九州大学学術情報リポジトリ
Kyushu University Institutional Repository

翻訳文法を用いた機械翻訳

柴田，雅博
九州大学大学院システム情報科学府知能システム学専攻 ： 博士後期課程

日高，達
九州大学大学院システム情報科学研究院知能システム学部門

冨浦，洋一
九州大学大学院システム情報科学研究院知能システム学部門

https://doi.org/10.15017/1515819

出版情報：九州大学大学院システム情報科学紀要. 8 (1), pp.61-66, 2003-03-26. 九州大学大学院システム情報科学研究院
バージョン：
権利関係：

九州大学大学院
システム情報科学紀要
第8巻 第1号 平成15年3月

Research Reports on Information Science and
Electrical Engineering of Kyushu University
Vol.8, No.1, March 2003

翻訳文法を用いた機械翻訳

柴田 雅博* ・ 日高 達** ・ 冨浦 洋一**

Machine Translation using Translation Grammar

Masahiro SHIBATA, Toru HITAKA and Yoichi TOMIURA

(Received December 13, 2002)

Abstract: This paper proposes the Machine Translation using newly proposed Translation Grammar. The grammar's production rules have the information for translation in addition to grammar rule, so syntactic tree parsed by this grammar has the information for translation too. The information for translation attached to a syntactic tree is used not only to derive the target sentence, but also to select the best syntactic trees for translation.

Keywords: Machine Translation, Transfer-based, Context Free Grammar, Lexical Cooccurrence

1. はじめに

トランスファー型翻訳（Transfer-based Machine Translation）[1] は代表的な機械翻訳手法の一つであり，商用の機械翻訳ソフトにも広く採用されている．従来のトランスファー型翻訳は，(1) 原言語文法を用いて原文を解析し，原言語の依存構造を求め，(2) 変換規則（語彙変換，構造変換）を用いて原言語の依存構造を目標言語の依存構造に変換し，(3) 目標言語の依存構造から訳文を生成する，という三つの過程から成る．ここで，一般に文を構文解析した際には，複数の構文木が得られるため，解析結果の絞り込みが重要となるのだが，従来のトランスファー型翻訳では，解析段階では原言語の情報しか利用されておらず，絞り込みに対し，翻訳結果を考慮に入れた評価基準を定めることができなかった．そのため，成句表現など原言語と目標言語とでその構造が著しく異なり，特殊な変換を必要とする文へは，例外操作でアドホックに対処するなどしていた．しかし，個々の例外操作に対する適用条件が繁雑になるなど，システム全体の見通しが悪くなっており，またシステムの拡張が困難であった．

そこで本稿では，解析と変換とで独立に用意されていた規則を統合し，解析に用いる文法に対し，翻訳に使用される情報（翻訳情報）を付与した文法（翻訳文法），および翻訳文法を用いた機械翻訳手法を提案する．翻訳文法は，二言語間翻訳を対象としており，特にここでは英日翻訳について考える．二言語間翻訳に限定するとき，翻訳情報として用意すべき情報は次の三つとなる．

- **訳語：** 原文中の各単語に対応する目標言語の単語または句
- **訳語順：** 訳文生成の際に，訳していく原文の句の順序
- **additional function phrase (AFP)：** 原言語では単語として現れないが，目標言語では単語または句として挿入する必要のある機能語の句

ここで，翻訳における主な問題の一つとして単語の訳し分けが挙げられる．一般にある単語に対しては，その訳語の候補が複数存在し，その中から妥当な訳語を選択しなければならない．訳し分けを行なうためには，その単語だけでなく周囲の情報まで考慮する必要がある．翻訳文法では，特に語彙の共起性を利用して訳語の絞り込みを行なう．

翻訳文法を用いた機械翻訳手法では，翻訳情報が付随した文法で解析を行なうので，解析結果の絞り込みに対して，翻訳情報を利用して，翻訳結果を考慮に入れた絞り込み基準を設定することができる．これにより，例外処理などでシステムを繁雑にすることなしに機械翻訳システムを構築することが可能となる．

2. 翻訳文法

2.1 翻訳文法の枠組

翻訳文法は，文脈自由文法（Context Free Grammar ; CFG）の生成規則に翻訳情報を付与したものである．ただし，語彙の共起性を利用した単語の訳し分けを実現するために，語彙共起制約文脈自由文法（Lexical Cooccurrence Constraints Grammar ; LCCG）[2)3)] を基に構成される．LCCG は，統語範疇をhead word（句の主辞となる単語）とfunction（係りの種類を規定する機能語）とで細分化したものを非終端記号とする文脈自由文法である．LCCGは非終端記号を細分化することにより，語彙の共起関係（係る句のhead word hとfunction f お

平成14年12月13日受付
* 知能システム学専攻博士後期課程
** 知能システム学部門

よび係られる句のhead word h'の三つ組(h, f, h')で表わされるものとする）を生成規則上に表現でき，その生成規則には句のhead wordや単語であるfunction（function word）といった語彙情報が明示されている．翻訳文法ではこの生成規則に付与された語彙に対し，直接訳語を指定することで，語彙共起パターンと一対一に対応する訳し分け規則を表現する．翻訳文法は次のように定義される．

定義 1: 翻訳文法

$G^t = (\Sigma, N, P^t, S)$

ただし，

Σ	:	（原言語）終端記号の集合
N	:	（原言語）非終端記号の集合
P^t	:	翻訳文法規則の集合
S	:	開始記号$(S \in N)$

ここで，P^tは翻訳文法規則の集合であり，翻訳文法規則はLCCGの生成規則である構文情報$p \in P$と，この構文情報で解析される句を翻訳する際の翻訳情報ξから成る．実際の翻訳文法規則は次のように表現される．

定義 2: 翻訳文法規則

$$Y \rightarrow \underset{c_1}{\overset{\eta_1}{X_1^{\#}}} \cdots \underset{c_i}{\overset{\eta_i}{X_i^{\#*}}} \cdots \underset{c_m}{\overset{\eta_m}{X_m^{\#}}} \tag{1}$$

ただし，$X^{\#}$は以下のいずれかの形式である．

$$X \quad or \quad \underset{h^t}{X(h)} \quad or \quad \underset{f^t \;\; h^t}{X(f\,,\,h)}$$

ここで，Xは統語範疇がXの句を導出する非終端記号，$X(h)$は統語範疇がXでhead wordがhである句を導出する非終端記号，$X(f,h)$は統語範疇がXでhead wordがh, functionがfである句を導出する非終端記号を表わす．また，右辺の句のうちで head phrase（右辺中で他のどの句にも係っていない句）には$*$を与える[†1]．

翻訳文法規則にはそれぞれ次のような翻訳情報が与えられる．右辺の句のうち，特定の単語が指定されているhead word h, およびfunction word fに対して，それぞれの訳語h^t, f^tを与える．翻訳文法規則は一つの語彙共起パターンを表わしており，生成規則上の単語情報に直接訳語を与えることで語彙共起パターンに応じた単語の訳し分け規則を表現する．

また，右辺の各句には訳語順η_j $(j = 1, \cdots, m)$が与えられる．右辺の句を左から順に自然数$1, \cdots, m$に対応させるとη_jは次のようなある置換として定義される．

$$\begin{pmatrix} 1 & \cdots & j & \cdots & m \\ \eta_1 & \cdots & \eta_j & \cdots & \eta_m \end{pmatrix}$$

英語では動詞は目的語の前にあるが，日本語では動詞は文末にあるというように，一般に原言語と目標言語とでは文の構造が異なり，翻訳の際には訳す順序を換える必要がある．訳文生成の際には，右辺の各句からそれぞれ生成された訳文の句（訳句）を訳語順にしたがって並び換えることで，目標言語の文として妥当な順番に直す．

$c_j (j = 1, \cdots, m)$はAFPである．AFPは原言語の文法では単語として現れない機能語のうち，目標言語の文法では単語として陽に表わす必要のある機能語の句である．例えば．英語では名詞句の格は，動詞の前なら主格，他動詞の直後なら目的格というように位置によって与えられるが，日本語では「が」や「を」といった助詞によって格が与えられる．英語から日本語に翻訳する場合には，訳語とは別に，句と句の間にこのようなAFPを挿入しなければならない．

以上のように，$Y \rightarrow X_1^{\#} \cdots X_i^{\#*} \cdots X_m^{\#} \in P$の右辺の各句$X_j^{\#}$には翻訳情報$(\eta_j, h_j^t, f_j^t, c_j)$が与えられ，$\xi$は句の翻訳情報を左から順に並べた列$\xi = (\eta_1, h_1^t, f_1^t, c_1) \cdots (\eta_m, h_m^t, f_m^t, c_m)$として与えられる．

例 1

次の翻訳文法規則を考える．

$$VP \rightarrow \underset{\varepsilon \quad 入る}{\overset{2}{VT(take)^*}} \; \underset{に \quad 風呂}{\overset{1}{NP(bath)}} \tag{2}$$

この翻訳文法規則では句$VT(take)$のhead word “take”の訳語が「入る」，句$NP(bath)$のhead word “bath”の訳語が「風呂」であることを表わす．また，$VT(take)$の訳語順は2，$NP(bath)$の訳語順は1である．ここで，句Xに対する訳句を$[X]^t$と表わすとすると，(2) を適用した場合の訳文は右辺の各句の訳句を訳語順に従って$[NP(bath)]^t, [VT(take)]^t$の順に並び換え，その間に$NP(bath)$に与えられたAFP「に」を挿入したもの，すなわち“$[NP(bath)]^t$に$[VT(take)]^t$”となる．

2.2 翻訳文法規則の分類

翻訳文法では，語彙共起パターンごとに翻訳文法規則を用意し，それぞれの語彙共起パターンに対して訳し分けを行なう．しかし，発生し得る全ての語彙共起パターンを網羅的に収集することは現実的には困難であるため，語彙共起パターンの一般化を考える．このように，与えられる情報の詳細さによって，翻訳文法規則を次の三つ

†1 head phrase は function を持たず，X^* もしくは $X(h_i)^*$ のいずれかである．

に分類する．

- **汎用規則**
  右辺の全ての句についてそのhead word，functionが特定されておらず，統語構造のみを表わす翻訳文法規則．汎用規則には翻訳情報として訳語順とAFPが与えられるが，訳語情報は与えられない．
- **語彙規則**
  生成規則右辺が単語（終端記号）である翻訳文法規則．語彙規則には翻訳情報として，右辺の単語に対する訳語が与えられる．
- **パターン規則**
  右辺の一つ以上の句について，そのhead word あるいはfunctionが特定されている翻訳文法規則．パターン規則には翻訳情報として，単語情報が与えられている句のhead word（function word）に対する訳語と，訳語順およびAFPが与えられる．

ある単語に対する訳語を，デフォルトの訳語とそれ以外の訳語とに分ける．デフォルトの訳語は，その単語に対する最も一般的な訳語であり，普通は対訳単語辞書の第一語義の訳語に当たる．周囲の状況が分からず，その単語単独だけを見て訳語を選択する場合にはデフォルトの訳語を選択する方が他の訳語を選択するよりも誤りが少ないと考えられ，また，デフォルト以外の訳語を選択すべき場合には，妥当な訳語を特定する何らかの情報（語彙共起パターン）が存在すると考えられる．そこで，語彙共起パターンのうちデフォルトの訳語だけで訳せるものについては一般化して汎用規則とし，訳し分けを必要とするもののみパターン規則として用意する．また，デフォルトの訳語は語彙規則上に与える．これにより，通常は語彙規則と汎用規則だけで翻訳を行なうことができ，また訳し分けを必要とする場合にはパターン規則を追加することで対処することが可能となる．

例 2

次の翻訳文法を考える．

$$S \rightarrow \overset{1}{NP}\ \overset{2}{VP^*} \quad \text{は}\ \ \varepsilon \tag{3}$$

$$VP \rightarrow \overset{2}{VT^*}\ \overset{1}{NP} \quad \varepsilon\ \ \text{を} \tag{4}$$

$$VP \rightarrow \overset{2}{VT}(take)^*\ \overset{1}{NP}(bath) \quad \varepsilon\ \ \text{入る}\ \ \text{に}\ \ \text{風呂} \tag{5}$$

$$NP \rightarrow DET\ \overset{1}{N^*} \quad \varepsilon\ \ \varepsilon \tag{6}$$

$$NP \rightarrow I \quad \text{私} \tag{7}$$

$$VT \rightarrow take \quad \text{取る} \tag{8}$$

$$DET \rightarrow a \quad \varepsilon \tag{9}$$

$$N \rightarrow bath \quad \text{風呂} \tag{10}$$

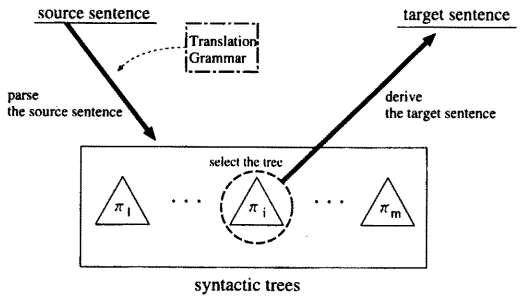

Fig.1 Machine Translation system using Translation Grammar.

上記翻訳文法規則のうちで，(3),(4),(6)は汎用規則である．(7)〜(10) は語彙規則である．また(5)はパターン規則である．

3. 翻訳文法を用いた機械翻訳手法

翻訳文法ではトランスファー型翻訳方式を採用する．従来のトランスファー型翻訳方式は(1)原文解析，(2)変換，(3)訳文生成という三つの過程から構成され，解析に用いられる生成規則と変換に用いる変換規則とを独立に用意していた．それに対し，翻訳文法では翻訳情報を生成規則が保持しており，解析結果の絞り込みに対して翻訳情報を利用することができる．また，解析結果得られた構文構造（構文木）には文法に与えられた翻訳情報が付随しており，構文木に与えられた翻訳情報を用いて，解析後直ちに訳文を生成する．すなわち，翻訳文法を用いた機械翻訳手法は（1）原文解析，(2）訳文生成 の二過程で構成される†2（**Fig.1**）．

3.1 原文解析

原文解析では，原文を構文解析して構文木を求める．構文解析は，翻訳文法の生成規則部分だけを使用して実行することができ，翻訳文法規則に与えられた翻訳情報の影響は受けない．従って，既存のCFG構文解析手法[4)5)]をそのまま適用すれば，解析は実現できる．

Fig.2は，例2の翻訳文法を用いて解析した結果の構文木の一例である．ここで，生成規則の両辺には，右辺の head phrase の head word が左辺のhead word であるという制約があり，ボトムアップに解析を行なうことで，この制約に従って，構文木の各句（構文木の各ノードを根とする部分木が表わす句）のhead wordおよびfunctionを全て特定することができる．また，構文木の各ノードには，翻訳文法規則によって訳語，訳語順，AFP といった翻訳情報が与えられることとなる．

†2 実装に対しては，後処理として日本語の活用処理を行なった後に最終的な訳文を出力する．

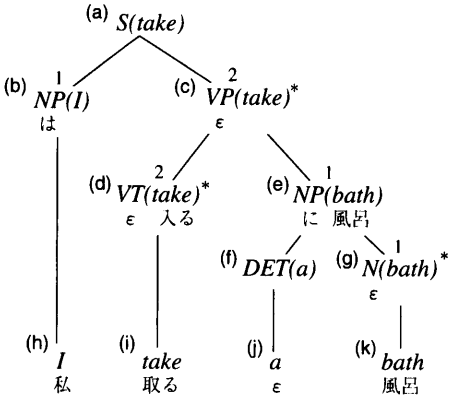

Fig.2 An example of the syntactic tree parsed by Translation Grammar (using rule (5) at the node (c)).

3.2 訳文生成

訳文生成では，原文解析で得られた構文木に与えられた翻訳情報を用いて訳文を生成する．

ここで，構文木に与えられた翻訳情報のうち，訳語順とAFPは句単位の情報なので問題ないが，訳語は句ではなく原文中の単語に対する情報なので，次のような問題がある．訳語は構文木の各ノードに対して，(翻訳文法規則に訳語が与えられていれば）それぞれ与えられる．そのため，同じ単語に対して，その訳語が複数箇所で与えられる可能性がある．例えば，**Fig.2**のノード(d)とノード(i)において，そのhead wordは共に同じ単語“take”であるが，“take”に対する訳語はノード(d)では「入る」，ノード(e)では「取る」と異なるものが与えられている．この問題に対し，狭い範囲の文脈を見て訳語を決定するよりは，広い範囲の文脈を見て訳語を決定した方が，より妥当な訳語を選択することができるという仮定を置き，head word が等しい句のうち，最も大きな句をカバーしているもの，つまりはできるだけ上位ノード（根からの深さが最も浅いノード）で与えられた訳語を選択する．

以上を踏まえて，訳文生成は以下のように実行される．ただし簡単のために，functionを含む句を考えず，head wordだけを持つ句についてのみ考えることとする．上位優先での訳語選択を実現するので，ある句のhead wordに対して，上位ノードで既に訳語が与えられている場合と，まだ訳語が与えられていない場合とに分けて考えなければならない．ノード$Y(h)$を根とする翻訳情報付き構文木をπ（ノードが葉のときは$\pi=\varepsilon$）とし，更に上位ノードで決定されたhead word hに対する訳語をu（上位でまだ訳語が与えられていないときは$u=*$）とするとき，構文木πから生成される訳文$[Y(h),\pi,u]^t$は次のように定義される．

定義 3: **訳文** $[Y(h),\pi,u]^t$

- $$[Y(h),\pi,u]^t = [X^{\#}_{\eta_1^{-1}},\pi_{\eta_1^{-1}},\varepsilon]^{t'}\cdots [X_{\eta_i^{-1}}(h)^*,\pi_{\eta_i^{-1}},u]^{t'}\cdots[X^{\#}_{\eta_m^{-1}},\pi_{\eta_m^{-1}},\varepsilon]^{t'}$$

ただし，$X^{\#}_{\eta_1^{-1}},\cdots,X_{\eta_i^{-1}}(h)^*,\cdots X^{\#}_{\eta_m^{-1}}$はそれぞれ構文木$\pi$における根ノード$Y(h)$の子ノード，$X_{\eta_i^{-1}}(h)^*$は*head phrase*，また各$\pi_j$は$Y_j$を根とする部分木である．また，$Y(h)$を作るときに適用した翻訳文法規則が定義2の(1)であったとすると，η_i^{-1}は次式を満足する．

$$\begin{pmatrix} 1 & \cdots & i & \cdots & m \\ \eta_1 & \cdots & \eta_i & \cdots & \eta_m \end{pmatrix} = \begin{pmatrix} \eta_1^{-1} & \cdots & \eta_i^{-1} & \cdots & \eta_m^{-1} \\ 1 & \cdots & i & \cdots & m \end{pmatrix}$$

- $$\underset{c}{[Y(w),\pi,u]^{t'}} = [Y(w),\pi,u]^t\, c$$

- $$\underset{c\ \ w^t}{[Y(w),\pi,u]^{t'}} = \begin{cases} [Y(w),\pi,u]^t\, c & (u\neq *) \\ [Y(w),\pi,w^t]^t\, c & (u=*) \end{cases}$$

- $$\underset{w^t}{[w,\varepsilon,u]^{t'}} = \begin{cases} u & (u\neq *) \\ w^t & (u=*) \end{cases}$$

例 3

Fig.2の翻訳情報付き構文木に対して訳文生成は次のように実行される．ただし式中で，π_xは**Fig.2**の構文木中のノード(x)を根とする部分木を表わすものとする．

$$
\begin{aligned}
&[S(take),\pi_a,*]^t \\
&= \underset{は}{[NP(I),\pi_b,*]^{t'}}\ \underset{\varepsilon}{[VP(take),\pi_c,*]^{t'}} \\
&= [NP(I),\pi_b,*]^t\ は\ [VP(take),\pi_c,*]^t \\
&= \underset{私}{[I,\varepsilon,*]^{t'}}\ は\ [VP(take),\pi_c,*]^t \\
&= 私\ は\ [VP(take),\pi_c,*]^t \\
&= 私\ は\ \underset{に\ 風呂}{[NP(bath),\pi_e,*]^{t'}}\ \underset{\varepsilon\ 入る}{[VT(take),\pi_d,*]^{t'}} \\
&= 私\ は\ [NP(bath),\pi_e,風呂]^t\ に \\
&\quad [VT(take),\pi_d,入る]^t \\
&= 私\ は\ \underset{\varepsilon}{[N(bath),\pi_g,風呂]^{t'}}\ に \\
&\quad [VT(take),\pi_d,入る]^t \\
&= 私\ は\ [N(bath),\pi_g,風呂]^t\ に \\
&\quad [VT(take),\pi_d,入る]^t \\
&= 私\ は\ \underset{風呂}{[bath,\varepsilon,風呂]^{t'}}\ に\ [VT(take),\pi_d,入る]^t \\
&= 私\ は\ 風呂\ に\ [VT(take),\pi_d,入る]^t \\
&= 私\ は\ 風呂\ に\ \underset{取る}{[take,\varepsilon,入る]^{t'}} \\
&= 私\ は\ 風呂\ に\ 入る
\end{aligned}
$$

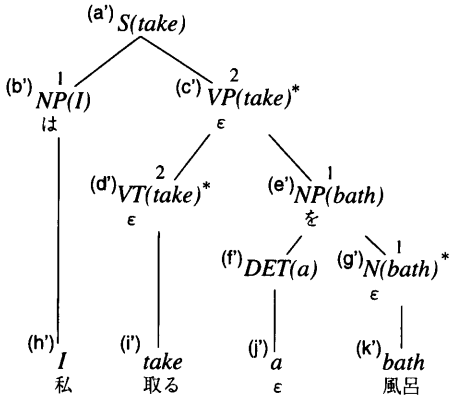

Fig.3 An example of the syntactic tree parsed by Translation Grammar (using rule (4) at the node (c')).

4. 解析結果の絞り込み評価基準

一般に構文解析の結果，複数の構文構造（構文木）が得られるため，解析結果の絞り込みが重要となる．例えば，例2の翻訳文法を用いて"I take a bath."という原文を解析すると，**Fig.2**の他に**Fig.3**のような翻訳情報付き構文木も得られる．**Fig.2**では，ノード(c)から(5)のパターン規則が適用されているのに対し，**Fig.3**では，ノード(c')から(4)の汎用規則が適用されており，この構文木からは「私は風呂を取る」という不自然な訳文が生成される．

翻訳文法では翻訳情報付き構文木によって生成される訳文が決定されるため，構文木の選択がそのまま訳文の選択に直結することとなる．絞り込みを行なうためには，解析結果に対して何らかの評価基準を設ける必要があるが，翻訳では，原文の意味を保存した訳文を得ることが目的であるから，最終的に生成される訳文が妥当となるように評価基準を定めるべきである．従来手法と異なり，翻訳文法を用いて解析した結果得られる構文木には翻訳情報が付随しているため，絞り込みに対して，この翻訳情報を利用することができる．

ここで，妥当な訳文についてだが，訳文が原文の意味を保存しているためには，少なくとも原文の各単語に対して適切な訳語が与えられている必要があると考えられる．そこで，本稿では訳文の妥当性を原文中の各単語に対する訳語選択の妥当性と置き換えて評価し，できるだけ妥当な訳語が選ばれている訳文を，またそのような訳文を生成する構文木を選択する．また，訳語選択の妥当性は，次のように考える．

人間が翻訳を行なう場合には，原文中に熟語や成句表現といった語彙共起パターンが見つかったら，まずそのパターンに照らし合わせて翻訳を試みるものと思われる．翻訳文法ではこの語彙共起パターンはパターン規則として用意されており，パターン規則を使って訳語を選択した方が妥当だと考えることができる．また，パターン規則の中でも右辺の長い規則は，より広範囲の文脈を詳細に表現しており，このような翻訳文法規則で与えられた訳語の方がより妥当であると仮定する．

そこで，原文sを解析して得られた翻訳情報付き構文木πに対する評価値$G(\pi)$を次のように定義する．

定義 4: 翻訳情報付き構文木πに対する評価値

$$G(\pi) = \sum_{i=1}^{n} g(w_i, \pi)$$

$$g(w_i, \pi) = l(p(w_i, \pi))$$

ただし，

$s = w_1 \cdots w_n$	:	原文（$w_i = \Sigma$）
$p(w, \pi)$	:	構文木πで単語wの訳語を決定した翻訳文法規則
$l(p)$	:	翻訳文法規則pの右辺長

原文sを構文解析した結果，得られる構文木の全体集合をΠ_sとすると，

$$\pi_{max} = \arg\max_{\pi \in \Pi_s} G(\pi)$$

なるπ_{max}を訳語選択に関して最も妥当な訳文が生成される構文木であるとし，π_{max}を選択する．

例 4

Fig.2と**Fig.3**の翻訳情報付き構文木について，それぞれ定義4の評価値を計算する．**Fig.2**の構文木をπ_1，**Fig.3**の構文木をπ_2とすると，それぞれの構文木に対する評価値$G(\pi_1), G(\pi_2)$ は

$$\begin{aligned} G(\pi_1) &= g(\text{"}I\text{"}, \pi_1) + g(\text{"}take\text{"}, \pi_1) \\ &\quad + g(\text{"}a\text{"}, \pi_1) + g(\text{"}bath\text{"}, \pi_1) \\ &= 1 + 2 + 1 + 2 \\ &= 6 \end{aligned} \tag{11}$$

$$\begin{aligned} G(\pi_2) &= g(\text{"}I\text{"}, \pi_2) + g(\text{"}take\text{"}, \pi_2) \\ &\quad + g(\text{"}a\text{"}, \pi_2) + g(\text{"}bath\text{"}, \pi_2) \\ &= 1 + 1 + 1 + 1 \\ &= 4 \end{aligned} \tag{12}$$

となり，パターン規則を適用したπ_1の方が評価値が高くなり，「私は風呂に入る」という訳文を生成する構文木の方を選択することができる．

上記のように，翻訳文法を用いた機械翻訳では原文解析結果の絞り込みに対し，訳語選択に対する妥当性と言う観点から，明確で統一的な評価基準を設けることができる．

5. お わ り に

本稿では，翻訳文法および翻訳文法を用いた機械翻訳手法を提案した．翻訳文法は，生成規則が翻訳情報が付

随しており，この文法を用いて原文解析を行なうので，解析結果の絞り込みに対して，訳語選択の観点から翻訳結果を考慮した評価基準を定めることができる．これにより，変換過程におけるシステムの繁雑さを解消することができ，見通しのよい機械翻訳システムの構築が可能となる．

この翻訳文法を用いて簡単な予備実験を行なったが[6]，十分なパターン規則を用意できなかったこともあり，十分な絞り込みが行なえなかった．これは，単語辞書である語彙規則を除くと，語彙レベルでのマッチングを要求する非常に詳細なパターン規則と，逆に非常に荒い汎用規則しかなく，また翻訳文法の評価基準は汎用規則に対して全く効かないことが原因である．これに対処するために，意味カテゴリーを利用し，汎用規則とパターン規則の中間に当たるカテゴリー規則を導入することを検討している．

参考文献

1) 長尾真 編: 自然言語処理, 岩波講座ソフトウェア科学 15, 岩波書店, 1996.
2) 田辺利文, 冨浦洋一, 日高達：係り受け文脈自由文法とその日本語への適用, 情報処理学会論文誌, Vol.41 No.01, pp.1234–1243, 2000.
3) 田辺利文: 語彙共起制約文脈自由文法, 九州大学大学院学位論文, 2000.
4) Earley, J.: An Efficient Context-Free Parsing Algorithm, CACM, Vol.13, No.2, pp.94–102, 1970.
5) Kay, M.: Algorithm Schemata and Data Structure in Syntactic Processing, Technichal Report CSL-80-12, XEROX PARC, 1980.
6) 柴田雅博, 冨浦洋一, 日高達: 翻訳文法による機械翻訳とその実装, 九州大学情報基盤センター年報 第 2 号, pp.71–79, 2002.