

2. 人間と機械における情報の表現と意味

一般化弁別ネットワーク

— 意味情報の新しい表現法 —

田中 穂積

東京工業大学工学部

奥村 学

北陸先端科学技術大学院大学情報科学研究科

弁別ネットワークを拡張し、任意の順序で制約が得られても、また、部分的に情報が得られなくても、たどることができる一般化弁別ネットワーク(GDN)をわれわれは提案している。本稿では、GDNを用いて、日本語の語義の曖昧性を漸進的に解消する手法について述べる。

1. はじめに

人間と機械との間の高次コミュニケーションの形態として、人間にとって最も自然で素直なものは、自然言語を用いることであろう。一般に、自然言語によるコミュニケーションの過程で意味情報がやり取りされるが、それは機械の内部になんらかの形で符号化され、表現されなければならない。意味情報の内部表現とその働きを理解することにより、初めてコミュニケーションのプロセスについての理解の道が開かれる。以上のことから、「人間と機械における情報の表現と意味に関する研究」の重要性が理解できると思われる。

ところで、人間の行う自然言語の意味理解の過程では、意味的曖昧性の解消が絶えず行われている。たとえば「あめが降る」という文を聞いた時、常識的には、「あめ」は「鉛」ではなく「雨」である、という曖昧性解消が行われている。こうした意味的曖昧性の解消は、文頭の「あめ」を聞いただけでは不可能で、「降る」という語を聞いて初めて可能になることに注意したい。曖昧性解消は文を読み進む(聞き取る)につれて、漸進的に行われるのである。このようなことを機械の側に行わせるためには、どうしたらよいだろうか。本章では、そのための一解決策を提案する。具体的

には、意味情報の表現形式として一般化弁別ネットワークを用い、それによる漸進的な意味的曖昧性解消モデルを提案する。

一般化弁別ネットワークの説明に入る前に、弁別ネットワークについて説明しておく。

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¹⁾。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^{2)~5)}。弁別ネットワークは有向非循環グラフであり、1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、入力された情報で満足される制約(枝)がたどる指針となる。

弁別ネットワークには、3節で述べるように、3つの利点があるが、同時に2つの問題点がある。この問題点を解決するため、われわれは一般化弁別ネットワーク(generalized discrimination network; GDN)を提案している⁶⁾。本稿では、GDNの日本語解析への応用について述べる。

自然言語解析において重要な問題の1つに、解析する文の曖昧性をどのように解消するかという問題がある。曖昧性を十分に解消できないと、文の

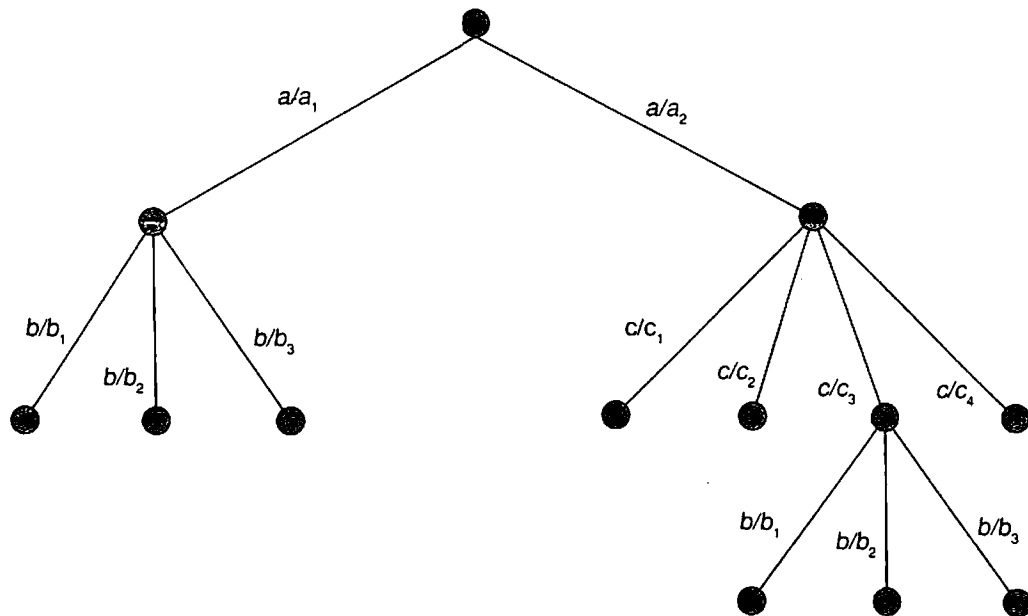


図1 弁別ネットワークの例

解析結果として妥当でないものが多数得られてしまうからである。自然言語解析で問題となる曖昧性には、構文解析レベルでの構造的曖昧性^{*}、意味解析レベルでの語義の曖昧性、文脈解析レベルでの照応の指示対象に関する曖昧性など多数が存在する。

その他に、日本語には次の特徴がある。日本語の文は文節(“自立語+付属語列”)の並びとして表現できるが、

- ・文節の出現順序(語順)が比較的自由である、
- ・文脈から容易に補うことができる文節は頻繁に省略される。

したがって、これらの特徴を自然に扱うことのできる解析モデルを考える必要がある。

本稿では、GDNを用いて、日本語の語義の曖昧性を漸進的に解消する手法について述べる。われわれが提案する漸進的曖昧性解消モデルは、文を解析する過程で、得られた情報をその時点で処理しながら、段階的に曖昧性を少しずつ解消していく計算モデルである⁷⁾。GDNを用いることにより、漸進的曖昧性解消過程で、日本語の語順の自由さにうまく対処できることを示す。

2. 漸進的曖昧性解消モデル

漸進的曖昧性解消モデルに関する研究には、他に Mellish⁸⁾、Hirst⁹⁾のものがある。Mellishは、「文を読み進みながら部分的に意味解析を行い、文全体を読み終った時点で意味解析結果が得られる」という、人間に近い意味解析のモデルを提案している。Mellishはこの手法を早期意味解析と呼んでいる。Mellishは、この早期意味解析を用いて名詞句の指示対象を漸進的に決定する手法について述べている。

Hirst⁹⁾は、ポラロイド語(Polaroid Word; PW)という概念を用いて意味的曖昧性を解消する手法を提案している。ポラロイド語とは、ポラロイド写真からの比喻で命名されたものであり、(ポラロイド語が表す)意味的对象の自律的現像「写真」が、徐々に望ましい意味的对象の絵として現像されていくことを意味する。ポラロイド語は、曖昧な単語の意味(意味解析の途中結果)をそのまま表現することが可能なデータである。また、ポラロイド語は、互いに情報を交換し合うことで周囲の単語の語義と両立しない語義を排除し、自分自身が表す意味的对象の曖昧性を解消していく、協調的かつ自発的な手続きでもある。

^{*}日本語の係り受けの曖昧性、英語の前置詞句付加(PP-attachment)の曖昧性は、この範疇に入る。

3. 弁別ネットワークと一般化弁別ネットワーク

3.1 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^{*}。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も、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枝は排除され、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。

弁別ネットワークには次のような利点がある。第1は、(複数の可能な解が存在する)曖昧性を表現する際に、選択枝として複数の候補をもつのではなく、(それらをすべて下位ノードとして包含する)1つの曖昧な表現(ノード)としても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が目的とする解の漸進的洗練過程と考えられる。

第2は、解を1つ1つ逐次的に調べていく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。

第3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1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、同じ制約を含む複数の規則における冗長な再計算を回避すること

ができる。

このように、弁別ネットワークには3つの利点がある一方で、2つの重大な問題がある。第1は、弁別ネットワークはあらかじ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。

第2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。

この問題点を解決するため、われわれは、任意の順序で制約が得られても、また、部分的に情報が得られなくても、たどることができ一般化弁別ネットワーク(generalized discrimination network; GDN)を提案している⁹⁾。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。

3.2 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、あるノード $1i_1i_2 \dots i_m$ に対して、その子ノードには $1i_1i_2 \dots i_mi$ (ここで、 i は1から n までの整数、

^{*}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。

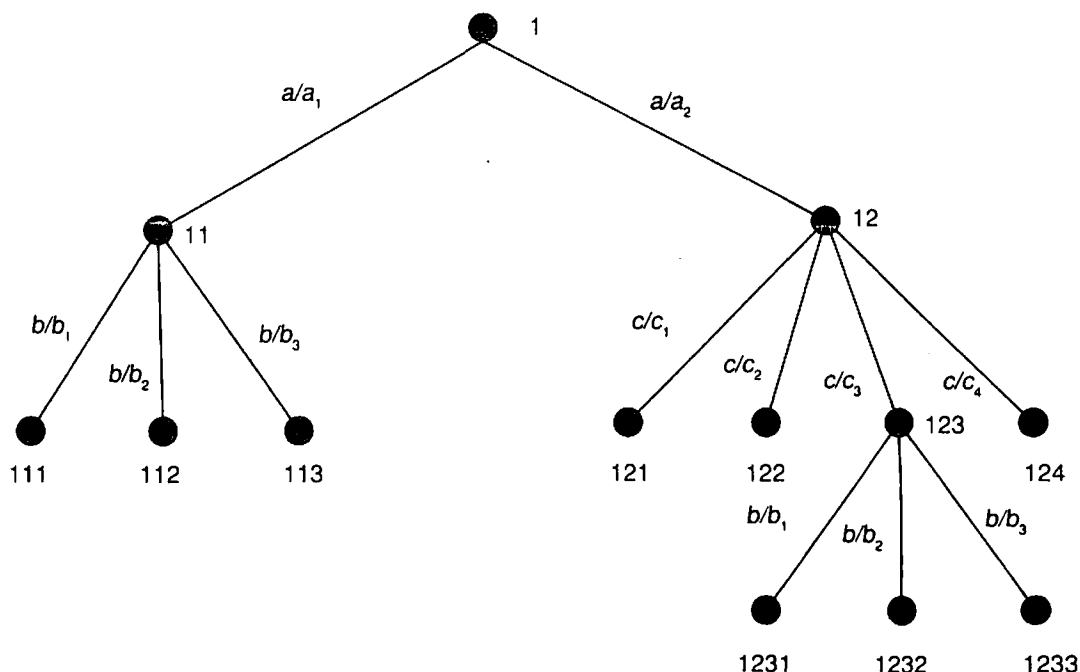


図2 ノードに識別子を付与した弁別ネットワーク

表1 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

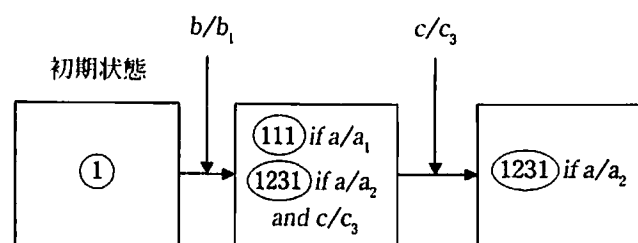


図3 GDNを用いた弁別過程

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、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、他の枝の制約に注意する必要がある。

上の例の識別子 122 の場合、根ノードから識別子のノードに到達するためには、他に制約 a/a_2 が満足される必要がある。したがって、上述した、制約 c/c_2 が満足された時の識別子 122 のノードへの可到達性は、「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」の可到達性であることが分かる。

このようにして、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、満足されなければなら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。上述した制約 c/c_2 に対応する条件付き識別子は、122 if a/a_2 になる。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ほしい。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: 1)を用いて、次のように計算される。

識別子同士(111, 1231と1)を比べると、111, 1231はどちらも1を文字列(数字列)の先頭に部分として含むので、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集合から、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。

ここで示した一般化弁別ネットワークをたどる手法が、通常の弁別木の動作とは以下の点で異なることに注意してほしい。われわれの手法では、制約集合 { c/c_1 , a/a_1 , b/b_1 } が得られた場合、それらの制約をすべて用いてネットワークをたどることができないため、どのノードにも到達できない。一方、通常の弁別木では、制約 c/c_1 を不要な情報として、たどる際に用いないことにより、(少なくとも)ノード111に到達することができる。通常の弁別木では、入力された制約を'don't care'として無視する場合があるのに対して、われわれの手法では、入力された制約はすべてネットワークをたどるのに用いる。

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 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 が得られたので、if節から取り除かれ、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

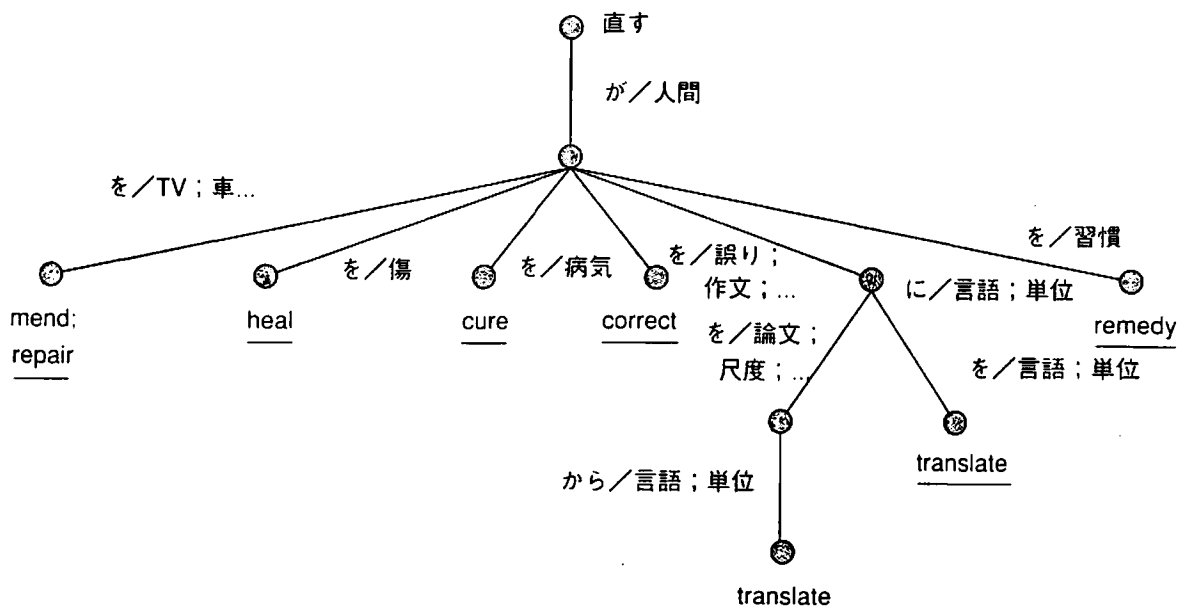


図4 動詞「直す」の語義弁別ネットワーク (一部)

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^{10),11)}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。

4. GDN を用いた漸進的語義曖昧性解消

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、その表層格と選択制限の組が記述される。図4では、「直す」が「が」、「を」などの格を取り得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)に到達し得ることから、到達し得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、異常な選択肢が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2~5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、3.1節で挙げたようなものが述べられている。

具体的に、Hirst のポラロイド語⁹⁾のような、従来の手法と比較してみる。ポラロイド語では、曖昧性を表現する際に、単語の複数の語義をリスト

形式で保持する。たとえば、「はしをかける」という文では、「はし」に「橋」、「端」、「箸」の3通りの、また、「かける」に「掛ける」、「賭ける」、「欠ける」、「駆ける」の4通りの曖昧性(同音異義性)がそれぞれ少なくとも存在するが、それらをそれぞれ(橋、端、箸)、(掛ける、賭ける、欠ける、駆ける)のように表現する。これらの表現を用いた、上文の解析(曖昧性解消)では、「かける」の4つの語義の「を」格の選択制限を、「はし」の3つの語義に対して単純な組み合わせで適用する。すなわち、4つの「を」格の選択制限のうち共通なものがあったとしても、 $3 \times 4 (=12)$ 通りの組み合わせの検査を実行する。

これに対して、弁別ネットワークを用いると、曖昧性を表現する際に、複数の候補をもつのではなく、(それらをすべて下位ノードとして包含する)1つの曖昧な表現(ノード)としてもつことができる。また、格フレーム集合を圧縮して表現することから、同一格の同じ選択制限は1回しか検査されず、同じ選択制限を含む複数の語義(格フレーム)における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)は組み合わ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⁸⁾は曖昧性解消のために望ましい方法であると考えられる。

しかし、漸進的に得られる制約の順序はあらかじめ定められたとお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、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実現する。英語の関係節や受動

態では、主語や目的語の出現する順序が本来の文と異なっていたり、また主語が欠落していたりするように、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はかなり大きい。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。

5. 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、語義の曖昧性解消がどのように行われるかについて述べた。

また、4節の語義の曖昧性解消の説明の際、語義として機械翻訳における訳語を用いたことから明らかのように、機械翻訳における動詞の意味の訳し分けにも GDN は応用可能であると考えられる。

文 献

- 1) Charniak E, Riesbeck CK, McDermott DV: Artificial intelligence programming, Lawrence Erlbaum Associates, 1980
- 2) Jacobs PS: Concretion: Assumption-based understanding, (In) Proc. of the 12th International Conference on Computational Linguistics, pp.270-274, 1988
- 3) Moerdler GD, McKeown KR: Beyond semantic ambiguity, (In) Proc. of the 7th National Conference on Artificial Intelligence, pp.751-755, 1988
- 4) Lytinen SL: Are vague words ambiguous? (In) Small SL, Cottrell GW, Tanenhaus MK (eds): Lexical Ambiguity Resolution: Perspectives from Psycholinguistics, Neuropsychology, and Artificial Intelligence, pp.109-128, Morgan Kaufmann Publishers, 1988
- 5) Adriaens G, Small SL: Word expert parsing revisited in a cognitive science perspective, (In) Small SL, Cottrell GW, Tanenhaus MK (eds): Lexical ambiguity resolution: Perspectives from Psycholinguistics, Neuropsychology, and Artificial Intelligence, pp.13-43, Morgan Kaufmann Publishers, 1988

- 6) Okumura M, Tanaka H: Towards incremental disambiguation with a generalized discrimination network, (In) Proc. of the 8th National Conference on Artificial Intelligence, pp.990-995, 1990
- 7) 奥村学, 田中穂積: 自然言語解析における意味の曖昧性を増進的に解消する計算モデル, 人工知能学会誌, Vol.4, No.6, pp.687-694, 1989
- 8) Mellish CS: Computer interpretation of natural language descriptions, Ellis Horwood, 1985, (訳) 田中穂積: 自然言語意味理解の基礎, サイエンス社, 1987
- 9) Hirst G: Semantic interpretation and the resolution of ambiguity, Studies in Natural Language Processing, Cambridge University Press, 1987
- 10) Hobbs JR, Stickel M, Martin P, Edwards D: Interpretation as abduction, (In) Proc. of the 26th Annual Meeting of the Association for Computational Linguistics, pp.95-103, 1988
- 11) Norvig P, Wilensky R: 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

コンピュータ科学 1月号

1992年 Vol.2 No.1

好評発売中

 定価1030円
 (本体価格1000円・税30円)

特集●メディア・スペース

責任編集 田中 譲

1 [序説]

メタメディアからメディア・スペースへ

田中 譲

2 [メディア・アーキテクチャ]

開放型統合メディア環境

田中 譲

3 [ヒューマンインタフェース]

メタウェアとエージェンシーモデル

田村浩一郎

4 [バーチャルリアリティ]

バーチャルリアリティとドラマ理論

村上公一

5 [ポータブルな環境空間]

パームトップとメディア環境

平山智史

6 [NETTOWN]

情報メディア空間と情報都市

野々垣 旦

7 [概念ブラウザと個人用情報データベース]

概念空間で断片情報を体系化する

藤澤浩道

8 [語彙空間]

電子化辞書は語彙の知識ベース

横井俊夫

9 [ゲーム]

ゲーム空間の創出するもの

中本伸一

特別寄稿

 エンジニアリングのための大規模知識ベース:
 スタンフォード知識システム研究所の
 How Things Workプロジェクト

E. A. Feigenbaumほか

コンピュータ応用の最前線6

数値予報 雨を降らせるコンピュータ

新田 尚

World Watching

ブレンダと愉快なVR仲間達(その2)

村上公一

Japan Watching

ローテク・ハイテクのベルリンより

Volker Grassmuck

My Favorite Tools

紺屋の白袴

竹内郁雄



秀潤社

 〒108 東京都港区白金台3-19-1 (第31興和ビル6F)
 TEL (03) 3440-6371 (代表) FAX (03) 3444-4092
 振替 東京3-7875