

4F-4

構文意味解析システム SAX における係り受け解析

- 構文意味情報の構成について -

福本文代

(財) 新世代コンピュータ技術開発機構

杉村領一

松下電器産業株式会社

1 はじめに

ICOTでは、自然言語処理の研究過程で得られた要素技術を汎用日本語処理系(Language Tool Box 以下LTB)としてまとめ上げている。SAXは、このLTBの一モジュールであり、論理型言語により高速に構文意味解析を行なうシステムであるが、[1]、今回この上に新たに文を文節と文節の係り受け関係で表現する係り受け文法を開発した。この文法は、係り受けの判定処理の細分化のみで文が解析でき、かつその処理は文法規則中、局所化しているため、効率よく日本語解析が行なえる。本文法では、文の構成単位である文節の意味を物または関係として定義し、出力する文の意味情報を、物と物との間の関係がネットワーク構造で表現されるようにする。また解析過程で得られた意味情報を蓄積するために、その文節の持つ構文情報を保持しながら、文節の意味情報を後文節へ伝播する。本稿では、この係り受け文法で意味を構築する手法、すなわち意味の蓄積伝播について述べる。

2 論理型言語による係り受け解析

本文法はDCG(definite clause grammar)と呼ばれる文法記述形式に0型文法を導入することによって実現された拡張記述形式を用い、文節と文節の係り受けを判定する係り受け文法である[2]。文を解析する際用いる文法規則は、

- 「AとBと」のような等位構造を認識し一文節にまとめるルール群【(9),(11)～(17)】
- まとめた等位構造を含む文節の依存関係(文節の係り受け)を調べるルール群【上記以外】

より成る。

<i>sentence</i>	→	<i>wordtt wba</i>	(1)
<i>wordt</i>	→	[<i>begin</i>]	(2)
<i>wordtt</i>	→	[<i>begin</i>]	(3)
<i>bar</i>	→	[<i>end</i>]	(4)
<i>wba</i>	→	<i>word bar wba</i>	(5)

Dependency Structure Analysis in SAX
-Composition of Syntactic and Semantic Information-
Fumiyo FUKUMOTO ICOT
Ryoichi SUGIMURA Matsushita Electric Industrial Co.,Ltd.

<i>wba</i>	→	<i>word bar</i>	(6)
<i>word bar</i>	→	<i>gapH bar</i>	(7)
<i>word bar word</i>	→	<i>word word</i>	(8)
<i>word bar pword</i>	→	<i>word pword</i>	(9)
<i>word bar word</i>	→	<i>word (bar; []) wba word</i>	(10)
<i>wordtt pword</i>	→	<i>wordt wordg</i>	(11)
<i>wordtt gapH</i>	→	<i>wordt wordg</i>	(12)
<i>word pword</i>	→	<i>gapH wordg</i>	(13)
<i>word gapH</i>	→	<i>gapH wordg</i>	(14)
<i>pword pword</i>	→	<i>pword wordg</i>	(15)
<i>pword gapH</i>	→	<i>pword wordg</i>	(16)
<i>pword</i>	→	<i>pword (bar; []) (wba; []) word</i>	(17)
<i>word</i>	→	<i>pword</i>	(18)

文節の持つ意味情報の蓄積は、この文法の補強項部分で行われている。ここでは、補強項でのこの処理を係り受け決定関数と呼ぶ。係り受け決定関数では、文節が等位を示し、かつ文節と文節が同種の等位であるかどうかの判定処理、あるいは文節と文節が係り受け可能か否かの判定処理、そしてこれらを判定した結果、同種の等位であり、また係り受け可能と判明した時に、文節の持つ意味情報を生成する処理を行なっている。

3 データ表現

本文法は、文の構成単位である文節の意味を、物あるいは関係として定義する。また扱う文節の構文意味情報は、形態素解析システムLAXの出力が与えられる。文法はこれを入力し、解析を行なった結果、その意味情報を物と物との間の関係がネットワーク構造で表現されるよう蓄積する。以下に文節「食べる」の構文意味情報の一例を示す。ここで、意味情報は、soas/で記述し、構文情報はそれ以外とする。

<i>sur</i> /[食, べ, る]	(1)
<i>moded</i> / cat/用言	(2)
<i>frame</i> /[(agt, が, 46], (obj, を, 47)]	(3)
:	
:	
<i>moding</i> / cat/連体詞	(4)
<i>soas</i> /[(45, heap: {食べる, 46, 47, yes})]	(5)

図1: 係り受け文法で扱う文節の構文意味情報

4 構文意味情報の蓄積

本文法は与えられた終端記号列に対し、順次文節と文節の係り受けの判定を行なう。この結果（係り受け）依存関係が成立すると、係る側の文節の意味情報をコピーし、それを受け側の文節のスタックに蓄積する。最終的に解析が終わった時点では、意味情報は文末の文節に蓄積される。この蓄積過程において、文法規則と文法カテゴリを極力減らしつつ、文節と文節の等位関係、依存関係により生成される新たな意味情報と、その文節の持つもとの構文意味情報とを保持するために、係った意味情報をスタックに蓄積する部分とその文節本来の構文意味情報を保持する部分とに分け、それら文節の構文情報を pair（組）で表した。以下では、意味の蓄積方法を同種の等位か否かを判定する箇所と、文節と文節の係り受けを判定する箇所とに分けて、述べる。

4.1 係り受けにおける意味の蓄積方法

係り受けにおける意味の蓄積方法は、pair で表された構文情報のどちらを係り受け判定処理へ渡すかの決定を行なう述語 pair - check で実現した。その際の係り受け判定タイプ分類表を（表 1）に示す。

```
w1(SEM11),bar,w2(SEM22)
  → w1(SEM1),w2(SEM2)
  { … pair_check(SEM1,SEM2), … }
```

```
pair_check([A,B], [C,D],X,Y,XX,YY,P,Q)
```

[A,B] : w1 の構文意味情報を pair のリストで表したものの

[C,D] : w2 の構文意味情報を pair のリストで表したものの

X : 係り受け処理に渡すもの (A または B)

Y : 係り受け処理に渡すもの (C または D)

XX : 係り受け処理結果 (X の結果)

YY : 係り受け処理結果 (Y の結果)

P : SEM11(A または B と XX とがリストで入力)

Q : SEM22(C または D と YY とがリストで入力)

表 1: 文節と文節の係り受け判定タイプ分類

No.	等位有無		W1	W2	処理	example
	W1	W2	W1([A,B])	W2([C,D])	X Y	
1	×	×	(S1,S1)	(S2,S2)	S1 S2	私が食べる
2	×	×	(S1,S1)	(S2,S2')	S1 S2'	私が物を食べる
3	×	×	(S1,S1')	(S2,S2)	S1' S2	私が物を食べる
4	×	×	(S1,S1')	(S2,S2')	S1' S2'	私が物を彼に与える
5	○	×	(S1,S1)	(S2,S2)	S1 S2	彼と美しい
6	○	×	(S1,S1)	(S2,S2')	S1 S2'	彼と速く歩く
7	○	×	(S1,S1')	(S2,S2)	S1' S2	物も家もある
8	○	×	(S1,S1')	(S2,S2')	S1' S2'	物も家も彼は持つ
9	×	○	(S1,S1)	(S2,S2)	S1 S2	食べる 物も
10	×	○	(S1,S1)	(S2,S2')	S1 S2	食べる 物も家も
11	×	○	(S1,S1')	(S2,S2)	S1' S2	私が食べる 物も家も
12	×	○	(S1,S1')	(S2,S2)	S1' S2	私が食べる 物も
13	○	○	(S1,S1)	(S2,S2)	S1 S2	花と森も
14	○	○	(S1,S1')	(S2,S2)	S1' S2	小さくて赤い花と
15	○	○	(S1,S1)	(S2,S2')	S1 S2'	かわいい花と蝶と
16	○	○	(S1,S1')	(S2,S2')	S1' S2'	小さくて赤い花と蝶と

上式あるいは表中、W1,W2 はそれぞれ係る側と受け側の文節とする。W1 又は W2 は、一文節または等位構造を表す。W1,W2 の構文意味情報は SEM1,SEM2 で表され、さらに SEM1 あるいは SEM2 は構文意味情報 A と B,C と D をそれぞれリストで表現したものから成る。各々の文節は、蓄積された構文情報 (s1' 又は s2')

と、その文節本来の構文情報 (s1 又は s2) とを pair で持つ。係り受けの判定を行なう際、上記表に従って、A 又は B, C 又は D のいずれか一方ずつを係り受け判定処理へ渡すことにより意味の蓄積が行われる。

4.2 等位構造における意味の蓄積方法

等位における意味の蓄積は、同種の等位か否かの判定を行う述語式 comp - Syn で実現した。

```
pword([SEM1L,SEM2RR])
→ pword([SEM1L,SEM1R],word([_,SEM2R])
{ … comp_Syn(SEM1R,SEM2R,SEM2RR) … }
```

```
comp_Syn(A,B,C)
```

A : pword で等位か否かの処理に渡すもの

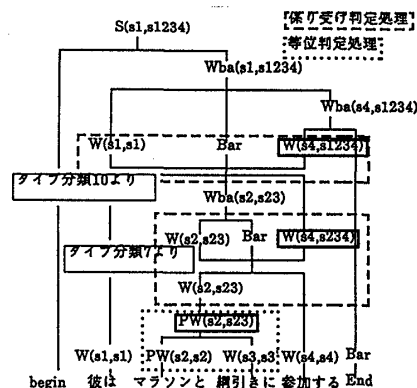
B : word で等位か否かの処理に渡すもの

C : 等位処理が成立した結果

等位構造では、等位を示す文節の次に出現した文節が同種の等位であり、等位構造を生成することが判明した時に始めて意味が蓄積される。上記は等位を示す文節の次に出現した文節が同種の等位であるか否かを判定するルールである。ルールが適応された場合には、等位構造が作成されるが、その際蓄積された構文情報は等位の文節の pair のうち常に右側に蓄積される。

5 解析例

ここでは、「彼は マラソンと 綱引きに 参加する。」を例にとり、その解析結果をノードで表す。



6 おわりに

本稿では、係り受け解析における意味の蓄積手法について述べた。本手法を SAX 上に構築した結果、日本語の係り受け解析を効率よく行なうことができた。今後は、係り受けの判定条件をより詳細化すると共に、係り受けで得られた文節の部分情報の談話情報への反映についても考えていく予定である。

参考文献

- [1] Y.Matsumoto and R.Sugimura, "A parsing system based on Logic Programming", *Proceedings of the International Joint Conference of Artificial Intelligence*, 1987.
- [2] 杉村領一, 「レイヤードストリームを用いた日本語解析処理」 ICOT-TR,1989.