情報学基礎 28－6
（1992. 11. 9）

# 一般化弁別ネットワークを用いた日本語意味解析

奥村 学*，秋葉友良**，田中 穂積**
* 北陸先端科学技術大学院大学 情報科学研究科，** 東京工業大学 工学部

[概要]

任意の順序で制約が得られてもたどることができるように弁別ネットワークを拡張した，一般化弁別ネットワーク (Generalized Discrimination Network) を我々は提案している．本研究では，一般化弁別ネットワーク形式に変換した格フレーム辞書を用いた日本語意味解析手法について述べる．一般化弁別ネットワークを用いることにより，語義や係り受けの曖昧性解消，省略の補完処理がどのように行なわれるかについて述べる．

# Japanese Semantic Analysis with a Generalized Discrimination Network

OKUMURA Manabu*, AKIBA Tomoyoshi** and TANAKA Hozumi**
* Faculty of Information Science, Japan Advanced Institute of Science and Technology
(Tatsunokuchi Ishikawa 923-12 Japan)
** Department of Computer Science, Tokyo Institute of Technology
(2-12-1 Ookayama Meguro-Ku Tokyo 152 Japan)

**Abstract**

We proposed a 'generalized discrimination network(GDN),' which is a variant of a discrimination network that can solve a big problem of a discrimination network that it can only be traversed in an a priori-fixed order of obtained constraints. In this paper, we show how GDN can neatly deal with various problems in natural language processing, such as word sense disambiguation, ellipsis resolution. Merits of using GDN for them are described.

# 1 はじめに

弁別ネットワークは決定木 [3] の一般化であり，問題解決に広く用いられている [5]．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 [8, 14, 11, 1]．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．葉ノードに解がそれぞれ位置する．枝にはラベルとして制約が付与されている．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，得られた制約で満足される枝がたどる指針となる．弁別ネットワークには次のような利点がある．一つは，(複数の可能な解が存在する) 曖昧性を表現する際に，選択肢として複数の候補を持つのではなく，(それらをすべて下位ノードとして包含する) 一つの曖昧な表現 (ノード) として持つことができることである．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 [6] が目的とする解の増進的洗練過程と考えられる．二つ目は，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よいことである．これは，解の探索が葉ノードに位置するそれぞれの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．可能な解の線形探索は，可能な解の個数を $n$ とすると，$O(n)$ の時間を要するのに対し，この探索は $O(l)(= O(\log n))$ の時間で行なえる [2]．ここで，$l$ は木の高さである．三つ目は，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) が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なわれないので，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．一つは，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．この順序は弁別ネットワークの構造に依存して決まる．二つ目の問題点はより深刻である．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．このような状況では，それまでに得られた制約はすべて何の役にも立たず捨てられてしまうことになる．

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 [16]．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 をたどることができる．

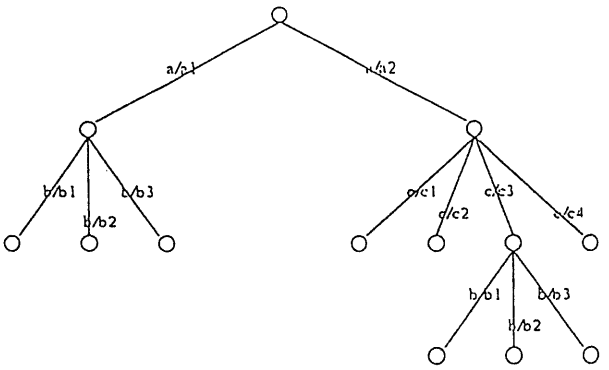


図 1: 弁別ネットワークの例

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，語義や係り受けの曖昧性解消，省略の補完処理がどのように行なわれるかについて述べる．

2 節では，弁別ネットワークと一般化弁別ネットワークについて概観する．3 節では，GDN を用いた日本語解析について述べる．

# 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[1]．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．その他のノードは，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，得られた制約で満足され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．この過程で異常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見つかったことを意味する．

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一方で，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 [1, 22]，[16] で指摘したように，これらはどれもそれぞれ問題があり，十分な解ではないと考える．

[1] 説明を簡単にするため，本稿では例として弁別木しか用いない．

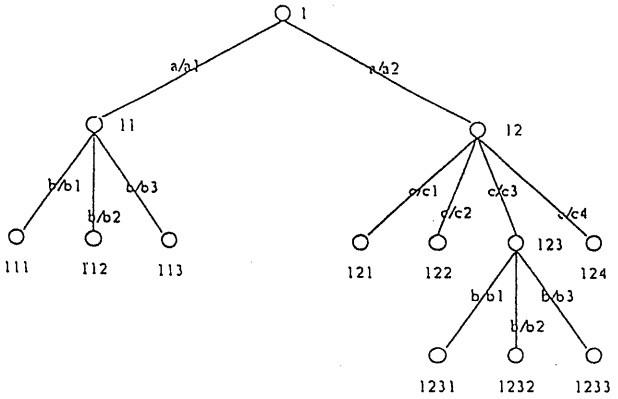


図 2: ノードに識別子を付与した弁別ネットワーク

## 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．まず，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 $1i_1i_2...i_m$ に対して，その子ノードには $1i_1i_2...i_mi$(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．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 1 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とを意味する．

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 1 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順序は本来 $a/a_2$, $c/c_3$, $b/b_1$ 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 $a/a_2$ が得られておらず，また，得られた制約についても順序が本来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 1 から得られる; $\{111\,if\,a/a_1, 1231\,if\,a/a_2\,and\,c/c_3\}$)，および現在の状態 (初期状態) を用いて，次のように計算される．

> 111,1231 はどちらも 1 を接頭部分文字列 (数字列) として含むので，長い方の文字列である 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，$\{111if\,a/a_1, 1231if\,a/a_2andc/c_3\}$ 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片方のノードへ可到達であるかどうかを調べる．これは，図 2 から明らかなように，あるノードから弁別ネットワーク上で可到達なノードには，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．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，解析は，識別子に関して一方がもう片方を接頭部分文字列として含まない場合に失敗する．

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なう．

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

表 1: 枝のラベル (制約) と対応する識別子

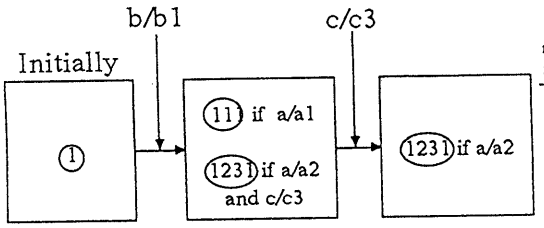


図 3: GDN を用いた弁別過程

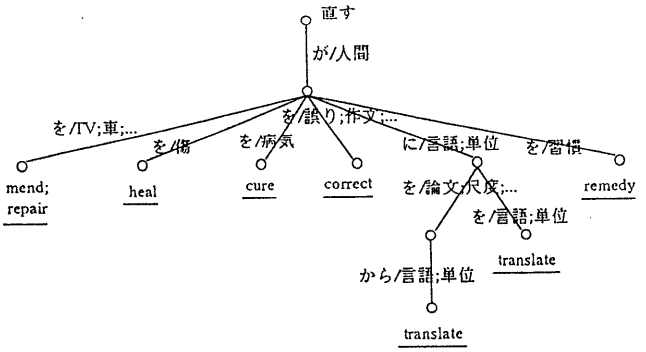


図 4: 動詞「直す」の語義弁別ネットワーク (一部)

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 $if\ a/a_2\ and\ c/c_3$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 $c/c_3$が得られ，if 節から取り除かれるので，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，「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．

このように，if 節中の制約は，得られることが仮定されており，後に得られると思われる制約の予測として使うことができる．また，if 節中の制約は，abduction[7, 15] における仮説と見なすことができる．(葉ノードに対応する) 規則の結論が成り立つためには，if 節中の前提条件 (制約) が成り立たなければならないということを if 節は意味しているからである．

## 3 GDN を用いた日本語解析

図 4 に動詞「直す」の語義および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ことが記述されている．弁別ネットワークのノードは，枝に記述された制約により弁別 (選択) された語義を表す．ネットワークの葉ノードは曖昧性のない 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．葉ノード以外のノードは，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，意味が未知であることを意味する．根ノードから葉ノードまでのパス中にある制約の集合が，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，得られた情報が満足する制約の枝に沿って弁別ネットワークを根ノードから葉ノード (曖昧性のない語義) 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，文中の回りの単語の情報 (動詞が取る格要素の情報) から，異常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，[8, 14, 11, 1] 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 節で挙げたような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増進的に行な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，増進的曖昧性解消手法 [13] は曖昧性解消のために望ましい方法であると考える．

また，文に含まれる曖昧性がその文中では十分解消できず，後続する文の情報を考慮しなければ解消できない場合にも，増進的曖昧性解消手法は自然に対処できると考えられる．増進的曖昧性解消過程は，曖昧性を含んだ (未決定の部分が残った) 意味解析結果を，新たに得られた制約により増進的に洗練し決定していく過程であり，1 文に対する意味解析結果として曖昧性を含んだ表現を許し，その曖昧性を含んだ意味解析結果を，後続する入力から得られる付加的情報により更新することが容易

に実現できるからである．

弁別ネットワーク中の葉ノード以外のノードはすべて曖昧性を含んだ意味表現と解釈できるので，未決定の部分が残った意味解析結果を，葉ノード以外のノードにより容易に表現できる．また，後続する入力から得られる付加的情報により新たな決定を行なうことは，現在到達しているノードから新たに得られた情報に基づいてさらにネットワークを下向きにたどることに自然に対応する．したがって，弁別ネットワーク表現は増進的曖昧性解消手法との整合性が大きい．

しかし，増進的に得られる制約の順序はあらかじめ定められた通りであるとは限らず，また，次に入力されるはずの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，増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，解析過程で増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実現する．英語の関係節や受動態の例から明らかなように，弁別ネットワークにとって問題となる状況はしばしば生じる．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からの逸脱の度合い，すなわち制約が得られる順序の非決定性がかなり大きい．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い．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．

次に，GDN によりどのように省略の補完処理を行な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 [4, 20] 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位置を検出したり，選択制限を用いて補う句の候補の妥当性を検査したりすることは容易に行なえるが，補う句の候補を発見したり，それらの間の優先性を文脈から決定したりするのには他の機構が必要である [19, 9]．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．

省略された句の位置は GDN では 2 つの方法で検出される．2 節で述べた if 節中の制約は得られていないが，得られるはずであることから，その位置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，到達している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考える．なぜなら，動詞の意味は本来曖昧性がない方が望ましいと考えられるからである[2]．この場合，省略の補完処理を行なうことにより，語義の曖昧性をさらに解消することが可能になる．補う句の候補に，省略された句の位置の選択制限を適用することで，補う句の意味的妥当性が検査される．

最後に，GDN を用いてどのように増進的曖昧性解消と省略補完処理が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，制約の得られた順序は，[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，省略された句を補う句の候補として，[論文，日本語，太郎] が得られる．ここでは，前にある要素の方が優先される．二番目の文は，次のように解析される．

> 「英語」は言語であり，「に」格の選択制限を満足する．よって，ノード 1 に到達できる．次に，「花子」が「が」格の選択制限を満足し，その結果，ノード 1 の到達可能性が無条件になる．「直す」の語義の曖昧性は，7 個から 2 個に減少するが依然曖昧である[3]．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，省略補完処理を試みる．そして，省略された句の位置の候補として，{ 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．補う句の候補として上のように得られたものから，「論文」，「日本語」がそれぞれ省略された句の候補である「を」，「から」格の選択制限を満足するので，省略補完処理が成功し，「直す」の語義は 1 つに決定される．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．

さて，GDN を用いた増進的解析には，他にどのような利点があるかを次に考える．図 4 に示したような一つの動詞だけではなく，動詞全体に対する弁別ネットワークを構成した場合を考えてみよう．動詞全体の弁別ネットワークでは，ネットワークの根ノードから葉ノードまでのパス中に，そのパス (格フレーム) がどの動詞のものであるかを表す，動詞の見出し語をラベルに持つ枝が新たに付加される．非常に trivial な例を図 5 に示す．動詞全体について弁別ネットワークを構成することにより，一つの動詞についての弁別ネットワークを構成する場合よりも，より多くの格フレーム集合を圧縮して表現でき，線形探索の比率を下げ，また，曖昧な語義をより少ないノードとして表現できるので (1 節参照)，解析の効率はさらに改善される．

GDN を用いることにより増進的な語義の曖昧性解消が実現できることを上で述べた．これは裏を返すと，文を解析している過程で得られる格要素の情報から，文の後方にある未解析の動詞の意味をある程度推測できる可能性があることを意味している．解析過程で段階的に得られる名詞句の意味と格情報を用いて動詞の弁別ネットワークをたどることで，動詞自体を解析する以前に，動詞の意味をある程度推測することが可能である

[2] 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他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．

[3] 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．

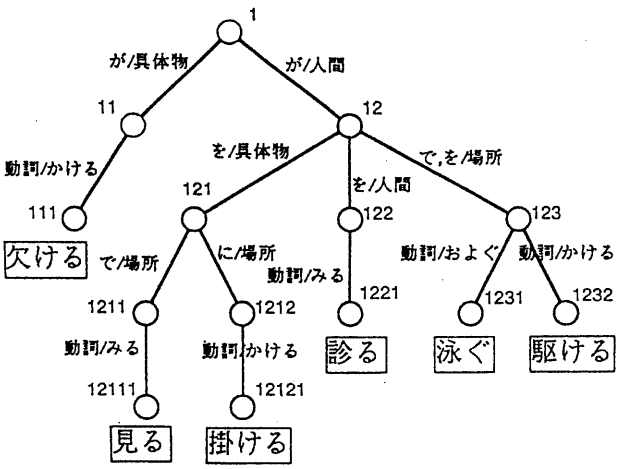


図 5: 動詞の語義弁別ネットワークの例

からである．たとえば,「医者が」,「患者を」という名詞句が次々に得られた場合を考える．この場合，図 5 のネットワークをたどることで，次に動詞として「診る」が得られるのではないかという予測が行なえるわけである．

また，GDN を用いることで，係り受けの曖昧性も増進的に解消することが可能になる．係り受けの曖昧性は，構造的曖昧性の一つとしてよく研究されており，曖昧性の数が組合せ的に爆発することが知られている [23]．この曖昧性の数の爆発は，局所的曖昧性が組み合わさることで生じていることを考慮して，それを防ぐため，局所的曖昧性のみを表現し，その上で曖昧性解消を行なう手法が提案されている [12, 21]．我々は，このような，一度文全体を解析した後曖昧性を解消していく二段階手法とは異なり，文を解析する過程で増進的に係り受けの曖昧性を解消する．できる限り早期に異常な係り受けの候補を排除することで，その候補から派生する多くの曖昧性を考慮しないで済むと考えるからである．

GDN を用いて増進的に係り受けの曖昧性を解消する手法の詳細は，[24] に譲る．ここでは，GDN を用いた係り受けの曖昧性解消の利点として，異常な係り受けを (従来より) 早期に排除できることについてのみ述べる．弁別ネットワークでは，上述したように，一つの格フレーム中で共起する名詞句の並びを根ノードから葉ノードまでのパス中の枝として表現している．このため，係り受けの解析の際に，今得られた名詞句がそれまでに得られた名詞句と同じ動詞に係るとして解析する場合は，現在到達しているノード[4]から，今得られた名詞句の情報を用いてさらにネットワークをたどることに対応する．たとえば,「太郎が」,「道を」という名詞句が次々に得られた場合，これらが同じ動詞に係ると解析することは，図 5 のネットワークで，ノード 12 を経て，ノード 123 までたどることに相当する．したがって，現在到達しているノードから，今得られた名詞句の情報を用いてネットワークをさらにたどることができない場合，それまでに得られた名詞句と今得られた名詞句を同じ動詞に係るとする係り受けの候補は排除することができる．たとえば,「太郎を」,「道を」というように名詞句が続いた場合[5]，ノード 122 から「道を」の情報を用いてネットワークをたどることができない．このような場合，係り受けの曖昧性が部分的に解消され，それらが別々の動詞に係るとする係り受けの候補のみが残ることになる．ここで，この解析 (到達しているノードからネットワークをさらにたどること) は，動詞が出てくる以前に行なえることに注意して頂きたい．係り受けが名詞句と動詞の間の関係であることから，係り受けの曖昧性解消は従来動詞が現れて初めて実行されることが多い．それに対し，GDN を用いることで，名詞句が次々に現れた過程で動詞が現れるのを待つことなく，増進的に係り受けの曖昧性解消を行なうことができる[6]．

[4] 過去に得られた名詞句の情報からこのノードに到達している．

## 4 おわりに

GDN の日本語解析への応用について述べた．GDN を用いることにより，語義や係り受けの曖昧性解消，省略の補完処理がどのように行なわれるかについて述べた．

最後に，GDN の他の応用法の一つについて述べる．我々は GDN が MOP を索引付けする手法として適していると考える．MOP(Memory Organization Packages)[17] は，script[18] を階層化したものであり，テロに関する新聞記事といった，特定の領域のテキスト理解によく用いられる．通常 MOP は multiple redundant discrimination network の形で索引付けされることが多い [10]．multiple redundant network とは，ラティスと同様に，同じスクリプトに到達可能な複数の冗長なパスが用意されているネットワークであり，制約が得られる順序が自由であることから，1 節で述べた弁別ネットワークの問題点を考えた場合必然的である．それに対して，GDN を用いれば，この問題を解消でき，MOP を冗長なリンクのない，より簡潔な形式で索引付けすることができる．

## 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, editors, *Lexical Ambiguity Resolution : Perspectives from Psycholinguitics, Neuropsychology, and Artificial Intelligence*, pp. 13–43. Morgan Kaufmann Publishers, 1988.

[5] 非常に人工的な印象を受けるが,「太郎を道を駆けて来た医者が診た.」といった文を考えて欲しい．

[6] 上で述べた二つの利点は，動詞全体についての弁別ネットワークを構成して初めて得られるものではない．なぜなら，動詞ごとの弁別ネットワークは，動詞全体のネットワークを一つ一つの動詞ごとに分割した表現に過ぎないからである．

[2] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *Data Structures and Algorithms.* Addison-Wesley, 1983.

[3] G. Brassard and P. Bratley. *Algorithmics.* Prentice Hall, 1988.

[4] 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.

[5] E. Charniak, C.K. Riesbeck, and D.V. McDermott. *Artificial Intelligence Programming.* Lawrence Erlbaum Associates, 1980.

[6] M. Dincbas. Constraints, logic programming and deductive databases. In *Proc. of the France-Japan Artificial Intelligence and Computer Science Symposium'86*, pp. 1–27, 1986.

[7] J.R. Hobbs, M. Stickel, P. Martin, and D. Edwards. Interpretation as abduction. In *Proc. of the 26th Annual Meeting of the Association for Computational Linguistics*, pp. 95–103, 1988.

[8] P.S. Jacobs. Concretion: Assumption-based understanding. In *Proc. of the 12th International Conference on Computational Linguistics*, pp. 270–274, 1988.

[9] M. Kameyama. *Zero Anaphora : The Case of Japanese.* PhD thesis, Stanford University, 1985.

[10] J.L. Kolodner. *Retrieval and Organizational Strategies in Conceptual Memory.* Lawrence Erlbaum Associates, 1984.

[11] S.L. Lytinen. Are vague words ambiguous? In S.L. Small, G.W. Cottrell, and M.K. Tanenhaus, editors, *Lexical Ambiguity Resolution : Perspectives from Psycholinguitics, Neuropsychology, and Artificial Intelligence*, pp. 109–128. Morgan Kaufmann Publishers, 1988.

[12] H. Maruyama. Structural disambiguation with constraint propagation. In *Proc. of the 28th Annual Meeting of the Association for Computational Linguistics*, pp. 31–38, 1990.

[13] C. S. Mellish. *Computer Interpretation of Natural Language Descriptions.* Ellis Horwood, 1985.

[14] G.D. Moerdler and K.R. McKeown. Beyond semantic ambiguity. In *Proc. of the 7th National Conference on Artificial Intelligence*, pp. 751–755, 1988.

[15] P. Norvig and R. Wilensky. A critical evaluation of commensurable abduction models for semantic interpretation. In *Proc. of the 13th International Conference on Computational Linguistics*, Vol. 3, pp. 225–230, 1990.

[16] 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.

[17] R. Schank. *Dynamic Memory: A Theory of Learning in Computers and People.* Cambridge University Press, 1982.

[18] R. C. Schank and R. P. Abelson. *Scripts, Plans, Goals and Understanding.* Lawrence Erlbaum Assoc., 1977.

[19] C. L. Sidner. Focusing in the comprehension of definite anaphora. In M. Brady and R. C. Berwick, editors, *Computational Models of Discourse*, pp. 267–330. MIT Press, 1983.

[20] 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.

[21] R. Sugimura, H. Miyoshi, and K. Mukai. Constraint analysis on japanese modification. In *Proc. of the 2nd International Workshop on Natural Language Understanding and Logic Programming*, pp. 5–18, 1987.

[22] W.A. Woods. Taxonomic lattice structures for situation recognition. In *Theoretical Issues in Natural Language Processing 2*, pp. 33–41, 1978.

[23] 奥津敬一郎. 「ボクハウナギダ」の文法 – ダとノ –. くろしお出版, 1978.

[24] 秋葉友良, 伊藤克亘, 奥村学, 田中穂積. 増進的曖昧性解消モデルに基づいた統合的日本語解析. 「自然言語処理における統合」シンポジウム論文集, pp. 93–100, 1991.