自然言語処理 129−14
(1999. 1. 21)

日本語学習支援における診断のための日本語処理系について

馬目 知徳　神田 久幸　掛川 淳一　長澤 直　伊丹 誠　伊藤 紘二

東京理科大学基礎工学部電子応用工学科

近年，コンピュータによる言語学習支援の分野において，自然言語処理技術を応用した様々なシステムが研究されている．我々は，これまで，具体的な場面設定のなかで学習者が行なう作文の診断を目的とした，LTAG(Lexicalized Tree Adojoining Grammar) に基づく誤り診断パーザについて研究してきた．本稿ではまず，LTAG について簡単の述べ，次に日本語 LTAG によるスタックパーザ，そして，誤り診断機構について述べる．

Diagnostic Processing of Japanese in Computer-Assisted Language Learning

TOMONORI MANOME　HISAYUKI KANDA　JUN-ICHI KAKEGAWA
TADASHI NAGASAWA　MAKOTO ITAMI　KOHJI ITOH

Department of Applied Electronics, Science University of Tokyo

In spite of the recent popularity of research on computer-assisted language learning, diagnosing the phrases composed by students remains a difficult task because diagnosis proceeds by parsing and generation interwoven dealing with semantics. In this paper we propose, with prototyping, a diagnostic processing of Japanese using LTAG(Lexicalized Tree Adjoining Grammar) in a shift-reduce stack parsing, recording such errors as detected in each of the reduction phases regarding the semantic heads of the pharses to be reduced.

1　はじめに

近年，コンピュータによる言語学習支援の分野において，自然言語処理技術を応用した様々なシステムが研究されている．

我々は，これまで，具体的な場面設定のなかで学習者が行なう作文の診断を目的とするシステムについて研究を行なってきた．

言語教育の現場では，コミュニカティブアプローチに代表されるように，文法や文型の教育は，それ単体で独立したものではなく，多様な具体的な状況に対応できる柔軟な言語能力を学習者が獲得できるように，場面設定を学習者に与え，そこでの表現の違いの比較を通じて学習するようになっている．

そこで，作文の診断については，誤りの指摘だけではなく，例えば，学習者の入力文が別の解釈をされる危険がある場合の指摘や，その状況において適切か不適切かというレベルでの診断を行うことを最終目標としている．

本稿では，現在開発している LTAG(Lexicalized Tree Adojoining Grammar) に基づく誤り診断パーザについて述べる．まず，LTAG を簡単に紹介し，次に日本語 LTAG によるスタックパーザ，そして，誤り診断機構について述べる．

2　LTAG

LTAG とは，ペンシルバニア大学の XTAG[1][2] で開発されている文法形式であり，文脈自由文法のような記号列を書き換える文法規則ではなく，木構造を書き換える文法規則を持っている．

2.1 木の種類

標準 TAG 形式には initial tree と auxiliary tree (図 1) の二つの型がある.

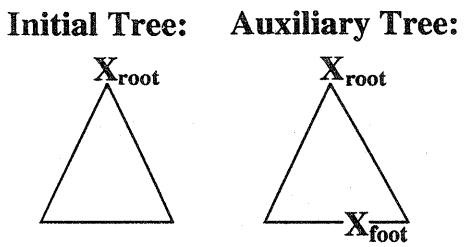

図 1: Elementary tree

init tree(initial tree):
initial tree は再帰を含まない言語学的に最小構造である. これは, root と同じ文法範疇の継ぎ手を持たない木, あるいは持っていても foot としてではない木を指す.

1. 全ての中間ノードは非終端でラベルづけされている.
2. 全ての葉ノードは終端か, もしくは substitution としてマークされた非終端ノードでラベルづけされている.

これは例えば, 単純な文, 名詞句, 前置詞句などの句構造を含んだ木である.

aux tree(auxiliary tree):
auxiliary tree は基本構造に付属物のついた再帰構造を含んでいる. これは, root と同じ文法範疇の継ぎ手を foot として持つ木を指す.

1. 全ての中間ノードは非終端でラベルづけされている.
2. 全ての葉ノードは終端か, もしくは foot ノードを除いた substitution としてマークされた非終端ノードでラベルづけされている.
3. foot ノードは木のその root ノードと同じラベルをもつ.

例えば, 形容詞相当語句, 副詞相当語句などの句構造を含んだ木である.

2.2 木の操作

TAG 形式では, substitution と adjunction の二つの操作が定義されている.

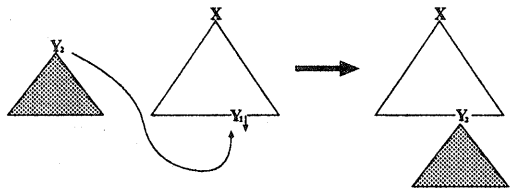

図 2: substitution

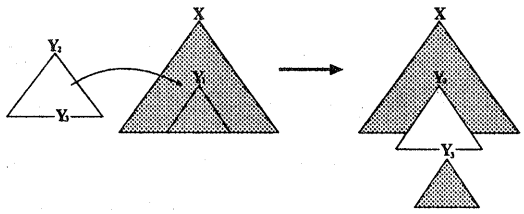

図 3: adjunction

substitution 操作:
substitution は initial tree の root ノードが他の initial tree の substitution するためにマーク (↓で表す) された非終端の葉ノードに併合され新しい tree を生成する操作である. このとき, root ノードと substitution ノードは同じ名前でなければならない. 図 2は二つの initial tree と substitution からできた tree を示している.

adjunction 操作:
adjunction は auxiliary tree を initial tree に, どこへでも非終端ノードに継ぎ木する操作である. auxiliary tree の root ノードと foot ノード (* で表す) は, auxiliary tree が結合したノードとマッチしなければならない. 図 3は auxiliary tree と initial tree そして, adjunction 操作の結果としてできた tree を示している.

2.3 TAG の辞書化 (Lexicalization)

辞書化 (Lexicalization) により, 各々の木の構造は, 辞書項目 (Lexical item) としての役割を果たしている.

木のノードには属性構造 (Feature Structure)

が対応づけられており，例えば，主辞変数の情報や意味制約などが書きこまれる．これらの情報は，adjunctionとsubstitutionの操作の際に，ユニフィケーションによりその操作が行なわれたノードの親ノードへと伝播していく．

図4に日本語の文の簡単な例を示す．ここでは，動詞「登る」の格支配の情報を「登る」の木(つまり辞書)に書きこんでおき，「僕は」と「山に」とを「登る」に係けるときの可能不可能の判定に利用する．また，「僕は」と「山に」の両者の継ぎ手には，それぞれ「僕」と「山」の継ぎ手の属性が，助詞「は」，「に」とsubstitutionしたときにユニフィケーションにより伝播する．

3 解析

LTAGを用いると，辞書項目に置かれた木構造の継ぎ手の単一化のみで文法形式を書き換えることができるため，係りに失敗した際に利用することができる非決定性が辞書項目のみでしか起きない．そのため構造的な組み替えを必要とせず大変扱いやすいものとなっている．

LTAGを利用した日本語の誤り診断機構にはスタック形式のシフトレデュースパーザを用いる．スタック形式のパーザを用いる上で，

- どの時点でシフト・レデュースのどちらを行うべきか？
- レデュース操作ときに，どのような生成規則を適応してスタックの内容のどこまでをレデュースするべきか？

が問題となるが，LTAGでは各操作が辞書項目に既に記述されているためにどちらも一意に決定が可能になる．

解析の手順の説明の前に，SAT(Saturated Auxiliary Tree)とSIT(Saturated Initial Tree)について定義する．SATとは，すべてのfootでない継ぎ手が充足したAuxiliary treeのことである．SATのデータ形式は，

sat(< rootの継ぎ手形式 >, < footの継ぎ手形式 >).

とする．木の継ぎ手だけを記述すれば良い．次に，SITとはroot以外の継ぎ手がすべて埋められた木であり，そのデータ形式は，

sit(< rootの継ぎ手形式 >).

とする．(図5)

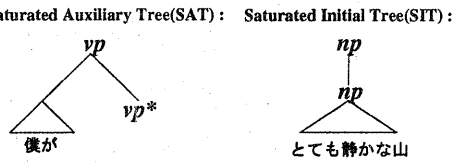

図5: SATとSITの例

これらの二種類の木とTAG形式の二種類の操作を用いて誤り診断機構を実現する．解析は以下の手順で行なう．

1. 表層の先頭の一語を辞書引きする．
2. 辞書引きした語がスタックの先頭の内容とadjunction操作ができるときはレデュースし，adjunctionする．adjunctionできなくなるまで，レデュースを繰り返す．
3. 2の結果がSATならばスタックにシフトする．
4. 2の結果がSITならば表層の次の語を先読みし，(1)文末であれば終了．(2)substitutionによりSATを作ることができればスタックにシフトする．(3)さもなければ活用形に対応する連体/連用空辞λを接続したSATの生成を試みる．
5. 1に進む．文末であれば終了．

解析の流れの例を図7に示す．

4 誤り診断機構

4.1 問題設定

誤り診断パーザは，具体的な場面設定での前後関係が与えられた穴埋め作文における誤り診断を行なう．

解析には以下の2つを制約として用いる．

- 正解の意味表現(具体的に正解が表層の文として学習者に提示されるわけではないことに注意)

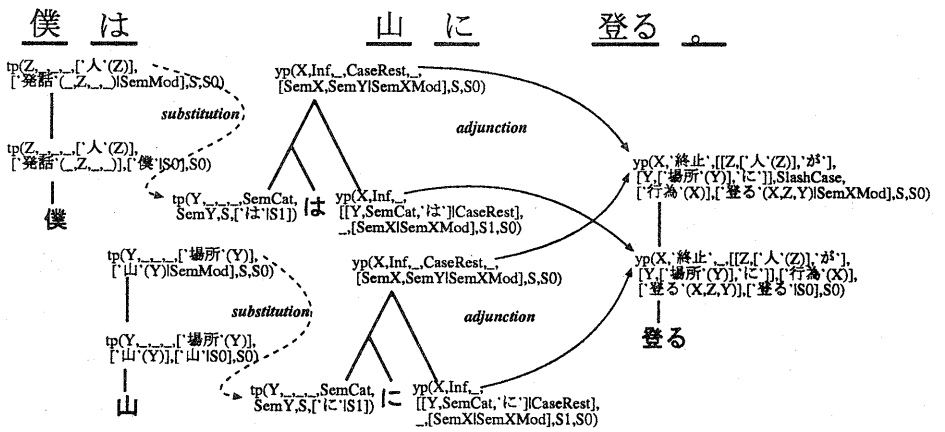

図 4: 辞書化された木

- 意味表現の各要素に対応する語彙の候補 (学習者が選択可能な語彙の種類)

学習者は，与えられた正解の意味表現とそれに対応する利用可能な語のリストを用いて作文を行ない，パーザは，その入力文における誤りを診断する．

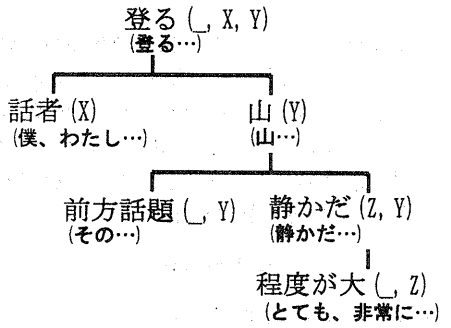

図 6: 正解の意味表現と対応する語彙の例

図 6に正解の意味表現と対応する語彙の例を示す．ここで，正解の意味表現は，多分木の構造となっており，各ノードの語に係る語がその子ノードに並ぶという形式になっている．

4.2 誤りの種類

図 6の正解の意味表現に対する作文として，例えば，以下の 3 文を考えると，1,2 が正解，3 が不適格となる．

1. 僕がそのとても静かな山に登る．
2. わたしがとても静かなその山に登る．
3. * その僕に静かだとても山を登る．

パーザが診断する誤りの種類としては以下のものを扱う．

語の不足 正解の意味表現が要求する意味に相当する語がない場合

余分な語 正解の意味表現にはない語がある場合

係りの誤り 交差係り，正解の意味表現と異なる係り関係

接続辞の誤り 助詞の間違い等

活用の誤り 動詞や形容詞の活用の間違い等

4.3 誤り診断の手順

学習者による入力文における誤りの診断は以下の手順で行なう．

1. 表層の先頭の一語を辞書引きする．
2. 辞書引きした語から，その語の正解の意味表現におけるノードを決定する．
3. 決定したノードの子ノードが，その語に係るべき主辞変数の意味表現のリストとなる．(以後 **要求リスト**と呼ぶ)

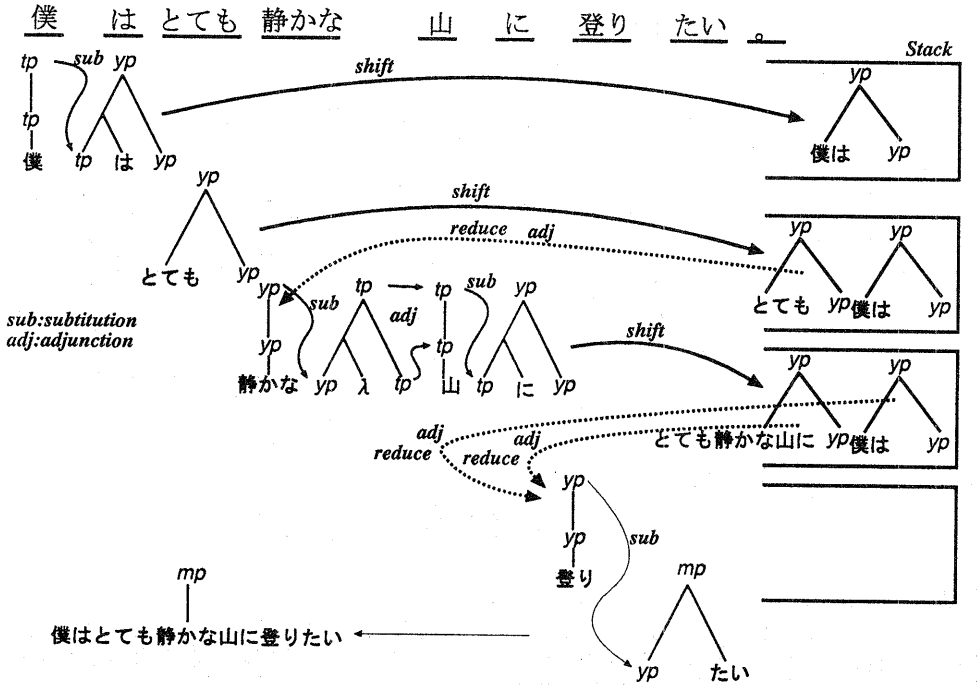

図 7: 日本語 LTAG を利用した構文解析の例

4. スタックの要素を調べて

   - adjunction 可能だが，要求リストにはない要素は削除し，余分な要素があったと記録する
   - 要求リストに適合する要素の手前に adjunction 不能な要素がある場合，交差係りであるので削除し，係りの誤りがあったと記録
   - 要求リストに適合する要素について，活用の誤り，助詞の誤りがある場合については修正し，その変更を記録する．
   - 要求リストの要素の内，対応するスタック要素が見出されないものについては語の不足として記録する．記録する．

5. 4 の結果が SAT ならばスタックにシフトする．

6. 4 の結果が SIT ならば表層の次の語を先読みし，(1) 文末であれば終了．(2)substitution により SAT を作ることができればスタックにシフトする．(3) さもなければ活用形に対応する連体 / 連用空辞 λ を接続した SAT の生成を試みる．

7. 1 に進む．文末であれば終了．

図 8に，誤り診断の例を示す．

5 おわりに

以上，本稿では，LTAG の概略を説明し，日本語 LTAG に基づくスタックパーザによる構文解析について述べ，それから，診断機構を導入したパーザの機構について述べた．

5.1 今後の課題

辞書の充実 現在のパーザで用いている辞書は，試作のための専用に人手で作成したものであり語彙数は限られる．入手可能な電子化された辞書を変換して利用することを検討している．

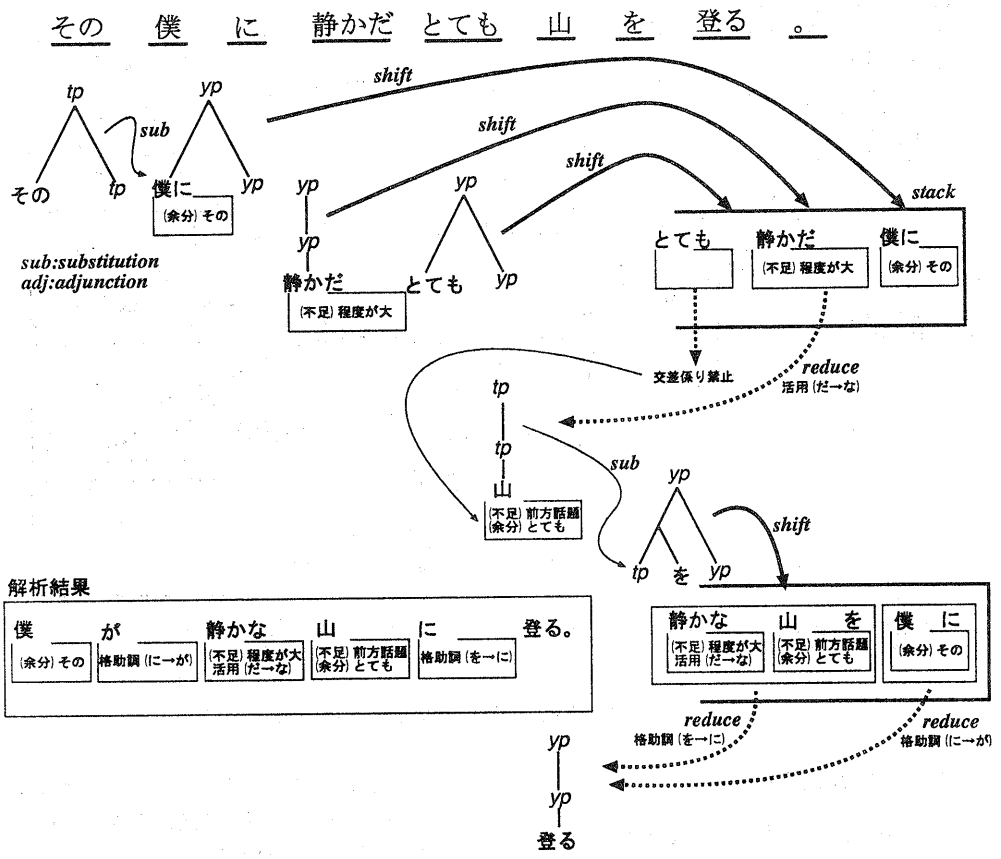

図 8: 誤り診断の例

コメント生成 パーザの出力結果から，学習者にとって意味のあるコメントを生成し提示する機構の実現について検討している．

参考文献

[1] The XTAG Research Group: "A Lexicalized Tree Adjoining Grammar for English", University of Pennsylvania, IRCS Report 95-03(1995).

[2] Chrity Doran,Dania Egedi: "XTAG System - A Wide Coverage Grammer for English", In Proceedings of the 15th International Conference on Computation Linguistics(COLING '94) pp.922-928(1994).

[3] 加藤伸隆 神田久幸 馬目知徳 伊丹誠 伊藤紘二: "日本語学習支援のための LTAG による文の生成と診断について", 言語処理学会第 4 回年次大会発表論文集,pp.658-661(1998).

[4] Owen Rambow and Aravind K. Joshi: "A Processing Model for Free Word Order Languages", In *Perspectives on Sentence Processing*, C.Clifton, Jr.,L.Frazier and K.Rayner, editors, Lawrence Erlbaum Associates(1994).

[5] 加藤芳秀 松原茂樹 外山勝彦 稲垣康善: "文法的不適格文に対する統語的制約を用いた漸進的解析手法", 言語処理学会第 4 回年次大会発表論文集,pp.290-293(1998).