

## 日本語学習支援システムにおいて 入力文の誤りを解析する為の単一スタックパーザ

劉 軼 榎本 圭孝† 加藤 伸隆 伊丹 誠 伊藤 紘二

東京理科大学

†富士ゼロックス(株)

### 1 はじめに

我々は、学習者がコンピュータエージェント(システムが用意する学習者の対話相手)との対話を通して、状況に応じた日本語表現を習得できるような日本語学習支援システムの研究を行っている。

システムは、学習者が状況と言語表現との対応がとりやすいように自然言語の語る世界を限定する。その中で、学習者とシステムはあらかじめ用意された機能指向型の対話を展開していく。可能な状況展開の場面毎に対話をあらかじめ用意することは、学習支援システムということを考えた場合、現在の自然言語処理技術ではいつも正しい文章が生成されるとは限らず、誤った表現を教えることを避けることが望ましいという考えからきている。

編集者が場面において使わせようとする主に自立語の語彙メニューを指定すると、システムは、パーザを利用して発話可能な文候補を生成する。その中で、編集者によるチェックをパスしなかったものについては、その理由をコメントとしてつけて貰う。

学習者はメニューの語彙を用い、機能語とムード(モダリティ)の部分、すなわち学習項目となる部分については、学習者が考えて補い、発話文を作成する。システムは、作成された発話文を前述のパーザに誤り解析機能をつけた機構によって解析し、その時の状況にそぐわなかったり、構文・意味的に誤りがあれば訂正して指導を行なう。

本論文では、このような枠組みのうち、発話文候補を生成し、かつ学習者の入力文の誤りを解析する為に単一のスタックを用いたパーザを提案する。具体的には、辞書引きと文法参照によって得られる格助詞句、連用修飾句、連体詞句と連体修飾句などは、位置拘束と構文構造と意味表現を組にして、スタックにプッシュする。文から用言(体言)が得られた時には、そこまでのスタックの中の用言を要求(体言を要求)する要素を、スタック上端から順に取り出して用言(体言)

への係りを調べ、可能なものまでをかけてまとめた句を作って、それをスタックに積み直す。

### 2 単一スタックパーザ

日本語を解析するには、様々な解析法があるが、今だなお、最適の解析方法が見つかったとはいえない。言語を解析するには、その言語の文法の規則性、意味論的な規則性などを考慮に入れる必要があり、状況意味コンテキストと照合する必要もある。

さて、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。この性質のために、日本語の解析は、スタックを用いることによって極めて把握しやすいものになる。すなわち、用言や体言に係る句をスタックに積んでゆき、用言や体言が現れた時に、それらに係ることができる句をスタックから下ろして用言句や体言句を作った上で、スタックに積み直す。係りうるかどうかを調べる際には、構文的、前後関係の、ならびに状況意味的拘束を利用する。この方法によれば、誤った文を解析する場合においても、見通しよく誤りを推定できることが予想される。スタックパーザは、汎用のものとして作られ、別に辞書情報を用意することによって、限定された世界の解析を行なうことができる。

#### 2.1 スタック

スタックパーザでは、句を形成させ、その句を要素(Element)としてスタック(Stack)にする。スタックに積む要素は句であって自体は、上で挙げたような要素を頭に積んでいく。その際の要素は、当面、用言句、体言句、格助詞句、副詞句、連用接続句、連体詞句、連体接続句を取り上げる。

◎ Stack ::= [Element|Stack]

Element ::= [Synstruct, Inflection, CaseInfo, Modality-Info, PrePost, Head, Semantic, SituationFocus]

但し、Synstruct: 統語構造。Inflection: 語形変化。

CaseInfo: 格または、接続情報。ModalityInfo: モダリティ情報。PrePost: 前後関係情報。Head: 主辞変数。Semantic: 意味表現。SituationFocus: 状況視点、例えば共感度条件。統語構造とは、要素である句の構文情報である。また、語形変化は、用言類の活用情報である。格情報、接続情報は、用言類の係り関係を知る際の「(人)は、(人)に、(物)を」などの情報である。モダリティ情報は、ある状況における話者の態度の情報のことである。前後関係情報は、連体詞句、連体接続句、連用接続句などにおいて、その句の主辞変数と、その後ろへ来るべき主辞変数をペアにしたもので、この句の意味表現に、両変数が含まれることによって、この後ろの接続部分との意味上の連絡をとるためのものである。主辞変数は、句における意味的な中心となる自立語に対する意味変数で統語解析を通じて意味表現を接続していく役割を担う。状況視点は、句が成立するために成立すべき状況内での条件などで表す。

## 2.2 パーザプログラムの概念

解析は、次のような辞書引きに始まり、そこから得られる統語・意味情報を、用言項、体言項、連用接続詞、接続詞など、品詞相当カテゴリ名の述語に引数として持たせ、それらを頭部に持つ以下に例示するような節を呼び出す。各節では、スタックから、辞書引きで得られた語に組合せられる句構造を取り出して、新たな句を作り、これをスタックに積み直す。

```
スタック解析 (Stack, S0, S):-
dic(Cat, Syn, Inflection, CaseInfo, ModalityInfo,
PrePost, Head, Sem, SituationFocus, S0, S1),
P=..[Cat, Stack, Syn, Inflection, CaseInfo,
ModalityInfo, PrePost, Head, Sem,
SituationFocus, S1, S],
call(P).
```

用言項 (Stack, Syn, Inflection, CaseInfo, Modality-Info, PrePost, Head, Sem, SituationFocus, S0, S1):-

用言係受け (Stack, Syn, Inflection, CaseInfo, ModalityInfo, Head, Sem, SituationFocus, YogenkuElement, StackRemain),

スタック解析 ([YogenkuElement|StackRemain], S0, S1).

用言項 (補助動詞、助動詞、終助詞の接続を含む) が来ると、スタックをポップしながら、格助詞、副助詞、用言句連用形、連用接続句などであって、意味制約上

も、この用言が受けられる要素まで (交差掛かり受けの禁止のため途中跳んではならない) を掛けて、用言句として、統語構造、意味表現などを作り、Elementとしてスタックに積み直す。

体言項 (Stack, Syn, Inflection, CaseInfo, Modality-Info, PrePost, Head, Sem, SituationFocus, S0, S1):-

体言係受け (Stack, Syn, Head, Sem, SituationFocus, TaigenkuElement, StackRemain),

スタック解析 ([TaigenkuElement|StackRemain], S0, S1).

体言項 (名詞、代名詞、指示詞) が来ると、スタックをポップしながら、用言句連体形、連体詞句、連体接続句等であって、意味制約上もこの体言が受けられるものまでを掛けて体言句とし、統語構造、意味表現を作ってスタックに積み直す。

このほか、以下に例示するようなルールが定義される。スタック先頭が、用言句の場合、辞書引きで、その活用形に見合った連用接続助詞が見い出されれば、連用接続句要素を作って、スタックにプッシュする。

スタック先頭が、体言句の場合、辞書引きで、格助詞があれば、一緒にして格助詞要素を作ってスタックに積む。

連体詞句がスタック先頭であるときに、格助詞が来ると、この連体詞句を「の」+空詞の体言句として、格助詞を加えて格助詞要素を作ってスタックに積む。

## 2.3 解析例

解析は上述のプログラムにおいて、辞書引きからはじめ、そこから得られた要素をスタックに積んでいく。ここで、「君が 僕が のぶちゃんに 買ってあげた おもちゃを 壊した。」の例文をあげ、次のように解析し、スタックパーザの動作を見てみることにする。

(1) 入力文は語彙メニューから選ばれて作られるので、[君, が, 僕, が, のぶちゃん, に, 買ってあげた, おもちゃ, を, 壊した, 。]

(2) 辞書引きにより [君] は体言項で、スタックが空なので、体言句 [君] 要素が積まれる。次の格助詞 [が] 見出されたところで格助詞句 [君が] 要素が作られてスタックに積まれる。以下、「僕, が」、「のぶちゃん, に」も同様であるから、結果として入力文の残り:

[買ってあげた, おもちゃ, を, 壊した, 。]

スタック:

[[[格助詞句,  
[体言句,[固有名詞,A, のぶちゃん]], [格助詞, に]],  
に,A,[nobuchan,A]],  
[[格助詞句,[体言句,[人称代名詞,B, 僕]], [格助詞, が]],  
が,B,[locuter,B]],  
[[格助詞句,[体言句,[人称代名詞,U, 君]], [格助詞, が]],  
が,U,[listener,U]]]

(3) [買ってあげた] は用言項であり、その格情報をスタックとの照合を行なう。

「が」格は「僕が」により、「に」格は「のぶちゃんに」により充足されるが、「を」格は非充足となるので、格情報にその旨のマーカnilをつけてスタックに積む。「君が」要素は結局は見出すのでスタックに残ったままとなる。結果として入力文の残り:

[おもちゃ, を, 壊した。]

スタック:

[[[用言句,[格助詞句,[人称代名詞,W, 僕],[格助詞,-, が]],  
[格助詞句,[固有名詞,Y, のぶちゃん],[格助詞,-, に]],  
[用言項,X, 買ってあげた]], [連体, 終止],  
[[W, が, に], [Y, に, が], [V, を, nil]], -, X,  
[[give\_benefit,X, benefactor,W, beneficiary,Y, by,Z],  
[buy,Z, agent,W, object,V, ], [locuter,W], [nobuchan,Y]],  
[empathy(X) ≥ empathy(Y)],  
[[格助詞句,[体言句,[人称代名詞,U, 君]], [格助詞, が]],  
が,U,[listener,U]]]

(4) [おもちゃ] は体言項であり、スタック先頭の「僕がのぶちゃんに買ってあげた」に対応した要素は用言句連体形なので、体言項に係り得るが非充足格情報の[Y, を, nil]が見出されるので、おもちゃ [toy, Y] がこの用言句の中から右外置されていると見ることができ

る。こうして「僕がのぶちゃんに買ってあげたおもちゃ」という体言句が作られてスタックに積まれる。あとは、この体言句に「を」がついた格助詞句が積み直され、残っていた格助詞句「君が」と共に、用言項句「壊した」の格情報を充足させることとなる。

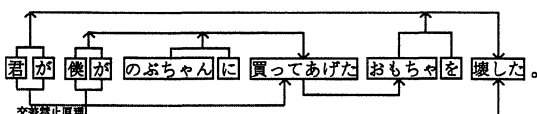


図 1: 係り受け関係

図 1 に上述の非交差係り受け、右外置の修飾処理、

または、飛び越し係り受けを示してある。

### 3 誤り解析について

本システムでは、インストラクタが、教育目標としての対話機能に合わせて、対話によって分岐する状況の流れを設定し、分岐場面毎に、そこで可能な表現のための自立語と機能語のメニューを作成する。

システムは、このメニューと状況設定に基づき、パーザを利用して分岐場面毎に可能な発話の候補を生成し、そのスタック構造とともに記録する。一方これをインストラクタに提示し、インストラクタは、それらの発話のうち、好ましくないものについて、学習者のためのコメントをつける。

学習の場面では、システムは、状況の流れを母語によって、学習者に伝えながら、状況が分岐する場面において、先にインストラクタによって作られた語彙のメニューを学習者に提示する。学習者は、メニューから選んだ語彙を用いて作文し入力する。入力された文が、先に挙げられている候補のうちの、コメントなしのものに該当すれば、正しいとみなす。コメントのあるものであれば、コメントを提示する。

入力文が候補文にないものである場合には、誤りカテゴリにより作られたバグルールと正しいルールとからなるパーザによって解析し、バグルールで解析できれば、そのバグをスタックに記録し、解析を続ける。もし、バグルールで解析できない場合は、システムが持っている可能な発話文を、表層における入力文との距離が近い順に、それらのスタックとともに並べ、入力文と発話候補の間の、ずれがある場所で、入力文のその部分を修正して正しい発話文にあわせたとき、スタック構造の上に該当する修正位置を特定して、そこに、その修正を記録する。

いずれの場合も、最終的に得られたスタック構造の上で、どこにどのような誤りがあったかを、母語によって学習者に伝える。このように誤りを検出するため、バグルールを用いるが、ルールは下記のカテゴリに分類される。

- 1) 機能語の欠落。
- 2) 用言の活用の誤り。
- 3) 機能語の誤り (例えば、「が」の代わりに「を」)。
- 4) 余分な機能語の挿入。
- 5) 自立語の誤り、欠落ないし余分。
- 6) 句の誤り、欠落ないし余分。

#### 4 おわりに

状況記述の仕方、予め可能な発話文候補を生成する部分で状況記述を用いて効率的に絞りこむための方法、また、誤り解析の為の情報を汎用のバグルールと辞書情報とに旨く分離できるかどうかについては未だ今後の課題である。

#### 参考文献

- [1] Enomoto, Liu, Itami and Itoh (1995.3): A Mechanism for Correcting Errors of Japanese Expressions According to the Discourse Situation – For the Computer Assistant for Learning Situated Expression in Japanese as Foreign Language –. *Technical Report, SIG-IES-9403-7, Japanese Society for Artificial Intelligence.*
- [2] 郡司 隆男 訳「制約にもとづく統語論と意味論」. 産業図書. 1994.5
- [3] Liu, Kato, Itami and Itoh (1995.12): Generation of Comments in an Interactive Environment for Learning Japanese as Foreign Language. – For Chinese Mother Tongue – *Natural Language Processing Pacific Rim Symposium '95.*
- [4] Sells, P.: *Lectures on Contemporary Syntactic Theories*, CSLI Lecture Notes Series No.3, 1985.