識/機械翻訳  
システム, 受賞: インド情報処理会議 Best  
Paper Award (1991年)。

奥 村 学 (おくむら・まなぶ, 1962年生)

北陸先端科学技術大学院大学情報科学研究科  
助教授, 東京工業大学大学院博士後期課程修了,  
工学博士。

研究課題: 自然言語理解/知識表現, 受賞: 情  
報処理学会第42回全国大会学術奨励賞 (1991  
年)。

文部省科研費重点領域研究

課題番号: 01633005