

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

加藤 伸隆 馬目 知徳 神田 久幸 伊丹 誠 伊藤 紘二

東京理科大学 基礎工学部

1 はじめに

日本語学習支援システムにおいて、使うべき自立語のリストを与えた上で対話例文の中につくられた穴を埋める作文を行わせ、学習者の入力文を診断する機能を我々は構築している。今まで構築してきたパーザ[1][2]はDCGを用いたボトムアップなスタック形式のパーザで、文法範疇の定義や状況意味的制約を自立語と機能語の辞書に記述しつくし、穴埋部分の状況機能的意味制約と共に、意味的、機能的に正しい文であるかという診断に対応してきた。この方法では、不適格文を診断するには構文制約と意味制約の構造的組み替えを必要とし柔軟性に欠ける。その点で、LTAG(Lexicalized Tree Adjoining Grammar)[3][4]は構造的な組み替えを必要とせず、継ぎ手の同一化のみであるために、継ぎ手に失敗した際のバックトラックが辞書項目のみでしか起きないことから、大変扱いやすいものとなっている。本稿では、LTAGを用いて日本語文の生成と誤り診断機構に対して、我々の研究の現状を報告する。

2 LTAGの形式

LTAGには以下の二種類の木(辞書項目)と二種類の操作でもって木を作りだすことができる。

2.1 木の種類

init tree(initial tree)

rootと同じ文法範疇の継ぎ手を持たない木、あるいは持っていないがfootとしてではない木

aux tree(auxiliary tree)

rootと同じ文法範疇の継ぎ手をfootとして持つ木

2.2 操作の種類

TAG形式では、substitution操作とadjunction操作の二つが定義されている。

substitution 操作

initial treeのrootノードが他のinitial treeにsubstitutionするためにマーク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。rootノードとsubstitutionノードは同じ名前でないといけない。

adjunction 操作

auxiliary treeはinitial treeにおいて、どこへでも非終端ノードに継ぎ木する。auxiliary treeのrootノードとfootノードは、auxiliary treeが結合したノードとマッチしなければならない。

3 LTAGを利用した日本語文の生成

我々のシステムは、学習者に、学習項目に応じた穴埋め問題を持った対話文を提供し、それを学習者がシステムにあらかじめ用意された単語メニューから入力してもらうことになっており、それらの正解に対応するために、その単語メニューからでき得る正解のバリエーションを作り出す仕組みを考える必要がでてきた。これによりシステムがあらかじめ用意した単語メニューから生成し得る正解文を教育現場の教師が知っておくことができ、またこれを診断にも利用することができる。以下に日本語LTAGを利用した文の生成機構のインプリメントの詳細を説明する。

3.1 生成機構

init treeとaux treeの二種類の辞書には、継ぎ手のリストも、terminationのリストも、rootに対するものを含めて作る。また対応する継ぎ手にterminationがないところにはnilを置く。いずれの種類の木であっても、nあるいはvあるいはaの辞書終端に終るnpあるいはvpあるいはapの枝は、トップから継ぎ手を出しておき、一般には、そこにaux treeをいくつかはさんで、枝自身は、terminationとして接続されて解決するが、はさむ必要がある(意味表現から見て未解決の)時には、その枝が解決するまで、aux treeをはさみ続ける。はさむ必要がない(意味表現から見て解決済みの)ときは、そこでterminationが接続されるので、その後は何の処理も必要がない。init treeでは、root継ぎ手がリストの先頭におかれる他、それ以外の継ぎ手は任意の順序で置かれて良い。また、aux treeでは、root継ぎ手をリストの先頭におき、それ以外

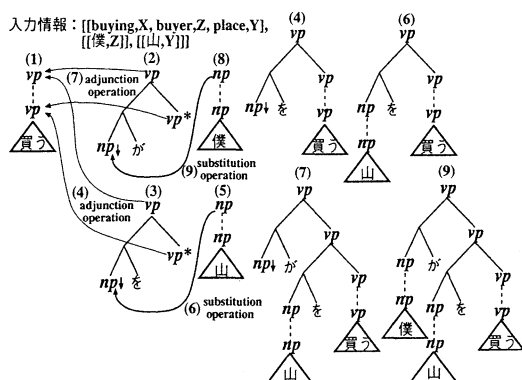


図 1: 日本語 LTAG を利用した文の生成機構の例

の継ぎ手については、表層の順序に関係なく(表層の差分リストが表層の語順を表しているので)foot 継ぎ手が先頭に、他の継ぎ手はその後ろにおかれる。

3.2 文法範疇の素性

各文法範疇には次のような素性を導入している。
 <head 変数>:X,
 <格支配などの文法的制約情報>:活用情報や必須格情報, 任意格情報,
 <意味表現>:スロット付き属性情報,
 <head の意味素性>:head の属性情報,
 <表層差分リスト>:表層表現情報

辞書中において、活用のない自立語や機能語といった文法範疇に対しては、<格支配などの文法的制約情報>は省略している。また、意味表現のデータ形式において、head に当たる意味表現は、それを保存しながら木に沿って降りてゆくためのリスト表現の都合から、それを最初の要素とするが、修飾句の意味表現は、処理の順序に従って意味的に遠い方から並べる。ただし、ここに遠い方からというのは、日本語の2分木構造で、上の方に来るもの(head から表層的に遠い)ほど意味的に head との関係が遠いとみなしてのことである。

3.3 文生成の例

図1の文の生成過程を以下のように追ってみる。

1. 始め入力情報として、意味表現としての [[buying,X, buyer,Z, place,Y], [[僕,Z], [[山,Y]]] を与える。

2. 次に1. で得られた意味表現の頭部 [buying,X, buyer,Z, place,Y] と同一化する init tree を辞書引きする。このときに対応するのが、init tree(買う)である。これを(1)とする。
3. 次に init tree(買う)の状況意味の意味表現としての<格支配などの文法的制約情報>に必須格情報が入っており、この場合は「が」格と「を」格であり、それに対応する(2)の aux tree(が)と(3)の aux tree (を)を辞書引きする。
4. 次に意味表現をみながら、aux tree(を)の foot の vp が空いていることから他の tree の中間ノードにおいて、文法範疇が同じものがあるなら、adjunction 操作され、(4)となる。但し「を」格に入るべき、意味素性が init tree(買う)に記述されているために、place(場所)という意味素性をみて、これに同一化する termination と同じものを root にもつ tree を探すと、この場合(5)の init tree(山)であり、これが substitution 操作され、(6)となる。
5. さらに、init tree(買う)の必須格情報の残りの「が」格の aux tree(が)が init tree(買う)へ adjunction 操作され、(7)を作る。(7)の tree の termination の np が空いているのをみて、4. と同様の操作で、この格に入るのが person(人)というのを init tree (買う)の意味素性に記述してあるのをみて、これに同一化する termination と同じものを root にもつ tree を探すと、この場合(8)の init tree(僕)であり、これが substitution 操作され、(9)となる。
6. 最後に root 以外の継ぎ手が全てなくなって終了し、出力結果「僕が山を買う」が生成される。

4 日本語 LTAG を利用した診断

日本語表現を解析し、診断するためには、日本語の特徴を知っておく必要がある。その幾つの特徴としては、(1)文頭から述語に至るまでの語順の自由度が高いけれども、それ自体話者の態度表明であること、(2)日本語の係り関係には自立語へ係る句がその自立語の直前に並ぶという極めて単純な性質があること、(3)その際の格関係が動詞に依存していること、(4)モダリティのように文の対話中の機能を表示するものが、主な述語よりも後部にくること、(5)副詞のモダリティ表現に対する呼応要素が存在すること、(6)て

いねい形やテンスについては述語とひとまとまりで扱う方がわかりやすいこと、などが挙げられ、それを実現するためにLATGを利用した診断機構の有効性を見るための考察をした。

4.1 木の種類と誤り診断機構

Saturated Auxiliary Tree(SAT): Saturated Initial Tree(SIT):

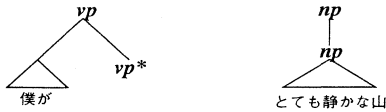


図 2: SAT と SIT の例

日本語の誤り診断機構にはスタック形式のパーズを用差し, Saturated Auxiliary Tree Stack(以下 SATS と略す)を用意する. termination を修飾する Saturated Auxiliary Tree(以下 SAT と略す)を蓄えるものとする. ここに Saturated Auxiliary Tree とは, すべての foot でない継ぎ手が充足した aux tree のことである. 基本的には, 文の生成機構で用いた init tree と aux tree を診断でも同様の木を使う. また Saturated Initial Tree(以下 SIT と略す)は root 以外の継ぎ手がすべて埋められた木である. これらの二種類の木(図 2)を TAG 形式の二種類の操作法を用いて誤り診断機構を構築する.

4.2 文解析の例

次に実際に積み上げていく例を示す(図 3). 入力文を「僕がとても静かな山を買う。」とする.

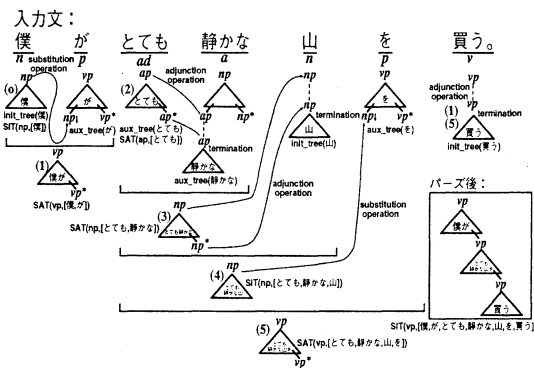


図 3: 日本語 LTAG を利用した構文解析の例

1. まずはじめは, 表層表現の先頭の形態素「僕」の init tree(僕) が辞書引きされる (0).
2. 次に「が」がきて, 辞書より aux tree(が) を引く. それがもつ termination の np に, (0) の init tree(僕) が substitution 操作によって, (1) の SAT(vp,[僕,が]) になり, SATS に積まれる. このとき, foot の vp が空いている.
3. 次に「とても」がきて, 辞書より aux tree(とても) を引く. foot の ap が空いた SAT(ap,[とても]) となり, SATS に積んで, (2) として待つ. ここまでの間, SATS には, SAT(ap,[とても]), SAT(vp,[僕,が]) が積まれている.
4. 次に「静かな」がきて, 辞書より aux tree(静かな) を引く. その中間ノードに ap が存在するので, (2) として待っていた aux tree(とても) が, aux tree(静かな) の中間ノード ap に adjunction 操作によって, SAT(np,[とても, 静かな]) を作り, (3) となる. まだ, foot の np が空いている.
5. 次に「山」がきて, 辞書より init tree(山) を引く. SATS に積まれている (3) の foot の np に adjunction 操作により, root 以外の terminal はすべて埋められた木 SIT(np,[とても, 静かな, 山]) となり, これを (4) とする.
6. 次に「を」がきて, 辞書より aux tree(を) を引く. 空いている termination の np に, np を root にもつ SIT(np,[とても, 静かな, 山]) を substitution 操作により, SAT(vp,[とても, 静かな, 山, を]) を作り, SATS に積む. このときの tree を (5) とする.
7. 現在, SATS にスタックされているのは, (5) の SAT(vp,[とても, 静かな, 山, を]), (1) の SAT(vp,[僕,が]) の二つである.
8. 次に「買う」がきて, 辞書より init tree(買う) を引く. この用言に書かれた<格支配などの文法的制約情報>中に書かれた必須格情報をもとにいままで SATS に積まれていたものを下ろす. 「買う」の必須格が, 「が」格と「を」格であることから, 状態は (1) の SAT(vp,[僕,が]) の foot の vp に init tree(買う) が adjunction 操作され, SIT(vp,[僕,が, 買う]) となる. さらに, (5) の SAT(vp,[とても, 静かな, 山, を]) の foot の vp に, SIT(vp,[僕,が, 買う]) の root の vp が adjunction 操作され, SIT(vp,[僕,が, とても, 静かな, 山, を, 買う])

9. 次は「。」なので、SATSに積まれているものがないことと表層表現のリストが空であることを確認したのち解析を終了する。

こうしてできたパーズ後の解析木は図3の右端に示した。この文は適格文であることが知られたが、この解析の視察により、誤りはSATをスタックからおろしてSITを作るところ、あるいはSITに対する接続要素を解析するときに診断されることが知られる。誤りとしては、係り関係の誤り・欠落・過剰や、接続要素の誤り・欠落・過剰や、SITの接続要素への接続形(活用)の誤りが検出される。また適格文でありながら、その場の状況や機能が役割を果たしていないことなどがあつた場合には、それに対して指摘のできる仕組みが必要である。当然、教育的に診断していくためには、言語形式的に間違っている、コミュニケーションが取れる文である場合にも解析ができる必要がある。

4.3 モダリティの扱い

4.3.1 伝達態度や真偽判断のモダリティの扱い

伝達態度のモダリティは表現タイプの違いにかかわらずなく、聞き手に対する話し手の文伝達の態度を示す働きをする。具体的には終助詞「よ」、「ね」である。また真偽判断のモダリティは、対象なる事柄の真偽に関する判断を表すもので、具体的には「だろう」、「ようだ」である。これらの扱いはip(intentional phrase)としてのaux tree(ip)を用意しておき、どんな文でもipで終わるようにし、必ず文解析後の一番後ろにあるSAT(vp)のあとに空のaux tree(ip)を付加し、aux tree(ip)がsubstitution操作されることを可能にする。図4では、伝達態度のモダリティはtreeの上層部のaux tree(よ)として、substitution操作によりtreeが作られる。また真偽判断のモダリティはaux tree(だろう)として、substitution操作によりtreeが作られる。

4.3.2 モダリティと依存関係構造

副詞の呼応関係が、モダリティと依存関係構造にあたる。その現れ方は[5]で指摘しているが、依存関係の主要素の地位を占めるものと、主要素と呼応する性格のもの、の二種類を区別することができる。例えば、「決して手紙を書かない」(図4)では、認め方のモダリティ「ない」が依存関係の主要素で、「決して」が「ない」と呼応する要素になっている。この「ない」扱いに関してはテンスやアスペクトでいい表現などのモダリティと同様に動詞やモダリティ助動詞の活

用の一部として扱い、動詞の辞書に書き込んでおくことにする。よって「書かない」という動詞辞書に認め方のモダリティを含めて扱う。このことは教育の現場で、そのように表現学習をさせていることにならつてのことである。また、モダリティにおいて、呼応要素を持たない、あるいは必要ない場合もあるので、依存関係の主要素と呼応要素の間になんらかの制約を書きしておく必要がある。

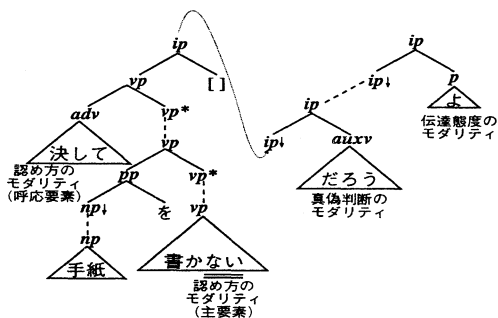


図4: モダリティを考慮した tree の例

5 結び

今回は日本語 LTAG を利用した文生成機構の試作と診断機構の考察をし、またモダリティなどの扱いについて報告した。今後特にモダリティをどう扱っていくかを研究する必要がある。例えば、モダリティのオーダリングの問題などは言語学的にはどのように組合さっているかといったことを考察する必要がある。

謝辞

本研究において、東京大学の辻井潤一氏より、LTAG に関する論文を紹介して頂いたことに感謝の意を表する。

参考文献

- [1] 劉鉄, 加藤伸隆, 馬目知徳, 伊丹誠, 伊藤紘二 (1996): “日本語学習支援システムにおける誤り文の解析,” 日本教育工学会第12回全国大会, pp.389-390(1996.11).
- [2] Nobutaka Kato, Liu Yi, Tomonori Manome, Hisayuki Kanda, Makoto Itami, Kohji Itoh(1997): “Use of Situation-Functional Indices for Diagnosis and Dialogue Database Retrieval in a Learning Environment for Japanese as Second Language”, Artificial Intelligence in Education, Knowledge And Media in Learning Systems, p.247-254,(1997-08), IOS Press, Inc. & Ohmsha, Ltd.
- [3] The XTAG Research Group(1995): “A Lexicalized Tree Adjoining Grammar for English”, Site of the NSF Science and Technology Center for Research in Cognitive Science, The Institute For Research In Cognitive Science, IRCS Report 95-03, March 1995.
- [4] Owen Rambow and Aravind K. Joshi(1994): “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.
- [5] 益岡隆志 (1991): “モダリティの文法”, くろしお出版.