日本語学習支援のための診断処理について

神田 久幸　馬目 知徳　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 電子応用工学専攻
{kanda,manome,kakegawa,naga,itami,itoh}@itlb.te.noda.sut.ac.jp

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。

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。

そこで我々は、日本語学習支援システムにおいて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LTAGを使った誤り診断パーザを用いて試作している。

2 LTAG(Lexicalized Tree Adjoining Grammar)

LTAGとはペンシルバニア大学のXTAGリサーチグループによって研究報告[1]されている文法形式である。このTAG(Tree Adjoining Grammar)とは、文脈自由文法(CFG:Context-Free Grammer)の記号列を書き換える文法規則ではなく、さらに拡張し、木構造を書き換える文法規則をもっている。

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。

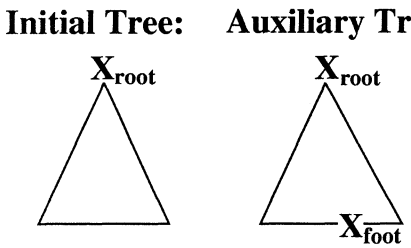

図 1: Elementary tree

initial tree:
initial treeは再帰を含まない言語学的な最小構造である。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。

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。

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。

substitution 操作:
substitutionはinitial treeのrootノードが他の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(図2)。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。

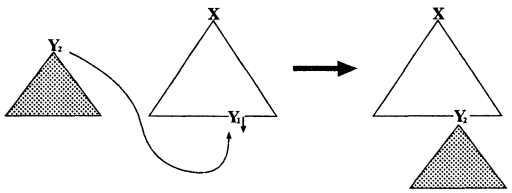

図 2: substitution

adjunction 操作:
adjunctionはauxiliary treeを他の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)。

2.3 TAGの辞書化 (Lexicalization)

TAG形式をLexicalizationすることで各々の木の構造に、辞書項目(Lexical item)を関連付けることがで

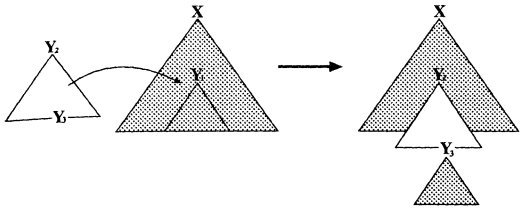

図 3: adjunction

きる。木のノードには素性構造 (Feature Structure) が対応づけられており、例えば、活用形、格情報、主辞変数の情報、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。

3 LTAG を用いた診断機構

LTAG を利用した日本語の誤り診断にはスタック形式のシフトレデュースパーザを用いている。LTAG には以下のような特徴があり、この特徴を利用して誤り診断を行う。

- 手続きが単純である。LTAG では文法項目があらかじめ辞書の中に記述されているので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。そのため、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ので、構造的な組み替えを必要とせず扱いやすい。
- LTAG の生成機構を利用している。LTAG は辞書項目のユニフィケーションだけで文の生成が可能である。誤り訂正においては、ローカルな部分 (局所的) で文の一部を生成し、学習者の入力した文と表層を比較して、診断を行う。

3.1 SAS と SIT

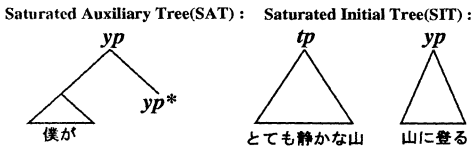

図 4: SAT と SIT の例

スタック形式のパーザを用いる上で、新たに 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。

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 4)

どちらも木の継ぎ手だけを記述すれば良い。

3.2 空辞 λ

空辞 λ は表層には表れない文法機能だけをもつ木である。空辞は体言に接続する連体空辞、用言に接続する連用空辞などを用意しておく。日本語の辞書項目の中には、initial tree と auxiliary tree の両方の性質をもつ語がある。例えば、動詞や形容詞などの連用形や連体形である。それらの語は initial tree と空辞という形で表現することで、木の構造を統一的に表現することができる。

また、動詞や形容詞などの連体形と終止形のように、同じ表層でありながら異なる文法機能をもつ語があるときに、その機能を限定せずに、空辞を使って SAT を作り、スタックに積むようにする。そして、入力文の語を先読みして、空辞を含んだ SAT をスタックから取り出すときに、その語がどのような機能をもつかを決定し、空辞の種類を判断する。

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)
- 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)

図 5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。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。

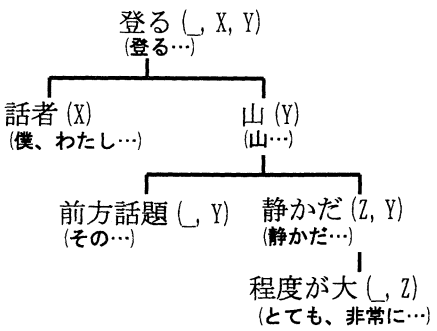

図 5: 正解の意味表現と対応する語彙の例

4.2 生成の利用

誤り診断における生成は、

1. 一組の係り関係だけからなる文(の一部)を生成。
2. 学習者の入力と表層のレベルで比較。

という形で用いる。生成された文と、学習者の入力した文の表層が異なる場合には、誤りがあったと判断し、訂正する。表層の比較を行うことで、動詞などの活用形や格助詞などの訂正を行う。

これは、係り関係は順序によらず独立であり、順序の違いによって意味は変らないとしているからである。語順の変化によってニュアンスが変化することはあるが、その取り扱いは、本研究の診断機能の範囲の外とする。

4.3 誤り診断の手順

診断は以下の手順に従って診断をすすめる。main スタックと tmp スタックの二つのスタックを用意する。

1. 表層リストが空ならば終了、さもなければ表層の先頭の一語を辞書引きし、得られる木を t とする。
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。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** $d-list$ と呼ぶ)
3. main スタックが空でなければ、pop して得られる木を e とする。
   - 要求リスト $d-list$ が空のとき、または e が要求リスト $d-list$ の要素ではないとき。main スタック内の語が adjunction 可能であれば、余分な係りとして記録する。tmp スタックに push し、3 に戻る。
   - e の主辞の意味表現が要求リスト $d-list$ の要素のとき、t に対して adjunction を行う。そのとき、表層から取りだした語と main スタックから取り出した語から生成を行い、表層のレベルで比較し、必要があれば修正をする。adjunction できないときは生成された文に基づいて修正を行う。修正した情報は記録しておく。そのとき、tmp スタックにあった要素は adjunction の際の障害になった(交差係りになる可能性あり)ことを記録する。また、e の意味表現を $d-list$ から取り除いた残りを改めて $d-list$ とする。こうして得られた木を改めて t とする。3 に戻る。
4. main スタックが空であるとき
   - 要求リスト $d-list$ が空でなければ、その中の要素に対する語が不足していることを記録する。
5. tmp スタックの全要素を main スタックに戻す。
6. t が SAT ならば main スタックに push する。
7. t 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 main スタックに push する。(3) さもなければ活用形に対応する (例えば、連体 / 連用) 空辞 λ を接続した SAT の生成し、main スタックに push する。
8. 1 に戻る。

誤り診断パーザは、図 6 ように正解の意味表現に適するよう、局所的に文を生成し、誤りに対して訂正を繰り返す。途中で訂正した箇所は記録しておき、学習者にコメントをする。つまり、このパーザは局所的に文を生成することで、局所的に見て検出できる限りでの誤りを検出し、訂正する機能をもつ。

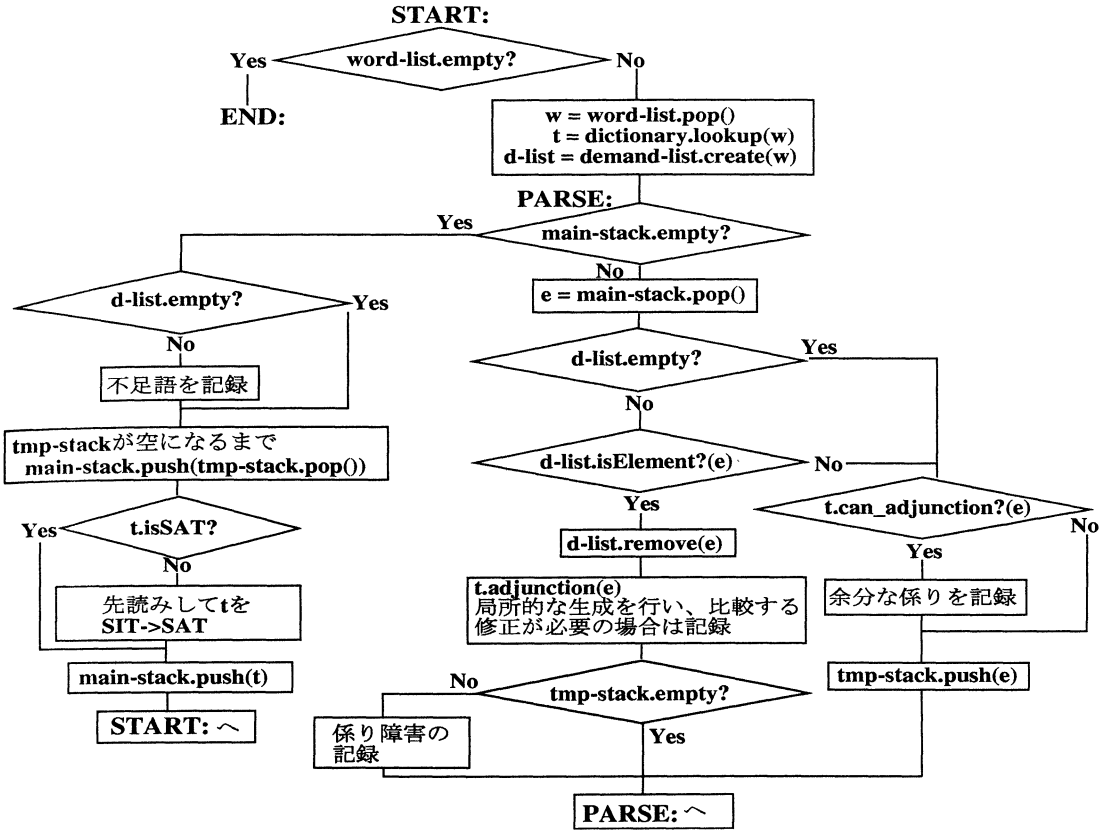

図 6: 誤り診断パーザのアルゴリズム

5 本システムの位置付けと今後の課題

- グローバルな診断
  このシステムではローカルな誤りを指摘するが、学習者の作文がこのシステムによる誤りの指摘を受けなくなった段階でその文はこのシステムから見てグローバルに正しい文とみなされる。それはシステムの生成機構が生成し、パーザの検証を通るような文の一つであることは保証されるが、人間が受け取って正しいと判断される文であることは保証されないので、この段階以降は学習者と教師のやりとりにゆだねられる。
- コメントの生成
  現在は誤り診断の結果記録しているだけだが、学習者に診断内容を提示する方法については工夫する必要がある。
- 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、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。

参考文献

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

[2] 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).

[3] 馬目知徳 神田久幸 掛川淳一 長澤直 伊丹誠 伊藤紘二: "日本語学習支援における診断のために日本語処理系について", 情報処理研究会報告 99-NL-129,pp.95-100(1999).