

増進的曖昧性解消モデルに基づいた統合的日本語解析

3C-3

秋葉友良 伊藤克巨 奥村 学 田中穂積
(東京工業大学)

1 はじめに

自然言語解析における大きな問題の一つは、解析の途中で生じる様々な曖昧性をどのように解消するかということである。自然言語に曖昧性が遍在するのは、文を構成する各部分の情報とどれも部分的で、文の他の部分の情報をいれないと曖昧性を解消できないことに起因している。我々は、意味の曖昧性を増進的に解消する手法を提案している [2]。

さらに、曖昧性を十分に解消するためには、形態素・構文・意味・文脈などの各解析レベルの処理に、他のレベルの情報をいなければならぬ。各レベルの処理を独立して行ない、処理終了後、次の処理に結果を渡すという方法では、各レベルで部分的な解析しか行なうことができないため、曖昧性の数が組合せ的に爆発してしまう。望ましい方法は、各解析レベルの処理が入力文に対して同時的に行なわれ、各処理が相互に他の処理の解析結果を利用して曖昧性を解消することである。すなわち、増進的曖昧性解消モデルにとって、レベルの統合は不可欠な方向である。

本稿では、増進的曖昧性解消モデルに基づいて、各処理レベルを統合する問題について検討する。構文解析・意味解析のレベルについて、このような立場にたった一解析手法について述べる。この手法では、構文解析レベルとして一般化 LR 構文解析法 [3] を、意味解析レベルとして一般化弁別ネットワーク (GDN) [1] を使用し、日本語の構文と意味の解析を統合し、増進的に曖昧性解消を行なう。

2 日本語の構造的特徴

日本語には、文節の出現順序が比較的自由であるという特徴がある。さらに、文脈から容易に補うことができる文節は頻りに省略されるため、文の構造を反映する形式で記述した文法規則は曖昧性解消の強い制約とはならない。一方、文節内の形態素の接続についてはほぼ正規文法で表現できるため、構文解析による効果が期待できる。

本稿で提案する手法では、文節内の構文解析に一般化 LR 構文解析アルゴリズム [3] を用いる。一般化 LR 構文解析アルゴリズムは、解析の曖昧性に対して複数のスタックを保持し、探索を横型に行なうことで、文脈自由文法を扱えるように LR 構文解析法を一般化したアルゴリズムである。並列横型探索なので、増進的曖昧性解消モデルとの整合性が良い。

一方、文節間は意味情報を主導的に用いて係り受け解析を行なう。3節では増進的に係り受け解析を行なう手法について概要を述べる。また、意味解析を増進的に行なうモデルと

して、一般化弁別ネットワーク (GDN) を利用する。(4節)

3 増進的係り受け解析

日本語の文節間の係り受け関係を修飾する文節から修飾される文節へ引いたアークで表現すると、増進的な係り受け解析における処理過程を表す状態表現は、解析中の文節の上を通過しているアークの集合で表される。文節間の係り受けが非交差であるとするならば、集合へのアークの付け加えや取り出しは最も最近付け加えたアークから取り出されるので、アークの集合はスタックを使って表現できる。また、同じ動詞に係る文節から出るアークは一本にまとめる。これは、4節で述べるように同じ動詞に係る文節を一般化弁別ネットワークを用いて増進的に処理するためである。「太郎が川で泳いでいる人を見た」という文を増進的に係り受け解析を行い、「太郎が川で泳いでいる...」まで処理した状態でのスタック表現を図1に示す。

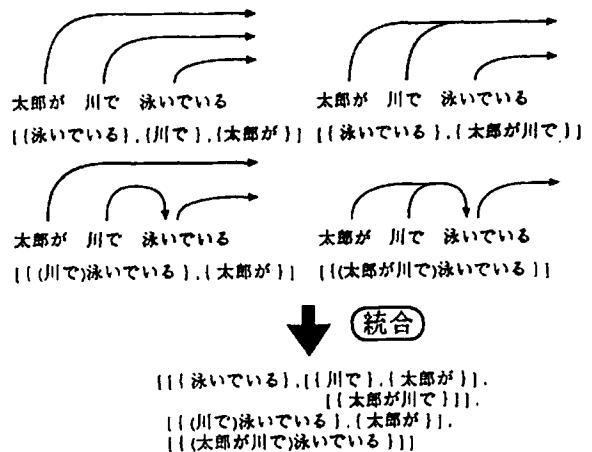


図1: 「太郎が川で泳いでいる...」のスタック表現

スタックは曖昧性に応じて複数個用意されるが、その内スタックのトップが同じものに対して統合を行なうことで、同じ処理が何度も行なわれることを避けることができる。これは、一般化 LR 構文解析アルゴリズム [3] で用いる木構造化スタックを用いて実現できる (図1)。したがって増進的係り受け解析は、LR 構文解析と同じくスタック操作に還元できる。

また、係り受け解析の制御を GDN を用いて行なうことにより、不適当な解析を早期に刈り定める。(5節)

4 GDN を用いた増進的意味曖昧性解消

意味的な曖昧性解消を、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がある。しかし、弁別ネットワークには、前もって決定された順序で制約が入力されないとたどれない問題がある。我々は [1] で、非決定的な順序で制約が

入力された場合でも弁別ネットワークを増進的にたどれる手法を提案している。制約の非決定的な順序に対処できる我々の弁別ネットワークを一般化弁別ネットワーク (Generalized Discrimination Network) と呼ぶ。

動詞全体の選択制限に対して GDN を構築することによって、増進的に意味的な曖昧性を解消できる。GDN を用いた曖昧性解消の過程は、ネットワークを根ノードから葉ノードへ向けて下向きにたどる操作に対応する。ネットワークの枝には格に対する制約 (選択制限) が付加される。葉ノードは動詞の各語義に対応する。ノードを下向きにたどると、到達可能な葉ノードが減少することから、解析中の文節に係る動詞の意味的な曖昧性が増進的に解消される。

図2は、動詞「みる(見る、診る)」「およぐ(泳ぐ)」「かける(掛ける、駆ける、欠ける)」によって構築した GDN の一例である。このネットワークを使って、「橋を川にかける」という文を解析することを考える。「橋を」を入力すると、制約“を/具体物”が満たされ、ネットワークをノード121までたどることができる。この状態において、到達可能な葉ノードは12111と12121であるから、動詞の意味は「見る」「掛ける」に絞り込まれたことがわかる。このように GDN を用いると、意味の曖昧性を、動詞が現れる前に早期に解消することができる。

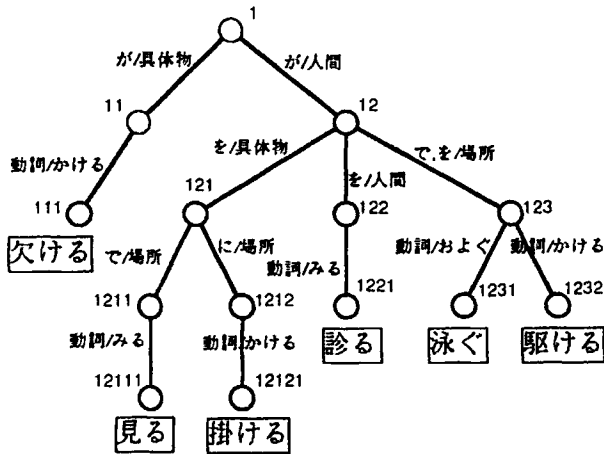


図2：一般化弁別ネットワーク

5 LR表とGDNによる構文・意味処理の統合

LR表とGDNは、共に計算しやすい形にコンパイルされた知識表現形式である。LR表は、スタックのトップに乗っている「状態」と入力から得られる先読みされた「終端記号」との関係から次の操作を決定し、最終的には次の「状態」に遷移する。一方GDNは、「状態」と入力から得られる「制約」との関係から、次の「状態」に遷移する。この類似性は、LRパーザがLR表を用いた解析を行なうのと同じように、LR表をGDNに置き換えて解析を行なう可能性を示唆している。

LR表とGDNの本質的な違いは、LR表が制約の入力順序を考慮に入れた状態遷移情報を表しているのに対して、GDNは制約の入力順序を自由にした場合の簡潔な状態遷移情報を表している点にある。文節間解析のために、LR表をGDNの代わりに用いるのは、ラティス表現の弁別ネットワーク (taxonomic lattice) に相当する。しかし、ネットワークをラ

ティス化したことにより、同じ情報量を表現するのに冗長な表現形式となり、計算機上で実現した場合に大量の記憶容量が必要になる[1]。

本稿で提案する手法では、語順が強い制約となる文節内の解析にはLR表を、語順が自由である文節間ではGDNを用いて日本語の文を解析する。その際、一般化LR構文解析アルゴリズムで使用されるスタックと、増進的係り受け解析(3節)で使用されるスタックとを区別せず、スタックをLR表と共にGDNでも操作することで、構文解析と意味解析を統合する(図3)。

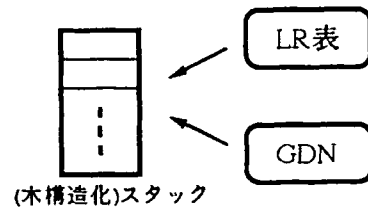


図3：LR表とGDNの統合

まず、LR構文解析によって文節内を形態素解析する。文節の終りまで解析が進むと、reduceプロセスが呼び出されて文節が得られる。文節間の解析もLR構文解析で行なう手法では、ここでgotoプロセスを呼び出し、得られた文節をスタックに積んで解析を進めるが、本手法では得られた文節をGDNに渡す。GDNは得られた文節とスタックに積まれている以前にたどったGDNの状態(ノード)を参照して係り受け解析を制御する。スタックに積まれている状態から得られた文節によってGDNを継続してたどれる場合はアークを統合し、同じ動詞に係ることを早期に決定する。また、スタックに積まれている状態からGDNをたどれないことは、係り受けのアークを統合できない、すなわち、埋め込み文の開始と見なすことができる。このように係り受け解析の制御にGDNを用いることで、動詞が現れる前に不適当な解析を防ぎ、早期に係り受けの曖昧性を解消することができる。

6 おわりに

増進的曖昧性解消モデルに基づいた日本語解析の一手法について述べた。本手法では日本語の特徴を考慮して、文節内ではLR表を用いた構文解析を行ない、文節間では意味主導に一般化弁別ネットワークを用いて係り受け解析を行なう。LR表と一般化弁別ネットワークは、共にスタックを操作する知識源として扱うことで同一視でき、処理の統合を実現できる。構文レベルと意味レベルを統合することにより増進的な日本語解析ができることを示した。

参考文献

[1] Manabu Okumura, Hozumi Tanaka. "Towards Incremental Disambiguation with a Generalized Discrimination Network". AAAI, 90.
 [2] 奥村学, 田中徳積. 「自然言語における意味的曖昧性を増進的に解消する計算モデル」. 人工知能学会誌, 4(6), 1989.
 [3] 沼崎浩明, 田村直良, 田中徳積. 「並列論理型言語による一般化LR構文解析アルゴリズムの実現」情報処理学会, NL74-5, 1989.