有限状態変換器を用いた漸進的構文解析

湊 恵一　　松原 茂樹　　稲垣 康善　　(名古屋大学)

1 はじめに

音声を聴きながらそれを同時に理解するには，実時間で，かつ，漸進的に構文解析する必要がある．そのための手法として，これまでに文脈自由文法に基づく漸進的構文解析手法が提案されているが [2]，実時間性については不十分であることが指摘されている．そこで本稿では，有限状態変換器 (finite state transducer: FST) を用いた漸進的構文解析手法について述べる．文脈自由文法を近似的に変換して作成した FST を構文解析に用いることにより，実時間での処理が可能となる．

2 有限状態変換器を用いた漸進的構文解析

本手法では，構文解析の入力は品詞列であるとする．構文解析に用いられる FST の例を図 1に示す．FST の各状態遷移には，入力と出力が割り当てられるが，入力として品詞を，出力として構文木を用いる．構文木は，DT や NN などの品詞と [NP や NP] のような括弧付き範疇の列で表現する．例えば，出力 '[S0 [S [NP PRP NP]' は図 2に示す構文木を表す．構文解析では，各単語の品詞入力に従って FST 上を状態遷移し，品詞入力ごとに出力される構文木を順次連接する．これにより，文の途中段階でもそれまでの入力に対する構文木を生成できる．

3 文脈自由文法から有限状態変換器への近似変換

文脈自由文法を有限状態オートマトンに変換する手法については，Kaiser によってすでに提案されている [1]．この手法では，文脈自由文法のグラフ表現である再帰遷移ネットワーク (recursive transition network: RTN) を使用し，その各ネットワークを結合することによりオートマトンを作成する．すなわち，開始記号 $S0$ に対するネットワークを初期状態として，その状態遷移を別のネットワークで置き換える操作を順次繰り返す．一般には，無限回の置き換え操作が行われることになるが，その回数を制限することにより，近似的に変換することができる．図 3に，範疇 NP の状態遷移を範疇 NP のネットワークで置き換える操作の例を示す．最終的に，ε 操作を除去することによりオートマトンを作成する．

本研究の目的である FST を作成するには，上述した置き換え操作を実行するときに対応する構文木を出力に付与すればよい．しかしながら，このような方法には，FST のサイズが膨大になるとともに，そこには，実際の解析処理において使用されない状態遷移が多数含まれるという問題がある．この理由の一つとして，構文木における文法規則の出現の仕方には，ある傾向が存在することが挙げられる．すなわち，構文木の導出過程において，構文木上のあるノードに適用されにくい規則があれば，それに相当する FST 上の状態遷移は，解析処理において使用されにくい．これに対して，状態遷移の置き換えに用いるネットワークに，実データをもとに制限を加えることが考えられるが，一方で，制限が強すぎると FST の汎用性が損なわれることになる．

そこで本手法では，構文木上のノードに対して，どのような規則が使用されやすいかは，そのノードの親ノードに対してどの規則が適用されたかに依存すると考える．すなわち，実データ上で実際に使用された文法規則に相当するネットワークのみを，FST を作成するときの置き換え操作に用いるものとする．例えば，作成途中の FST に，文法規則 $X \rightarrow X_1 X_2 \ldots Y \ldots X_m$ に対応する状態遷移の系列が存在したとする．このとき，範疇 Y を入力とする状態遷移に対して，規則 $Y \rightarrow Y_1 Y_2 \ldots Y_n$ に対応するネットワークで置き換え可能であるとは，図 4に示すような関係がなりたつ構文木が少なくとも一つ実データに存在するときであるとする．これにより，実データを反映し，かつ，サイズが抑えられた FST の作成が可能となる．

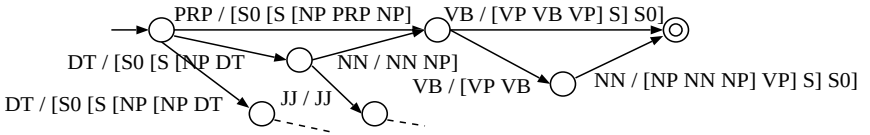

図 1: 有限状態変換器例

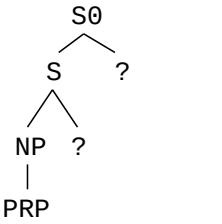

図 2: PRP 入力時の構文木

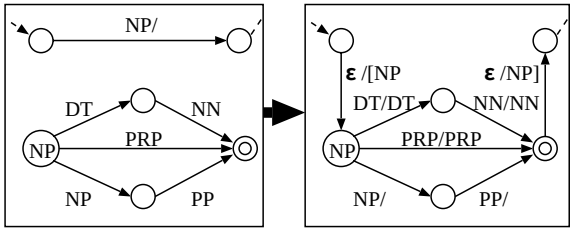

図 3: 置き換え操作

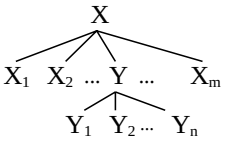

図 4: 構文木の一部

表 1: FST の状態数・状態遷移数

	制限なし	本手法
状態数	3,964,057	50,727
状態遷移数	11,378,287	76,510

4 解析実験

本手法を C 言語を用いて実装し，解析実験を行った．実験では，Penn Treebank の ATIS コーパス 528 文に付与された構文木を用い，構文木から文法規則を抽出した．獲得した 483 個の文法規則，76 個の非終端記号に対して，上述した手法で高さ 7 以下の構文木を生成可能な FST を作成した．比較のため，制限を用いない手法でも同様の FST を作成した．FST の状態数，状態遷移数の比較を表 1 に示す．これらの FST を用いて，ATIS コーパスの FST の作成に使用していない 50 文 (平均文長 8.6 単語) を構文解析した結果，制限を用いない手法では 22 文，本手法では 20 文の解析に成功した．本手法により，解析精度はやや減少するものの，状態数，状態遷移数ともに大きく減少している．

5 おわりに

本稿では，漸進的構文解析のための FST の生成手法と，その ATIS コーパス解析への適用について述べた．今後は，データ量を増やして実験を行い，さらに統計的特性を反映できる手法についても考えていきたい．

参考文献

[1] E.C.Kaiser, "Robust,Finite-State Parsing for Spoken Language Understanding", Proceedings of ACL'99, pp.572-578 (1999).

[2] S.Matsubara, et al., "Chart-based Parsing and Transfer in Incremental Spoken Language Translation", Proceedings of NLPRS'97, pp.521-524 (1997).