■2 群（画像・音・言語）- 10 編（自然言語処理）

3 章 構文解析

【本章の構成】

本章では，依存構造解析（3–1 節），句構造解析（3–2 節）について述べる．

■2 群 - 10 編 - 3 章

3-1 依存構造解析

■2 群 - 10 編 - 3 章

3-2 句構造解析

（執筆者：宮尾祐介，松崎拓也）[2009 年 8 月 受領]

句構造解析とは，文を入力とし，その句構造を出力する構文解析技術である．句構造解析は様々な自然言語処理アプリケーションで必要とされる基礎技術の一つであり，自然言語処理研究の初期から続く中心的な研究課題である．

文の句構造は，名詞句・動詞句など句の種類を表す記号を内部ノード，入力文中の各単語を葉ノードとする木構造（構文木）で表現される．例えば，図 3･1 は英語の文 “John ate the fish on the table” の 2 つの異なる句構造木の例である．NP は名詞句，VP は動詞句，PP は前置詞句，S は文を表す．2 つの句構造では前置詞句 “on the table” の係り先が異なっており，句構造の違いは文の意味の解釈の違いに対応している．

句構造をどのように形式化するかによって様々な文法理論があるが，現在最も一般的なものは文脈自由文法（CFG）を用いた句構造解析である．文脈自由文法は，自然言語の多くの構文構造を記述することができ，かつ効率的な処理が可能なことから，句構造解析を行うための数理モデルとして広く用いられている．

文脈自由文法 G は 4 つ組 (N, Σ, R, S) として定義される．ここで，N は非終端記号の集合，Σ は終端記号の集合，R は生成規則の集合である．R の各要素は $A \rightarrow \alpha$，ただし $A \in N$，$\alpha \in (N \cup \Sigma)^*$ という形をとる．また，S は開始記号と呼ばれる特別な非終端記号である．開始記号から終端記号列を導出する過程を木構造で表したものが構文木である．構文木中の 1 段の部分木は 1 つの生成規則に対応しており，親ノードの記号が規則の左辺，子ノードの記号を左から並べたものが規則の右辺に対応する．図 3･1 の構文木を生成する CFG を図 3･2 に示す．

ある文法 G と文 w が与えられたとき，文法 G による w の構文木を 1 つ，あるいは全て見つける問題を解析と呼ぶ．文脈自由文法のための効率的な解析アルゴリズムは複数知られており，代表的なものでは Cocke-Younger-Kasami 法（CYK 法），Earley 法，チャート法，一般化 LR 法が知られており，現在の構文解析においても広く利用されている．

文脈自由文法の各生成規則に，その規則の適用確率を割り当てた確率文脈自由文法 (PCFG) は応用上重要である．生成規則 $A \rightarrow \alpha$ の適用確率 $P(A \rightarrow \alpha | A)$ は，非終端記号 A が現れたときに，その記号に対し規則 $A \rightarrow \alpha$ が適用される条件付き確率と解釈される．構文木 T の確率 $P(T)$ は，T の中で規則 $A \rightarrow \alpha$ が適用された回数を $c(A \rightarrow \alpha)$ と表すとき，$P(T) = \prod_{A \rightarrow \alpha \in R} P(A \rightarrow \alpha | A)^{c(A \rightarrow \alpha)}$ となる．

図 3･1 に示したように，一般に，文に対して複数の構文木が与えられる（構造曖昧性）．しかし，句構造解析を利用するアプリケーションは通常一つの解析結果しか必要としないので，文法が与える複数の構文木の中からその文の解釈に対応する構文木を一つ選択すること（構造曖昧性解消）が必要である．近年の構文解析の研究では，曖昧性解消を正確に行い，解析精度を向上させる手法がさかんに研究されている．特に，現在の句構造解析技術ではコーパスから学習した統計モデルを用いて曖昧性解消を行う手法が主流である．Penn Treebank[1] や京都テキストコーパス[2]に代表される構文構造付きコーパス（ツリーバンク）が整備されたことにより，それを学習データとして教師付き学習を応用することで高精度を達成している．

句構造解析の曖昧性解消は，一般に以下のように定式化される:

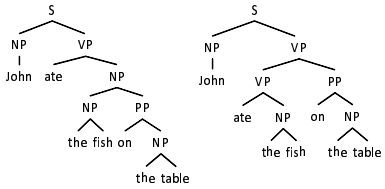

図 3·1　句構造の例

S → NP VP
VP → VP PP
VP → ate NP
PP → on NP
NP → NP PP
NP → John
NP → the fish
NP → the table

図 3·2　文脈自由文法の例

$$T = \underset{T' \in G(w)}{\operatorname{argmax}} f(T', w)$$

ここで w は入力文，T は w に対する構文木，$G(w)$ はある文法 G が w に与える構文木の集合，$f(T,w)$ は w に対する T の適切さを測るスコア関数である．スコア関数 $f(T,w)$ としては，PCFG に代表される確率的生成モデル $P(T,w)$ や，条件付き確率モデル $P(T|w)$ が代表的なものであったが，近年は最大マージン法に基づくものなど，確率モデル以外のスコア関数を用いる手法も発展している．

PCFG を用いた句構造解析は，大規模ツリーバンクから文法を抽出する手法や，句の主辞や親ノードの記号などの情報を非終端記号に付加して細分化する手法の開発によって 90 年代に大きく精度が向上した [3]．例えば，“*eat an apple*” という動詞句に対して，動詞が右側の名詞句と結合して動詞句を作る，という規則 VP→VB NP の各記号に主辞を付加すると VP_{eat}→VB_{eat} NP_{apple} という規則が得られる．このように規則を詳細化することで，“*eat*” が他動詞として使われている，という動詞の下位範疇化フレームに関する情報や，“*apple*” が “*eat*” の目的語になりやすいという選択選好の情報を規則確率に反映することができる．

ツリーバンクから抽出した大規模な文法に対して更に上のような詳細化を行うと規則数が膨大になり，規則確率の推定の際の疎データ性が問題となる．これに対しては，規則のマルコフ化，すなわち CFG 規則 $A \to \alpha$ による A から α への書き換えを，記号列 α を生成するマルコフ過程とみなして規則確率を平滑化する方法や，最大エントロピー法に着想を得た特殊な推定法などが開発された．

PCFG 以外の確率モデルを用いる手法としては，Left-corner 法や一般化 LR 法などの解析アルゴリズムの各ステップの動作（shift，reduce など）の確率を推定し，それらの積として構文木の条件付き確率を $P(T|w) = \Pi_{i=1}^{n} P(a_i|w, h_{i-1})$ のように定義する方法がある．ここで，a_i はアルゴリズムの 1 ステップの動作，h_i は動作列 $a_1, \ldots, a_i$ を入力文に適用した結果できる，部分的な構文木などの中間的な処理状態を表す．このような手法の利点としては，動作選択の確率 $P(a_i|w, h_{i-1})$ が中間的な処理状態に依存することを自然に表現できることが挙げられる [4]．

ここまでで述べた，確率モデルによるスコア関数をより一般化したものとして，構文木 T と入力文 w の組から様々な特徴量（素性）を取り出し，それらの重み付き和をスコア関数とする線形識別モデルが考えられる．

$$f(T,w) = \Sigma_i \lambda_i \phi_i(T,w)$$

ここで，ϕ_i は素性関数と呼ばれるもので，例えば $\phi_i(T,w)$ =“T の中の名詞句で ‘the’ から始まるものの数” といった特徴量を表す．重み λ_i を推定する方法としては，最大マージン法に基づく方法や，対数線形モデルの条件付き尤度 $P(T|w) = Z(w)^{-1} \exp(\Sigma_i \lambda_i \phi_i(T,w))$（$Z(w)$ は正規化係数）を最大化する方法などがある．PCFG モデルは，素性関数 $\phi_i(T,w)$ として各規則 $A \rightarrow \alpha$ が構文木 T の中で出現する回数，重み λ_i として規則確率の対数 $\log P(A \rightarrow \alpha|A)$ を用いた場合の線形識別モデルとみなせる．

PCFG を用いる場合，解析アルゴリズム，すなわち $\mathrm{argmax}_T f(T,w)$ を求める手法としては CYK 法に基づく Viterbi アルゴリズムないしビームサーチなど，動的計画法を用いることが多い．より一般的な線形識別モデルを用いる場合，同様に動的計画法に基づく手法を用いるには，構文木中の比較的狭い領域（一段の部分木など）だけに依存する素性関数に限定する必要がある．

この制約を乗り越え，より広汎な素性関数を用いるための手法として構文木のリランキングがある [5]．リランキングによる解析では，まず，入力文 w に対してベースとなる解析器，例えば PCFG に基づく構文解析を適用し，スコアの高い順に上位 N 個の構文木の集合 $N(w)$ を求める．次に，より複雑なスコア関数 $f(T,w)$ を用いて $N(w)$ 中の各構文木を再度評価し，最大のスコアを持つ構文木を最終的な出力とする．

CFG に基づく句構造は文の表層的な構造しか表現していないため，論理構造などの意味表現を得るためには更に複雑な後処理が必要となる．この問題に対して，CFG より詳細な情報を付加した句構造を用いて文法を記述することにより，直接意味表現を計算するような文法記述枠組みが提案されている．代表的なものでは主辞駆動句構造文法（HPSG）[6], 語彙機能文法（LFG）[7], 組合せ範疇文法（CCG）[8]などがあり，現在ではこれらに基づく構文解析器も広く利用されている．

句構造解析に関する今後の研究課題としては，これまで中心的に研究されてきた英語などの言語以外への取組みや，様々な種類のテキスト（特許文書やウェブなど）に対する適応の問題などが挙げられる．

■参考文献

1) M. Marcus, B. Santorini, and M. A. Marcinkiewicz：“Building a large annotated corpus of English: The Penn Treebank,” *Computational Linguistics* vol.19, no.2, pp.313-330, 1994.
2) 黒橋禎夫, 長尾　眞：“京都大学テキストコーパス・プロジェクト,” 言語処理学会第 3 回年次大会, pp.115-118, 1997.
3) M. Collins：“Head-driven statistical models for natural language parsing,” Ph.D Thesis, University of Pennsylvania, 1999.
4) A. Ratnaparkhi：“Learning to Parse Natural Language with Maximum Entropy Models,” *Machine Learning*, vol.34, no.1-3, pp.151-175, 1999.
5) E. Charniak and M. Johnson：“Coarse-to-fine n-best parsing and MaxEnt discriminative reranking,” in *ACL-05*, pp.173-180, 2005.
6) I. A. Sag, T. Wasow, and E. M. Bender：“*Syntactic Theory: A Formal Introduction*,” CSLI Publications, 2003.
7) J. Bresnan：“*The Mental Representation of Grammatical Relations*,” MIT Press, 1982.
8) M. Steedman：“*The syntactic process*,” MIT Press, 2000.