一般化LR構文解析法による文中の複数箇所の誤りの検出と修正

今井 宏樹* THEERAMUNKONG Thanaruk 奥村 学
北陸先端科学技術大学院大学 情報科学研究科

田中 穂積
東京工業大学 情報理工学研究科

1 はじめに

従来の自然言語処理システムでは，システムが用意した文法に適合する文のみを解析するものが多い．そのため，文法に適合しない入力は扱うことができず，柔軟性に欠けるものであった．また，文法を拡張しても，それに適合しない自然言語の文は存在し，完全な解決には至らない．したがって，実用性の高いシステムを作成するためには，非文をも解析できる能力が必要である．

これまで，chart法をベースとした非文解析の研究はいくつか行われているが([4], [3], [6])，一般化LR法(GLR法)をベースとした研究は多くない．入力が音素や文法のレベルで不適格であることが多い音声認識の分野では解析にGLR法がよく用いられており，GLR法の非文解析手法は有用であると考えられる．また，発話文を対象とする対話システムにおいても，この解析手法が利用できると考えられる．

GLR法を基に過去に行われた研究としては，Saitoらの行ったノイズのある音素列を解析する研究がある[5]．しかしながら，各入力単位ごとに誤りの可能性を考慮する方法をとっているため，正しい入力に対しても無駄な解析をする問題があった．我々は，この問題を回避しかつ文中の1箇所の誤り検出・修正を行う手法を提案した[7]．本稿では，この手法を拡張し，文中に複数箇所の誤りが出現する場合にも対応できるアルゴリズムの提案を行う．

また，誤り修正処理を行うと，文法自身が持つ曖昧性の他に，修正された誤りの種類による曖昧性も加わることになる．そのため，得られる構文木の数が非常に膨大になってしまい，実際の自然言語処理システムに組み込むことが難しくなる．そこで，本稿では，修正した誤りの形を反映するような解析候補の絞り込みの手法を合わせて提案する．

*E-mail: h-imai@jaist.ac.jp

2 誤りの定義

自然言語処理の分野においてはさまざまな種類の誤りが存在するが，本研究では，「システムに与えられた文法に適合しない文」を非文として扱うことにする．また，検出された誤りに対して施す処理として，文法を修正する，単語のカテゴリを修正する，という2つのアプローチがあるが，本研究では後者の処理を行う．

次に，本研究で扱う誤りの定義を示す．ここでは文献[4], [3], [6]で用いられている，終端記号レベルの誤りの定義を基本とする．また，それ以外にも，対話文などで見られる句構造(非終端記号)レベルの挿入や脱落についても考慮することとした．これらをまとめると，定義は以下のようになる．

終端記号の置換誤り
文中のある語が違う語に置換されている．

終端記号の挿入誤り
文中に余分な語が含まれている．

終端記号の脱落誤り
文中に必要な語が抜けている．

未知語
置換が起こっている語がシステムにとって未知の語である場合．

非終端記号の挿入誤り
文中に余分な句構造が含まれている．

非終端記号の脱落誤り

文中に必要な句構造が抜けている.

3 文中の複数箇所の誤り検出・修正アルゴリズム

本手法の概要は，通常の GLR の解析において途中で解析が失敗した場合に，1 箇所の誤り修正処理を行い，修正が成功したら再び解析を続け，その後解析が失敗するごとに 1 箇所の誤り修正処理を起動し，文末に到達するまで繰り返し行う，というものである (図 1).

図 1 中の 1 箇所の誤り検出・推定処理は，文献 [7] で提案され，誤り位置の推定処理，および推定された位置における修正処理の大きく 2 つの処理からなる.

誤り位置の推定では，エラー発生までに部分木を構成している語は誤りの可能性が低い，というヒューリスティックを導入し，

1. 誤りの起こった先読み記号
2. まだ部分木を構成していない語
3. 隣接する部分木の間
4. 部分木を構成している語

の順に，誤りと思われる語 (位置) を候補として推定する[1]. 本手法では，1 回の処理での推定範囲を，前回の誤り修正処理で復帰した位置から現在誤りの起こった位置としている.

誤り修正の処理では，推定された位置に対して，2 節で示した誤りの種類についてそれぞれの可能性を考慮し，以下の処理を行う[2].

終端記号の置換誤り

LR 表を参照して，誤りと推測された記号の代わりに動作可能な記号で解析を続ける.

終端記号の挿入誤り

誤りと推測された記号をスキップし，その次の記号から解析を続ける.

終端記号の脱落誤り

LR 表を参照して，誤りと推測された記号の直前に動作可能な記号を補う.

非終端記号の挿入誤り

誤りと推測された記号の直前にある非終端記号を除去し，再び推測された記号での解析を試みる.

非終端記号の脱落誤り

LR 表を参照して，誤りと推測された記号の直前に可能な非終端記号を挿入し，推測された記号での解析を試みる.

このアルゴリズムには，以下のような特徴がある.

- 正しい文に対しては通常の解析を行い，無駄な処理はしない.
- エラー発生までに解析した情報を利用可能.
- 出力結果は修正回数が最も少ないものだけが得られる.
- 1 回の誤り検出・修正の範囲内に 1 箇所の誤りの出現を仮定している.

4 解析候補の絞り込み

非文解析によって得られる構文木の数は，構文的曖昧性だけでなく誤りの種類による曖昧性も含むため，非常に多くなる. したがって，意味解析などの他のレベルの解析に構文解析の結果を利用する際に，全ての解を渡すのではなく，優先順位をつけ尤もらしい解に絞り込んでおくことが重要となる.

順位付けは，誤りの出現位置・誤りの種類によりそれぞれスコアを加算し，その和が最も小さいものから順に選び出す，という方針で行う.

誤りの出現位置によるスコア

誤りの出現位置によるスコア S_P は，修正された各誤りに対して，修正された位置が誤り位置推定処理において候補とされた順位を足し合わせることで計算する.

誤りの種類によるスコア

誤りの種類によるスコア S_T は，誤りの種類と修正した記号の種類にそれぞれ基本値を設け，それらをかけあわせることで計算する.

- 誤りの種類の基本値: s_{type}
  置換 (s_{subst})・挿入 (s_{extra})・脱落 (s_{omit}) のいずれか. これらの値はあらかじめ与えておく.

[1] 各項目において複数候補がある場合は，誤り発生位置から近い順に候補を返す.

[2] 未知語に対する処理は，終端記号の置換・挿入誤りの処理に含まれる.

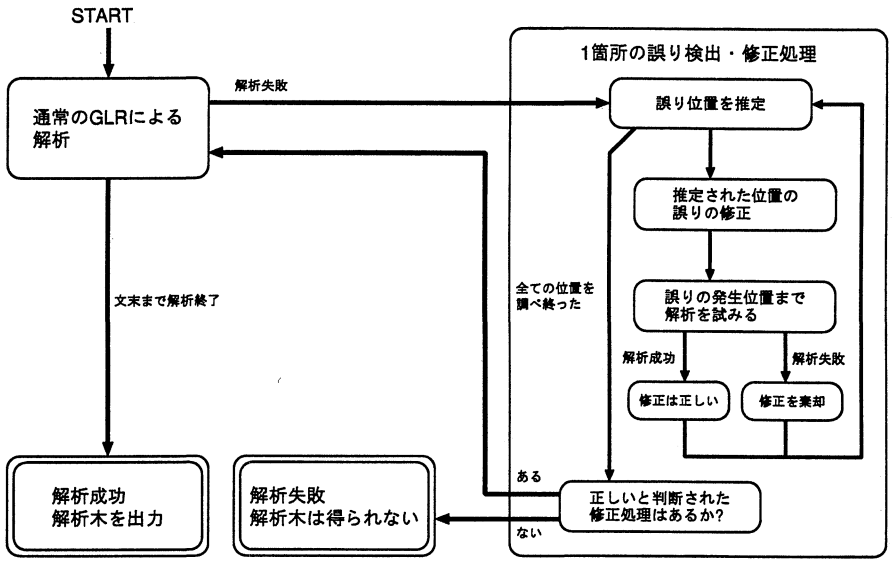

図 1: 誤り検出・修正処理の流れ

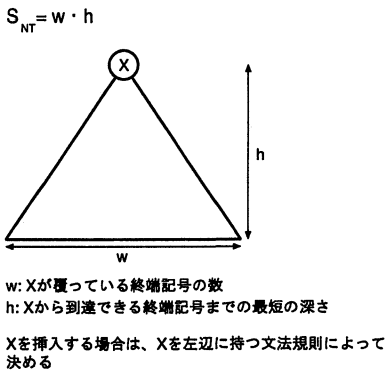

図 2: 非終端記号の基本値

- 修正した記号に対する基本値: s_{sym}
  終端記号 $term$ に対する基本値 ($s_{term} = 1$) または非終端記号 $nonterm$ に対する基本値 ($s_{nonterm}$) のいずれか．非終端記号に対する修正の方が終端記号に比べ修正の影響が大きいと考えられるので，$nonterm$ が覆っている面積に相当する値を計算することで反映させる (図 2).

これらを用いて，次のように S_T を計算する．

$$S_T = \sum_{\text{for errors}} s_{type} \cdot s_{sym}$$

解析木のスコア

S_P と S_T を足し合わせ，解析木全体のスコア S とする．

$$S = \alpha S_P + \beta S_T \ (\alpha, \beta \text{ は各スコアの重み})$$

5 実験

提案したアルゴリズムを用いて，対話文の解析において，非文解析を用いて解析精度がどの程度向上するかを調べる実験を行った．

入力文には ATR の英語対話コーパス [1] から 300 文を使用した．そして，このコーパスに Brill による Rule-based tagger [2] を用いてタグ付けを行い，生成された品詞列を入力とした．

文法は，Rule-based tagger で使用されている品詞のセットを終端記号とした英語の文法を用意した．文法の規則数，終端記号の数，非終端記号の数はそれぞれ 260, 34, 34 である．さらに，この文法を基に各文に対して正解となる解析木を人手で用意し，評価基準とした．

まず，通常解析・非文解析で候補を得た文数，正解が含まれていた文数の内訳を表 1 に示す．解析失敗の文は，1 箇所の誤り修正範囲に複数誤りがあることが原因であり，正解を得られない文は tagger の品詞割当の間違いによるものであった．また，非文解析で正解を含んでいた 117 文のうち，2 箇所以上の修正が行われた文は 24 文であった．

	候補の出力	正解を含む
通常解析で成功	140	134
非文解析で成功	132	117
解析失敗	28	——
合計	300	251

表 1: 誤り修正アルゴリズムの実験結果

次に，この 117 文に対して，スコアリングによる順位付けの実験を行った．評価は，絞り込まれた候補の上位 50 個に正解が含まれているかを基準とした．4 節でのパラメータは以下のように設定した．

$$\alpha = \beta = 0.5$$
$$s_{subst} = s_{extra} = s_{omit} = 1$$

絞り込みの結果，上位 50 個に正解が含まれていた文数を表 2 に示す．パラメータを一様に設定したため，上位の候補のスコアが横並びとなり，あまり良い結果は得られなかった．よって，適切なパラメータの設定方法を検討する必要があると考えられる．

非文正解	絞り込み	(絞り込み)/(非文正解)	(絞り込み)/(入力全体)
117	55	47.0%	18.3%

表 2: 解析候補の絞り込みの実験結果

6 まとめ

本稿では，一般化 LR 法を用いて文中に複数箇所の誤りが含まれる場合に誤りを検出し修正を行う手法を提案した．また，誤りの種類による曖昧性の増加に対応するため，解析で得られた複数の木に順位付けを行う方法を提案した．対話文を用いた実験では，通常解析で 45% 程度の解析制度を提案したアルゴリズムを用いて約 85% に向上させることができた．スコアリングによる候補の絞り込みについては，適切なパラメータの設定が必要であるが，現在は正解の解析木から誤りの種類の出現頻度を抽出し，それを用いてパラメータを学習する方法を検討・評価している．

今後の課題としては，1 回の解析失敗に対して複数箇所の誤りが存在する場合に対する処理を検討すること，提案したアルゴリズムの理論的な計算量の検討，などが考えられる．

参考文献

[1] ATR 自動翻訳電話研究所. ATR 対話データベースの内容, 1990.

[2] E. Brill. A simple rule-based part of speech tagger. In *Proceedings of the Third Conference on Aplied Natural Language Processing, ACL*, pp. 152–155, 1992.

[3] T. Kato. Yet another chart-based technique for parsing ill-formed input. In *4th Conference on Applied Natural Language Processing, ACL*, pp. 107–112, 1994.

[4] C. S. Mellish. Some chart-based techniques for parsing ill-formed input. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 102–109, 1989.

[5] H. Saito and M. Tomita. Parsing noisy sentences. In *Proceedings of the International Conference on Computational Linguistics*, pp. 561–566, 1988.

[6] T. Theeramunkong and H. Tanaka. A parallel chart-based parser for analyzing ill-formed inputs. 人工知能学会誌, Vol. 10, No. 4, pp. 531–541, 7 1995.

[7] 今井宏樹, 田中穂積, 徳永健伸. 一般化 LR 法を用いた非文の処理. 第 8 回人工知能学会全国大会, pp. 539–542, 1994.