

汎用日本語処理系 (LTB) 文生成部の生成方式 Sentences Generation in LTB

池田 先生^{*}、幡野 浩司^{**}、福島 秀顕^{**}、重永 信一^{*}
松下電器東京研究所、新世代コンピュータ技術開発機構^{**}

表層文の構文構造を表現するために二分木をもとにした基本表現を設定し、これに対するマクロ表現の形で生成用中間表現とマクロ展開処理を定義した。また、本生成部のデフォルト処理を切り分けることで処理規則が簡潔になり、ユーザによる処理規則の定義が容易になった。以上により本生成部の汎用性を高めることができた。

1. はじめに

我々は、これまでの自然言語処理に関する成果を汎用的なツールとしてまとめて、LTB (Language Tool Box) に組み込む文生成部の研究開発を行っている。本生成部では、構文情報だけを参照して表層文を生成する。

本生成部の入力データの表現を、生成用中間表現と呼ぶ。生成用中間表現には生成に必要な構文情報がすべて与えられることを想定する。態変化にともなう助詞変換、格助詞、接続助詞などの連結語、語彙などが、中間表現に書かれていない場合には、システムがデフォルト処理を行う。

システムの汎用性を高めるためには、入力形式はユーザが自由に設定できることが望ましい。また、語彙決定などのデフォルト処理もユーザごとに定義できたほうがよい。そこで、本生成部では表層文の構文構造を二分木により表した基本表現を設定し、この基本表現を得るためにマクロ表現を用いることによって、ユーザごとに自由に生成用中間表現の記述形式を定義可能にしている。システムのデフォルト処理もマクロ表現のマクロ展開処理に取り込むことによって、ユーザごとの定義ができる。

生成用中間表現はすべて基本表現に展開されるため、どのレベルのマクロ表現で記述されていても、同様の生成結果が得られる。

昨年度の試作システム^[2]では、接辞の決定、サ変名詞の動詞化などのデフォルト処理を一度に行っていたが、各処理で互いに処理結果を参照し合うため困難であった。本生成部では各処理を切り分け、何段階かの処理に分けた。このことにより、処理規則が簡潔になり、デフォルト処理のユーザ定義がしやすく、汎用的になった。

2章では、基本表現とマクロ定義の例について述べ、3章では、マクロ展開において行われる生成処理と、基本表現からの形態素処理について述べる。最後に4章で、まとめと今後の課題について述べる。

2. 汎用日本語文生成システムの間接表現

2.1 基本表現

文生成部の入力となる生成用中間表現は、生成すべき文や節や句の構文情報を表したものである。生成用中間表現をユーザが自由に設定できれば、システムの汎用性を高めることができる。そこで、生成に必要な構文情報を二分木をもとにして表した基本表現と呼ぶ表現形式を設定し、この基本表現に対するマクロという形で、様々なレベルの生成用中間表現を設定することで本生成部を汎用的なものにした。

本生成部の基本表現では日本語の構文構造を表すのに二分木による表現を採用した^[3]。基本表現では、すべての語彙がすでに決められており、節や句の語順が二分木の構造の形で決定されている。また、基本表現では、構文的な情報のみが記述されていて、意味的な情報はもはや記述されていない。したがって、基本表現の節や句には、述語が持っている格に対応する「行為者」「対象」などのラベルはつけられていない。基本表現は、次の形式に従って書かれる。

```
<文> ::=
    { head / <文>.comp / <文> } |
    { cont / <文>.symbol / <記号> } |
    set ( [ <文> . <文> . . . . . ] ) |
    { lex / <表層表現> }

<記号> ::=
    { type / <TYPE> . right / <SYMBOL> . left / <SYMBOL> }

<TYPE> ::=
    right | left | right left
```

句読点や引用符などの記号の処理を行うために、symbolおよびcontというラベルを用意した。contの値として与えられた文にsymbolの値として与えられた記号が付加される。

また、「AとBと」や「AしたりBしたり」など

の並立関係を記述するためにset()を用いる。

図1に基本表現による記述例をあける。

(例)人間は、自然の恵みに頼らなければならない。

```
{head/{head/{head/{lex/なければならない},
  comp/{lex/頼る}},
  comp/{head/{lex/に},
    comp/{head/{lex/恵み},
      comp/{head/{lex/の},
        comp/{lex/自然}}}}},
  comp/{symbol/{type/right,right/",""},
    cont/{head/{lex/は},
      comp/{lex/人間}}}}
```

図1 基本表現による記述例

本生成部の入力は後述のマクロ表現である。生成処理を行った結果、基本表現に“suf”属性を付加したものが得られる。“suf”属性には語彙データのリストが与えられている。

2.2.2 マクロ表現

2.1で述べた基本表現はマクロ表記を用いて記述することができ、マクロ表現はユーザが自由に設定することが可能である。すべてのマクロ表現は基本表現に展開されるため、どのレベルのマクロ表現で記述されていても同様の生成結果を得ることができる。

ここでは、概念的なマクロ表記の例をあけ、さらに次章でマクロ展開処理についても説明する。

マクロ表現においては、語順があいまいであったり、接辞が未決定であったりする場合など、構文情報を明示しない書き方をすることができる。このように構文構造が一意に定まらない部分については、システムがマクロ展開を行う過程でデフォルト処理を行う。マクロ展開処理はユーザが定義することが可能であり、したがって各デフォルト処理はユーザごとに設定することが可能である。

また、ヴォイス、モタリティ、アスハクトなどを基本表現のように構造として記述することはむしろ難しく、記述量も多くなる。これらもマクロで表すことによって簡潔な記述が可能である。

今回提案するマクロ表現は、述語によって表される関係を中心として、その格構造に対応するロールを記述することによって文を表現する表現形式である。

日本語の構文的な情報を属性－属性値の対からなる属性構造で表し、ヴォイス、アスハクト、モタリティの指定ができるようになっている。また、語順

や、接辞は必ずしも決定されている必要はなく、指定のない場合にはデフォルト処理を行う。マクロ表現は以下のような形式によって記述される。スペースの関係で一部割愛した。

```
<文> ::= {<接続詞> } {<文核> } {<名詞節> }
<文核> ::= <関係> {<関係子> } {<ロール> }
           {<モーダル> } {<表層> } {<語順> } {<名詞化> }
           {<時制> } {<認め方> } {<アスハクト> }
           {<ヴォイス> } {<記号> } {<取り立て詞> }
<名詞節> ::= <語彙> {<連体修飾> } {<取り立て詞> }
           {<記号> }
<関係> ::= 関係 / ( <文> | 並立 | 特殊 )
<関係子> ::= 関係子 / ( の | も | たり | ... )
<ロール> ::=
           ロール / ( {<ロール名> / <補語> , ... } |
           <補語> | set( {<補語> , ... } ) )
<モーダル> ::=
           モーダル / {<態度> } {<ムード> } {<丁寧> }
           {<認め方> } {<時制> }
<補語> ::= <補語> / <文> {<取り立て詞> } {<表層> }
           {<記号> }
<連体修飾> ::= 連体修飾 / <文>
```

関係属性の値で「並立」は並立する句や節の記述に用いる。関係属性の値が並立の場合は関係子属性を設け、それぞれの句や節に付加すべき接辞を指定する。

例えば、「食物と衣服と」のマクロ表現は次のようになる。

```
{関係/並立,
 関係子/と,
 ロール/set({補語/{語彙/食物},
             補語/{語彙/衣服}})}
```

「特殊」という関係は連体修飾句の記述に用いる。特殊関係の関係子の値としては「の」「という」「についての」などがある。例えば、「地球の上」のマクロ表現は次のようになる。

```
{語彙/上,
 連体修飾/{関係/特殊,
            関係子/の,
            ロール/{補語/{語彙/地球}}}}
```

次に「人間は、自然の恵みに頼らなければならない」という表層文に対応するマクロ表現の例を図2に示す。

(関係/語彙/頼る),
 モーダル/ムード/必然),
 ロール/
 (行為者/補語/語彙/人間),
 取り立て詞/は,
 記号/読点,
 対象/補語/語彙/恵み,
 連体修飾/関係/特殊,
 関係子/の,
 ロール/
 (補語/語彙/自然))]]

図2 マクロ表現による記述例

3. 生成エンジンの処理方式

3. 1 マクロ展開処理

3. 1. 1 生成エンジンの処理手順

本生成部の生成エンジンは、2. 2で述べたマクロ表現を2. 1で述べた基本表現に展開する。図1の基本表現は図2のマクロ表現を展開したものである。辞書検索、感変化による助詞変換、語順決定などのデフォルト処理は、このマクロ展開の中で行う。マクロ展開はユーザが自由に定義することができる。ここでは、2. 2で述べたマクロ表現に対する展開処理について説明する。

昨年度の試作システムは、生成用文法の中から生成処理手続きを呼ぶ方式で、生成用文法に従った一回の手続き実行の中ですべての生成処理を行い直接語彙データを得る方式であった。しかし、この方式は各生成処理手続きが互いに情報を参照し合うため困難であった。

例えば、述語の格に対応するロールの接辞を決定するために、そのロールが持つ名詞化属性の値や、ロールの主辞の品詞を参照することがある。一方、名詞化属性の値を決定する手続きでは、そのロールの接辞を参照したり、主辞の品詞を参照したりする。

このように、各生成処理手続きを生成用文法という手続き実行制御プログラムの中で制御することは困難であった。これに対して本生成部では、マクロ表現の各々の既件に対してマクロ展開を行う手続きを独立に用意し、各手続きを完全に切り分けた。これらのマクロ展開手続きを、マクロ表現のレベルや、マクロ展開処理の進捗に応じて、呼び出すことによりマクロ展開処理を行うことができる。このことにより、マクロ展開規則を非常に簡潔に書くことができた。

各生成処理では、マクロ展開の結果得られた語彙データのリストを"suf"属性の値として与える。マク

ロ展開がすべて終わった後の基本表現の中の"suf"の値を取り出して、形態素生成処理部へ渡し、表層文の生成を行う。

3. 1. 2 関係属性の処理

関係属性のマクロ展開処理では、関係属性の値として与えられた述語の辞書検索を行い、名詞化属性の付加や、述語性名詞の動詞化などを行う。

関係属性の値として述語性の名詞を与えることができる。関係属性の値が名詞であり、かつ派生名詞化の指定や、品詞の指定がされていなければ動詞化を行う。形容名詞及び状態性の副詞と連体詞には、「た」を付加して活用型をタ系にする。サ変名詞には「する」を付加して活用型をウ系サ変にする。

関係属性の値は、通常は、動詞、形容詞、述語性名詞であるが、マクロ表現ではこれ以外に「特殊」「並立」という特別な関係を用意している。特殊関係は連体修飾における「の関係」や「同格関係」などを記述するのに用いる。また、並立関係は節や句や語の並立関係を記述するのに用いる。

さらに、関係属性の値として「自然に頼る」のような「文」に相当するマクロ表現を与えることができる。関係属性をマクロ展開することによって、名詞化処理や動詞化処理などの結果が"head"の値として与えられる。

3. 1. 3 ヴォイスの処理

マクロ表現のヴォイス属性の値には、「受動」「使役」などがある。また、間接受動などの場合には、「受動("受け身主体")」のように書く。

まず、ヴォイス属性の値から語彙を決める。受動なら「れる」、使役なら「せる」である。

次に、"head"の値を取り出し、ヴォイス成分を付加したものを新たに"head"の値として与える。ヴォイス属性の処理時には、関係属性はすでにマクロ展開されていることを前提とする。例えば、"head"の値が「頼る」でヴォイス属性の値が「受動」のとき、ヴォイス属性のマクロ展開の結果、"head"の値は「頼られる」となる。

次に、感変化による格助詞変換規則を用いてロールの格助詞の変換を行う。この時点で、各ロールの格助詞は決定されていなければならない。

3. 1. 4 アスペクトの処理

マクロ表現のアスペクト属性には、「ている」「つつける」のようにアスペクト辞を直接与える。

"head"の値を取り出して、アスペクト成分を加えたものを新たに"head"の値として与える。ヴォイスの場合同様、"head"の値として、述語で表される関係が記述されていることを前提とする。例えば、

"head"の値が「頼られる」でアスペクト属性の値が「ている」のとき、アスペクト属性のマクロ展開の結果、"head"の値は「頼られている」となる。

3. 1. 5 認め方、時制の処理

マクロ表現の認め方属性の値は「肯定」または「否定」である。認め方属性の値が指定されない時は、「肯定」とする。時制属性の値は、「現在」または「過去」である。時制属性の値が指定されない時は、「現在」とする。認め方属性の値が、「肯定」であったり、時制属性の値が「現在」であるときは、何もせずに処理を終わる。

認め方属性の値が、「否定」のときは、"head"の値を取り出して、「ない」を加えたものを新たに、"head"の値として与える。例えば、"head"の値が「頼られている」のとき、認め方属性のマクロ展開の結果、"head"の値は「頼られていない」となる。

時制属性の値が、「過去」のときは、"head"の値を取り出して、「た」を加えたものを新たに"head"の値として与える。例えば、"head"の値が「頼られていない」のとき、時制属性のマクロ展開の結果、"head"の値は「頼られていなかった」となる。

3. 1. 6 モタリティの処理

モータル属性の値は、態度、ムート、丁寧、認め方、時制からなる。モータル属性の中の認め方属性と時制属性はモタリティに関する認め方、時制の値を表している。両方モタリティの場合は、リストを用いて表し、リストに書かれた順番に生成する。例えば、「でもよいのだろうか」という表層表現に対応するマクロ表現は次のように表される。

モータル/ムート/[でもよい、のた、う]、
態度/か]

モタリティの処理では、"head"の値を取り出して、モタリティ成分を付加し、新たな"head"の値として与えるということを繰り返すことによって処理を行う。

3. 1. 7 ロールの処理

"head"の値として与えられる関係の格成分に対応するロール属性の値をマクロ表現から取り出して、辞書の格情報を参照しながら、ロール成分に付加すべき接辞を決定する。

次に、ロール成分が動詞である場合にはロール名と接辞とを参照して、名詞化属性の値を付与する。マクロ表現に名詞化の方法が指定されているときはそれに従う。指定のない場合は名詞化属性付与規則によって属性値を付加する。

3. 1. 8 語順の処理

マクロ表現には語順属性があり、ロールや連体修飾句の語順を指定できる。語順の指定のない場合は、語順規則により決定する。本生成部では、ロール名を7つのグループに分けて、そのグループ間の順序を定めている。また、取り立て詞の「は」を持つロールは他のロールよりも前に置く。

基本表現では、語順は二分木の形式で表される。語順の処理では、決められた語順に従って、二分木を構成していき、基本表現を作る。

3. 2 形態素生成処理

形態素生成部では、マクロ展開の結果得られた基本表現の"suf"属性の値を取り出して、表層文を生成する。

"suf"属性の値は語彙データのリストになっていて、形態素生成部では、個々の活用語について活用を変化させるための後続する最初の語彙データを求め、後続表を用いて活用形を決める。

4. まとめと今後の課題

汎用日本語処理系(LTB)文生成部の生成方式について述べた。基本表現を設定し、これに対するマクロ表現をユーザが定義可能にすることによって、本生成部の汎用性が高まった。また、生成処理を切り分けることにより、処理規則を簡潔に記述することかでき、ユーザによる変更がしやすくなった。

現在、ユーザ定義するためにはプログラムを書く必要があるがこれをツール化して使いやすいうのにすること、さらに、これを発展させて各生成処理を並列に実行することが今後の課題である。

参考文献

- [1] 木村 睦子 空閑 茂起
「態による格助詞変換」
計量国語学第十五巻第二号 1985 pp54-62
- [2] 重永 信一 安川 秀樹
「自然言語処理システムのための日本語文生成
ツールの試作」
情報処理学会第35回全国大会 1987
- [3] 日井英俊 郡司隆男 橋田浩一 原田康世
「原所的制約による文法記述」
電子情報通信学会言語処理とコミュニケーション
研究委員会資料,NL87-1,1987