Estimation of the PCFG Building the Dependency Constraint

田辺, 利文
知能システム学専攻 : 博士後期課程

冨浦, 洋一
知能システム学専攻

日高, 達
知能システム学専攻

https://doi.org/10.15017/1485040

出版情報：九州大学大学院システム情報科学紀要. 2 (1), pp.93-97, 1997-03-26. Faculty of Information Science and Electrical Engineering, Kyushu University
バージョン：
権利関係：

九州大学大学院
システム情報科学研究科報告
第2巻 第1号 平成9年3月

Research Reports on Information Science and
Electrical Engineering of Kyushu University
Vol.2, No.1, March 1997

係り受け制約を組み込んだPCFGの評価

田辺利文* ・ 冨浦洋一** ・ 日高 達**

Estimation of the PCFG Building the Dependency Constraint

Toshifumi TANABE, Yoichi TOMIURA and Toru HITAKA

(Received December 24, 1996)

Abstract: In Natural Languages Processing, there are lots of syntax trees corresponding to an input sentence. If we can choose the correct syntax tree meaningfully, the quality of the processing is improved. We proposed the method of building the dependency constraint into a context free grammar by subdividing nonterminals according to the meaning of the phrase generated from the nonterminal and by grasping production rules from superordinate-subordinate relation of their meanings in the thesaurus. This paper shows estimation of the PCFG building the Dependency Constraint by the experiment of disambiguating which of N_2 or N_3 governs N_1 in the noun phrase "N_1の N_2の N_3" .

Keywords: Context Free Grammar, The Dependency Constraint, Head word, Function, Thesaurus, Superordinate-Subordinate relation

1. はじめに

自然言語処理における構文解析では，一般に入力文に対応する構文構造がたくさん存在し，それらからどのようにして構文構造を選択するかが問題点の一つである．構文構造の中には誤った意味のものも含まれる．意味的に正しい構文構造を選択できるかどうかが，仮名漢字変換や機械翻訳などの以後の処理の質を大きく左右する．従来は文節数最少法のような粗い経験則によって構文構造の間に優先順位を付け，優先順位の高い構文構造に基づき出力を合成していた．しかし，この方法は質の高いものではなかった．そこで，さらに質を上げるには，意味処理の導入が自然言語の機械処理の大きな課題になっている．

意味処理の実用的な導入として，係り受け制約を文脈自由文法（CFG）に組み込むことが考えられる．この組み込み法として我々は非終端記号から導出される句の概念（意味）により非終端記号を細分化し，さらにシソーラスを文法規則として捉えることにより，係り受け制約を文脈自由文法の生成規則として組み込む手法を提案した[1)]．

本論文ではそれを確率化した確率文脈自由文法(PCFG) を用いた名詞句「N_1のN_2のN_3」におけるN_1の係り先の曖昧さ解消実験により提案した係り受け制約のCFGへの組み込み法を評価する．

平成8年12月24日受付

* 知能システム学専攻博士後期課程
** 知能システム学専攻

2. 係り受け制約を組み込んだ文脈自由文法

本節では，文献1)に従って，係り受け制約の文脈自由文法への組み込み法を概説する．

2.1 係り受けの表現

係り受け関係を解析する際にポイントとなるのは，係り受け関係を構成するときの係る語，係られる語と，係りの種類を決定する情報である．句において，修飾語，被修飾語になり得る単語をその句の*head word*，修飾句における係りの種類を決定する情報をその句の*function* と定義する．今後，句Xを，非終端記号Xで用いる他に，Xから導出されている単語列としても用いることにする．

この場合，句は，

- *head word*と*function*の両方を持つ句
- *head word*を持ち，*function*を持たない句
- *function*を持ち，*head word*を持たない句

の3種類に分類することが出来る．このような句を導出する非終端記号の集合をそれぞれN_{HF}，N_H，N_Fと表すものとする．

係り受けの観点から述べると，N_{HF}の要素は他の句の単語に係り得る単語とその係りの種類を決定する情報を含む句(修飾句)を導出する非終端記号であり，N_Hの要素は受けることしか出来ない単語をその句の*head word*として含む句(被修飾句)を導出する非終端記号である．N_Fの要素は単独で係りも受けも出来ない単語を含む句を導出する非終端記号であるが，その句において係りの種類を決定する単語を含む句を導出する．自然言語の文法では，その非終端記号の分類をもとにすると，一般に次のよう

に生成規則を分類することが出来る．

1. $X \longrightarrow Y_1 \cdots Y_{k-1}\quad Z\quad Y_k \cdots Y_m$
   $(X, Z \in N_H, Y_k \in N_{HF})$
   $Y_i(i = 1, 2, \cdots, m)$から導出される句がZから導出される句を修飾している場合．Xの*head word*は，Zの*head word*になる．
2. $X \longrightarrow Y\quad Z\qquad (X \in N_{HF},\ Y \in N_H,\ Z \in N_F)$
   Xが修飾句になり得る場合．Xの*head word*はYの*head word*，Xの*function*はZの*function*である．
3. $X \longrightarrow \alpha\quad X\quad \beta\qquad (X \in N_F)$
   単語列の結合の時にどれが*function*になるのかを規定している．左辺のXの*function*は右辺のXの*function*である．

文が与えられ，それに対する構文木に

$$X \longrightarrow Y_1 \cdots\quad Y_{k-1}\quad Z\quad Y_k\quad \cdots Y_m \qquad (2.1)$$

という規則(但し，$Y_i \in N_{HF}$，$Z \in N_H$，$X \in N_H$)が含まれている時，その規則においてY_iの*head word*がh_i，Zの*head word*がh，Y_iの*function*がfであるとき，h_iはhに係りの種類fで構造的に係っていると定義する．

しかし構造的な係り受け関係が意味的に適格であるとは限らない．CFGでは，規則(2.1)を用いてXを$Y_1 \cdots\quad Y_{k-1}\quad Z\quad Y_k\quad \cdots Y_m$に書き換えた場合，$Y_i$からの導出と$Z$からの導出は独立である．従って，CFGで意味的にも適格に係り受け関係を表現するためには，単語と単語とを依存させることが出来るような非終端記号の細分化が必要となる．

ここで，係られる単語h_iとその係りの種類fは，係る単語hに依存すると考え，係り受けを生成規則に取り込むことを考えると，規則(2.1)において，$Z \in N_H$から導出される語hで，$Y_i \in N_{HF}$から導出される語h_i及びf_iをコントロールすることが出来れば，意味的に適格な係り受け関係も記述することが出来る．従って，規則(2.1)は，

$$X(h) \longrightarrow Y_1(-h)\quad \cdots\quad Z(h)\quad \cdots\quad Y_m(-h) \qquad (2.2)$$

及び

$$Y_i(-h) \longrightarrow Y_i(h_i, f_i) \qquad (2.3)$$

と2つの生成規則とすれば良い．規則(2.2)，(2.3)において，各非終端記号は以下のような意味を持つ．

- $X(h, f)$　*head word*がhであり，*function*がfであるカテゴリXの句を導出する非終端記号
- $X(-h)$　*head word*がhである句に係るカテゴリXの句を導出する非終端記号
- $X(h)$　*head word*がhであるカテゴリXの句を導出する非終端記号
- $X(f)$　*function*がfであるカテゴリXの句を導出する非終端記号

規則(2.3)はh_iが係りの種類fでhに係りうることを表している．

ただし，提案した文法では，非交差性を満足しない文，並列句を含む文，終助詞，受身，使役の助動詞を取り扱わないものとする．

2.2 シソーラスの生成規則への組込み

2.1節で提案した文法において問題になるのは，生成規則の数である．それにより処理時間も増えてしまうため，意味的に適格な係り受け関係の表現精度にあまり影響を及ぼさないように非終端記号を減らす方法を考える必要がある．単語w_1とw_2の間に意味的に適格な係り受け関係が成立していれば，w_1の下位語であるw_1'と，w_2の下位語であるw_2'の間にも意味的に適格な係り受け関係が成立するものと仮定する．

シソーラスにおいて概念 W_u と W_d が上位-下位関係であるとき，

$$W_u \longrightarrow W_d \qquad (2.4)$$

の生成規則として捉え，単語 w の概念が W であるとき，

$$W \longrightarrow w \qquad (2.5)$$

の生成規則として捉える．

従って，シソーラスを用いた場合の文法の生成規則のパターンは次の8通りとなる．

1. $Y(H, f) \longrightarrow X(H)\quad Z(f)$
2. $X(H) \longrightarrow Y_1(-H)\quad \cdots\quad Z(H)\quad \cdots\quad Y_n(-H)$
3. $Y(-H) \longrightarrow Y(H', f)$
4. $Z(f) \longrightarrow \alpha\quad Z(f)\quad \beta$
5. $Z(f) \longrightarrow f$
6. $Y(H) \longrightarrow H$
7. $H \longrightarrow H'$
8. $H \longrightarrow h$

但し，記号Hはシソーラス中のある概念記号であり，これを*head word*として用いている．h，fは単語，α，βは単語列である．パターン6～8の生成規則により，*head word*（概念）がHであるカテゴリYの句（語）が，Hの下位語に書き換えられることを表現している．

3. 確率文脈自由文法とそのパラメタ推定

3.1 確率文脈自由文法

確率文脈自由文法PCFGは次のような5つ組で定義される．

$$G = (\Sigma, V, P, S, p) \qquad (3.1)$$

- Σ：終端記号の有限集合
- V：非終端記号の有限集合
- S：開始記号$(S \in V)$
- P：生成規則の有限集合
- p：$P \rightarrow (0, 1]$ (Pから(0,1]への写像)

$p(X \rightarrow \alpha)$を$X \rightarrow \alpha$の適用確率という．左辺が同一の

非終端記号(Xとする)であるすべての書き換え規則を

$$X \longrightarrow \alpha_i \qquad (i = 1, 2, \ldots, I_X) \tag{3.2}$$

とすると,

$$\sum_{i=1}^{I_X} p(X \to \alpha_i) = 1 \tag{3.3}$$

が成り立つ.

PCFGでは文$s \in L(G)$,sの導出木Tに対し,Tの生起確率$P_r(T)$は,Tの導出に適用された書き換え規則の適用確率の(重複を含めた)積である.

3.2 パラメタ推定 (最尤推定法)

PCFG $G = (\Sigma, V, P, S, p)$において, $p : P \to (0, 1]$は標本(事例データ)に基づいて推定することが出来る.ここでは,パラメタ推定法のうちの最尤推定法について述べる.

N個の標本(構文木)を$T_1, T_2, \cdots, T_N$とし書き換え規則$X \to \alpha \in P$が構文木Tの導出に適用された回数を$n(T, X \to \alpha)$で表す.標本の採集が互いに独立に行なわれたと仮定すると$T_1, T_2, \cdots, T_N$が標本として収集される確率(尤度)は,

$$\prod_{k=1}^{N} P_r(T_k) \tag{3.4}$$

であり,これを最大にする$p(X \to \alpha_i)$の値は次の式で与えられる.

$$p(X \to \alpha_i) = \frac{\sum_{k=1}^{N} n(T_k, X \to \alpha_i)}{\sum_{i=1}^{I_X} \sum_{k=1}^{N} n(T_k, X \to \alpha_i)} \tag{3.5}$$

4. 実　　験

4.1 実験方法

名詞が「の」で連結した名詞句(「名詞の名詞」)を,前述により作成したPCFG で解析してその正解率を評価する.

実験では日本電子化辞書研究所が作成したEDRコーパスを用いる.このコーパスには,文と,その文の構文情報や形態素情報などが格納されている.EDRコーパスから,名詞が「の」で連結された名詞句と,個々の名詞の概念(語義)およびその係り受けを抽出する.例えば「谷の激流を身もだえしてサケが上る.」に対する,形態素データ,構文木データ,概念関係データから,名詞句「谷の激流」,この名詞句における「谷」の概念記号が3cec8a,「激流」の概念記号が3cf2cf であること,および,「谷」が「激流」に係ることが抽出できる.このようにして,コーパスの中の「名詞の名詞」を抽出する.抽出したデータおよびシソーラスを用いて生成規則を作成する.今回の実験では,シソーラスの root node からl 段下の概念記号を*head word*として行なう.但し概念記号がすでに root からl 段目以内にあるような単語の場合,その概念記号を*head word*とする.

作成する生成規則のパターンは次の通りである.

1. $S \longrightarrow NP(C)$
2. $NP(C) \longrightarrow PP(-C) \quad NP(C)$
3. $PP(-C) \longrightarrow PP(C', の)$
4. $PP(C, の) \longrightarrow NP(C) \quad P(の)$
5. $NP(C) \longrightarrow C$
6. $C \longrightarrow C'$
7. $C \longrightarrow w$
8. $P(の) \longrightarrow の$

但し,S は開始記号,C, C'は概念記号,wは単語を表す.

ここで,一つの概念の上位概念が必ずしも一つではないので,構文木が複数存在する場合がある.3.2節で挙げた最尤推定式,式(3.5)は,正しい構文木を標本として収集する必要がある.しかしこの場合は複数の構文木のどれが正しいかの保証がないため,そのような場合は,構文木の数をNとすると,各構文木の頻度を$1/N$とする.その頻度つきの構文木の列を学習データとして,最尤推定によって生成規則の適用確率(パラメータ)を推定する.推定式は式(4.1)のように与えられる.

$$p(X \to \alpha_i) = \frac{\sum_{k=1}^{N} n(T_k, X \to \alpha_i) f(T_k)}{\sum_{i=1}^{I_X} \sum_{k=1}^{N} n(T_k, X \to \alpha_i) f(T_k)} \tag{4.1}$$

但し,$f(T_k)$はT_kの頻度である.

EDRコーパスから「名詞の名詞の名詞」の概念およびその係り受けを抽出し,それに対して先に作成したPCFGを用いて以下のようにして係り先を判定した結果と比較する.「N_1のN_2のN_3」(=「名詞の名詞の名詞」)において,係り受けのパターンはN_1がN_2に係る場合とN_1がN_3に係る場合の2 種類が考えられる.テスト文「N_1のN_2のN_3」を構文解析してそれらの構文木を,係り受けのパターンにより2種類に分類する.それぞれのパターンにおいて構文木の確率の和を算出し,確率の大きい方のパターンを構文解析における係り受け判定とする.これがテスト文のコーパス中の係り受けパターンと合致していれば正解として,全テスト文に対する正解の割合を求める.今回は学習ずみのテスト文と学習なしのテスト文に対して行なった.学習データはEDRコーパスから抽出した「名詞の名詞の名詞」と「名詞の名詞」である.実験は学習ずみのテスト文と学習なしのテスト文の両方に対して行なった.ここで学習ずみのテスト文とはその抽出した「N_1のN_2のN_3」の一部を意味し,学習なしのテスト文とは,学習のために抽出されていない「N_1のN_2のN_3」を意味する.

Table 1 The number of rules and head words

l	生成規則の数 (個)	*head word*の数 (個)
3	30228	940
4	33493	1103
5	41286	1769
6	50703	2947
7	59879	5110

Table 2 The rate of acceptable input sentences by the grammars

l	学習ずみテスト文	学習なしテスト文
3	100 %	61.5 %
4	100 %	61.1 %
5	100 %	55.4 %
6	100 %	43.3 %
7	100 %	29.0 %

Table 3 The number of test sentences belonging to each group

グループ	学習ずみテスト文	学習なしテスト文
1	417	61
2	0	13
3	10	3
0	0	403

4.2 実験結果

今回の実験では，使った「名詞の名詞」の数は20000個に固定し，段数lを3〜7に変化させて行なった．学習ずみのテスト文は427個，学習なしのテスト文は480個であった．

段数lに対応した生成規則の数，*head word*の数を**表−1**に示し，テスト文に対応する構文木が作成出来た割合は**表−2**の通りであった．

テスト文を係り受けで4つのグループに分類した．分類はテスト文の係り受け関係と学習データによるものとし，具体的には，テスト文「N_1のN_2のN_3」で「N_1がN_2」に係っており，学習データに「N_1のN_2」があり「N_1のN_3」がない場合はグループ1に属し，逆に「N_1のN_3」があり「N_1のN_2」がない時はグループ2に属させる．グループ1はテスト文と同じ係り受け関係のみが学習されている文であり，グループ2は逆の係り受け関係のみが学習されている文である．学習データに「N_1のN_2」も「N_1のN_3」も含む時はテスト文をグループ3に属させ，「N_1のN_2」も「N_1のN_3」もない時はグループ0に属させる．学習ずみのテスト文と学習なしのテスト文における各グループに属する文の個数を**表−3**に示す．学習量が増加するにつれてグループ3に属する文が増加し，グループ0は減少する．グループ3に属する文は，構文解析結果の係り受け構造に曖昧さが出てくる可能性があることを意味しており，そのような文の数は**表−3**より学習ずみのものは10個，学習なしのものについては3個であった．それらの全テスト文に対する割合が低く実験の結果には信頼性がないと思われるため，今回は学習ずみのテスト文と学習なしのテスト文の2種類に対して実験を行なった．

Table 4 The rate of correct dependency by PCFG

l	学習ずみテスト文	学習なしテスト文
5	79.3 %	64.4 %
7	94.2 %	68.3 %

構文木が作成されたテスト文の中で，学習ずみのテスト文及び学習なしのテスト文に対する構文解析結果が正しい係り受けと判定された割合は**表−4**の通りであった．

4.3 考察

表−4の結果では，段数lが大きい場合係り受けが正解である割合が高くなることを示している．これはlが大きくなると係り受け制約の表現が細かくなるため，学習ずみのテスト文と学習なしのテスト文の両方において正しい構文木をある程度正しく導出出来ることを意味している．

言語現象の一つとして，単語の係り先の単語に曖昧さがある時は，単語から一番近い位置にある単語に係りやすいと言われている．コーパス中にある「N_1のN_2のN_3」の個数は8623個であり，これによると「N_1がN_2」に係る方が「N_1がN_3」に係るより可能性が高いはずで，実際「N_1がN_2」に係る方が6230個で全体の72.25%を占めた．従って，係り受け解析をするときに係り先の単語の曖昧さがある文では，無条件に単語から係り得る単語の中で一番近い単語に係るものとしてもある程度の結果は期待できる．今回の実験では，**表−4**の学習ずみの結果を見ると，7段の時は94.2%という正解率を得ており，学習なしの方は68.3%で，72.25%より若干落ちる結果となったが，学習量を増やすことで，構文木が作成される割合や正解率が増加するものと期待できる．

5. おわりに

我々が提案した係り受け制約を組み込んだ文脈自由文法に対して，名詞句「N_1のN_2のN_3」と*function*を「の」に限定して実験を行なった結果，有効性を確認した．次は一般の日本語文に対して実験を行ない有効性を確認したい．

参考文献

1) 田辺利文，冨浦洋一，日高達: 係り受け制約の文脈自由文法への組み込み法，九州大学大学院システム情報科学研究科報告，1996